

On Simple Moving Grid Methods for One-Dimensional Evolutionary Partial Differential Equations

J. G. BLOM

*Centre for Mathematics and Computer Science,
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands*

J. M. SANZ-SERNA

*Departamento de Ecuaciones Funcionales,
Facultad de Ciencias, Universidad de Valladolid, Valladolid, Spain*

AND

J. G. VERWER

*Centre for Mathematics and Computer Science,
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands*

Received October 13, 1986; revised February 13, 1987

Two moving grid algorithms for the numerical integration of evolutionary one-dimensional partial differential equations are considered. Both algorithms discretize the equation on trapezoidal space-time elements and combine a prediction step with a remeshing technique in order to determine the orientation of the sides of the trapezoids joining nodes at consecutive time levels. The behavior of the discretizations employed is forecast by means of the modified equation technique and then studied in a series of numerical experiments. © 1988 Academic Press, Inc.

1. INTRODUCTION

Many evolutionary problems involving linear or nonlinear partial differential equations (PDEs) have solutions with sharp transitions such as boundary layers, shock layers, or steep wave fronts. Nowadays there is a consensus, at least for problems in one space dimension, that adaptive or moving grid methods are capable of resolving these sharp transitions to acceptable degrees of accuracy, while avoiding the use of excessive numbers of grid points. Such methods use nonuniform spatial meshes and, as time proceeds, automatically concentrate the grid in spatial regions of high activity. It is thus possible to keep the number of space points at a minimum.

In recent years the interest in adaptive grid techniques has increased enormously (see, e.g., Furzeland [7] and the references therein). One may distinguish at least *two approaches*. In the *first approach* the grid moves continuously in the space-time

domain and the discretization of the PDE and the grid selection are intrinsically coupled. A prominent example is the moving finite element method proposed by Miller and Miller [13] and later extended by several authors (see, e.g., Baines [1]). Another example is the technique suggested by White [19], based on a transformation of the independent variables which is essentially defined by a grid selection criterion.

In the *second approach* the grid moves only at discrete time levels and no intrinsic coupling exists between the discretization of the PDE and the grid selection. Here the solution is advanced in time on a fixed nonuniform grid, while after each step, or series of steps, a regridding is carried out which is in turn followed by an interpolation to generate the initial values for the next step. Some recent contributions in this area are Bieterman and Babuška [2], Fuzzeland [7], Sanz-Serna and Christie [17], and Revilla [15].

Continuously moving grid methods are potentially very powerful in that they adapt the grid with the evolving solution in a certain optimal way. This implies that very sharp profiles can be computed with them. A disadvantage is that, due to the intrinsic coupling of the computations of the grid and the solution, the size of the system of nonlinear equations to be solved at each step becomes larger. Moreover, this discrete system is likely to be very ill-conditioned, due, among other things, to mesh-tangling. Often, the ill-conditioning is so severe that the time integration cannot be completed. Garcia-Archilla [8] reports this sort of problem for the original White's technique [19] and the present authors have had similar experiences with several modifications thereof. It is clearly conceivable to regularize the nonlinear equations by adding penalty terms meant to prevent the grid points from jumping into each other. However, we have found situations where, in practice, such penalty terms were unable to render the nonlinear system easily solvable. On the other hand, recent developments in the moving finite element method literature indicate that mesh-tangling can be successfully avoided in some cases (see Miller [12] and Baines [1]).

An attractive feature of the discrete time level approach is that the whole solution procedure can be split up into three clearly defined and more or less independent tasks. Namely, the integration step on a fixed nonuniform grid for which well-developed existing codes can be used (see, e.g., Bieterman and Babuška [2]), the regridding and the interpolation. A drawback, due to the step advancement on a fixed grid and the interpolation, is that methods of this type are not always successful in generating very sharp profiles.

The technique followed in the present paper is in a sense intermediate between the two approaches outlined above. We discuss two algorithms which advance the solution in time on a moving grid, but without coupling the computations of the grid and the solution as in the first approach and without using interpolation as in the second approach.

The discretization used in the first algorithm is related to the finite element Galerkin discretization using piecewise linear approximations over trapezoidal space-time elements first suggested by Bonnerot and Jamet [3] in the context of

one-dimensional Stefan problems (see also Jamet and Bonnerot [11] and Jamet [10]). Varoğlu and Finn [18] studied the Burgers' equation by employing the Bonnerot and Jamet technique and using the characteristics in order to determine the orientation of the sides of the trapezoids joining nodes at consecutive time levels. Davis and Flaherty [6] have also developed two adaptive finite element procedures using trapezoidal elements, one based on linear (cf. [3, 11]), the other on cubic approximations. One of the main differences between our first algorithm and those used by Davis and Flaherty lies in the grid selection part. More precisely, we use, aiming at a feasible and efficient generation of very sharp profiles, a *prediction step* for the estimation of a near-optimal grid at the next time level t_{n+1} , prior to the solution step from t_n to t_{n+1} . In [6] zero order extrapolation is favored, i.e., the optimal grid computed at time level t_n is used at t_{n+1} , a device which may imply the use of too small time steps and may not fully exploit the advantages of the grid motion capabilities allowed by the trapezoidal space-time elements.

In our second algorithm, which uses very simple finite differences, the discretization operates on the Lagrangian form of the time derivative. The grid selection method is identical for both algorithms.

In Section 2 of this paper we describe the two suggested discretizations. Section 3 is devoted to the grid selection procedure which relies on the equidistribution of u_{xx} via the variable knot spline procedure of de Boor [4]. A heuristic analysis based on the modified equation technique (see, e.g., Griffiths and Sanz-Serna [9]) is presented in Section 4. The results of our analysis are confirmed in Section 5, where we test the two algorithms in a series of numerical experiments and compare them with a third scheme of the finite difference-interpolation type constructed along the lines of the aforementioned second approach. In Section 6 we summarize our conclusions and outline our future plans in this area.

2. THE DISCRETIZATION SCHEMES

Both adaptive grid algorithms studied in this paper consist of two parts, a solution or discretization part and a grid selection part. We first discuss the discretization schemes and postpone to the next section the discussion of the grid selection procedure (identical in both algorithms).

In the remainder of the paper we restrict ourselves to the PDE

$$u_t + L(u) \equiv u_t + f(u)_x - \varepsilon u_{xx} = 0, \quad x_L < x < x_R, t > 0, \quad (2.1)$$

subject to the initial and boundary conditions

$$u(x, 0) = u^0(x), \quad x_L \leq x \leq x_R, \quad (2.2)$$

$$u(x_L, t) = b_L(t), \quad u(x_R, t) = b_R(t), \quad t \geq 0. \quad (2.3)$$

Here ε is a positive constant and f represents a linear or nonlinear real function. We wish to note that cases involving vector-valued functions, other boundary conditions, and even more general one-dimensional PDEs can be treated by the methods considered in this paper.

The following notations will be used in the descriptions of the schemes:

$$\{x_L = x_0^n < x_1^n < \dots < x_{m-1}^n < x_m^n = x_R\}$$

denotes a grid at $t = t_n$ and

$$\{x_L = x_0^{n+1} < x_1^{n+1} < \dots < x_{m-1}^{n+1} < x_m^{n+1} = x_R\}$$

a (different) grid at $t = t_{n+1}$. Here t_n and $t_{n+1} = t_n + \tau$ are consecutive time levels. The notation u_i^n represents the discrete approximation to the value $u(x_i^n, t_n)$. Throughout the paper we suppose that m is fixed during the time stepping, i.e., the number of grid points is not changed. It is useful to connect by a straight line each point x_i^n to the corresponding point x_i^{n+1} , thus originating m trapezoids in the strip $x_L < x < x_R$, $t_n < t < t_{n+1}$, as depicted in Fig. 2.1.

