NASA Technical Memorandum 84402

N84-24276

An Explicit Predictor-Corrector Solver with Applications to Burgers' Equation

Suhrit K. Dey and Charlie Dey

September 1983

National Aeronautics and
Space Administration

NASA Technical Memorandum 84402

An Explicit Predictor-Corrector Solver with Applications to Burgers' Equation

Suhrit K. Dey, Ames Research Center, Moffett Field, California
Charlie Dey, Country Lane School, San Jose, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field. California 94035

AN EXPLICIT PREDICTOR-CORRECTOR SOLVER WITH APPLICATION TO BURGERS' EQUATION

Suhrit K. Dey and Charlie Dey*

Ames Research Center

SUMMARY

Forward Euler's explicit, finite-difference formula of extrapolation is used as a predictor and a convex formula as a corrector to integrate differential equations numerically. An application has been made to Burgers' equation.

INTRODUCTION

The single-step, explicit, forward Euler scheme is possibly the simplest algorithm with which to solve initial-value problems. However, it has very restricted stability properties which require small step sizes so that this formula of extrapolation may be useful. Implicit schemes, in which large step sizes may be used, have better stability properties. But iterative methods are generally used for computational solution. Most predictor-corrector algorithms require multistep operations in which correctors use iterative methods for convergence.

In this work, a filtering formula has been introduced by the second author so that extrapolated values obtained by an explicit formula may be appropriately corrected using one iteration which requires no computation of derivatives or inversion of a matrix. Mathematically, the corrector introduces a convex mapping, and as such it may be called a <u>convex</u> corrector. The explicit formula of extrapolation used in this work is the forward Euler difference scheme. Linearized stability analysis by Lomax (private communication) showed that the method is stable for step sizes that could be much larger than those required for stability of the forward Euler scheme. This will be discussed further on. With regard to the solution of nonlinear models, the limitations of this scheme are still being investigated.

Let us first develop the algorithm, discuss its stability properties, and look into some of its simple applications.

THE ALGORITHM

Let us consider an initial-value model:

$$\left.\begin{aligned} \frac{du}{dt} &= f(u,t) \\ u(t_o) &= u_o \end{aligned}\right\} \qquad (1)$$

*Country Lane School, San Jose, Calif.

The forward Euler scheme is:

$$U_{n+1} = U_n + \Delta t\ f(U_n, t_n) \tag{2}$$

where Δt = time-step

$U_n = U(t_n)$ = the net function corresponding to $u(t_n)$

We may now set up a predictor-corrector formula as follows:

$$\hat{U} = U_n + \Delta t\ f(U_n, t_n) \quad \text{(predictor)} \tag{3}$$

$$U_{n+1} = (1 - \gamma)\hat{U} + \gamma[U_n + \Delta t\ f(\hat{U}, t_{n+1})] \quad \text{(corrector)} \tag{4}$$

Gamma is herein called a filtering parameter. Here we have restricted γ to

$$0 < \gamma \leq 1 \tag{5}$$

Thus, γ is essentially a convex parameter. If $\gamma = 0$, $U_{n+1} = \hat{U}$. The corrector is ineffective.

If the model (eq. (1)) is linear, then for a given step size it may be possible to find a γ such that the method could be effective. If the model is nonlinear, it may be locally linearized to obtain a γ (for a given step size) so that the method may remain effective. This stability analysis, done by Lomax (private communications), is our next topic of discussion.

STABILITY ANALYSIS

Let us consider a simple linear model:

$$\left.\begin{aligned} \frac{du}{dt} &= \lambda u \\ u(t_o) &= u_o \end{aligned}\right\} \tag{6}$$

If we combine equations (3) and (4) for this model we get

$$U_{n+1} = (1 + z + \gamma z^2)U_n \tag{7}$$

where $z = \lambda\Delta t$. Since λ may be complex, so is z. It is well known that for stability

$$|\sigma| \leq 1 \tag{8a}$$

where

$$\sigma = 1 + z + \gamma z^2. \tag{8b}$$

Using z as $z = x + iy$, a complex variable, when we plot the equality part of equations (8) in a complex plane we get a stability contour for a given value of γ. Some of these stability contours (drawn by Lomax) have been shown here in the figures 1-3 for $\gamma = 0.095$, 0.175, and 0.25, respectively. Outside these stability contours, the inequality (eqs. (8)) is not valid and as such the algorithm (eq. (7)) is unstable.

If we set $\gamma = 0$, we get the stability contour for the forward Euler method. This represents a unit circle passing through the origin and $x = -2$. (This is included in fig. 3). These contours show that knowing λ, which must be real (in this case), and Δt, we can choose a γ such that the method will remain stable. This scheme will be demonstrated in some simple examples. First we will do the truncation error analysis.

In order to make a simple analysis of truncation error we consider a simple linear model as follows:

$$\frac{du}{dt} = \lambda u + a\,, \qquad u(0) = u_o \tag{9}$$

The present scheme is

$$\hat{u} = u^n + zu^n + \Delta ta + (TE)^{(n)} \tag{10}$$

$$u^{n+1} = (1 - \gamma)\hat{u} + \gamma[u^n + z\hat{u} + a\Delta t + (TE)^{(n)}] \tag{11}$$

where $z = \lambda\Delta t$, $(TE)^{(n)} = 0(\Delta t^2)$ = truncation error at t_n. We know that $\lim_{\Delta t \to 0} (TE)^{(n)} = 0$. This gives the consistency of the forward Euler scheme (eq. (10)).

We will now examine how errors for the combined scheme behave as $\Delta t \to 0$.

Combining equations (10) and (11) we get

$$u^{n+1} = (1 + z + \gamma z^2)u^n + (1 + \gamma z)\Delta ta + (1 + \gamma z)(TE)^{(n)} \tag{12}$$

The term, neglected in the algorithm, is $(1 + \gamma z)(TE)^{(n)}$, which tends to zero as $\Delta t \to 0$.

We may study now the accuracy of the steady-state solution. Let us consider equation (9) again. The steady-state solution is

$$u = -a/\lambda \tag{13}$$

Neglecting the truncation error part in equation (12) we get

$$U^{n+1} = (1 + z + \gamma z^2)U^n + (1 + \gamma z)\Delta ta \tag{14}$$

where U^n = the net-function corresponding to u^n.

At the steady-state condition, $U^{n+1} = U^n$. Thus, from equation (14), at steady-state

$$U^n = -\frac{(1 + \gamma z)\Delta t}{z(1 + \gamma z)} \cdot a = -\frac{a}{\lambda} \qquad (z = \lambda\Delta t)$$

Thus, the present numerical scheme leads to the same analytical steady-state solution as the one given by the differential equation itself.

APPLICATIONS

Let us consider two simple applications, one linear and the other nonlinear.

Application No. 1:

$$\left.\begin{aligned} \frac{du}{dt} &= -100(u - \sin t) + \cos t \\ u(0) &= 0 \end{aligned}\right\} \tag{15}$$

The analytical solution is given by

$$u = \sin t \tag{16}$$

Here: $\lambda = -100$. If we choose $\Delta t = 0.1$, $z = \lambda\Delta t = -10.0$. From figure 1, $z = -10$ is within the stability contour for $\gamma = 0.095$. The computational results obtained here are given in table 1.

Obviously, if we set $\gamma = 0$, the method reduces to the forward Euler scheme which was unstable for $\Delta t = 0.1$. There is an interesting phenomenon here: if we reduce the step size and choose $\Delta t = 0.06$, then $z = \lambda\Delta t = -6.0$ which is <u>exterior</u> to the stability contour for $\gamma = 0.095$. Thus, the method must diverge. This has been verified computationally.

Application No. 2:

$$\left.\begin{aligned} \frac{du}{dt} &= -25(u - 1/u) \\ u(0) &= \sqrt{2} \end{aligned}\right\} \tag{17}$$

The analytical solution is given by

$$u(t) = [1 + \exp(-50t)]^{1/2} \tag{18}$$

Obviously, $\lim_{t\to\infty} u(t) = 1$ which gives the steady-state solution.

To obtain the value of λ we may linearize the model near $t = 0$. Here,

$$f(u,t) = -25(u - 1/u)$$

$$\lambda = \left.\frac{df}{du}\right]_{u=u_0} = -25(1 + 1/2) = -37.5$$

Choosing $\Delta t = 0.1$, $z = \lambda\Delta t = -3.75$. Now, according to figure 1, if we choose $\gamma = 0.095$, since $z = 3.75$ is outside the contour of stability, the method must diverge (verified computationally). But $z = -3.75$ is well within the contour of stability for $\gamma = 0.175$ (fig. 2). Computational results for this case are given in table 2.

BURGERS' EQUATION

One primary objective in this work is to see how effective the method could be with regard to the solution of mathematical models represented by nonlinear partial differential equations. A very common test problem for numerical methods is Burgers' equation, which is

$$\partial u/\partial t + u\ \partial u/\partial x = \nu\ \partial^2 u/\partial x^2. \tag{19}$$

This is a nonlinear, parabolic, partial differential equation and as such its solution has a special significance in applied mathematics. Since, in general, closed-form solutions of this equation that are subject to a given set of initial-boundary conditions are difficult to obtain, the equation is solved numerically.

We tested the effectiveness of our algorithm with regard to solution of equation (19) subject to three distinct sets of initial-boundary conditions. Some of our findings will be discussed now.

Let us approximate u_t by the forward Euler difference formula and the space derivatives u_x and u_{xx} by central differences. The explicit finite-difference predictor is then

$$\hat{U}_j = U_j^n + a\ U_j^n(U_{j-1}^n - U_{j+1}^n) + b(U_{j-1}^n - 2U_j^n + U_{j+1}^n) \tag{20}$$

where

$U_j^n = U(x_j, t_n) =$ the net function corresponding to u_j^n

$a = \Delta t/(2\Delta x)$, $\quad b = \nu\Delta t/\Delta x^2$

$\Delta t =$ time-step , $\quad \Delta x =$ net spacing

$\hat{U}_j =$ predicted value of U_j at t_{n+1}.

