Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.G. Blom, J.M. Sanz-Serna, J.G. Verwer

A Lagrangian moving grid scheme for one-dimensional evolutionary partial differential equations

Department of Numerical Mathematics Report NM-R8713 June

A LAGRANGIAN MOVING GRID SCHEME 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

Dpto. Ecuaciones Funcionales, Facultad de Ciencias,

Universidad de Valladolid, Valladolid, Spain

J.G. Verwer

Centre for Mathematics and Computer Science,

Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

A Lagrangian moving grid finite difference method for one-space dimensional, evolutionary partial differential equations which exhibit sharp transitions in space and time is developed. The method is based on a Crank-Nicolson type difference scheme derived via a co-ordinate transformation governed by equidistribution of the second space derivative. Each time step of our method involves two stages. First, a static grid numerical integration is carried out, immediately followed by a de Boor type redistribution of nodes at the forward time level. This stage serves only to compute the transformation. Second, a moving grid numerical integration is carried out with the Crank-Nicolson scheme. Numerical experiments show that the method automatically concentrates the grid in regions of high spatial activity and is also able to step in time with stepsizes larger than those needed by static methods, that is, methods which for intervals of time work on a fixed, nonuniform grid. As a result the method achieves high accuracy with few gridpoints in space and time.

1980 Mathematics Subject Classification: Primary: 65M50, Secondary: 65M99.
1982 CR Categories: 5.17.
Key Words & Phrases: numerical analysis, partial differential equations, time dependent problems, moving grid methods, space-time finite differences.
Note: This report will be submitted for publication elsewhere.

Report NM-R8713
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1. INTRODUCTION

Many important problems in time-dependent partial differential equations (PDEs), such as combustion problems, possess solutions with sharp transitions in time and space. Finite difference and finite element methods involving grids in space which are static, for intervals of time, often perform badly when applied to such problems. On the other hand, moving grid methods which at each time step adjust the space grid to high spatial activity normally perform more effectively, in that they avoid the need of excessive numbers of space points, while often they also can take significantly larger time steps than static methods.

This paper deals with a moving grid method for problems in one space variable. The method is based on a Crank-Nicolson type difference scheme and belongs to the important class of methods which is somewhat "*intermediate* " to the static regridding methods, where nodes remain fixed for intervals of time [9,10], and continuously moving grid methods, where the space node movement and the PDE integration are fully coupled (like in Miller's finite element and White's finite difference method [7,8,11]). Advancing from an initial space grid with m nodes, each time step of our "intermediate" method involves two stages: *The grid prediction stage.* Here a numerical integration step is performed on the strip of m rectangular space-time elements like in a static method, followed by a redistribution of the m nodes at the forward time level with a de Boor type regridding algorithm, which equidistributes a chosen monitor function. Hereby the static solution is used as input.*The integration stage.* The PDE is integrated by applying the moving grid Crank-Nicolson scheme over the strip of m trapezoidal space-time elements found after the regridding. Thus, the Crank-Nicolson scheme underlies a co-ordinate transformation governed by the equidistribution relation.

The work presented here is a continuation of our earlier work [1]. There we have compared various methods and presented some first, preliminary results of the Crank-Nicolson difference scheme. We have shown that this scheme is closely related to a finite element Galerkin scheme using piecewise linear approximations over trapezoidal space-time elements first suggested by Bonnerot & Jamet [2] in the context of Stefan problems (see also Davis & Flaherty [5] for more references). Unfortunately, the Bonnerot-Jamet scheme may suffer in practice from an annoying form of instability, which can also be analytically forecast [1]. The Crank-Nicolson scheme overcomes this limitation.

The above mentioned "intermediate" approach has two clear advantages. The space node movement is easier to deal with than in a continuously moving grid method where mesh tangling and ill-conditioning of the systems of algebraic equations to be solved are well-known threats. With de Boor's technique, employed at the grid prediction stage, points cannot cross or leave the domain. When compared with static regridding methods, larger time steps can be taken as the method employed at the integration stage also underlies a co-ordinate transformation as, for example, in White's technique.

A potential drawback of the "intermediate" approach is that to some extent it prohibits us to take full advantage of the moving grid difference formula, due to the fact that the solution on the rectangular space-time grid is used for regridding. For sharp transitions in time the errors of static, rectangular grid solutions are normally significantly larger than those of moving grid solutions. A too large time step then may result in too large errors in the static solution, which of course can lead to a less favourable grid positioning at the forward time level. This, in turn, will result in non-smooth trajectories for the grid points, which can be detrimental to the accuracy of our Crank-Nicolson scheme. However, our experiments clearly indicate that in practice our intermediate approach allows the use of sufficiently large stepsizes.

The contents of the paper read as follows. In Section 2 we describe the Crank-Nicolson scheme applied at the afore mentioned "integration stage". Section 3 is devoted to the "grid prediction stage". Here we give the governing equidistribution transformation and outline our version of the de Boor regridding algorithm. In Section 4 some implementation details concerning the numerical solution of the arising systems of nonlinear algebraic equations and the determination of the start grid are discussed. In Section 5 we list three different types of solutions of the nonlinear Burgers' equation, which we have used in our numerical experiments. In this fifth section we also illustrate the attractive "slowing down property" of the chosen transformation. Then, in Section 6, numerical results are presented. Here our Lagrangian moving grid method is shown to compare favourably with a representative from the class of static regridding methods. The final Section 7 is devoted to some conclusions.