2.1. The Bonnerot–Jamet–Crank–Nicolson (BJCN) Scheme

This is given by

$$\begin{aligned} & u_i^{n+1}(x_{i+1}^{n+1} - x_{i-1}^{n+1}) - u_i^n(x_{i+1}^n - x_{i-1}^n) \\ & - \frac{1}{2}[(u_{i+1}^{n+1} + u_{i+1}^n)(x_{i+1}^{n+1} - x_{i+1}^n) - (u_{i-1}^{n+1} + u_{i-1}^n)(x_{i-1}^{n+1} - x_{i-1}^n)] \\ & + \frac{1}{2}\tau[(f_{i+1}^{n+1} - f_{i+1}^n) + (f_{i-1}^{n+1} - f_{i-1}^n)] \\ & - \tau\varepsilon \left[\left(\frac{u_{i+1}^{n+1} - u_i^{n+1}}{x_{i+1}^{n+1} - x_i^{n+1}} - \frac{u_i^{n+1} - u_{i-1}^{n+1}}{x_i^{n+1} - x_{i-1}^{n+1}} \right) \right. \\ & \left. + \left(\frac{u_{i+1}^n - u_i^n}{x_{i+1}^n - x_i^n} - \frac{u_i^n - u_{i-1}^n}{x_i^n - x_{i-1}^n} \right) \right] = 0, \end{aligned} \quad (2.4)$$

a discretization that on a fixed grid (i.e., $x_i^n = x_i^{n+1}$) reduces to the Crank–Nicolson central difference scheme. The formulae (2.4) follow those considered by Bonnerot and Jamet [3, 10, 11] (cf. also Davis and Flaherty [6]) and can be derived with the

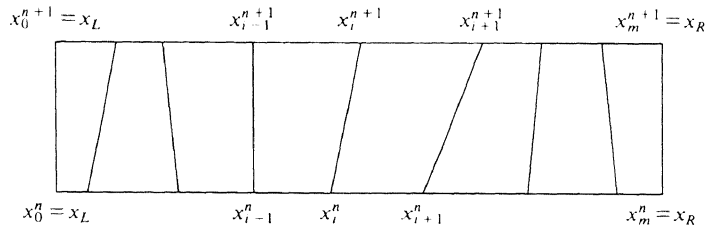


FIG. 2.1. Trapezoids covering the strip $x_L < x < x_R$, $t_n < t < t_{n+1}$.

help of the Galerkin approach, with basis functions defined by means of local coordinate transformations which map each of the trapezoids represented in Fig. 2.1 onto a reference rectangle $-1 \leq \tilde{x} \leq 1$, $0 \leq \tilde{t} \leq 1$. For brevity, the derivation is omitted here and the interested reader is referred to the papers just quoted. It should be pointed out that the discretization technique we have just outlined is attractive in that it easily allows the systematic construction of high-order schemes.

2.2. The Implicit-Euler Lagrangian (IEL) Scheme

In order to describe our second discretization we first rewrite Eq. (2.1) in its "Lagrangian" form. More precisely, we introduce new independent variables (s, T) linked with the old variables (x, t) through relations of the form $x = x(s, T)$, $t = T$. Although the new time T equals the old time t , the derivatives u_t and u_T are different: The former measures the changes of u as a function of t at a *fixed* x value (Eulerian description); the latter at a *fixed* s value (Lagrangian description). The Lagrangian form of (2.1) is obtained by expressing u_t in terms of u_T as follows

$$u_T - u_x x_T + L(u) \equiv u_T - u_x x_T + f(u)_x - \varepsilon u_{xx} = 0, \quad (2.5)$$

where we should note that x -derivatives are still present. Now (2.5) is discretized on the stencil of Fig. 2.1 to yield

$$\begin{aligned} & \frac{u_i^{n+1} - u_i^n}{\tau} - \frac{u_{i+1}^{n+1} - u_{i+1}^n}{x_{i+1}^{n+1} - x_{i+1}^n} \frac{x_i^{n+1} - x_i^n}{\tau} \\ & + \frac{f_{i+1}^{n+1} - f_{i-1}^{n+1}}{x_{i+1}^{n+1} - x_{i-1}^{n+1}} - \varepsilon \frac{2}{x_{i+1}^{n+1} - x_{i-1}^{n+1}} \left(\frac{u_{i+1}^{n+1} - u_i^{n+1}}{x_{i+1}^{n+1} - x_i^{n+1}} - \frac{u_i^{n+1} - u_{i-1}^{n+1}}{x_i^{n+1} - x_{i-1}^{n+1}} \right) = 0. \end{aligned} \quad (2.6)$$

We note that (2.6) reduces to the implicit Euler-central difference scheme if the space grid is not changed in time. The formulae (2.6) represent the simplest conceivable implicit scheme for (2.5) and more sophisticated discretizations of the Lagrangian form could also be envisaged.

3. THE GRID SELECTION PROCEDURE

Suppose that the integration has reached the level t_n and that approximations u_i^n to $u(x_i^n, t_n)$ have been computed on the spatial grid x_i^n , $1 \leq i \leq m-1$. Before formulae (2.4) or (2.6) can be used to compute the new approximations $\{u_i^{n+1}\}$ at level t_{n+1} , the new grid $\{x_i^{n+1}\}$ must be chosen. In this section we describe an adaptive grid selection procedure which can supplement either the BJCN scheme (2.4) or the IEL scheme (2.6). The procedure is composed of two parts: a prediction step and a regridding step. The prediction step delivers the input for the regridding step in the form of approximations \tilde{u}_i to $u(x_i^n, t_{n+1})$, $1 \leq i \leq m-1$. These approximations are then employed in a regridding step to yield the new grid points

x_i^{n+1} , $1 \leq i \leq m-1$. The intermediate solution \tilde{u} plays no further role as the final approximations $\{u_i^{n+1}\}$ are computed by (2.4) or (2.6) in terms of $\{u_i^n\}$, $\{x_i^n\}$, $\{x_i^{n+1}\}$, and τ .

3.1. The Regridding Step

We shall describe first the regridding step, which is based on ideas borrowed from the field of two-point boundary value problems in ODEs (see [4, 7, 14]).

3.1.1. Equidistribution of the Second Derivative

It is both common practice and illuminating to introduce nonuniform grids via the notion of a coordinate transformation to new independent variables $s = s(x, t)$, $T = t$. The transformation should be such that in the new variables the problem at hand can be successfully integrated on a uniform s -grid. A suitable nonuniform x -grid can then be simply obtained by transforming the s -grid according to the inverse change of variables $x = x(s, T)$. The transformations used in practice are of the form

$$s(x, t) = \int_{x_L}^x M(\xi, t) d\xi / \eta(t), \quad \eta(t) = \int_{x_L}^{x_R} M(\xi, t) d\xi, \quad (3.1)$$

where M is a so-called monitor function. Typically $M(x, t)$ represents the value of a power of a spatial derivative of the solution u at the point (x, t) . Our specific choice for M will be given below. Note that $\eta(t)$ is only a normalizing factor introduced in order that $s(x_R, t) = 1$, so that s varies in the fixed interval $0 \leq s \leq 1$. If the grid points $x_i = x_i(T)$, $1 \leq i \leq m-1$, arise from a uniform s -grid via the change of variables, then clearly

$$\int_{x_i}^{x_{i+1}} M(\xi, t) d\xi = \eta(t)[s(x_{i+1}, t) - s(x_i, t)] = m^{-1}\eta(t), \quad 0 \leq i \leq m-1, \quad (3.2)$$

i.e., the x -grid has the property that, on each of the subintervals $x_i \leq x \leq x_{i+1}$, $0 \leq i \leq m-1$, the integral of the monitor function has the constant (i -independent) value $m^{-1}\eta(t)$ (*equidistributing property*). Thus, regions with large values of M receive more grid points than regions with small values of M . For the role of M we have successfully used the second derivative monitor

$$M(x, t) = (\alpha + |u_{xx}(x, t)|)^{1/2}, \quad \alpha > 0, \quad (3.3)$$

(cf. Revilla [15]), but other choices are clearly possible and have been tried in the literature. Approximation of the integral in (3.2) by the midpoint quadrature rule, shows that, when $\alpha = 0$, the following equidistribution relation holds

$$(x_{i+1} - x_i)^2 \left| u_{xx} \left(\frac{x_{i+1} + x_i}{2}, t \right) \right| \approx \text{constant}(t), \quad 0 \leq i \leq m-1. \quad (3.4)$$

Note that the left-hand side of (3.4) measures the interpolation error of u by piecewise linear interpolants based on the grid at hand. The parameter α in (3.3) has been introduced to regularize the transformation in regions where $u_{,xx}$ is zero or nearly vanishes. In the experiments we have always taken $\alpha = 1$, but it is clear that a more careful tuning of this parameter will improve the results. In situations where the unknown u is a vector, the foregoing ideas still apply, provided that the bars in (3.3)–(3.4) are understood to denote some suitable vector norm.

It should be stressed that, although the introduction of the new variables (s, T) is helpful in the previous discussions, the computation of the grid $\{x_i^n\}$, at least for $n \geq 1$, can be completely achieved in terms of the old variables (Furzeland [7], Sanz-Serna and Christie [17]). To this effect it is enough to replace (3.2) by the midpoint quadrature, leading to the set of relations

$$(x_{i+1} - x_i)M\left(\frac{x_{i+1} + x_i}{2}, t\right) = c(t), \quad 0 \leq i \leq m-1, \quad (3.5)$$

for the implicit determination of the grid points. Here $c(t)$ is an i -independent quantity which equals m^{-1} times the (composite) midpoint approximation to $\eta(t)$. It is perhaps helpful to note that, since $c^2(t)$ is connected to the expression in (3.4), $c(t)$ may provide the basis for a heuristic error monitor, which would suggest when to increase or decrease m (node creation or node deletion). We have not explored this possibility and, as mentioned before, we always keep m fixed during the integration in time.