A corrector may now be set up as follows:

$$U_j^{n+1} = (1 - \gamma)\hat{U}_j + \gamma[U_j^n + a\ \hat{U}_j(U_{j-1}^{n+1} - \hat{U}_{j+1}) + b(U_{j-1}^{n+1} - 2\hat{U}_j + \hat{U}_{j+1})] \tag{21}$$

$$(j = 1,2,\ldots, \mathrm{JMAX} - 1)$$

where boundary conditions are given at $j = 0$ and $j = \mathrm{JMAX}$. As before if we now set $\gamma = 0$, then $U_j^{n+1} = \hat{U}_j$, which means the corrector is not in use. We may see now that equation (21) retains accuracy for the steady-state solution.

At the steady state, $U_j^n = U_j^{n+1} = \hat{U}_j$ for all j. Thus, if we replace in equation (20) $\hat{U}_j$ by U_j^n, we get the difference equation for the steady-state condition as

$$a\ U_j^n(U_{j-1}^n - U_{j+1}^n) + b(U_{j-1}^n - 2U_j^n + U_{j+1}^n) = 0 \tag{22}$$

In equation (21), if we replace U_j^{n+1} and $\hat{U}_j$ by U_j^n, we get precisely the same equation (22). Thus, accuracy of the solution is retained by the algorithm (eq. (21)).

Let us examine now how the truncation error will behave under this convex mapping analysis. Assuming that $u(x,t) \in C^{4,2}$ (four times continuously differentiable with respect to x and twice differentiable with respect to t) we have

$$\hat{u}_j = u_j^n + G_j(u^n) + (TE)_j^n \tag{23}$$

where

$$G_j(u^n) = a\, u_j^n(u_{j-1}^n - u_{j+1}^n) + b(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

$$(TE)_j^n = \Delta t\left\{\frac{\Delta t}{2}\,(u_{tt})_j^n + \frac{\Delta x^2}{6}\left[(uu_{xxx})_j^n - \frac{\nu}{2}\,(u_{xxxx})_j^n\right]\right\} + \ldots$$

As $\Delta t \to 0$ and $\Delta x \to 0$, $(TE)_j^n \to 0$.

Let $\hat{u} = (\hat{u}_1, \hat{u}_2, \ldots \hat{u}_{JMAX-1})^T$, $u = (u_1, u_2, \ldots u_{JMAX-1})^T$. The corrector gives

$$u_j^{n+1} = (1-\gamma)\hat{u}_j + \gamma[u_j^n + H_j(\hat{u},u^{n+1}) + (TE)_j^n] \tag{24}$$

Obviously $G_j(u^n)$ may be replaced by the functional $H_j(u^n,u^n)$ where

$$H_j(\hat{u},u^{n+1}) = a\,\hat{u}_j(u_{j-1}^{n+1} - \hat{u}_{j+1}) + b(\hat{u}_{j+1} - 2\hat{u}_j + u_{j-1}^{n+1})$$

From equations (23) and (24)

$$u^{n+1} = u^n + H(u^n,u^n) + \gamma[H(\hat{u},u^{n+1}) - H(u^n,u^n)] + (TE)^n \tag{25}$$

If we linearize and assume that the relation

$$H(\hat{u},u^{n+1}) - H(u^n,u^n) = A_n(\hat{u} - u^n) + B_n(u^{n+1} - u^n) \tag{26}$$

(A_n and B_n are square matrices of the order JMAX - 1) is valid at each time-step, we get, from equations (25) and (26),

$$u^{n+1} - u^n = H(u^n,u^n) + \gamma A_n(\hat{u} - u^n) + \gamma B_n(u^{n+1} - u^n) + (TE)^n \tag{27}$$

Since from equation (23) $\hat{u} - u^n = H(u^n,u^n) + (TE)^n$, from equation (27) we get

$$u^{n+1} - u^n = (I - \gamma B_n)^{-1}(I + \gamma A_n)[H^n + (TE)^n] \tag{28}$$

where $H^n \doteq H(u^n,u^n)$.

We assumed that $(I - \gamma B_n)$ is invertible for all n. Since as $\Delta t \to 0$ and $\Delta x \to 0$, $(TE)_j^n \to 0$, *equation (28) shows that the present scheme maintains its consistency property.* It is also clear that as steady state is approached,

$$(I - \gamma B_n)^{-1}(I + \gamma A_n)H^n = 0$$

giving $H_j^n = G_j(u^n) = 0$, which is true, provided $(I + \gamma A_n)$ is nonsingular.

In order to make some assessment of the stability properties, let us consider the linear Burgers' equation in the next section. An attempt will be made in the future to analyze the stability properties of the nonlinear Burgers' equation, using explicit D-Mappings, as discussed in reference 5.

LINEARIZED STABILITY ANALYSIS

Let us first consider the linearized stability properties of this method. We consider

$$\frac{\partial u}{\partial t} + c\,\frac{\partial u}{\partial x} = \nu\,\frac{\partial^2 u}{\partial x^2} \tag{29}$$

Then the present method may be expressed as

$$\hat{U}_j = U_j^n + a(U_{j-1}^n - U_{j+1}^n) + b(U_{j-1}^n - 2U_j^n + U_{j+1}^n) \quad \text{(predictor)} \tag{30}$$

$$U_j^{n+1} = (1-\gamma)\hat{U}_j + \gamma[U_j^n + a(U_{j-1}^{n+1} - \hat{U}_{j+1}) + b(U_{j-1}^{n+1} - 2\hat{U}_j + \hat{U}_{j+1})] \quad \text{(corrector)} \tag{31}$$

where

$$a = \frac{c\Delta t}{2\Delta x}, \qquad b = \frac{\nu\Delta t}{\Delta x^2} \tag{32}$$

The predictor may be expressed as

$$\hat{U}_j = (b+a)U_{j-1}^n + (1-2b)U_j^n + (b-a)U_{j+1}^n$$

In matrix notation, this becomes

$$\hat{U} = A \cdot U^n \tag{33}$$

where A is a tridiagonal band matrix, and may be expressed as

$$A = B(b+a,\ 1-2b,\ b-a) \tag{34}$$

The corrector may also be expressed as

$$U_j^{n+1} - \gamma(a+b)U_{j-1}^{n+1} = (1-\gamma-2b\gamma)\hat{U}_j + \gamma(b-a)\hat{U}_{j+1} + \gamma U_j^n$$

In matrix notation, we express this equation as

$$\Lambda U^{n+1} = \Theta\hat{U} + \Gamma U^n \tag{35}$$

where

$$\Lambda = B[-\gamma(b+a),\ 1,\ 0] \tag{36}$$

$$\Theta = B[0,\ 1-\gamma-2b\gamma,\ \gamma(b-a)] \tag{37}$$

$$\Gamma = \gamma I \tag{38}$$

Combining equations (33) and (35), we get

$$U^{n+1} = \Lambda^{-1}(\Theta \cdot A + \Gamma)U^n \tag{39}$$

The amplification matrix M is thus

$$M = \Lambda^{-1}(\Theta \cdot A + \Gamma) \tag{40}$$

Stability is obtained if M is a convergent matrix which is true if we can find a norm such that

$$||M|| < 1 \tag{41}$$

Some computational experiments have been conducted to find $\rho(M)$ numerically; $\rho(M)$ is the spectral radius of the amplification matrix M for some given Δt, Δx, ν, and $\gamma \in (0,1)$. (For $\Delta t = \Delta x = 0.05$, $\nu = 10^{-3}$, $\rho(M) < 1$ for $\gamma = 1.01$.) The results are given in table 3 ($c = 1$ in all the cases).

If we consider the predictor alone (eq. (30)), for stability we need $b \le 1/2$ and $a \le b$. These give, respectively,

$$\frac{\nu\Delta t}{\Delta x^2} \le 1/2 \quad \text{and} \quad \frac{\Delta x}{\nu} \le 2 \quad (c = 1) \tag{42}$$

If we consider $\Delta t = \Delta x = 0.05$, $\nu = 0.01$, the first criterion is satisfied; however, $\Delta x/\nu = 5 > 2$.

The predictor alone is unstable, whereas the present predictor-corrector is stable (first row in table 3). Also if we consider $\Delta t = 0.05$, $\Delta x = 0.025$, $\nu = 0.01$, both inequalities (eq. (42)) are violated.

Again, whereas the predictor itself is unstable when it is combined with the corrector for $\gamma = 0.28$ (table 3), the method becomes stable. Table 3 has been formed by computing $\rho(M)$, where M is given in equation (40) for a given Δt, Δx, and ν, and choosing $0 \le \gamma \le 1$ (in general). Table 3 was done by Strate (private communication).

Fourier stability analysis will be done in the future.

COMPUTATIONAL RESULTS

At each t_n, we first computed U_j for $j = 1,2,\ldots,(JMAX - 1)$, using the predictor (eq. (20)); then in a SUBROUTINE we corrected all the predicted values by using equation (21), where the most recent corrected values were used.

The code and its description appeared in reference 8. It is written in Extended BASIC for the TRS-80 color computer made by Radio Shack by the second author. The boundary conditions were kept fixed at $j = 0$ and $j = JMAX$.

Case 1:

$$u(x,0) = \sin \pi x$$

$$u(0,t) = u(1,t) = 0 \tag{43}$$

This represents a sine wave which decays as $t \to \infty$. The analytical solution as given in references 2 and 7 is

$$u(x,t) = 2\pi\nu\ S_1/S_2 \tag{44}$$

where

$$S_1 = \sum_{n=1}^{\infty} n \exp(-n^2\pi^2\nu t)A_n \sin(n\pi x) \tag{45}$$

$$S_2 = \sum_{n=0}^{\infty} \exp(-n^2\pi^2\nu t)A_n \cos(n\pi x) \tag{46}$$

$$A_0 = \int_0^1 \exp[(\cos \pi x - 1)/2\pi\nu]dx \tag{47}$$

$$A_n = 2\int_0^1 \exp[(\cos \pi x - 1)/2\pi\nu] \cos(n\pi x)dx \tag{48}$$

It may be noticed that for all n and $\nu \neq 0$ the integrals are convergent and $\lim_{n\to\infty} A_n = 0$. The rate of convergence slows down as ν gets smaller. Thus, for small values of ν, it is extremely difficult to compute $u(x,t)$ using the analytical expression (eq. (44)).