2. THE CRANK-NICOLSON SCHEME APPLIED AT THE INTEGRATION STAGE

We first describe the Crank-Nicolson scheme applied at the integration stage and postpone to the next section the description of the grid prediction stage. The scheme is applied to the PDE problem

(2.1) $$u_t = L(u), \quad u = u(x,t), \quad x_L < x < x_R, \quad t > 0,$$

subject to initial and boundary conditions. L represents a linear or nonlinear operator involving only spatial derivatives. Here we suppose that L is such that truly discontinuous solutions (shocks) are excluded. L may depend explicitely on the variables x and t. This dependence, however, is suppressed in the general notation (2.1). The boundary points are supposed to be fixed. The dependent variable u may be vector-valued for the method we discuss, although in the present paper attention shall be confined to the scalar case.

The most natural way to set up a moving grid is to do it via a co-ordinate transformation. We make the hypothesis that the grid selection is governed by such a transformation to the new pair of independent variables $(s,t) = (s(x,t),t)$ with $s(x_L,t) = 0$, $s(x_R,t) = 1$ for all t in the domain of computation. So the space interval is mapped onto the unit interval in the new co-ordinate system. Observe that we only introduce a new space variable.

The transformation should be chosen such that in the variables (s,t) the problem is easier to handle numerically than in the original pair (x,t). Ideally, in the new variables any rapid transition should be absent because we then can take acceptable stepsizes in the temporal direction while using a coarse uniform s-grid in space. A suitable nonuniform x-grid then exists according to the inverse change of variables $x = x(s,t)$. As noted above, we discuss the actual transformation in the next section.

We write $v(s,t) = u(x(s,t), t)$. Problem (2.1) can then be written in the Lagrangian form

(2.2) $$x_s v_t - v_s x_t = x_s L(u), \quad 0 < s < 1, \quad t > 0.$$

Note that we have transformed only u_t to v_t. Also note that although the time variable t has not been transformed, these derivatives are different. The former measures the change of u as a function of t at a fixed value of x (Eulerian description), and the latter at a fixed s-value (Lagrangian description). The following notation will be used for the grids. The grid

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

denotes a grid at $t = t_n$. The x-grids are obtained from the inverse change of variables $x_i = x(s_i,t)$, where it is supposed that s_i is a point belonging to an equidistant s-grid ($s_i = ih$, $h = 1/m$, $0 \le i \le m$). In the description of the scheme we shall use the notation u_i^n for representing the discrete approximation to $u(x_i^n, t_n)$.

Our Crank-Nicolson scheme is obtained by standard second order central differencing on the s-grid the expressions $x_s v_t - v_s x_t$ and x_s occurring in the Lagrangian form (2.2) . Collecting terms yields (formula (4.10) of [1])

$$(2.3)\quad ((x_{i+1}^{n+1} - x_{i-1}^{n+1}) + (x_{i+1}^{n} - x_{i-1}^{n}))\left(\frac{u_i^{n+1} - u_i^{n}}{\tau}\right) -$$

$$((u_{i+1}^{n+1} - u_{i-1}^{n+1}) + (u_{i+1}^{n} - u_{i-1}^{n}))\left(\frac{x_i^{n+1} - x_i^{n}}{\tau}\right) =$$

$$(x_{i+1}^{n+1} - x_{i-1}^{n+1})\, L_{h,i}(u^{n+1}) + (x_{i+1}^{n} - x_{i-1}^{n})\, L_{h,i}(u^{n}).$$

$L_{h,i}$ represents an appropriate finite difference discretization of the spatial differential operator L at the gridpoint x_i and the index i varies between 1 and m-1. The scheme must be supplemented with the boundary conditions at x_0 and x_m and with the initial condition at t = 0. We shall introduce these conditions into the scheme not before we really need them.

3. THE GRID PREDICTION STAGE

Suppose that the integration has reached the n-th time level and that with scheme (2.3) approximations u_i^n to $u(x_i^n, t_n)$ have been computed on the grid x_i^n, $1 \le i \le m-1$. Before (2.3) can be used to compute the new approximations u_i^{n+1} at the next time level, the new grid points x_i^{n+1} must be chosen. We perform this task in what in the introduction is called the grid prediction stage. This stage itself is composed of two procedures : a *static integration step* (in [1] called the prediction step), and a *regridding step* . The static step delivers input for the regridding step in the form of approximations to $u(x_i^n, t_{n+1})$, $1 \le i \le m-1$. The static solutions play no further role in proceeding from time level n to time level n+1, as the final approximations u_i^{n+1} are computed by the moving grid scheme (2.3) only in terms of u_i^n, x_i^n, x_i^{n+1} and τ.

3.1 *The regridding step*

We shall first describe the computations carried out in the regridding step. Apart from a few changes our present regridding works similarly as in [1] and on the same theoretical basis. To save space we therefore will be very brief here and confine ourselves to the main computational aspects.

The regridding algorithm is based on the following equidistribution transformation which we hypothesized in the previous section:

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

For the monitor function M we take the second space derivative expression

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

If the x-grids arise from an equidistant s-grid, via the inverse transformation $x = x(s,t)$, then

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

for $0 \le i \le m-1$. Hence the x-grid has the property that on each of its subintervals the average of the monitor function has the same value (equidistribution of M over $[x_L, x_R]$). Regions with large values of M thus receive more grid points than regions with smaller values.

The parameter α has been introduced in order to regularize the transformation in regions where the solution u is very flat, i.e., where u_{xx} is nearly zero or truly zero. Hence its magnitude determines the number of points in regions where, relative to the regions of high spatial activity, the solution is flat. In all our experiments we have taken $\alpha = 1$, for reasons of simplicity. A more careful tuning of this parameter may improve the results somewhat. We also note that $\eta(t)$ may provide the basis for an heuristic spatial error monitor, which would suggest when to increase or decrease m. We have not explored this possibility in our present investigation.