To sum up, the grid that we define as optimal with respect to the equidistribution of $u_{,xx}$ is defined by the equations (3.3) and (3.5) with $\alpha = 1$. In the next subsections we describe the actual computation of the grid points.

3.1.2. Computation of the Grid for $n \geq 1$

At time levels $n \geq 0$, we must compute the grid $\{x_i^{n+1}\}$ so as to satisfy (3.5) with the monitor (3.3) evaluated at the *prediction* \tilde{u} . Since this prediction is only available at the grid points $\{x_i^n\}$ we use a finite difference approximation to compute $\tilde{u}_{,xx}$. In our experiments we have employed the following relations for $\tilde{u}_{,xx}$ at the midpoints $x_{i+1/2} = (x_i + x_{i+1})/2$

$$\begin{aligned} \tilde{u}_{,xx}|_{x_{1/2}} &\approx 2 \left(\frac{\tilde{u}_2 - \tilde{u}_0}{x_2 - x_0} - \frac{\tilde{u}_0 - \tilde{u}_1}{x_0 - x_1} \right) / (x_2 - x_1), \\ \tilde{u}_{,xx}|_{x_{i+1/2}} &\approx \left(\frac{\tilde{u}_{i+2} - \tilde{u}_i}{x_{i+2} - x_i} - \frac{\tilde{u}_{i+1} - \tilde{u}_{i-1}}{x_{i+1} - x_{i-1}} \right) / (x_{i+1} - x_i), \quad 1 \leq i \leq m-2, \\ \tilde{u}_{,xx}|_{x_{m-1/2}} &\approx 2 \left(\frac{\tilde{u}_m - \tilde{u}_{m-1}}{x_m - x_{m-1}} - \frac{\tilde{u}_m - \tilde{u}_{m-2}}{x_m - x_{m-2}} \right) / (x_{m-1} - x_{m-2}). \end{aligned} \quad (3.6)$$

The computation of the new grid points is best achieved by means of a well-known *explicit* procedure due to de Boor [4]. See Sanz-Serna and Christie [17] for

implementation details of de Boor's technique. These authors refer to equidistribution of arclength, but their material can be extended in a straightforward way to the equidistribution of (3.3). We wish to emphasize that the computational cost of the de Boor algorithm is very low when compared with the cost of the integration step itself. It is also of interest to keep in mind that, due to the construction, the knot ordering is maintained so that no two grid points can cross. In practice one should check whether the trapezoidal elements do not show excessive distortion (i.e., their angles do not depart significantly from 90°). Davis and Flaherty [6] have given a simple algorithm by which the distortion can be monitored and, if necessary, suppressed. The experiments in Section 5 were carried out without such a control as severe distortion was not perceived.

3.1.3. Computation of the Grid for $n=0$

For the computation of an equidistributing grid at $t=0$, where the solution u (and therefore its second derivative u_{xx}) is known analytically, three possibilities come to mind:

(i) Starting from a suitable initial grid, (that can be taken to be uniform if nothing better is available) approximate u_{xx} by differencing as in (3.6), apply de Boor's algorithm to find an improved grid; difference on this improved grid and apply de Boor's algorithm to find an even better grid. Iterate this cycle. This is the procedure followed by Revilla [15] and has the drawback that often in practice the iteration does not converge (Revilla, private communication).

(ii) Rewrite (3.5) as the (nonlinear) system

$$(x_{i+1} - x_i)M\left(\frac{x_{i+1} + x_i}{2}, t\right) = (x_i - x_{i-1})M\left(\frac{x_i + x_{i-1}}{2}, t\right), \quad 1 \leq i \leq m-1, \quad (3.7)$$

for the $m-1$ unknown grid points, and solve (3.7) by means of some sort of Newton method. Unfortunately we have experienced that this process is not very robust, as the Newton iteration is likely to fail.

(iii) The third possibility, which we advocate since it is simple and cannot fail if properly employed, is to integrate numerically, by any standard code used with a demanding tolerance, the initial value problem for the single ODE problem

$$\frac{dx(s, 0)}{ds} = \frac{\eta(0)}{M(x(s, 0), 0)}, \quad 0 \leq s \leq 1, \quad x(0, 0) = x_L, \quad (3.8)$$

which is a trivial consequence of Eq. (3.1). (Note that $M(x, 0)$ is available in analytic form.) The values $x(i/m, 0)$, $i=0(1)m$, returned by the code clearly equidistribute the monitor function M (cf. (3.2)). Before the code can be applied, it is necessary to compute accurately the integral $\eta(0)$. This can be done by means of the same code used to integrate (3.8), as every quadrature problem can obviously be recast as a problem demanding the integration of an initial value problem for an

ODE. An accuracy check for the whole procedure is afforded by the discrepancy between the approximation $x(1, 0)$ provided by the code and the theoretical value x_R .

3.2. The Prediction Step

The remeshing scheme described in Section 3.1 computes the new grid points from a numerical approximation \tilde{u} to the solution at t_{n+1} . It is obvious that if the remeshing is to work satisfactorily the prediction should approximate, within reasonable bounds, the true profile at t_{n+1} . This requirement is particularly important for problems with very sharp transitions in time.

We have experimented with three different predictions:

(i) $\tilde{u}_i = u_i^n$. The grid computed at t_n is then used at t_{n+1} , so that the remeshing procedure lags one time-step behind. This zero-order extrapolation is favored by Davis and Flaherty [6], who reject extrapolation formulae based on several past time levels.

(ii) \tilde{u}_i computed by the explicit discretization of (2.1) resulting from the replacement, on the fixed grid $\{x_i^n\}$, of u_i by forward differences and of the x -derivatives by central differences.

(iii) Use on the fixed grid $\{x_i^n\}$ of the implicit discretization (2.4) or (2.6) employed to compute u_i^{n+1} , i.e., the prediction is computed from (2.4) or (2.6) with $x_i^n = x_i^{n+1}$. This prediction, being implicit, is significantly more expensive than (i) or (ii).

In our experiments the zero-order approach turned out to be unreliable. Using this predictor both schemes, (2.4) and (2.6), really falsified the solution in one example. The explicit discretization predictor (ii) performed significantly better. However, the best results were obtained with the implicit predictors. It is our impression that the extra costs, needed to solve the implicit equations for \tilde{u}_i , pay off for difficult problems. This observation is made more plausible when taking into account that, as shown in the next section, moving-grid schemes like (2.4) or (2.6) work best when the grid points x_i^{n+1} are at their optimal location. The results presented in Section 5 were all obtained using the implicit predictors.

4. ANALYSIS

This section is devoted to an analysis of the BJCN and IEL schemes. The *rigorous* treatment of finite difference discretizations on *fixed nonuniform grids* contains already a number of subtle points (see, e.g., Sanz-Serna [16]) and the case of *moving* nonuniform grids, such as those considered in this paper, would be intrinsically more difficult. Jamet [10] has proved the convergence of the scheme (2.4) as applied to the heat equation (i.e., $f \equiv 0$) under rather restrictive hypotheses on the

grid motion. We limit ourselves to some considerations based on the modified equation approach (see Griffiths and Sanz-Serna [9] for a survey). Our treatment, albeit heuristic, will provide a good insight on the practical performance of the schemes. In fact, in this area, the derivation of rigorous error bounds may turn out not to be very helpful, because the advantages of a moving grid would be felt in the size of the error constants rather than in the order of convergence, and error constants can seldomly be sharply estimated (see de Boor [4] for a similar comment). Therefore the heuristic approach can be expected to be, to some extent, more illuminating. The case of the IEL scheme will be studied first.

4.1. The IEL Scheme

We begin by observing that, in going from (2.5) to (2.6), the errors in the approximation of the time derivatives by *one-sided differences* can be expected to dominate over those in the approximation of the space derivatives by *central differences*. If in the left-hand side of (2.6) we substitute u_i^n, u_i^{n+1} , etc. by the nodal values of a smooth function $v(x, t)$, Taylor expand around (x_i^{n+1}, t_{n+1}) , and neglect the errors in approximating the space derivatives and higher order terms in τ , we arrive at the expression

$$v_t + L(v) - \frac{1}{2\tau} [(x_i^{n+1} - x_i^n)^2 v_{xx} + 2\tau(x_i^{n+1} - x_i^n)v_{tx} + \tau^2 v_{tt}] = 0, \quad (4.1)$$

where all the functions are evaluated at (x_i^{n+1}, t_{n+1}) . We need the (reasonable) assumption that, as the mesh is refined,

$$x_i^{n+1} - x_i^n = \tau d(x_i^{n+1}, t_{n+1}) + \text{higher order terms}, \quad (4.2)$$

with d a smooth function, which measures the velocity of the nodes. On taking (4.2) into (4.1), suppressing again higher order terms, and equating to zero, we arrive at the modified equation

$$v_t + L(v) - \frac{1}{2}\tau(d^2 v_{xx} + 2dv_{tx} + v_{tt}) = 0. \quad (4.3)$$

Roughly speaking, the main issue here is that the solutions of (2.6) are expected to be closer to the solutions of the auxiliary modified equation (4.3) than to the solutions of Eq. (2.1). Note that (2.1) and (4.3) differ in $O(\tau)$ terms.