The computational results obtained by the present scheme have been plotted in figure 4. Here $\nu = 0.01$, $\gamma = 0.25$, $\Delta x = 0.05$, and $\Delta t = 0.1$. The time = 2.5 sec. Evidently, as $t \to \infty$, $u(x,t) \to 0$. (The method failed if we set $\gamma = 0$.) The sharp overshooting near $x = 1$ has been caused by the boundary conditions. Such solution patterns are quite well known; they are not time-accurate.

Case 2:

$$\left.\begin{aligned} u(x,0) &= 1 \qquad 0 \le x \le 0.1 \\ &= 0 \qquad x > 0.1 \end{aligned}\right\} \tag{49}$$

This represents a moving shock. Here we chose $\gamma = 0.75$ and $\nu = 0.01$. Figure 5 describes the solution. Because of the nonconservative form of the algorithm, locations of the shock were not traced accurately. As expected, overshooting was found at $x = 1$. It was not, however, propagated upstream. With $\gamma = 0$ the oscillations were unbounded, which is true for Euler's forward scheme by eigenvalue analysis. In figure 6 an appropriate scale was used to show some of these oscillations.

Case 3:

$$\left.\begin{aligned} u(x,0) &= \phi(x,0) \\ u(0,t) &= \phi(0,t) \\ u(1,t) &= \phi(1,t) \end{aligned}\right\} \tag{50}$$

where

$$\phi(x,t) = \frac{0.1e^{-A} + 0.5e^{-B} + e^{-C}}{e^{-A} + e^{-B} + e^{-C}} \tag{51}$$

$$\left.\begin{aligned} A &= (0.5/\nu)(x - 0.5 + 4.95t) \\ B &= (2.5/\nu)(x - 0.5 + 0.075t) \\ C &= (5/\nu)(x - 0.375) \end{aligned}\right\} \tag{52}$$

This model is discussed in references 3 and 4. It represents the motions of two shocks which merge into one at the steady state. With $\nu = 0.022$ and $\gamma = 0.25$, U_j's were computed for up to 15 time-steps and plotted in figure 7. Similar profiles were given in references 4 and 5. With $\gamma = 0$, the method failed. Oscillations are shown in figure 8.

In order to remove instabilities, $\partial u/\partial x$ was approximated by upwind differencing (MacCormack). The predictor is now

$$\hat{U}_j = U_j^n + a\ U_j^n(U_{j-1}^n - U_j^n) + b(U_{j-1}^n - 2U_j^n + U_{j+1}^n) \qquad \text{(upwind-predictor)} \tag{53}$$

The corrector is

$$U_j^{n+1} = (1 - \gamma)\hat{U}_j + \gamma[U_j^n + a\ \hat{U}_j(U_{j-1}^{n+1} - \hat{U}_j) + b(U_{j-1}^{n+1} - 2\hat{U}_j + \hat{U}_{j+1})] \tag{54}$$

where $a = \Delta t/\Delta x$. Other parameters are the same as discussed before.

Applying this formula to case 1, with $\Delta x = 0.05$, $\Delta t = 0.1$, $\nu = 10^{-6}$, and $\gamma = 0.4$ we obtain figure 9. Instabilities were removed; however, with $\gamma = 0$, the method failed (fig. 10).

The same was true for cases 2 and 3. The motion of the shock is shown in figure 11 for case 2, in which $\Delta x = 0.05$, $\Delta t = 0.1$, $\nu = 0.01$, and $\gamma = 0.25$. There was no overshooting. However, when $\gamma = 0$, the method failed after six time-steps; this is shown in figure 12.

Figure 13 shows no overshooting for case 3 ($\Delta x = 0.05$, $\Delta t = 0.1$, $\nu = 0.022$, $\gamma = 0.15$, number of time steps = 22). Figure 14 shows that the method collapsed with $\gamma = 0$.

In figure 15 it is shown that even when $\Delta t = 4\Delta x$ ($\Delta t = 0.2$, $\Delta x = 0.05$), the method may still be effective for representing the motion of the shock. Figure 16 shows somewhat interesting results for case 2. Here $\Delta x = 0.05$, $\Delta t = 4\Delta x$, $\nu = 0.01$, and $\gamma = 0.355$. Numerical solutions initially showed some strong oscillations. Later they were damped out. Although the results were not time-accurate, the converged form of solution was correct.

In figures 17-19, it is shown that for the present explicit predictor-corrector scheme, Δt, which is much larger than Δx, may be used for small values of ν if we use the upwind predictor (eq. (27)). It has also become clear that although steady-state solutions are correct here, intermediary solutions at various time-steps are not acceptable.

Larger Δt does not necessarily mean a faster rate of convergence to steady state; the latter depends on how fast residuals converge to zero. This has been done in the past with regard to this method for solution of Burgers' equation expressed in the potential form. Details on this may be found in reference 9.

DISCUSSION

The primary objective of the predictor-corrector formulas discussed here is to obtain the steady-state solution. The analysis given in the Stability Analysis section shows that this objective is fulfilled.

The method is only first-order accurate unless $\gamma = 1/2$. Stability analysis for the linear Burgers' equation shows that we may use Δt much larger than Δx with a suitable γ, such that the numerical solution may remain stable (table 3). Computationally, this has been found to be true for the nonlinear Burgers' equation. In the future, nonlinear error analysis done in reference 5 using D-mappings will be applied to this problem.

More analyses and computational experiments are needed in order to make a more realistic assessment of the present predictor-corrector schemes for numerical solution of nonlinear models.

APPENDIX

THE CODE IN TRS-80 COLOR-EXTENDED BASIC

```
1Ø REM--------------------------------------------
11 REM
2Ø REM   BURGERS' EQUATION
25 REM
3Ø REM   DU/DT+U*DU/DX=NU*D2U/DX2
35 REM
4Ø REM   A PREDICTOR-CORRECTOR METHOD
45 REM
5Ø REM   U=VALUE OF U AT THE RECENT TIME STEP
55 REM   UØ=VALUE OF U AT THE PREVIOUS TIME STEP
57 REM   G=A CONVEX PARAMETER
6Ø REM   JMAX=FIELD SIZE
61 REM   NTSTEPS=NO. OF TIME STEPS
63 REM   K=LEVEL OF ITERATION
89 REM
9Ø REM--------------------------------------------
99 REM
1ØØ DIM U(51),UØ(51)
1Ø9 REM
11Ø REM-------------------------------------------
115 REM   INPUT PARAMETERS
117 REM-------------------------------------------
12Ø DX=Ø.Ø5:DT=Ø.1:JMAX=21:NTSTEPS=2ØØ
125 NU=Ø.Ø1:PI=3.1416
13Ø A=Ø.5*DT/DX:B=NU*DT/(DX*DX)
132 K=Ø
14Ø REM-------------------------------------------
142 REM   CHOOSE G
145 REM-------------------------------------------
15Ø G=Ø.25
16Ø REM-------------------------------------------
162 REM   SET INITIAL & BOUNDARY CONDITIONS
165 REM-------------------------------------------
17Ø FOR J=1 TO JMAX:X=(J-1)*DX:U(J)=SIN(PI*X):NEXT J
179 REM
18Ø REM-------------------------------------------
182 REM   RECORD PREVIOUS TIME STEP SOLUTIONS
185 REM-------------------------------------------
19Ø FOR J=1 TO JMAX:UØ(J)=U(J):NEXT J
198 REM
2ØØ REM-------------------------------------------
212 REM   ITERATIONS START
215 REM-------------------------------------------
22Ø K=K+1
229 REM
23Ø REM-------------------------------------------
232 REM   EULER FORWARD:PREDICTOR
234 REM-------------------------------------------
24Ø FOR J=2 TO JMAX-1
```

```
245 U(J)=UØ(J)+A*UØ(J)*(UØ(J-1)-UØ(J+1))+B*(UØ(J-1)
        -2*UØ(J)+UØ(J+1))
25Ø NEXT J
26Ø GOSUB 1ØØØ
27Ø REM----------------------------------------
272 REM OUTPUT
275 REM----------------------------------------
28Ø PRINT "AT TIME STEP=";K
29Ø FOR J=1 TO JMAX:X=(J-1)*DX
295 PRINT X,U(J)
3ØØ NEXT J
32Ø IF K<NTSTEPS THEN 19Ø
8ØØ END
9ØØ REM----------------------------------------
9Ø2 REM   CONVEX CORRECTOR:CHARLIE
9Ø5 REM----------------------------------------
1ØØØ FOR J=2 TO JMAX-1
1Ø1Ø TEMP=(1.-G)*U(J)
1Ø3Ø V=UØ(J)+A*U(J)*(U(J-1)-U(J+1))+B*(U(J-1)-2*U(J)
        +U(J+1))
1Ø5Ø U(J)=TEMP+G*V
1Ø6Ø NEXT J
1Ø9Ø RETURN
```

REFERENCES

1. Ames, W.: Numerical Methods for Partial Differential Equations. Barnes & Noble, Inc., New York, 1969.

2. Dey, S. K.: Perturbed Iterative Solution of Nonlinear Equations with Applications to Fluid Dynamics. J. Comp. Appl. Math., vol. 3, no. 1, 1977.

3. Seider, W. D.; White, C. W., III; and Prokopakis, G. J.: Stiff Ordinary Differential Equations in Chemical Process Analysis. Proceedings of the International Conference on Stiff Computations, Apr. 12-14, 1982, Park City, Utah.

4. Schiesser, W. E.: Some Characteristics of ODE Problems Generated by the Numerical Method of Lines. Proceedings of the Interational Conference on Stiff Computations. Apr. 12-14, 1982, Park City, Utah.

5. Dey, S. K.: Nonlinear Discretization Errors in Partial Difference Equations. BIT, vol. 20, 1980.

6. Dey, Charlie: Simulation of Geometrical Curves on a Microcomputer. Simulation, Mar. 1982.

7. Miller, E. L.: Predictor-Corrector Studies of Burgers Model of Turbulent Flow. U. of Delaware, Newark, Del., 1966.

8. Dey, Charlie; and Dey, S. K.: A Microcomputer Code on Burgers' Equation. Proceedings of International Conference on Mini- and Microcomputers. May 16, 1983, San Francisco, Calif.

9. Dey, Charlie; and Dey, S. K.: Explicit Finite Difference Predictor and Convex Corrector with Applications to Hyperbolic Partial Differential Equations, Computers and Mathematics with Application, vol. 9, no. 3, 1983.