The new pair of variables (s,t) is central in the theoretical development of our moving grid method. It should be stressed, however, that in the actual application the computation of the equidistributing x-grids is completely achieved in terms of the old variable pair (x,t). This is easily done by means of a well-known explicit procedure due to de Boor [3]. Our version of this regridding algorithm is composed of the following four computations. (I) Approximate, for $0 \le i \le m-1$, the monitor function values $M_{i+1/2}$ at all the midpoints $x_{i+1/2} = (x_i + x_{i+1}) / 2$ using finite difference approximations (cf. (3.6) in [1]). (II) Construct an approximate monitor function $M_a(x, t_{n+1})$ by linearly interpolating $M_{i+1/2}$ $(0 \le i \le m-1)$ in $(x_{1/2},x_{m-1/2})$. In $[x_0,x_{1/2}]$, respectively $[x_{m-1/2},x_m]$, $M_a(x,t_{n+1})$ takes the constant value $M_{1/2}$, respectively $M_{m-1/2}$. (III) Form the approximating function, $s_a(x,t_{n+1})$ say, to $s(x,t_{n+1})$ by exact integration. Note that the function s_a is piecewise quadratic and that $s_a(x_R,t_{n+1})$ is an approximation to $\eta(t_{n+1})$. Normalize as in equation (3.1). (IV) Carry out the inverse transformation on s_a to obtain the grid at t_{n+1}. This involves the solution of the quadratic equations $s_a(x,t_{n+1}) = i/m$ for $1 \le i \le m-1$.

Observe that the computational cost of the above algorithm is negligible when compared with the cost of an integration step with an implicit method. Further it is of interest to note that, due to the construction, the knot ordering is maintained so that no two gridpoints can cross or leave the domain. It may be advisable to check whether the grids on two consecutive time levels do not show excessive distortion (the angles of their connecting lines should not depart significantly from each other). A simple algorithm can be employed to monitor this distortion and, when decided necessary, to suppress it. The experiments reported by us were carried out without such a control as severe distortion was not perceived.

3.2 *The static integration step*

The regridding algorithm computes new grid points from a numerical prediction to the true solution at $t = t_{n+1}$. It is obvious that if the regridding is to work satisfactorily, the prediction should approximate, within reasonable bounds, the true profile at the new time level. In particular this is important for problems with very sharp transitions in time and space. Here the grid should move such that it is sufficiently dense in regions with large spatial gradients. Specifically, the center of a sharp layer should remain in the center of the finely meshed zone within one time step.

In [1] we have successfully employed the implicit Crank-Nicolson scheme (2.3) in its static step mode, that is, (2.3) with $x^n_{i+1} = x^n_i$. In this contribution we use the static implicit Euler scheme, as experimentation has revealed that with few gridpoints it generates smoother x-trajectories due to its better stability properties [6].

By using an implicit integration scheme, the computational effort put into implementing the co-ordinate transformation is considerable. Our philosophy is that the determination of the grids is as important as the computation of the solution on the grids. If the grid location does not accurately enough correspond to the true solution profile, the final application of any moving grid scheme will result in too large errors (see [1] for more comments).

4. IMPLEMENTATION DETAILS

Given for $n \geq 0$ the grid vector u^n, any complete step to the next time level (n+1) involves the application of two implicit integration formulas, plus one application of the regridding algorithm. The costs of our algorithm are mainly the costs involved in solving the arising systems of nonlinear algebraic equations.

4.1 *Solution of the nonlinear algebraic equations*

For this purpose we employ the true Newton process using, both in the *static* and *moving* step, the vector u^n as start vector. In the experiments reported in the next section we terminate the Newton iteration if

(4.1) $\qquad \| \text{difference of two successive iterates} \|_\infty < 10^{-8}.$

This is sufficiently safe with regard to the discretization errors we make. In fact, one can say we solve the implicit relations exactly. The prediction generated by the static step schemes is *not* used in the integration stage. We use it only as input vector for the regridding scheme. We have decided to do so since for problems with rapid temporal transitions, errors in the static step normally are significantly larger than in the moving grid step. In our tests we have experienced that for this reason there will in general be hardly any advantage in interpolating the static step result onto the new grid to generate the start vector in the Newton process of the moving step.

Likewise we have experienced that for difficult problems (steep gradients in time) the Newton convergence substantially slows down if we iterate with an old Jacobian (modified Newton). Even one Jacobian per integration step then may readily turn out to be more costly. For this reason we have decided to implement the genuine Newton process in the computer program used for the experiments below. Hence, the total costs of one Newton iteration amounts to

(4.2) *a Jacobian evaluation, an LU-decomposition*

an evaluation of L_h, *a forward backward substitution.*

In our tables of result we shall list the number of Newton iterations, thus considering (4.2) as a unit of costs. One may consider these numbers, in a loose way, as a measure of nonlinearity. We have computed the Jacobian matrices by standard numerical differencing. Finally, the possibility of using to advantage a more sophisticated Newton process does exist of course. We plan to pay more attention to this aspect in our future investigations.

4.2 *Computation of the start grid*

So far our discussion concerning the grid selection for the step n to n+1, assumes that the integration is underway. Also at the initial line (t = 0) we need grid points x_i^0, which (approximately) equidistribute the chosen monitor function. For this we have the exact initial function u(x,0) at our disposal. For the actual practice the following procedure is advocated (different from what we used in [1]). (i) Select a *trial start grid* $\{x_L = x_0 < x_1 < \dots < x_{m-1} < x_m = x_R\}$ with $m = m_{trial}$. (ii) Carry out step (I) - (III) from the de Boor algorithm using $u_{trial}(x_i)$, $0 \le i \le m_{trial}$, as input function. Here $u_{trial}(x)$ is the given initial function or an appropriate modification thereof. Reset m and carry out step (IV) of the de Boor algorithm. (iii) Carry out again (I)-(IV) to obtain the final *start grid..*

The introduction of the auxiliary function u_{trial} and the trial start grid is based on the following considerations:

(a) A general *rule of thumb* is that in the construction of the start grid one should reckon with the solution behaviour for small times. For example, if u is constant for t = 0, yet a layer evolves for t just greater than zero, then it is desirable that u_{trial} contains this layer, rather than that $u_{trial}(x)$

= u(x,0) for all x. Otherwise it may turn out to be necessary to start the integration process with very small time steps to avoid distortion.