In order to further the discussion we first restrict our attention to the situation where (2.1) is given by the model advection equation

$$u_t + cu_x = u_t + L(u) = 0, \quad (4.4)$$

whose solutions satisfy

$$u_{tt} = c^2 u_{xx}, \quad u_{xt} = -cu_{xx}. \quad (4.5)$$

These relations also hold, except for $O(\tau)$ terms, for the solutions of (4.3), and thus, upon neglecting once more higher order terms, (4.3) may be rewritten as

$$v_t + L(v) - \frac{1}{2}\tau(d-c)^2 v_{xx} = 0, \quad (4.6)$$

an equation which is readily analyzed to yield the following conclusions: (i) If no hypothesis is made on the function $d(x, t)$, then the IEL scheme is first-order accurate, due to the $O(\tau)$ term in (4.6). (ii) The bulk of the error will take the form of excessive dissipation, due to the heat-equation nature of the offensive term in (4.6). (iii) This dissipation can be reduced by choosing the function $d(x, t)$, governing the nodal velocities, to be an approximation to the velocity c of the true solution u , or, in other words, by moving the nodes along the characteristics. (iv) If the IEL scheme is used in combination with an appropriate grid selection strategy, as those described in the previous section, then d can be expected to approximate, except for $O(\tau)$ terms, the velocity c . In this case the scheme becomes a second-order discretization.

The four conclusions above hold not only for solutions of (4.4), but also whenever we are dealing with a theoretical solution of the *travelling wave* form $u(x - ct)$, regardless of the equation being solved. This can be proved by observing that in this more general case the relations (4.5) are still valid. On the other hand, it is also possible to consider the nonlinear equation $u_t + f(u)_x = 0$ instead of (4.4). On eliminating v_{tt} and v_{tx} from (4.3) one arrives at an equation similar to (4.6), where now the $O(\tau)$ terms are not only dissipative but also involve a term in $(u_x)^2$. It is still true that, provided that the nodal velocity $d(x, t)$ is an $O(\tau)$ approximation to the slope of the local characteristic, the scheme affords a second-order discretization. However, it should be emphasized that, except in the simple situations considered so far (travelling wave solutions of any equation/nonlinear hyperbolic equation) the IEL scheme is only *first-order* accurate.

4.2. The BJCN Scheme

A conspicuous feature of the formulae (2.4) is the unfamiliar expression arising from the discretization of u_t via the Galerkin approach on trapezoidal elements a la Bonnerot and Jamet. In order to gain some insight into those terms we begin by noticing that (2.4) is the Crank-Nicolson discretization in time of the system of continuous time ODEs

$$\begin{aligned} & [u_i(x_{i+1} - x_{i-1})]_T - (u_{i+1}[x_{i+1}]_T - u_{i-1}[x_{i-1}]_T) \\ & + f_{i+1} - f_{i-1} - \varepsilon \left(\frac{u_{i+1} - u_i}{x_{i+1} - x_i} - \frac{u_i - u_{i-1}}{x_i - x_{i-1}} \right) = 0. \end{aligned} \quad (4.7)$$

If we conceive the subindex i ($i = 0((1)m)$) as referring to a nodal value at the point $s_i = i/m$ of an equidistant s -grid in $0 \leq s \leq 1$, then (4.7) equals $2\Delta s$ times the central difference discretization of

$$(ux_s)_T - (ux_T)_s + f_x x_s - \varepsilon(u_x)_s = 0, \quad (4.8)$$

an equation, which upon rearrangement reads

$$u_T x_s - u_s x_T + f_x x_s - \varepsilon(u_x)_s = 0. \quad (4.9)$$

Now multiplication of (4.9) by s_x readily leads to the Lagrangian form (2.5) of Eq. (2.1) being integrated. Therefore the BJCN scheme, which was derived via the trapezoidal element technique, can alternatively be derived by successively (i) going from (2.1) to its Lagrangian form (2.5), (ii) introducing s -derivatives in (2.5) to arrive at (4.9), (iii) rearranging as in (4.8), and (iv) discretizing on a fixed, uniform s -grid by central differences and employing the Crank–Nicolson time-stepping.

Obviously one could question the utility of the stage (iii) above. In this connection, the central difference (Crank–Nicolson) scheme

$$\begin{aligned} & (u_i^{n+1} - u_i^n) \left(\frac{1}{2}(x_{i+1}^{n+1} + x_{i+1}^n) - \frac{1}{2}(x_{i-1}^{n+1} + x_{i-1}^n) \right) \\ & - \left(\frac{1}{2}(u_{i+1}^{n+1} + u_{i+1}^n) - \frac{1}{2}(u_{i-1}^{n+1} + u_{i-1}^n) \right) (x_i^{n+1} - x_i^n) \\ & + \frac{1}{2} \tau (f_{i+1}^{n+1} - f_{i-1}^{n+1}) + (f_{i+1}^n - f_{i-1}^n) \\ & - \tau \varepsilon \left(\left(\frac{u_{i+1}^{n+1} - u_{i+1}^n}{x_{i+1}^{n+1} - x_{i+1}^n} - \frac{u_i^{n+1} - u_i^n}{x_i^{n+1} - x_i^n} \right) + \left(\frac{u_{i+1}^n - u_i^n}{x_{i+1}^n - x_i^n} - \frac{u_i^n - u_{i-1}^n}{x_i^n - x_{i-1}^n} \right) \right) = 0, \quad (4.10) \end{aligned}$$

resulting from direct discretization of (4.9) seems to deserve some attention.

It is possible to derive modified equations for (2.4) or (4.10) by considering them as discretizations on a uniform s -grid. The derivation is rather heavy and we only present here the results for the semidiscrete (continuous T) versions, i.e., the results when the time-stepping errors are not taken into account. For (4.7), the semidiscrete version of (2.4), the result is

$$\begin{aligned} & v_T x_s - v_s x_T + f_x x_s - \varepsilon(v_x)_s + [(\Delta s)^2/6](v_T x_{sss} - x_T v_{sss}) \\ & - [(\Delta s)^2/2](v_{ss} x_{Ts} + v_s x_{Tss}) \\ & + O((\Delta s)^4) + (\Delta s)^2 \{ \text{terms arising from } L \} = 0, \quad (4.11) \end{aligned}$$

while for the semidiscrete version of (4.10) the modified equation reads

$$\begin{aligned} & v_T x_s - v_s x_T + f_x x_s - \varepsilon(v_x)_s + [(\Delta s)^2/6](v_T x_{sss} - x_T v_{sss}) \\ & + O((\Delta s)^4) + (\Delta s)^2 \{ \text{terms arising from } L \} = 0. \quad (4.12) \end{aligned}$$

These equations differ only in the term $- [(\Delta s)^2/2](v_{ss} x_{Ts} + v_s x_{Tss})$ and the most interesting feature in (4.11) is the presence of the second derivative v_{ss} whose coefficient x_{Ts} may take both positive and negative values. In fact, if we assume for simplicity that the nodes x_i are numbered in their natural order (i.e., $x_i < x_{i+1}$ for

$i = 0(1)m$) then $x_s > 0$ and, in regions where the nodes approach each other, $x_{Ts} < 0$. In those regions the term v_{ss} , which would act in a backward heat equation manner, is a source of *instability*. Admittedly this instability may be offset by the presence of the dissipative term in $L(v)$. However, in the applications of interest ε is very small and the stabilizing effects of the term εv_{ss} would not be felt unless a very large number m of grid points, leading to a small Δs , were employed. It is of importance to note that the term causing instability is absent from (4.12), a result which hints that the central difference scheme (4.10) could be more useful than the BJCN discretization. In the next section we shall support this observation in a numerical experiment.

5. NUMERICAL EXPERIMENTS

5.1. Three Adaptive Grid Methods

The following three adaptive grid methods will be examined:

BJCN. Formulae (2.4), combined with the regridding procedure described in Section 3 with the implicit predictor.