TABLE 1.- COMPARISON BETWEEN ANALYTICAL AND COMPUTATIONAL VALUES OF U: $\nu = 0.095$, $\Delta t = 0.1$

Time	U (Analytical)	U (Computed)
2.5	0.598472144	0.597493063
5.0	-0.958924274	-0.957553061
7.5	0.937999977	0.936781981
10.0	-0.544021111	-0.543440746
12.5	-0.0663218955	-0.0660338101
15.0	0.650287837	0.649245878

TABLE 2.- COMPARISON BETWEEN ANALYTICAL AND COMPUTATIONAL VALUES OF U: $\nu = 0.175$, $\Delta t = 0.1$

Time	U (Analytical)	U (Computed)
0.1	1.00336332	1.12706991
0.5	1.00	1.04808666
0.9	1.00	1.00621156
1.3	1.00	1.0001474
1.7	1.00	1.00000293
2.1	1.00	1.00000006

TABLE 3.- STABILITY FOR LINEAR BURGERS' EQUATION

c = 1.0			ν = 0.01	
Case	Δt	Δx	ρ_{MIN}	GAMMA
1	0.05	0.05	0.6741081217	0.44
2	0.05	0.025	0.2634833436	0.28
3	0.1	0.05	0.5591445292	0.39
4	0.1	0.025	0.4932129687	0.22
5	0.2	0.05	0.8777235062	0.32
6	0.2	0.025	0.731106526	0.13
c = 1.0			ν = 0.1	
1	0.05	0.05	0.8349436353	0.18
2	0.05	0.025	3.8000485810	0.05
3	0.1	0.05	1.5907523689	0.1
4	0.1	0.025	8.0442946351	0.03
5	0.2	0.05	3.7391180937	0.05
6	0.2	0.025	24.990423418	0.01
c = 1.0			ν = 0.001	
1	0.05	0.05	0.95710316836	1.010
2	0.05	0.025	0.8299183481	0.77
3	0.1	0.05	0.8320843664	0.89
4	0.1	0.025	1.7710418247	0.37
5	0.2	0.05	1.9420702213	0.38
6	0.2	0.025	4.5407595319	0.18

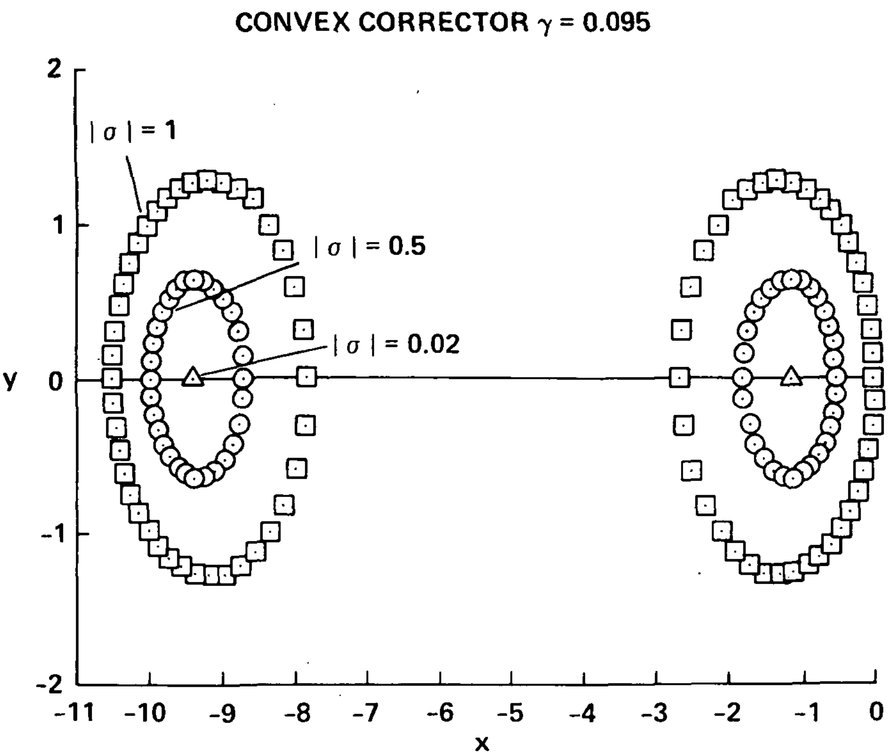

Figure 1.- Stability contours, convex corrector: γ = 0.095.

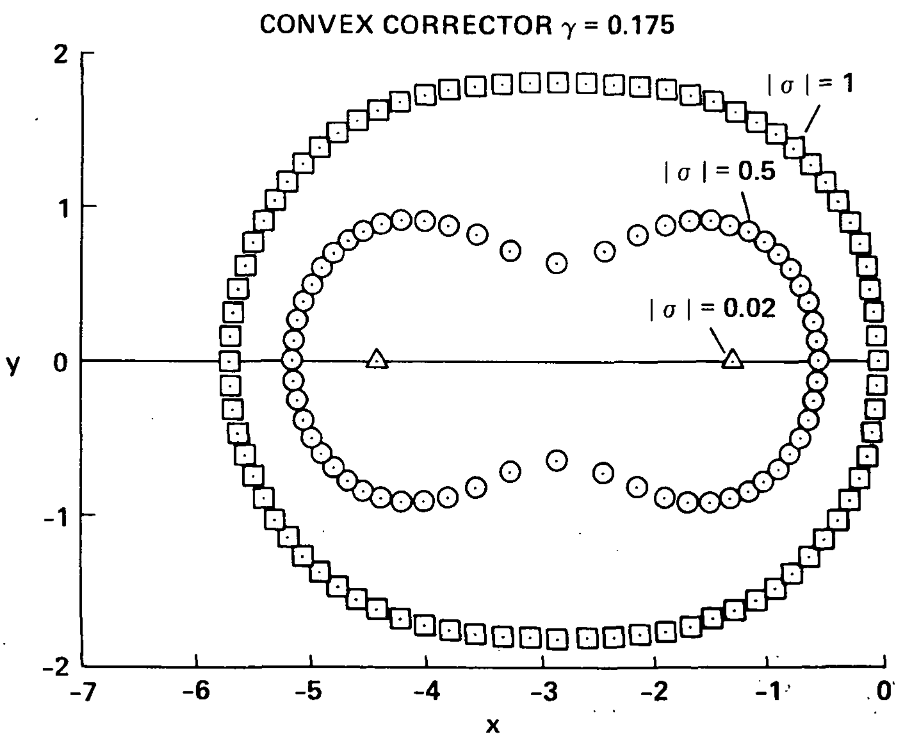

Figure 2.- Stability contours, convex corrector: γ = 0.175.

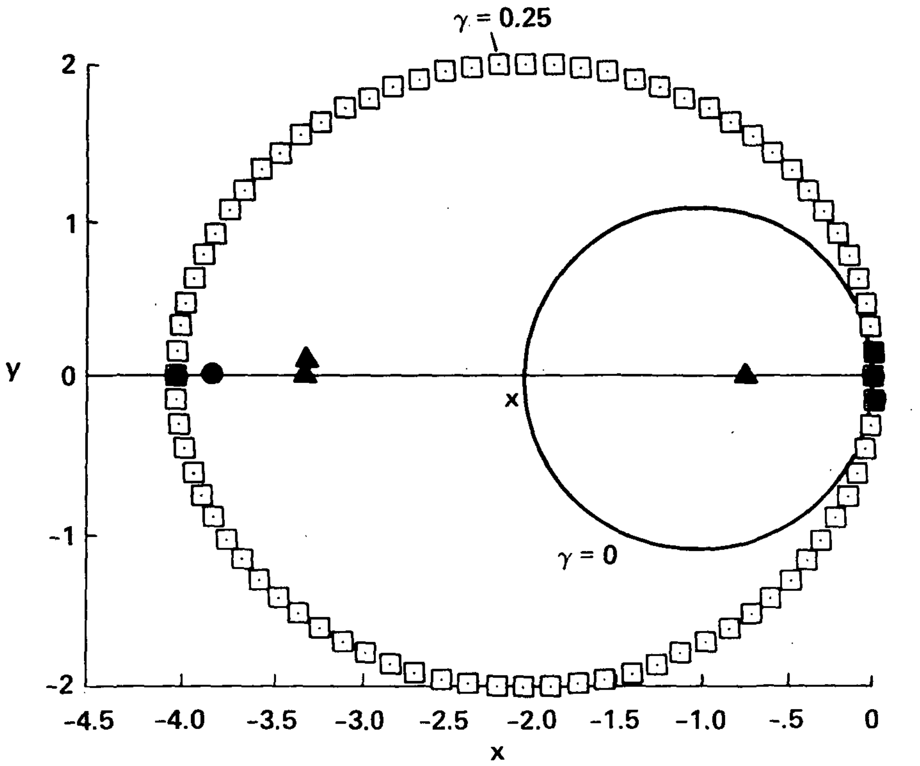

Figure 3.- Stability contours for forward Euler's method ($\gamma = 0$) and that for the present scheme with $\gamma = 0.25$, $|\sigma| = 1$.