(b) Loosely speaking, the *optimal start grid* is that grid which best resembles the numerical grids generated by the method itself during the first few steps. If the first few grids deviate too much from the start grid, then we have grid distortion which usually gives rise to larger errors. It is our experience that the optimal start grid (in the above sense) is difficult to find and that in the initial phase of the integration always some distortion will occur, unless the solution for times near the initial time is free from large gradients. If from start on we have large gradients, the monitor values derived from the numerical solutions in the first few steps may differ substantially from those taken from the exact solution (the effect of numerical differentiation). This, of course, influences the distribution of the grid points. At later steps, this influence usually rapidly diminishes.

(c) For the reason just given, even when the exact equidistributing function x(s,0) is availabletogether with u(x,0), it is usually advantageous to carry out step (iii) of the above procedure. In doing so the actual start grid thus is generated by the regridding algorithm which is also used at later steps. The effect of this *recipe* is that the start grid is better adjusted to the grids generated at these later steps.

In our experiments reported below we have used this recipe too. More precisely, using $u_{trial}(x)$ = u(x,0), we have carried out steps (i),(ii) above on a fine, uniform trial start grid, thus approximating x(s,0) accurately. Next we have carried out step (iii) with the desired number of points m. It should be emphasized that in practice we do not advocate to approximate the exact function s(x,0) up to a very high accuracy, because this may require too much computational effort and is entirely redundant. The procedure given above works satisfactorily if a modest accuracy in the approximating curve $s_a(x,0)$ is provided.

5. ILLUSTRATION OF THE EQUIDISTRIBUTION TRANSFORMATION

5.1 *The test set*

Following [1] we have done experiments using three different solutions of the Burgers' equation

$$(5.1) \qquad u_t = -f(u)_x + \varepsilon u_{xx}, \quad 0 < x < 1, \quad t > 0, \quad f(u) = u^2/2, \quad \varepsilon = 0.001.$$

In all three cases we have used Dirichlet boundary conditions. The boundary values and the initial function are derived from the solutions specified below.

Problem 1. The first solution we examine is given by $u(x,t) = c - d \tanh(d(x - ct - x_0)/(2\varepsilon))$ where $c = (u_- + u_+)/2$, $d = (u_- - u_+)/2$ and $u_- > u_+$. It describes a travelling front joining the upstream state u_- and the downstream state u_+. The smaller the diffusion parameter, the steeper the front. The front travels with velocity c and its initial position is x_0. We select $u_- = 1$, $u_+ = 0$, $x_0 = 0.25$ and let $0 \leq x,t \leq 1$. A plot is given in Fig. 5.1.

Problem 2. The second solution is $u(x,t) = 1 - (0.9r_1 + 0.5r_2) / (r_1 + r_2 + r_3)$ where $r_1 = \exp(-(20x + 99t - 10)/400\varepsilon)$, $r_2 = \exp(-(4x + 3t - 2)/16\varepsilon)$ and $r_3 = \exp(-(x - 3/8)/2\varepsilon)$. This solution is also a travelling wave front, except that here initially two thin layers exist which merge in the subsequent evolution. For moving grid schemes this is an additional difficulty. A precise investigation reveals that at $t = 0$ the position of the layers is $x = 0.25$ and $x = 0.5$, while the speed of the slowest is 3/10 and of the fastest 3/4. Hence they merge at $t = 5/9$. Beyond this time the speed is 11/20. From here the solution is similar to that of problem 1. We again let $0 \leq x,t \leq 1$. A plot is given in Fig. 5.1.

Problem 3. The third solution has the smooth initial function $u(x,0) = \sin(\pi x)$ and homogeneous Dirichlet conditions at $x = 0,1$. This is a wave that first steepens and moves to the right until a layer is formed at the right end point $x = 1$. This takes place for $t \approx 0.6$. Then the solution slowly decays to zero while the layer remains in the same position near $x = 1$. This behaviour makes the problem different in nature from the two previous ones. The exact solution is available in the form of an infinite series [4]. However, for small values of e, the evaluation of this series is not practical. An accurate numerical approximation is given in Fig. 5.1. The instability of the Bonnerot-Jamet scheme, which we mentioned in the introduction, is clearly observed for this problem [1].

We emphasize that for our testing purposes the choice of the governing PDE is of secondary importance. The form of the solution is the important factor here. In this connection we note that our three problems are different in nature. Each of it encompasses its own difficulty when moving grid methods are applied.

5.2 *Illustration of the equidistribution transformation*

The advantage of the equidistribution transformation is that it is generally applicable and easy to implement in a very cheap algorithm. As stipulated before, the rationale behind the co-ordinate transformation is that in the new variables the problem is easier to handle numerically than in its original physical variables. Hence it is necessary that the exact solution v(s,t) = u(x(s,t),t) is significantly smoother in the (s,t)-plane than in the (x,t)-plane. To illustrate that the equidistribution transformation is suitable in this respect, we present plots of the exact v(s,t) of Problem 1 (in Fig. 5.2) and Problem 2 (in Fig. 5.3).