IEL. Formulae (2.6), combined with the regridding procedure of Section 3 with the corresponding implicit predictor.

FDI. This is a method based on the second approach mentioned in the Introduction and is considered here for reasons of comparison. When the grid $\{x_i^n\}$ and corresponding approximations $\{u_i^n\}$ have been found, the FDI method first uses the implicit Euler–central difference scheme to compute, on the fixed grid $\{x_i^n\}$, intermediate approximations $\{\tilde{u}_i\}$ at t_{n+1} . The remeshing scheme then provides a mesh $\{x_i^{n+1}\}$. (Note that, so far, the FDI method is identical to the IEL method.) Finally the new approximations u_i^{n+1} are found as the value at x_i^{n+1} of the linear interpolant of the set of data (x_i^n, \tilde{u}_i) . The FDI method just described is very similar to those considered by Sanz-Serna and Christie [17] and Revilla [15].

All the computations were performed with a constant time step τ and a constant number $m+1$ of space points (including the boundary points x_L, x_R). Our main interest is to examine the accuracy and reliability of the three methods. Therefore, we have not paid much attention to efficiency aspects. For instance, we find \tilde{u}_i and u_i^{n+1} by two completely uncoupled applications of the Newton method iterated to convergence, but perhaps there is no need for solving very accurately for the predicted values \tilde{u}_i and information computed in the \tilde{u}_i loop could be used for the u_i^{n+1} loop. With our simple implementation the costs of FDI are approximately 50% lower than those of either BJCN or IEL, but this ratio would not be so marked if better implementations of latter methods were used.

5.2. Results for the Burgers' Equation

We present results for three well-known problems for the Burgers' equation

$$\begin{aligned} u_t + \left(\frac{1}{2}u^2\right)_x - \varepsilon u_{xx} &= 0, & x_L < x < x_R, t > 0, \\ u(x, 0) &= u^0(x), & x_L \leq x \leq x_R, \\ u(x_L, t) &= b_L(t), & u(x_R, t) = b_R(t), & t \geq 0. \end{aligned} \quad (5.1)$$

In all three problems the viscosity parameter ε equals 10^{-3} and $x_L = 0$, $x_R = 1$. We have pictured the true solutions of these problems in Fig. 5.1.

PROBLEM I. Our first problem has the exact solution

$$u(x, t) = c - d \tanh \left\{ \frac{d}{2\varepsilon} (x - ct - x_0) \right\} \quad (5.2)$$

where $c = \frac{1}{2}(u_x^- + u_x^+)$, $d = \frac{1}{2}(u_x^- - u_x^+)$, $u_x^- - u_x^+ > 0$. It describes a travelling front joining the upstream state u_x^- and the downstream state u_x^+ . The front travels with velocity c and has an initial position governed by x_0 . Following [15] we used the parameter values $u_x^- = 1$, $u_x^+ = 0$, $x_0 = \frac{1}{4}$.

PROBLEM II (Whitham [20, Chap. 4]). This has the solution

$$u(x, t) = 1 - 0.9 \frac{r_1}{r_1 + r_2 + r_3} - 0.5 \frac{r_2}{r_1 + r_2 + r_3}, \quad (5.3)$$

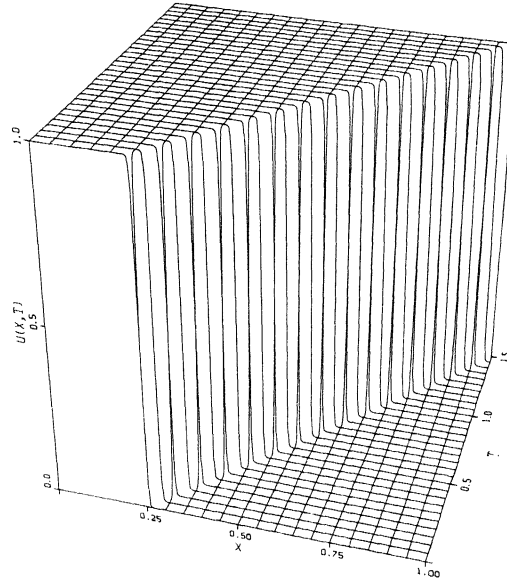
where

$$r_1 = \exp\left(-\frac{x-0.5}{20\varepsilon} - \frac{99t}{400\varepsilon}\right), \quad r_2 = \exp\left(-\frac{x-0.5}{4\varepsilon} - \frac{3t}{16\varepsilon}\right), \quad r_3 = \exp\left(-\frac{x-3/8}{2\varepsilon}\right).$$

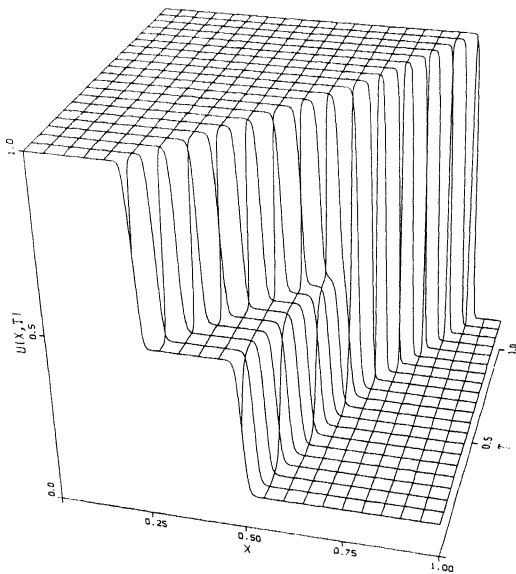
For our value of ε this solution initially contains two layers which merge in the subsequent evolution.

PROBLEM III. Now the initial function is given by $u^0(x) = \sin(\pi x)$ and the boundary conditions are homogeneous. The corresponding solution is a wave that first steepens and moves to the right until a layer is formed at $x = 1$. Then the solution slowly decays to zero while the layer, for the values of t of interest here, remains at the same position. The exact solution is available in the form of an infinite series (Cole [5]) whose evaluation, for our value $\varepsilon = 10^{-3}$, is not practical. Therefore, in this case, we have assessed the accuracy of our numerical solutions with the help of a reference solution computed on a very fine grid.

PROBLEM I



PROBLEM II



PROBLEM III

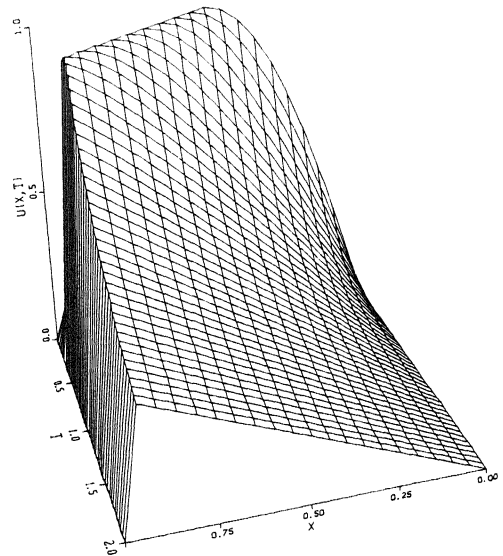


FIG. 5.1. Solution of Problems I-III. The solution of Problem III has been computed with IEL using $\tau^{-1} = m = 320$.

In what follows, results will be presented in the form of plots and, in the cases of Problems I and II, also in error tables, showing the following norms of the errors ($e_i^n = u_i^n - u(x_i^n, t_n)$):

$$\|e^n\|_\infty = \max_{0 \leq i \leq m} |e_i^n|, \quad \|e^n\|_2 = \left(\sum_{i=0}^{m-1} \left(\frac{x_{i+1}^n - x_i^n}{2} \right) (|e_i^n|^2 + |e_{i+1}^n|^2) \right)^{1/2}. \quad (5.4)$$

Results for Problem I. Table I contains the normed errors (5.4) at $t=1$ and $t=1.5$ for the nine combinations formed out of the choices $\tau = \frac{1}{20}, \frac{1}{40}, \frac{1}{80}$ and $m = 20, 40, 80$. The $\tau = \frac{1}{20}$ results for the IEL scheme are not given, as for this value of τ the Newton iteration failed for most values of m . Figure 5.2 shows plots of the exact solution and of the approximations for $\tau = \frac{1}{40}, m = 20$ and $\tau = \frac{1}{80}, m = 40$. The results of the BJCN scheme are excellent perhaps for the overshooting at $t=1.5$, when $\tau = \frac{1}{40}, m = 20$. The $v_{ss} x_{T_S}$ instability mentioned in the previous section is not detectable, undoubtedly as a consequence of the simple structure of the solution, which is more or less composed of flat portions (where $v_{ss} = 0$) and a steep front