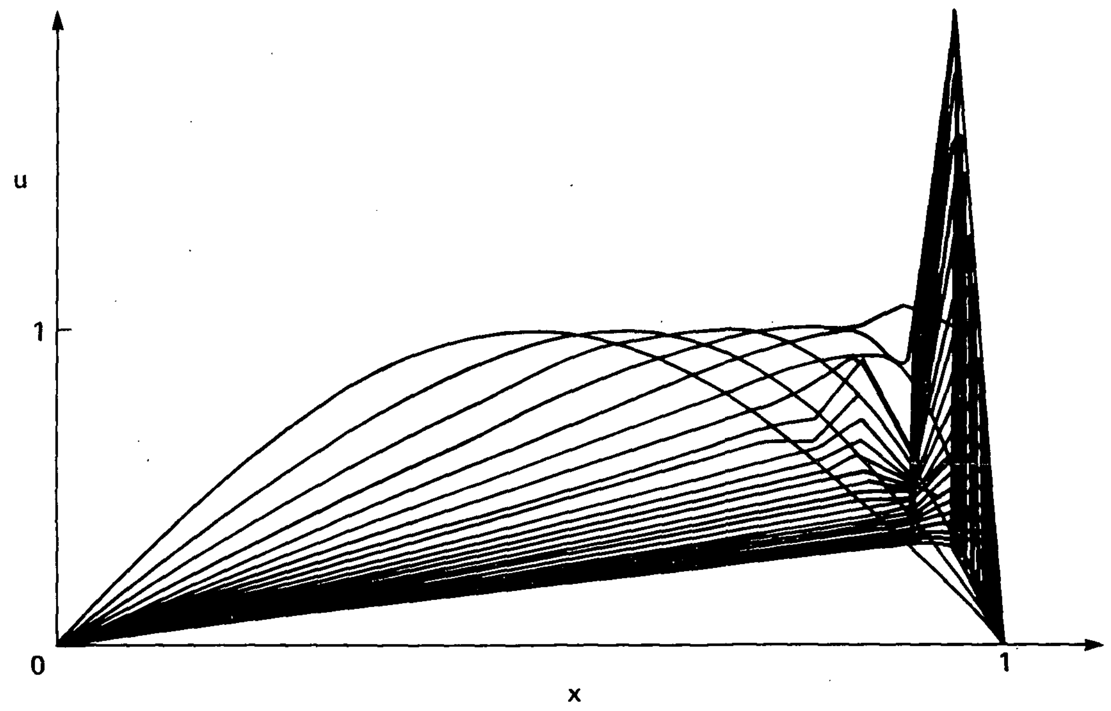

Figure 4.- Velocity profiles (case 1): $\nu = 0.01$, $\gamma = 0.25$, $\Delta x = 0.05$, $\Delta t = 0.1$.

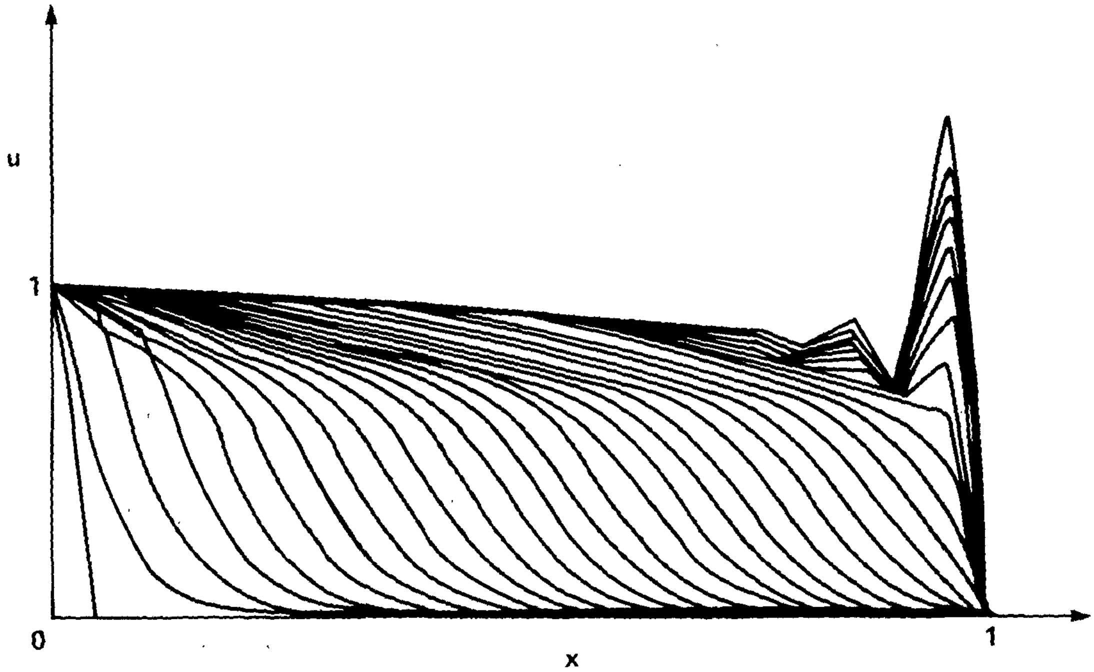

Figure 5.- Velocity profiles (case 2): $\nu = 0.01$, $\gamma = 0.75$, $\Delta x = 0.05$, $\Delta t = 0.1$.

Figure 6.- Method failed: $\gamma = 0$, other parameters are the same as those in fig. 5.

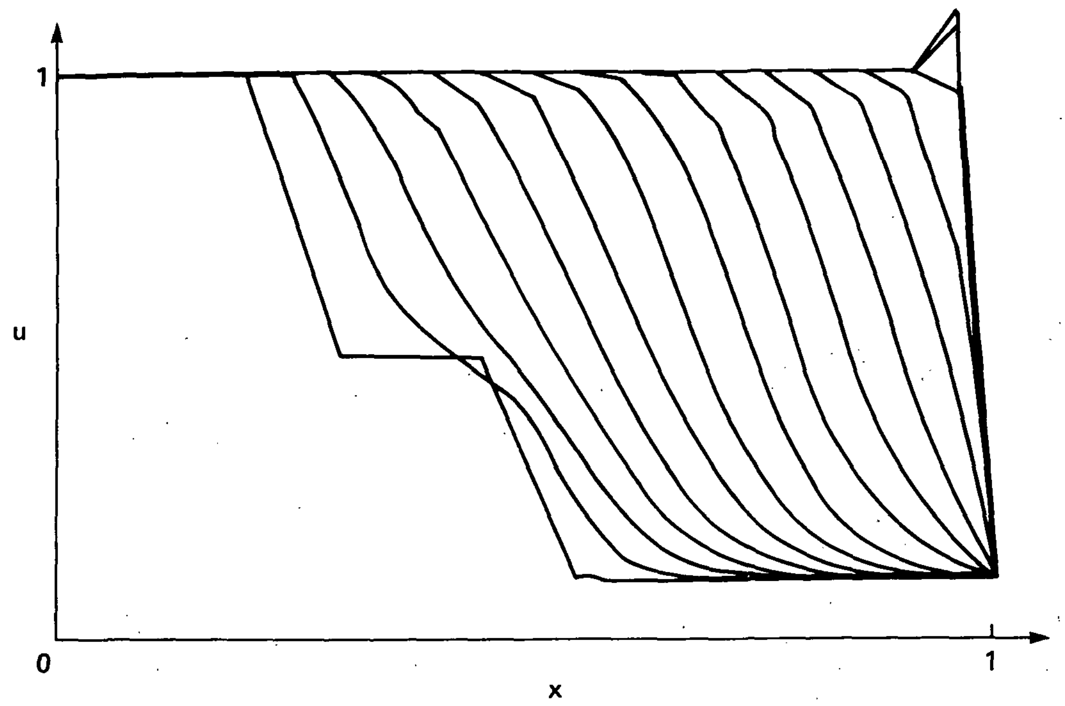

Figure 7.- Velocity profiles (case 3): $\nu = 0.022$, $\gamma = 0.25$, $\Delta x = 0.05$, $\Delta t = 0.1$.

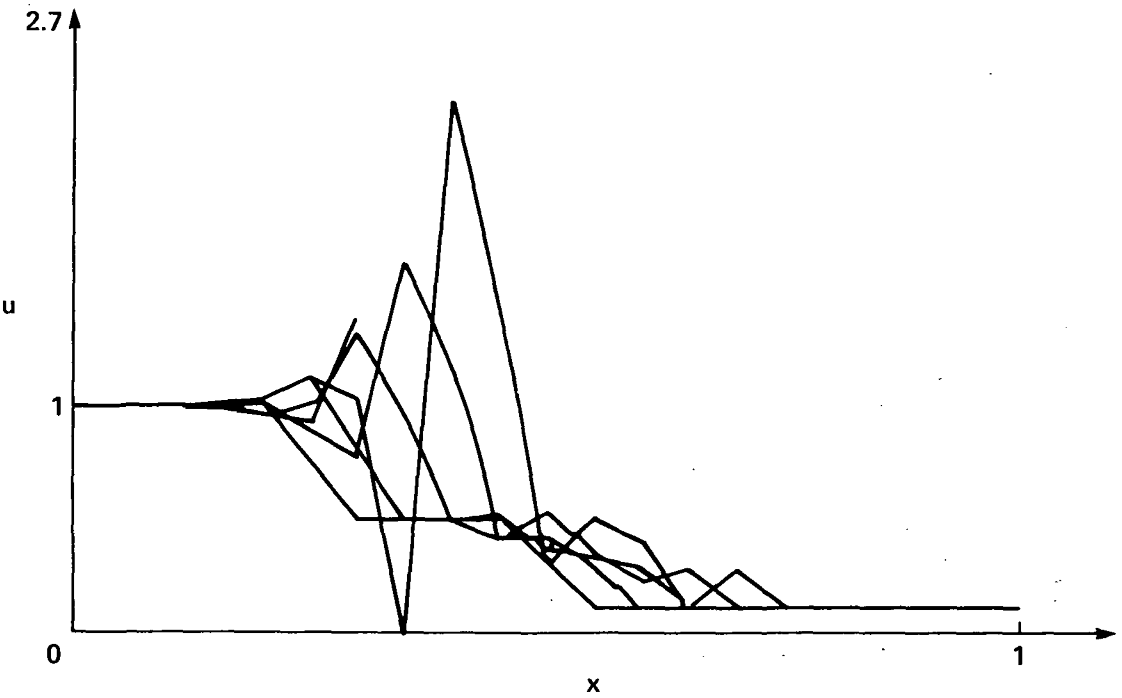

Figure 8.- Method failed: $\gamma = 0$, other parameters are the same as those in fig. 7.