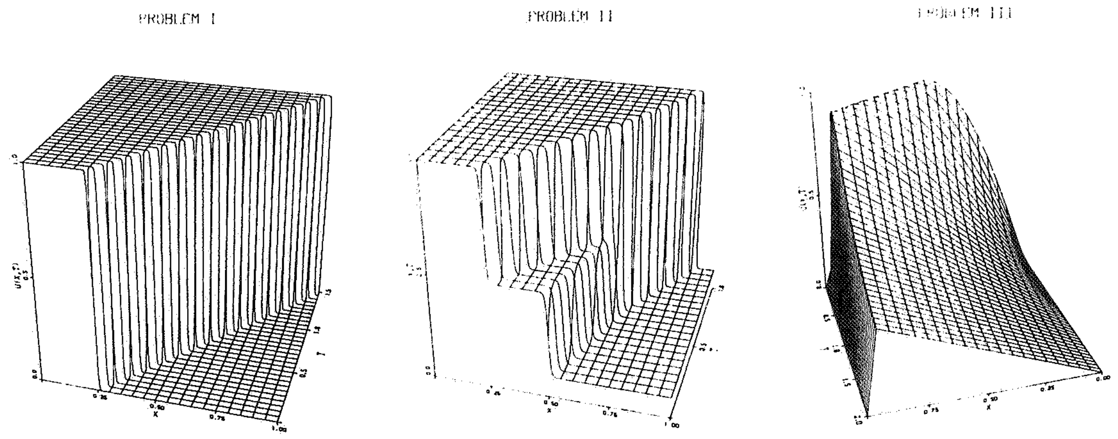

Fig. 5.1 Solutions of Problems 1-3. In the plot of Problem 3 the time has been reversed

Inspection of Fig.5.2 reveals that the equidistribution transformation has successfully removed the steep gradients of the travelling wave front u. In the spatial direction the very thin layer has been stretched considerably and in the temporal direction v is nearly constant. This means that the speed of propagation has been reduced to nearly zero. There is no doubt that the transformed equation can be solved up to normal accuracy on a standard equidistant space-time grid. To achieve this, it is of course a prerequisite that in the numerical algorithm the co-ordinate transformation is properly simulated.

For Problem 2 the following remarks are in order (see Fig. 5.3). The fact that here two thin layers exist on the initial line t = 0 and especially their subsequent merging into one makes the present problem more difficult to transform. The plots of v(s,t) show this nicely. In the spatial direction v is about as smooth as in the previous example. However, the merging of the two layers causes v(s,t) to be less smooth in the temporal direction. There is still a weak layer near t = 5/9. Despite this layer, if we are successful in computing the transformation it is still true that the transformed problem shall be much easier to solve than its original, the passing of the line t = 5/9 being the only difficult task left.

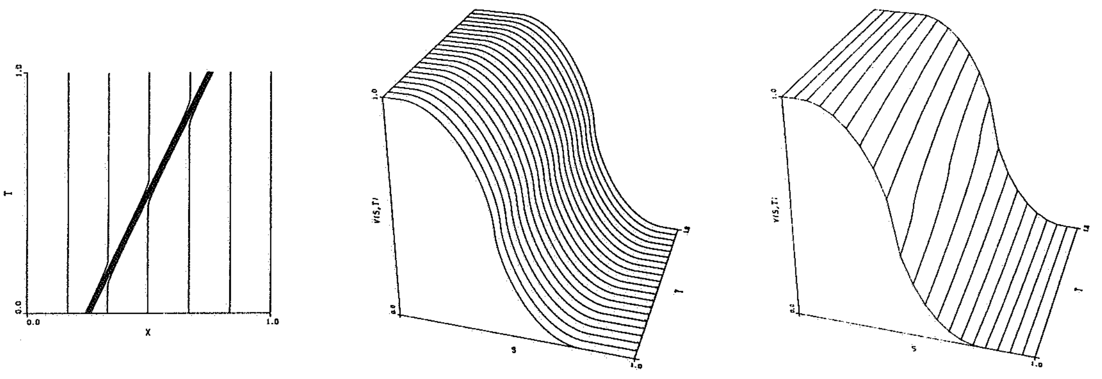

Fig. 5.2. (Problem 1) (a) shows the exact trajectories x(s, .) for s = 0(1/20)1. (b) contains the transformed, true solution curves v(.,t) for t = 0(1/20)1 and (c) the curves v(s, .) for s = 0(1/20)1.

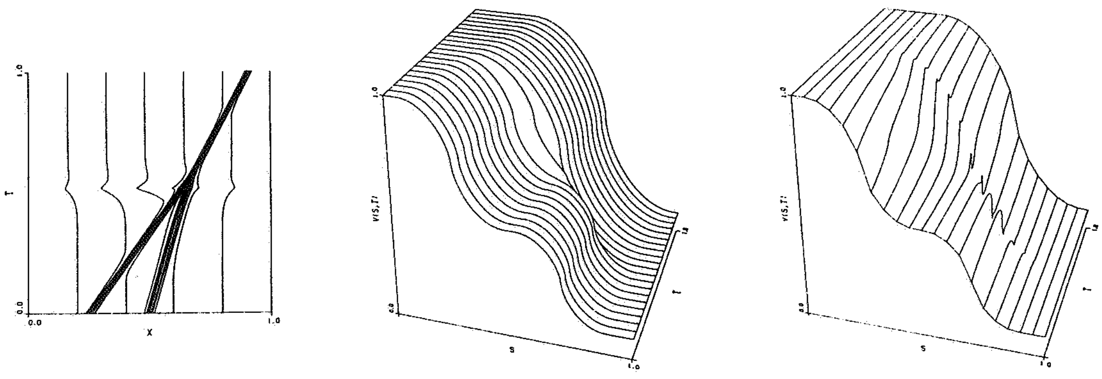

Fig 5.3. (Problem 2) (a) shows the exact trajectories x(s,.) for s = 0(1/20)1. (b) contains the transformed, true solution curves v(.,t) for t = 0(1/20)1 and (c) the curves v(s, .) for s=0(1/20)1.

It is instructive to examine the exact trajectories of x(s, .) for some values of s. Observe that for problem 2 the three different speeds of propagation are reproduced exactly in the movement of the dense grid regions. This shows that the equidistribution transformation works satisfactorily. We emphasize that the cusps are genuine and correspond to the weak layer behaviour at t = 5/9.