TABLE I
Error Table for Problem I for $t = 1, 1.5$

		$t = 1.0$			$t = 1.5$			
		m	20	40	80	20	40	80
τ^{-1}								
BJCN	$\ e^n\ _\infty$	20	0.0456	0.1560	0.0218	0.2770	0.2613	0.2450
		40	0.0567	0.0084	0.0036	0.1392	0.1283	0.1090
		80	0.0579	0.0064	0.0040	0.0537	0.0423	0.0400
	$\ e^n\ _2$	20	0.0048	0.0141	0.0020	0.0227	0.0213	0.0205
		40	0.0042	0.0005	0.0002	0.0094	0.0081	0.0068
		80	0.0039	0.0003	0.0002	0.0034	0.0023	0.0022
IEL	$\ e^n\ _\infty$	20						
		40	0.1603	0.0999	0.0353	0.0985	0.1635	0.1500
		80	0.2815	0.0790	0.0406	0.0442	0.0494	0.0804
	$\ e^n\ _2$	20						
		40	0.0127	0.0084	0.0024	0.0056	0.0127	0.0118
		80	0.0215	0.0058	0.0029	0.0034	0.0035	0.0060
FDI	$\ e^n\ _\infty$	20	0.3896	0.4039	0.4235	0.3399	0.3183	0.3079
		40	0.3017	0.2976	0.3246	0.3508	0.2854	0.2589
		80	0.4652	0.2135	0.2129	0.6220	0.2462	0.2031
	$\ e^n\ _2$	20	0.0669	0.0645	0.0646	0.0482	0.0426	0.0411
		40	0.0412	0.0403	0.0406	0.0396	0.0307	0.0277
		80	0.0478	0.0251	0.0243	0.0641	0.0222	0.0182

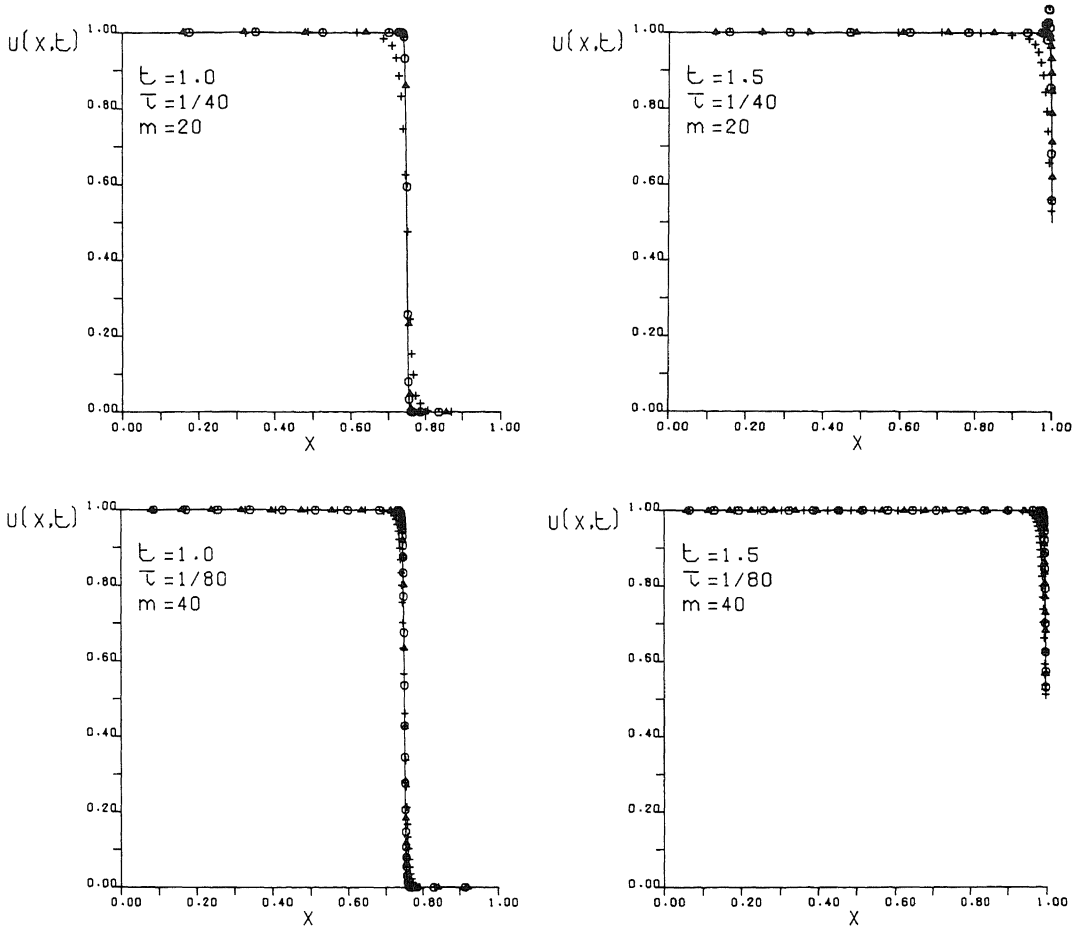


FIG. 5.2. Problem I. Comparison of exact solution (solid line) and numerical solutions, BJCN(\circ), IEL(\triangle), FDI(+).

(where the nodes are close but do not become closer to each other so that $x_{T_s} = 0$). (Note also that in a very steep front the modified equation approach is too naive, due to the role played there by the higher order terms of the truncation error, ignored in deriving the modified equation, see [9].) The IEL scheme also performs very satisfactorily, but locates the front slightly ahead of its true position with the result that the error norms are worse than those of BJCN. Both IEL and BJCN are very successful in generating a very sharp profile. In this connection we note that we are dealing with a travelling wave solution, one of the cases where, as shown in the previous section, the dissipation errors in IEL are of higher order. The plots show that the FDI algorithm smears the shock, a behavior typical for this sort of scheme and due to the interpolation. (Cubic interpolation yields less smearing but

TABLE II
Results of Convergence Test for Problem I at $t = 1$

$m = \tau^{-1}$	BJCN		IEL		FDI	
	$\ e^n\ _x$	$\ e^n\ _2$	$\ e^n\ _x$	$\ e^n\ _2$	$\ e^n\ _x$	$\ e^n\ _2$
40	0.008415	0.000529	0.099866	0.008396	0.2976	0.0403
80	0.004028	0.000192	0.040596	0.002920	0.2129	0.0243
160	0.001337	0.000037	0.004681	0.000332	0.1314	0.0138
320	0.000986	0.000027	0.000400	0.000027	0.0748	0.0074

may readily introduce oscillations, cf. Revilla [15].) Experiments with $m = 10$ were also conducted. BJCN and IEL generated an accurately positioned and sharp front, but introduced small oscillations.

We have also examined numerically the orders of convergence of the three schemes for Problem I upon halving τ and doubling m (see Table II). FDI, as

TABLE III
Error Table for Problem II for $t = 0.25, 1.0$

		$t = 0.25$			$t = 1.0$			
		m	20	40	80	20	40	80
	τ^{-1}							
BJCN	$\ e^n\ _x$	20	0.0556	0.0221	0.0248	0.0287	0.0353	0.0143
		40	0.0484	0.0087	0.0070	0.0191	0.0100	0.0041
		80	0.0589	0.0141	0.0060	0.0172	0.0072	0.0019
	$\ e^n\ _2$	20	0.0097	0.0043	0.0035	0.0033	0.0024	0.0010
		40	0.0077	0.0010	0.0006	0.0016	0.0008	0.0003
		80	0.0089	0.0014	0.0007	0.0013	0.0004	0.0001
IEL	$\ e^n\ _x$	20	0.0899	0.1183	0.0576	0.1206	0.6007	0.1477
		40	0.0300	0.0379	0.0084	0.1360	0.0154	0.0228
		80	0.0444	0.0126	0.0083	0.3527	0.0105	0.0040
	$\ e^n\ _2$	20	0.0184	0.0147	0.0069	0.0134	0.0618	0.0121
		40	0.0052	0.0044	0.0011	0.0123	0.0011	0.0017
		80	0.0078	0.0014	0.0008	0.0300	0.0007	0.0003
FDI	$\ e^n\ _x$	20	0.1895	0.1912	0.2032	0.3388	0.3940	0.4029
		40	0.1587	0.1762	0.1716	0.3146	0.2924	0.3142
		80	0.1690	0.1508	0.1390	0.3409	0.2436	0.2111
	$\ e^n\ _2$	20	0.0634	0.0593	0.0561	0.0672	0.0697	0.0700
		40	0.0496	0.0417	0.0417	0.0506	0.0438	0.0441
		80	0.0463	0.0304	0.0286	0.0427	0.0290	0.0268

expected, behaves as a first-order method (cf. the results in Sanz-Serna and Christie [17]). The IEL scheme, which uses one-sided differences in time, exhibits an order of convergence definitely higher than 1, a behavior we forecast in the previous section and due to the fact that we are dealing with a travelling wave solution. The BJCN errors do not behave too satisfactorily: they decrease in a rather slow manner, thus anticipating a failure of the scheme that will show up in later experiments.