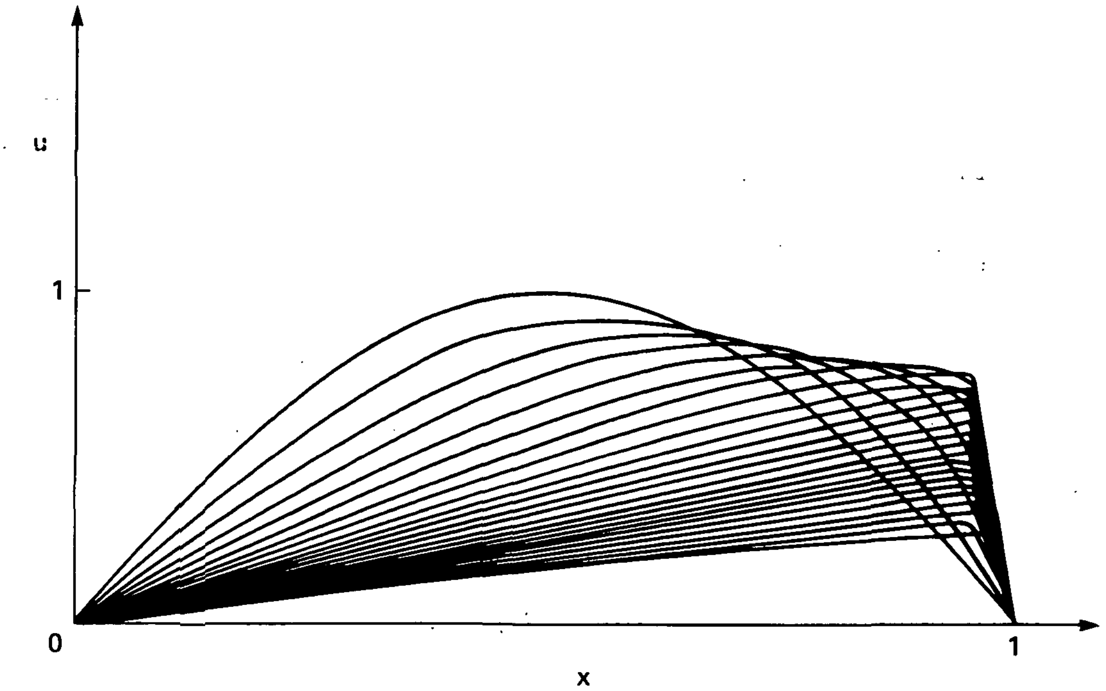

Figure 9.- Velocity profiles using upwind differencing (eqs. (53) and (54)) for case 1: $\nu = 10^{-6}$, $\gamma = 0.4$, $\Delta x = 0.05$, $\Delta t = 0.1$.

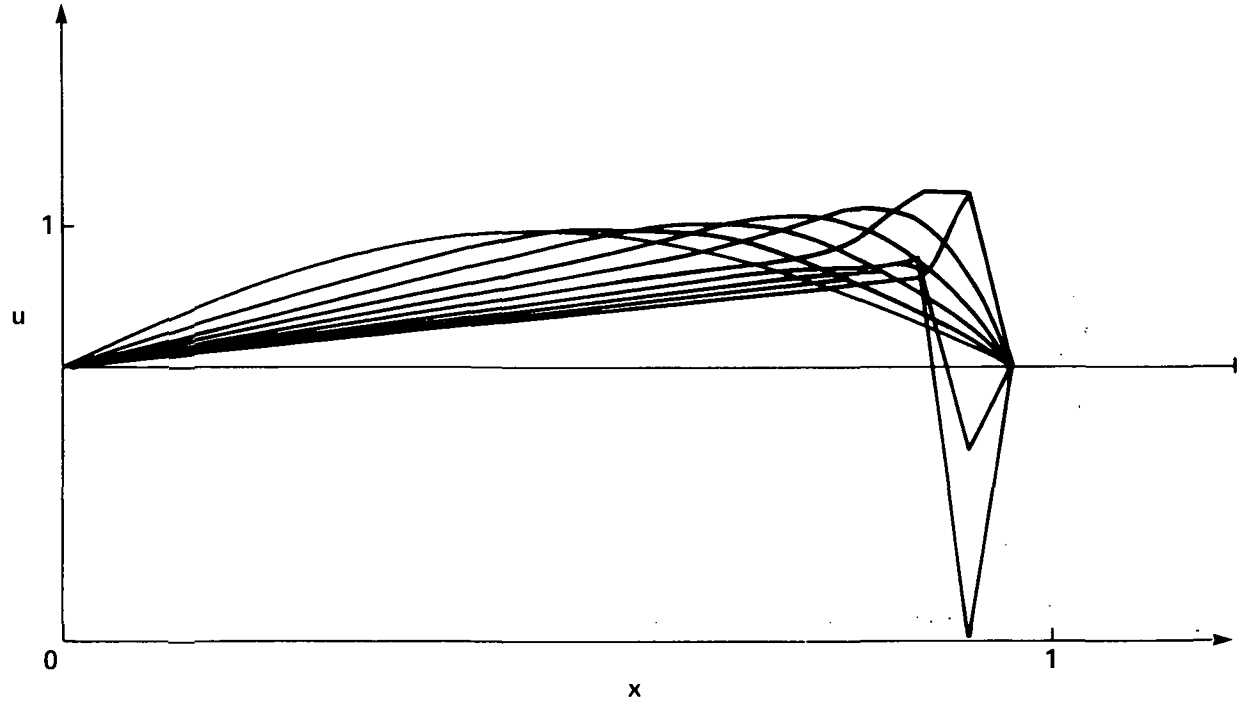

Figure 10.- Method failed (eq. (53)): $\gamma = 0$, other parameters are the same as those in fig. 9.

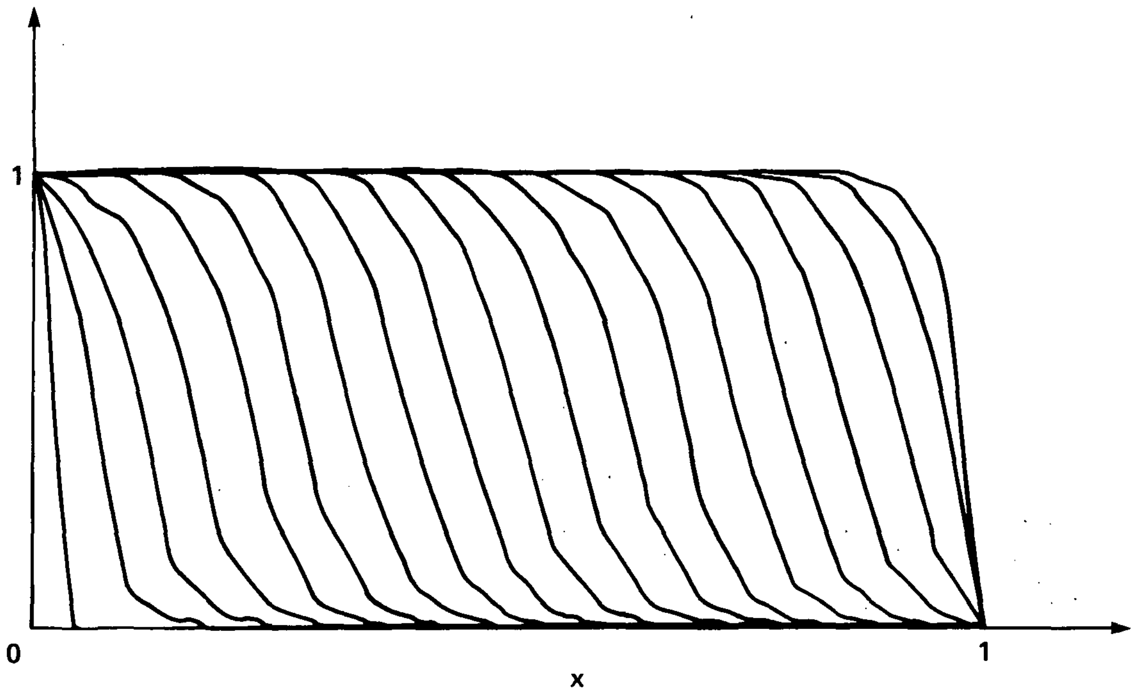

Figure 11.- Velocity profiles using upwind differencing for case 2: $\nu = 0.01$, $\gamma = 0.25$, $\Delta x = 0.05$, $\Delta t = 0.1$.

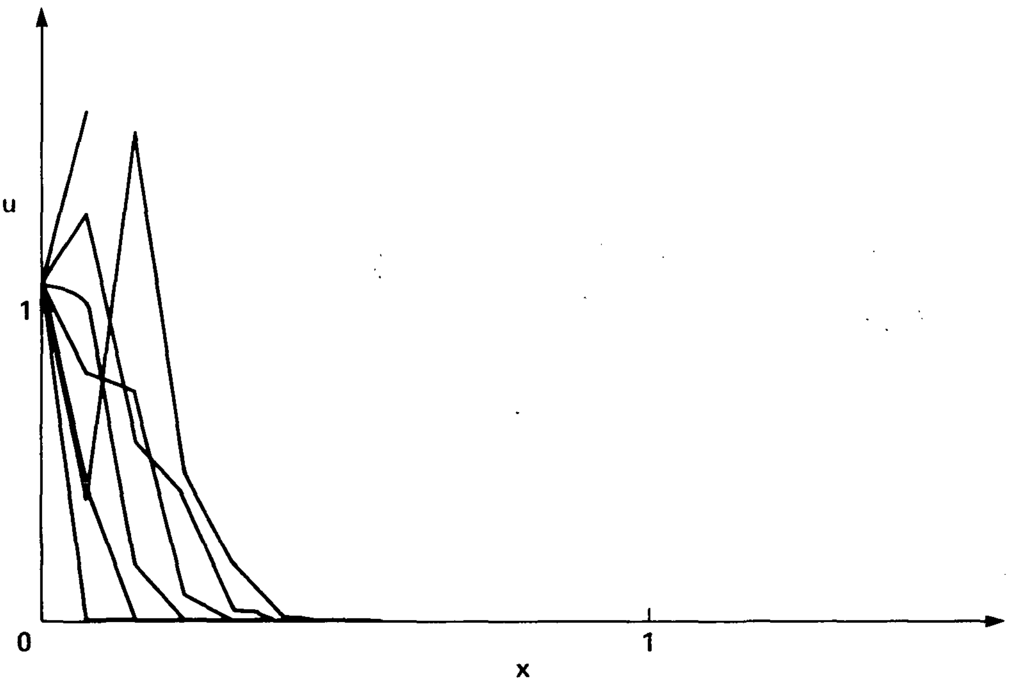

Figure 12.- Velocity profiles upwind differencing for case 2: $\gamma = 0$, other parameters are the same as those in fig. 11.