6. NUMERICAL EXPERIMENTS

In our tests all the computations were carried out with a constant stepsize τ in time and a number of space points $m = 1/h$ fixed in time, to be specified later. Further, we have employed the standard 3-point replacement for the spatial operator

$$L_{h,i}(u) = -\frac{f(u_{i+1}) - f(u_{i-1})}{x_{i+1} - x_{i-1}} + \varepsilon\,\frac{\dfrac{u_{i+1} - u_i}{x_{i+1} - x_i} - \dfrac{u_i - u_{i-1}}{x_i - x_{i-1}}}{(x_{i+1} - x_{i-1})/2}\,.$$

This operator naturally arises by central differencing on the equidistant s-grid the right hand side of the transformed equation (2.2) if we bring it in the form $x_s v_t - v_s x_t = -f(v)_s + \varepsilon(v_s/x_s)_s$.

For comparison, we have also implemented a second method based on static regridding (similar to those in [9,10] and to method FDI of [1]). This reference method differs from the Lagrangian method suggested in this paper only in that the final approximations u_i^{n+1} at the forward time level are obtained by linearly interpolating the static step values, rather than by applying formula (2.3). In the remainder we shall use the abbreviations BE/CN (Backward Euler/Crank-Nicolson) and BE/IP (Backward Euler/Interpolation) for our Lagrangian method and for the static regridding method, respectively.

In the tables of result we have listed maximum errors on the computed grids at the specified times. Hence these values include pointwise errors within or near the fronts. The integer numbers in italic are the rounded, averaged numbers of Newton iterations per step. For BE/CN the numbers before the slash symbol are those of the static step, and the numbers after the slash belong to the moving step (see Section 4.1). We note that for given τ and m, BE/CN is expected to be approximately twice as expensive as BE/IP, as the application of the latter involves per step one implicit computation instead of two.

Problem 1. Table 6.1 shows maximum errors for the specified values of τ and m. Two plots are given in Fig. 6.1. The results for BE/CN are excellent. For BE/IP the errors are significantly larger due to smearing, a behaviour typical for static regridding schemes and due to the interpolation (Cubic interpolation yields less smearing, but may readily introduce oscillations [9]).

m	20	40	80	160	320
BE/CN;τ^{-1}=2m	.0613	.0071	.00083	.00016	.000046
	6/5	*5/4*	*5/3*	*4/3*	*4/3*
BE/CN; τ^{-1}=4m	.0307	.0032	.00057	.00014	.000041
	5/3	*5/3*	*4/3*	*4/3*	*3/3*
BE/IP; τ^{-1}=2m	.3293	.2348	.1346	.0754	.0407
	5	*5*	*4*	*4*	*4*
BE/IP; τ^{-1}=4m	.4866	.2340	.1005	.0478	.0231
	5	*4*	*4*	*4*	*3*

Table 6.1. Maximum errors for the single wave front problem at t = 1.0.

m	20	40	80	160	320
BE/CN; τ^{-1}=2m	.0624	.0064	.0015	.00023	.000047
	5/4	*5/4*	*5/3*	*4/3*	*4/3*
BE/CN; τ^{-1}=4m	.0246	.0092	.0011	.00020	.000042
	5/4	*5/3*	*4/3*	*4/3*	*3/3*
BE/IP; τ^{-1}=2m	.3243	.2431	.1499	.0835	.0443
	4	*4*	*4*	*4*	*3*
BE/IP; τ^{-1}=4m	.3725	.2736	.1344	.0583	.0267
	4	*4*	*4*	*3*	*3*

Table 6.2. Maximum errors for the double wave front problem at t = 1.0.

From the table one also observes that both methods converge with increasing m, but in a somewhat odd way. The static scheme BE/IP should converge with order one, while in the (s,t)-reference frame the Crank-Nicolson scheme is of order two. However, the respective factors of 2 and 4 associated to simultaneous grid halving in space and time will show up only on unrealistically fine grids. This is due to the numerical determination of the grids and, of course, to the nature of the problem. Because the front is very steep, a small change in a gridpoint may result in quite a large change in the computed solution, not necessarily in a larger error. This "ill-conditionedness" affects the convergence behaviour of any type of moving grid scheme. Despite this situation, BE/CN is very successful in generating a very sharp front at the right location using few gridpoints in space and time.

Finally, the required number of Newton iterations to meet criterion (4.1) is acceptable when realizing that the accuracy requirement of 10^{-8} is rather stringent (and certainly redundant in practice). Some additional tests have shown that the number of iterations reduces considerably if the criterion is relaxed.

Problem 2. For this problem the results are shown in Table 6.2 and Fig. 6.2. The figure shows two plots at $t = 0.25$, where two layers must be resolved, and two plots at $t = 1.0$ where only one layer remains. The table contains maximum errors only for $t = 1.0$. We emphasize that in all runs the errors near the difficult point of merging of the two waves are only marginally larger than at $t = 1.0$. The results are similar to those of the previous problem. BE/CN is very accurate and positions the sharp front very nicely. BE/IP is again significantly less accurate due to smearing.

Problem 3. The third problem differs from the two previous ones in that its initial solution is very smooth. Hence, initially the grid is almost uniform. At later times the method must refine the grid near the right boundary and keep it there (see the description given in Section 5). We here only show plots of numerical solutions at $t = 0.6$ and $t = 2.0$ (see Fig. 6.3). Again the Lagrangian method performs very satisfactorily and is to be preferred to the static regridding scheme BE/IP. However, when taking into account that per step BE/IP is approximately twice as cheap as BE/CN, the difference in performance is not as large as in the two previous cases.