Results for Problem II. We present results for $t = 0.25$, where two layers must be resolved, and $t = 1.0$, where only one layer remains (see Table III and Fig. 5.3). BJCN results are very good, with some oscillations. IEL suffers a little from smearing: the solution does not consist any longer of a single travelling wave and in this situation we know we should expect dissipative $O(\tau)$ errors. The results of FDI

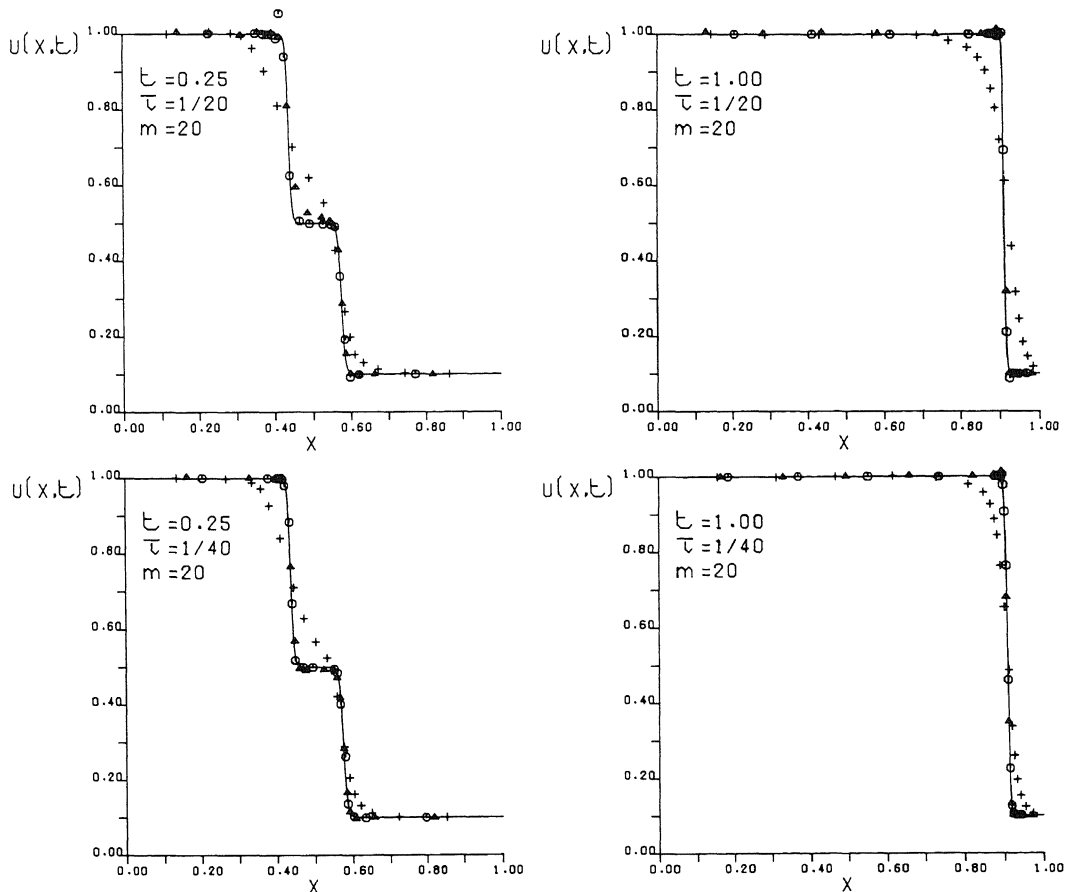


FIG. 5.3. Problem II. Comparison of exact solution (solid line) and numerical solutions, BJCN(\circ), IEL(Δ), FDI(+).

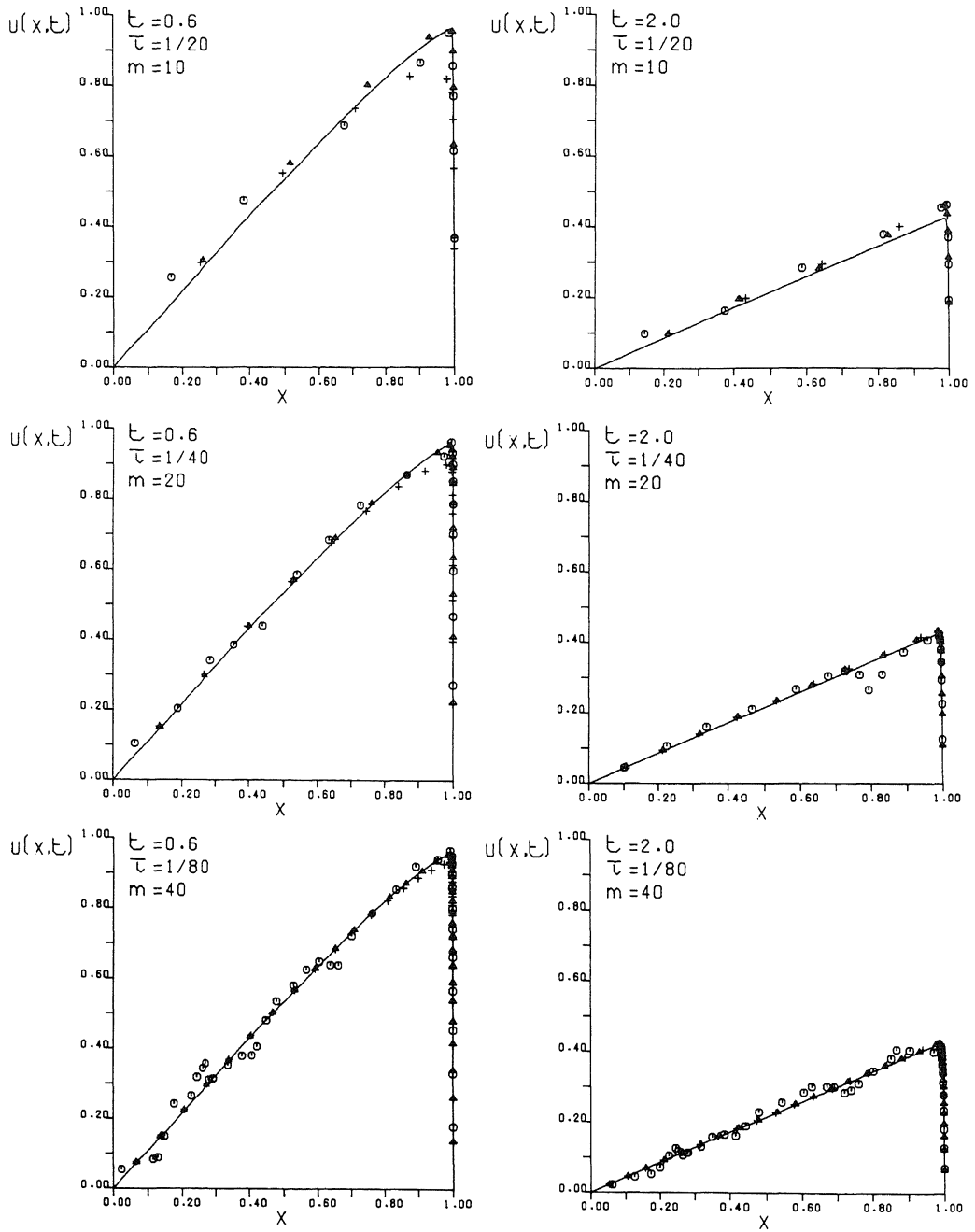


FIG. 5.4. Problem III. Comparison of computed reference solution (solid line) and numerical solutions, BJCN(○), IEL(△), FDI(+).