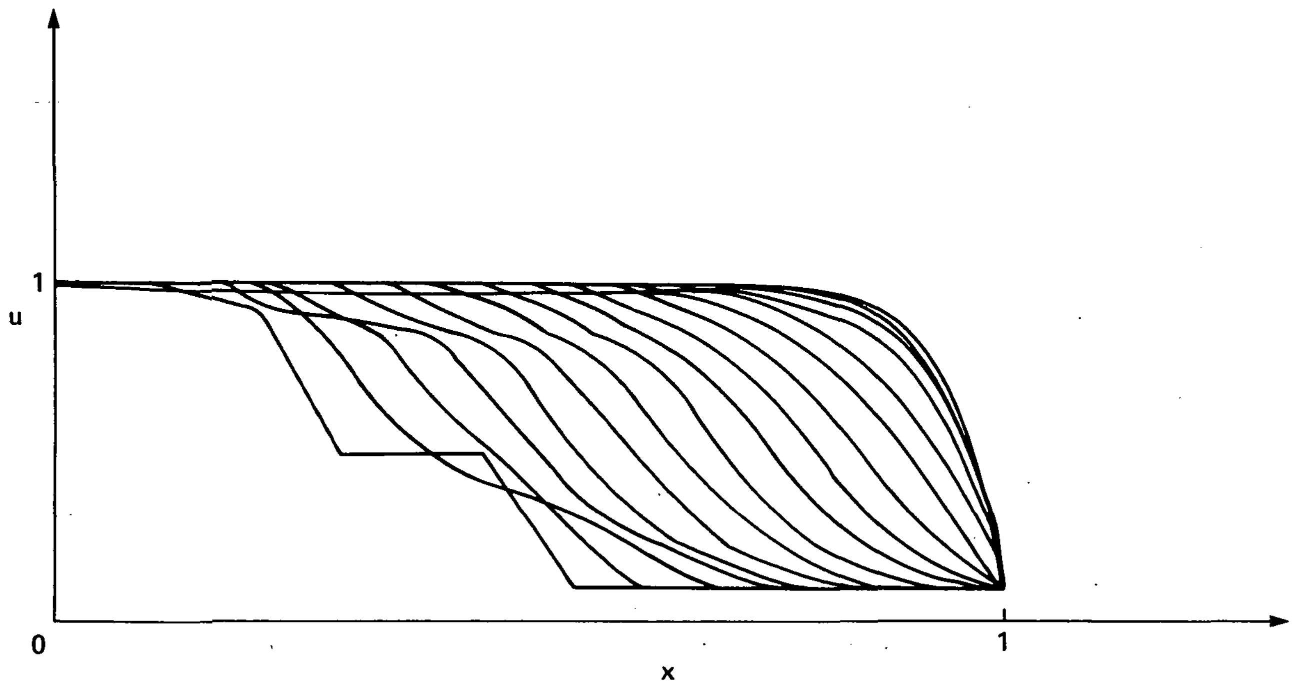

Figure 13.- Velocity profiles using upwind differencing for case 3: $\nu = 0.022$, $\gamma = 0.15$, $\Delta x = 0.05$, $\Delta t = 0.1$.

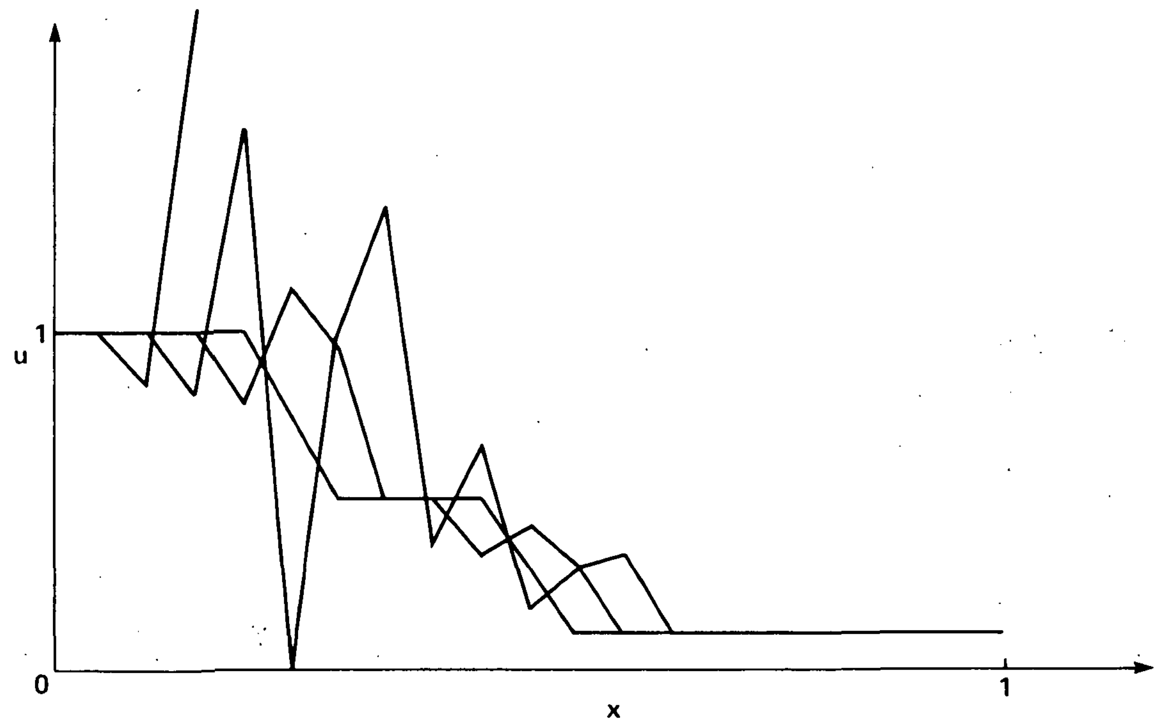

Figure 14.- Velocity profiles using upwind differencing for case 3: $\gamma = 0$, other parameters are the same as those in fig. 13.

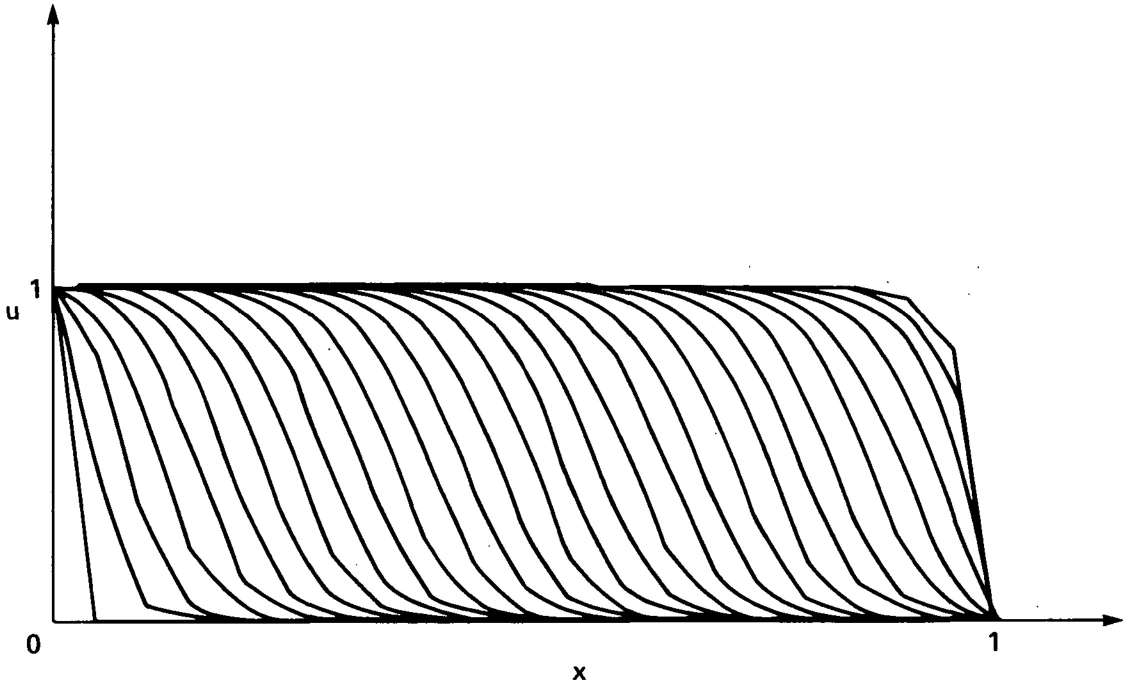

Figure 15.- Velocity profiles using upwind differencing for case 2: $\nu = 0.01$, $\gamma = 0.15$, $\Delta x = 0.05$, $\Delta t = 0.2$.

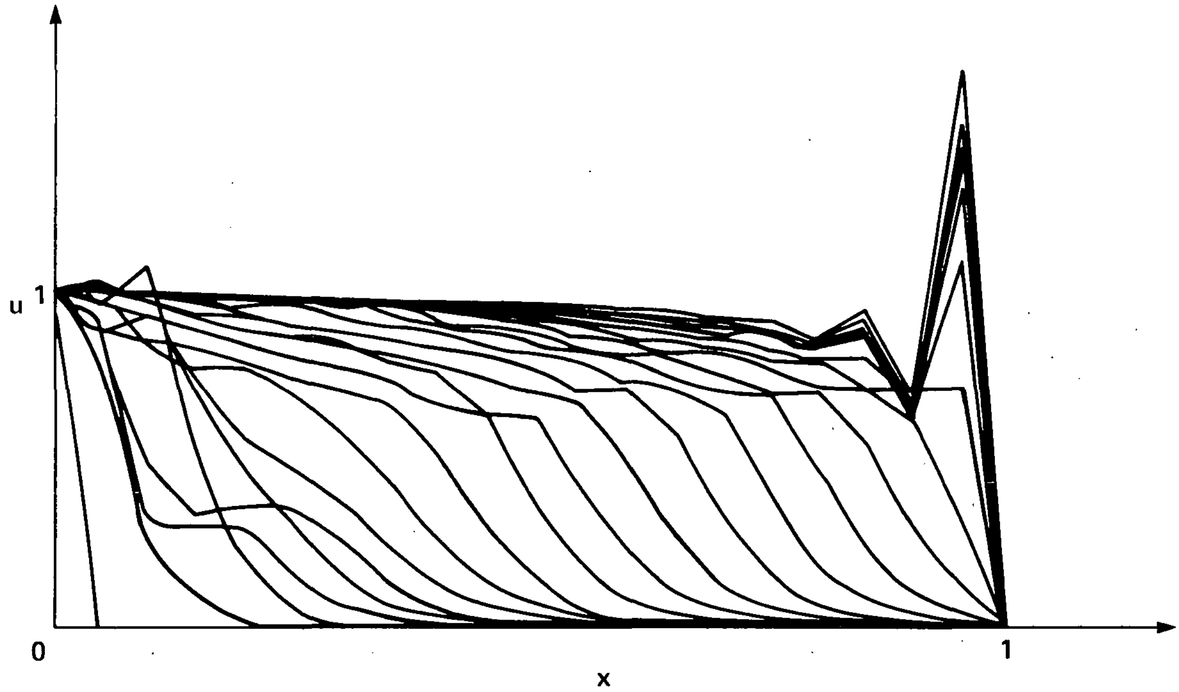

Figure 16.- Velocity profiles using central differencing for case 2: $\nu = 0.01$, $\gamma = 0.355$, $\Delta x = 0.05$, $\Delta t = 0.2$.