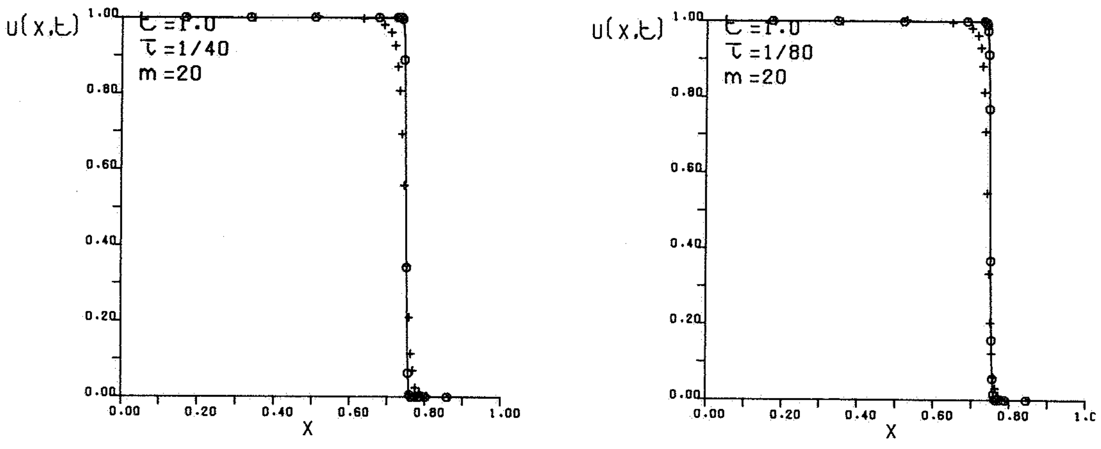

Figure 6.1. Problem 1. Comparison of exact solution (solid line) and numerical solutions. BE/CN(o), BE/IP(+).

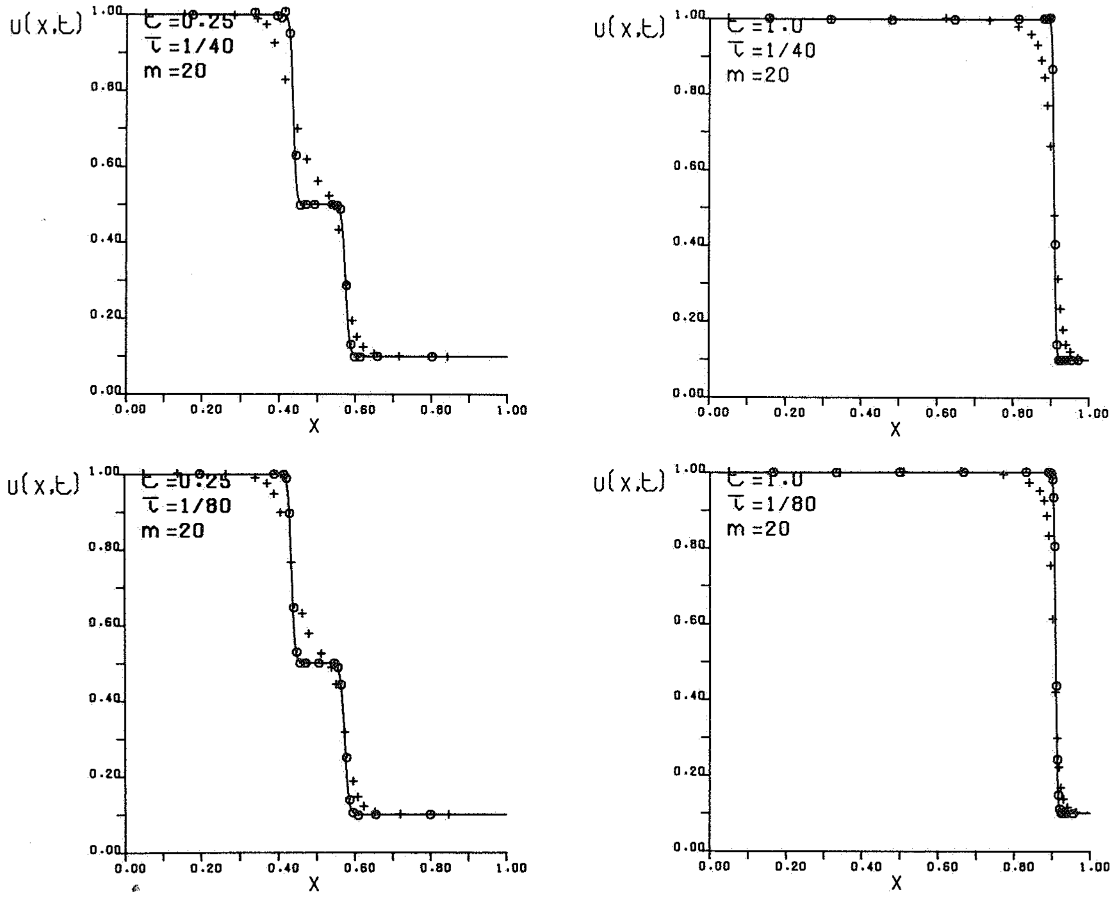

Figure 6.2. Problem 2. Comparison of exact solution (solid line) and numerical solutions. BE/CN(o), BE/IP(+).