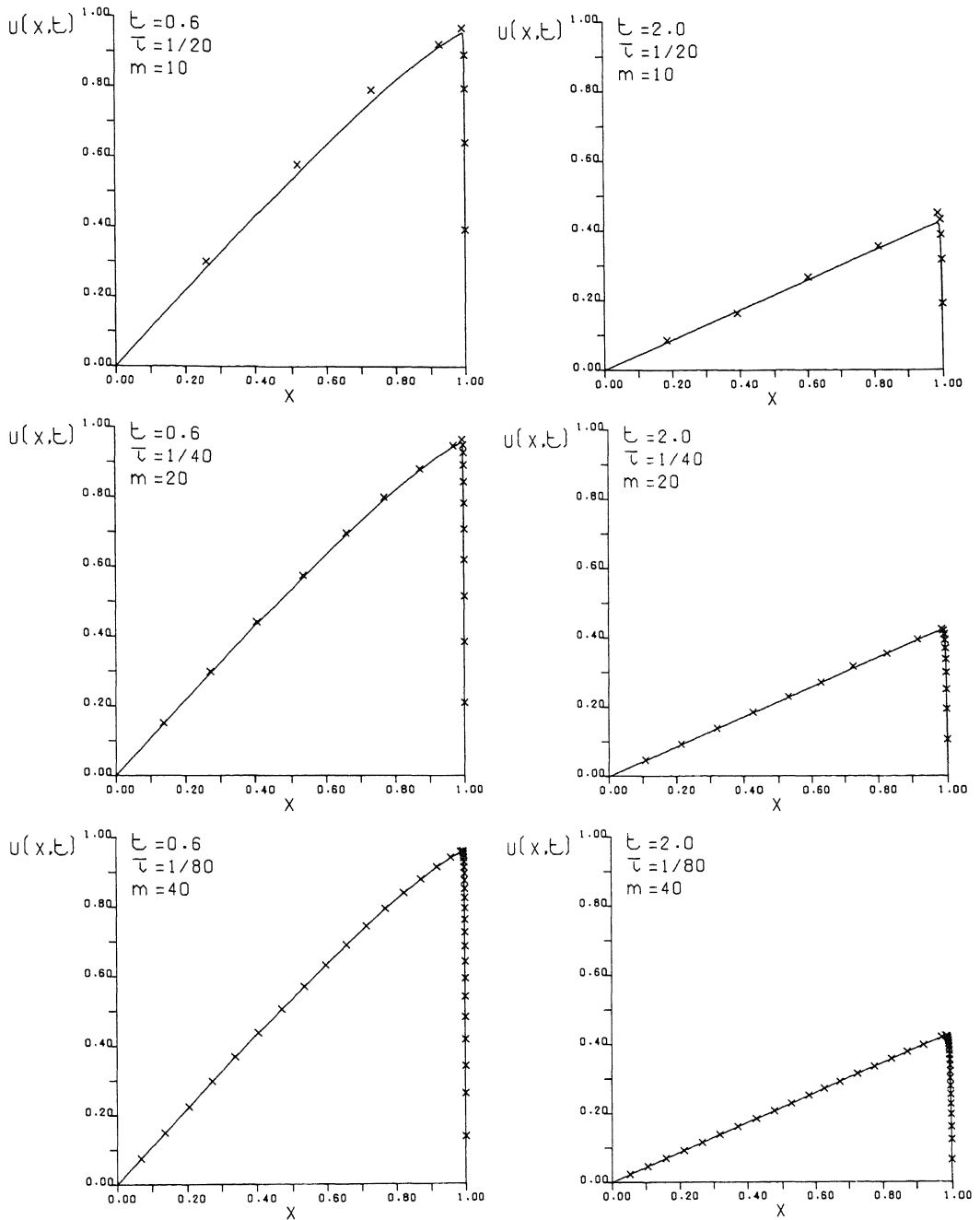


FIG. 5.5. Problem III. Comparison of computed reference solution (solid line) and the Crank-Nicolson solution (4.10).

are similar to those of the previous problem: wiggle-free and smeared. In our opinion this scheme is rather reliable and can be used when much accuracy is not of interest.

Results for Problem III. In Fig. 5.4 we show results at $t=0.6$ and $t=2.0$ for $(2\tau)^{-1} = m = 10, 20, 40$. While IEL gives extremely good results and FDI behaves as in the previous examples, the BJCN instability we had forecast is clearly apparent (note that, for these values of t , the nodes tend to leave the layer, thus becoming nearer to each other in the smooth region). In order to check that the instability is in fact due to the term $v_{ss}x_{T_s}$ in (4.11) we integrated our third problem by means of the scheme (4.10) supplemented by the grid selection strategy of Section 3. We recall that the scheme (4.10), while very similar to the BJCN discretization, does not give rise to the $v_{ss}x_{T_s}$ term. The results in Fig. 5.5 clearly show that the instability is not present, thus backing our analysis. We have also successfully applied the scheme (4.10) to Problems I and II, but the investigation of (4.10) is beyond our scope here.

Finally we would like to point out that the regridding strategy employed in the paper has worked very satisfactorily throughout the examples, as the limitations of the algorithms have always been attributable to the discretization formulae themselves.

6. DISCUSSION

We have described two moving grid algorithms which are intermediate between the continuous moving grid and regridding interpolation approaches described in the Introduction. One algorithm incorporates a discretization based on the Bonnerot–Jamet technique, while in the other the discretization was derived via the Lagrangian form of the equation. Both algorithms were successful in following and resolving very sharp profiles without coupling the grid selection and the computation of the solution, performing better in this respect than a scheme of the interpolatory kind. We have also found that the modified equation technique can be successfully applied to the investigation of the properties of the various discretizations. We therefore feel that the present research should be furthered. Some aspects to be explored are discussed next.

While the mesh selection procedure worked satisfactorily in all numerical experiments, the discretization schemes have shown some shortcomings. The IEL scheme, due to its backward Euler nature, is too inaccurate in general, while the BJCN discretization suffers from instability. Thus, new discretizations should be considered. It has also been shown that the rather lengthy derivation à la Bonnerot and Jamet can be avoided, as a simpler derivation technique can be used. This simpler technique led us to the scheme (4.10), of the Crank–Nicolson type, which will be the object of a forthcoming report. The scheme (4.10), while very similar to BJCN, is not subject to instability. In this connection we point out that *part* of the

BJCN instability can be eliminated by introducing a grid-distortion control (this keeps a check on x_{T_3}), but our experiments in this area (not described in Section 5) show that the instability cannot be totally suppressed. Clearly there is much work to be done on the implementation side, particularly in the solution of the nonlinear equations and in the introduction of variable time-steps. Finally, testing on a greater variety of problems, including systems, should be taken into consideration in future work.

REFERENCES

1. M. J. BAINES, University of Reading Numerical Analysis Report 1/85, 1985 (unpublished).
2. M. BIETERMAN AND I. BABUŠKA, *J. Comput. Phys.* **63**, 33 (1986).
3. R. BONNEROT AND P. JAMET, *Int. J. Numer. Methods Eng.* **8**, 811 (1974).
4. C. DE BOOR, in *Conference on the Numerical Solution of Differential Equations, Dundee, Scotland, 1973*, edited by G. A. Watson (Springer-Verlag, Berlin, 1974), p. 12.
5. J. D. COLE, *Q. Appl. Math.* **9**, 225 (1951).
6. S. F. DAVIS AND J. E. FLAHERTY, *SIAM J. Sci. Stat. Comput.* **3**, 6 (1982).
7. R. M. FURZELAND, Report TNER.85.022, Thornton Research Centre, Shell Research Limited, 1985 (unpublished).
8. J. B. GARCIA-ARCHILLA, MSc. Thesis, Facultad de Ciencias, Universidad de Valladolid, 1986 (unpublished).
9. D. F. GRIFFITHS AND J. M. SANZ-SERNA, *SIAM J. Sci. Stat. Comput.* **7**, 994 (1986).
10. P. JAMET, *SIAM J. Numer. Anal.* **17**, 530 (1980).
11. P. JAMET AND R. BONNEROT, *J. Comput. Phys.* **18**, 21 (1975).
12. K. MILLER, in *Adaptive Computational Methods for Partial Differential Equations, College Park, Maryland, 1983*, edited by I. Babuška *et al.* (SIAM, Philadelphia, 1983), p. 165.
13. K. MILLER AND R. MILLER, *SIAM J. Numer. Anal.* **18**, 1019 (1981).
14. V. PEREYRA AND E. G. SEWELL, *Numer. Math.* **23**, 261 (1975).
15. M. A. REVILLA, *Int. J. Numer. Methods Eng.* **23**, 2263 (1986).
16. J. M. SANZ-SERNA, "Stability and Convergence in Numerical Analysis I: Linear Problems—A Simple Comprehensive Account," in *Nonlinear Differential Equations*, edited by J. K. Hale *et al.* (Pitman, Boston, 1985), p. 64.
17. J. M. SANZ-SERNA AND I. CHRISTIE, *J. Comput. Phys.* **67**, 348 (1986).
18. E. VAROGLU AND W. D. LIAM FINN, *Int. J. Numer. Methods Eng.* **16**, 171 (1980).
19. A. B. WHITE, *SIAM J. Numer. Anal.* **19**, 683 (1982).
20. G. B. WHITHAM, *Linear and Nonlinear Waves* (Wiley-Interscience, New York, 1974).