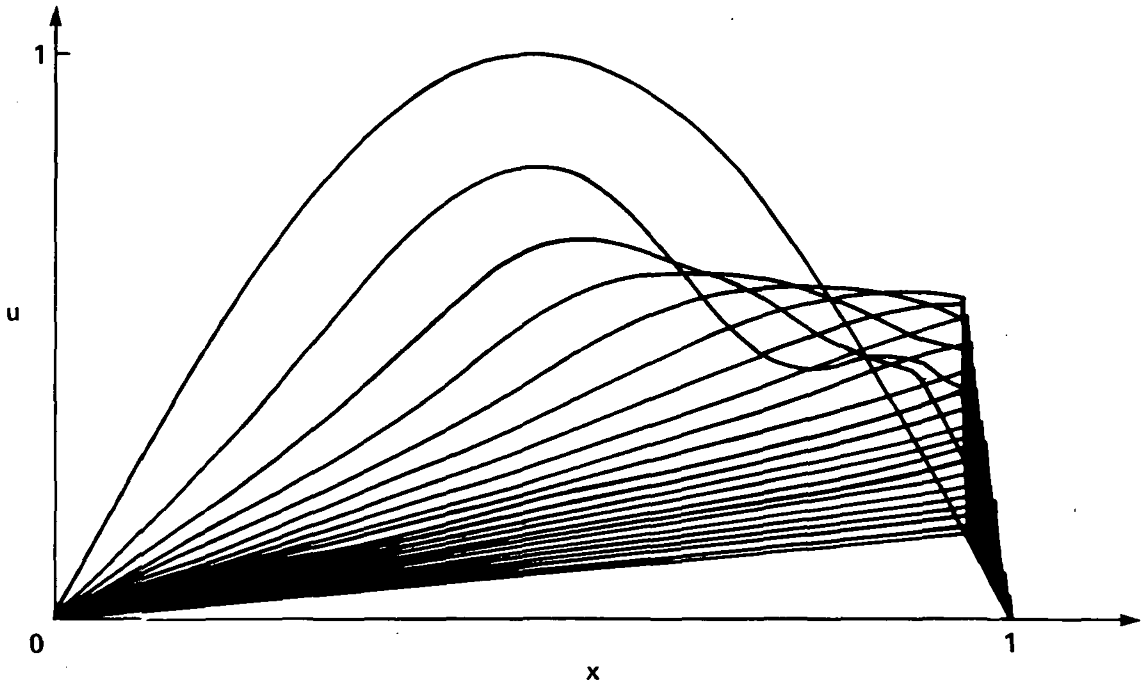

Figure 17.- Velocity profiles using upwind differencing for case 1: $\nu = 0.001$, $\gamma = 0.25$, $\Delta x = 0.05$, $\Delta t = 0.2$.

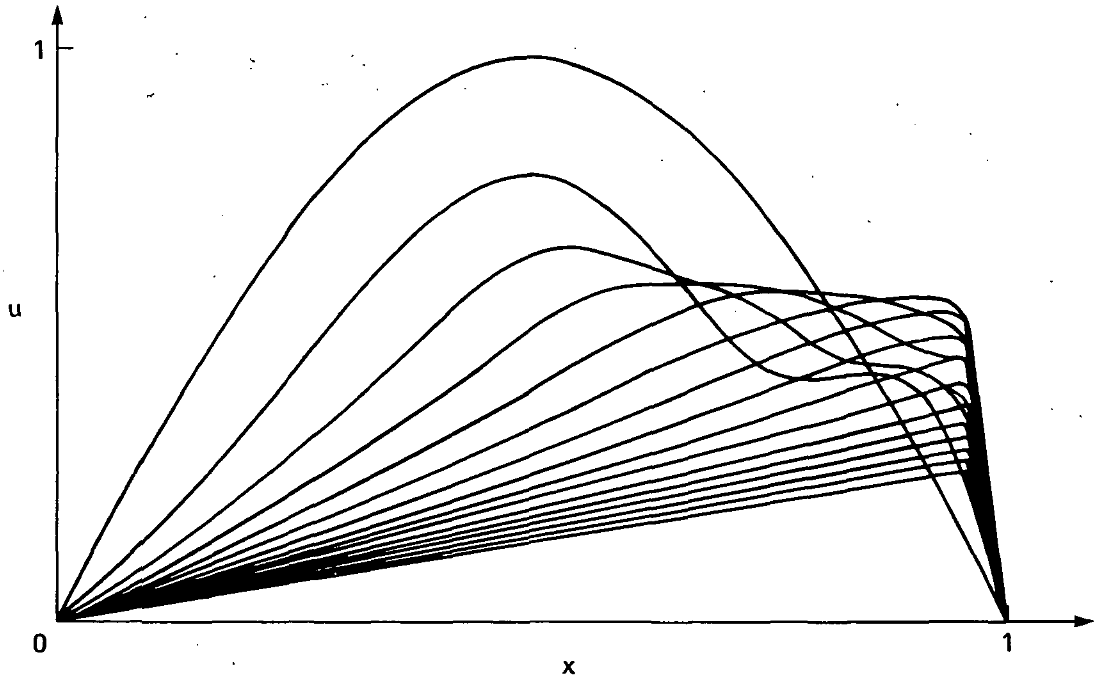

Figure 18.- Velocity profiles using upwind differencing for case 1: $\nu = 0.00001$, $\gamma = 0.25$, $\Delta x = 0.05$, $\Delta t = 0.2$, number of time steps = 30.

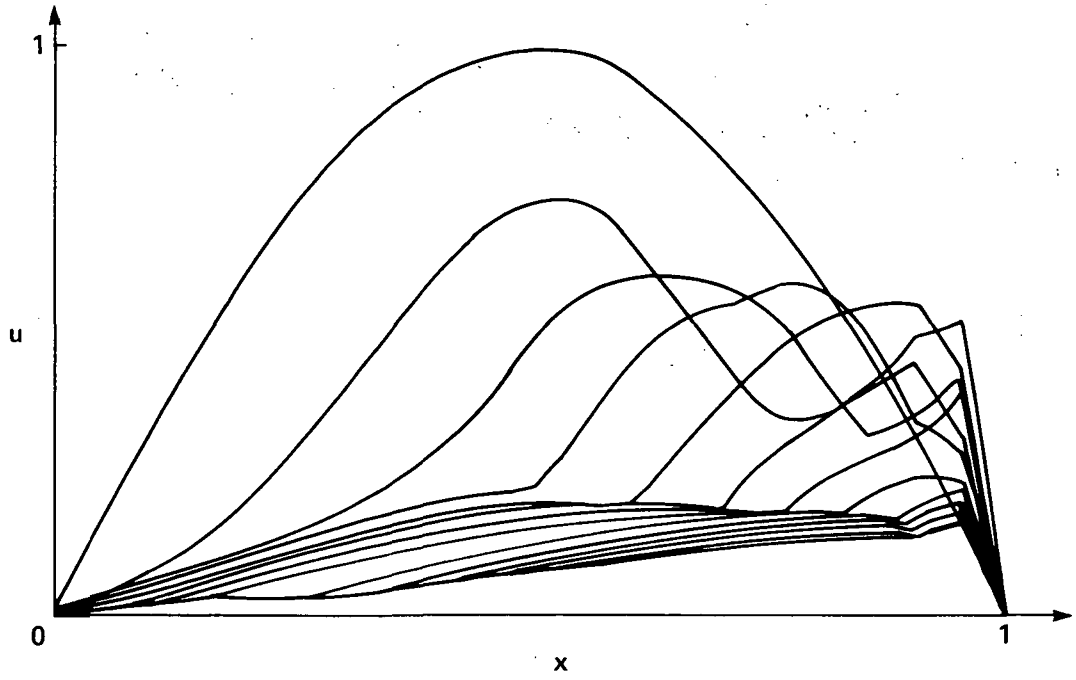

Figure 19.- Velocity profiles using upwind differencing for case 1: $\nu = 0.00001$, $\gamma = 0.152$, $\Delta x = 0.05$, $\Delta t = 0.3$, number of time steps = 40.

1. Report No. NASA TM 84402	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle An Explicit Predictor-Corrector Solver with Applications to Burgers' Equation		5. Report Date September 1983
		6. Performing Organization Code ATP
7. Author(s) Suhrit K. Dey and Charlie Dey*		8. Performing Organization Report No. A -9475
		10. Work Unit No. T-6465
9. Performing Organization Name and Address Ames Research Center, Moffett Field, CA. 94035 *Country Lane School, San Jose, CA		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC, 20546		14. Sponsoring Agency Code 505-31-01-01-00-21

15. Supplementary Notes

Point of contact: Howard Lomax, Ames Research Center, MS 202A-1, Ames Research Center, Moffett Field, CA. 94035 (415) 965-5124 or FTS 448-5124

16 Abstract

Forward Euler's explicit, finite-difference formula of extrapolation, is used as a predictor and a convex formula as a corrector to integrate differential equations numerically. An application has been made to Burgers' equation.

17. Key Words (Suggested by Author(s)) Explicit finite difference Burgers' equation Predictor-corrector	18. Distribution Statement Unlimited distribution Subject category:64

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Uncl.	Uncl.	29	A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161