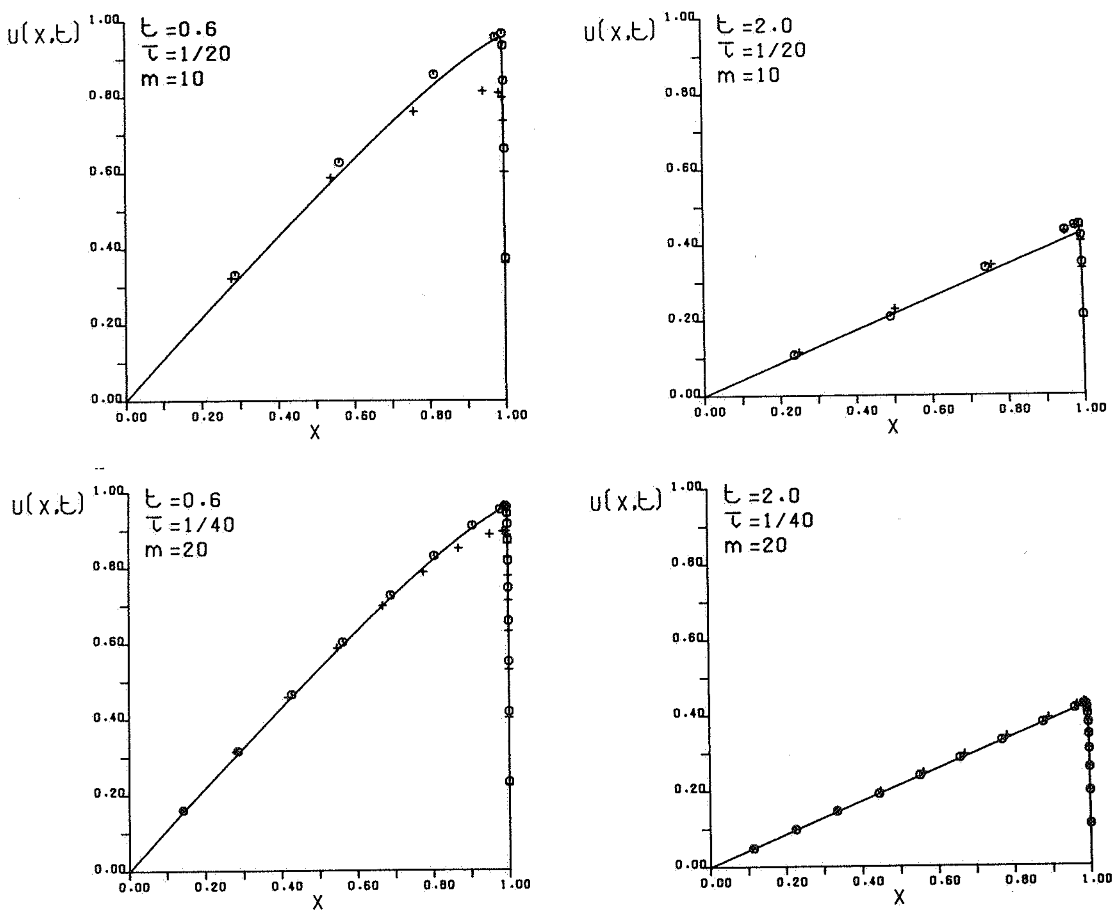

Figure 6.3. Problem 3. Comparison of exact solution (solid line) and numerical solutions. BE/CN(o), BE/IP(+).

7. CONCLUSIONS

We have described a Lagrangian type moving grid algorithm which is intermediate between the continuous moving grid and static regridding/interpolation approaches. Our algorithm incorporates a Crank-Nicolson discretization which is related to a discretization proposed by Bonnerot and Jamet [2]. Using three solutions of the nonlinear Burgers' equation, each of a different type, we have shown that our algorithm is 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 static regridding/interpolatory kind.

Our computational experience with the Lagrangian moving grid scheme, laid down in the present paper, looks promising, in our opinion. In the near future we therefore plan to continue investigating methods based on the present "intermediate" approach. Points which deserve further attention are automatic stepsize control in space and time and optimization of the solution of the nonlinear systems of algebraic equations arising in the application of the implicit integration formulas. Further, testing on a greater variety of problems, including systems, is necessary, and no doubt it would certainly be valuable to carry out a comparison with a well developed representative from the class of continuously moving grid methods and static regridding methods, and perhaps with a mixture of both.

REFERENCES

[1] J.G. Blom, J.M. Sanz-Serna & J.G. Verwer, *On Simple Moving Grid Methods for One-Dimensional Evolutionary Partial Differential Equations*, Report NM-R8620, Centre for Mathematics and Computer Science, Amsterdam, 1986 (to appear in J. Comput. Phys.).

[2] R. Bonnerot & P. Jamet, *A Second Order Finite Element Method for the One-Dimensional Stefan Problem*, Int. J. Numer. Methods Eng. 8, 811-820, 1974.

[3] C. de Boor, *Good Approximation by Splines with Variable Knots II*, **in:** Conference on the Numerical Solution of Differential Equations, Ed. G.A. Watson, Lecture Notes in Mathematics 363, pp. 12-20, 1973.

[4] J.D. Cole, *On Quasi-Linear Parabolic Equations Occurring in Aerodynamics*, Quarterly of Applied Mathematics 9, 225-236, 1951.

[5] S.F. Davis & J.E. Flaherty, *An Adaptive Finite Element Method for Initial-Boundary Value Problems for Partial Differential Equations*, SIAM J. Sci. Stat. Comput. 3, 6-27, 1982.

[6] K. Dekker & J.G. Verwer, *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*, North-Holland, 1984.

[7] K. Miller & R.N. Miller, *Moving Finite Elements I* , SIAM J. Numer. Anal. 18, 1019-1032, 1081.

[8] K. Miller, *Moving Finite Elements II* , SIAM J. Numer. Anal. 18, 1033-1057, 1981.

[9] M.A. Revilla, *Simple Time and Space Adaptation in One-Dimensional Evolutionary Partial Differential Equations*, Int. J. Numer. Methods Eng. 23, 2263-2275, 1986.

[10] J.M. Sanz-Serna & I. Christie, *A Simple Adaptive Technique for Nonlinear Wave Problems*, J. Comput. Phys. 67, 348, 1986.

[11] A.B. White, *On the Numerical Solution of Initial Boundary-Value Problems in One Space Dimension*, SIAM J. Numer. Anal. 19, 683-697, 1982.