CHAPTER 12

A Moving Grid Method for One-Dimensional PDEs Based on the Method of Lines*

J. G. Verwer†
J. G. Blom†
R. M. Furzeland‡
P. A. Zegeling†

Abstract. Even in one space dimension the numerical solution of time-dependent partial differential equations is often complicated due to large local gradients in the solution that evolve in time. The sharp moving gradients limit the efficient application of easy-to-use method of lines schemes that work on a fixed space grid. In such a situation the use of an adaptive or moving grid can often improve the efficiency and accuracy of the numerical computation. The method described in this paper integrates in a moving reference frame. The grid movement is based on the principle of spatial equidistribution of nodes and is regularized by employing a grid-smoothing technique in space and time. The spatial grid-smoothing ensures that the ratio of adjacent grid intervals is restricted, thus controlling clustering and grid expansion. The temporal grid-smoothing serves to obtain a smooth progression of the grid for evolving time. The spatial discretization is based on standard central differencing since we aim at a large problem class. For the numerical integration in time we use a sophisticated BDF code. In many cases this stiff solver can be used in a similar easy way as on a fixed grid. In other, more difficult cases, some parameter tuning may be required to optimally govern the grid movement. The performance of the method is numerically illustrated.

1. Introduction. We consider systems of partial differential equations (PDEs) in one space dimension,

$$u_t = f(u,x,t), \qquad x_L < x < x_R, \quad t > 0, \tag{1.1a}$$

with the initial and boundary conditions

* For the CWI/Shell project 'Adaptive Grids', P.A. Zegeling has received support from the 'Netherlands Foundation for the Technical Sciences' (STW), future Technical Science Branch of the Netherlands Organization for the Advancement of Research (NWO) (contract no. CWI55.092). J.G. Verwer has received financial support from the US Army Research Office in London which made it possible for him to participate in the Workshop at RPI (ref. no. R&D 6168-MA-06).

†Centre for Mathematics and Computer Science (CWI), P.O. Box 4079, 1009 AB Amsterdam, The Netherlands.

‡Koninklijke/Shell-Laboratorium, Amsterdam (Shell Research B.V.), P.O. Box 3003, 1003 AA Amsterdam, The Netherlands.

$$u(x,0) = u^0(x), \quad x_L < x < x_R \quad \text{and} \quad b(u,x,t) = 0, \quad x = x_L,\ x_R, \quad t > 0. \tag{1.1b}$$

Here f and b are spatial differential operators and it is tacitly assumed that the problems under consideration are well-posed and that they possess a unique solution. The differential operator f is supposed to be of at most 2-nd order. In particular, we are concerned with problems with disparate space and time scales giving rise to solutions with large space-time gradients. However, we do not consider genuinely discontinuous shock solutions as those arising in first order hyperbolic problems. Problems with disparate space and time scales occur in many applications from the engineering sciences and often an adaptive or moving grid can improve the efficiency and accuracy of a numerical computation.

The method described here is based on the method of lines (MOL) which is a well-known approach for numerically solving PDE problems such as (1.1). In the MOL approach the discretization of the PDE is carried out in two stages. In the first stage the space variables are discretized on a selected space mesh, normally chosen a priori for the entire calculation, so as to convert the PDE problem into a system of, usually stiff, ordinary differential equations (ODEs) with time as independent variable. The second stage then deals with the numerical integration in time of this stiff ODE system to generate the desired numerical solution. With this MOL approach in mind, several sophisticated PDE packages have been developed in recent years, notably for one-space-dimensional problems (see e.g. [2, 3, 8, 10, 11, 14, 15]). These MOL packages greatly benefit from the very successful developments of automatic stiff ODE solvers. In particular, the implicit Gear-type BDF solvers play a prominent role here. Gear-type solvers have proved to be efficient, robust and reliable, in that they work for a broad class of problems and usually solve the stiff ODE system under consideration in an accurate and efficient way. The experiences with MOL packages have revealed clearly that this is also true of semi-discrete PDE problems on fixed space grids. However, for solutions possessing large space-time gradients, like travelling wave fronts or emerging boundary and interior layers, a grid held fixed for the entire calculation can be computationally inefficient, since this grid will almost certainly have to contain a very large number of nodes. In such cases, a moving grid procedure that attempts to adjust automatically both the space and the time-stepsizes is likely to be more successful in efficiently resolving critical regions of high spatial and temporal activity.

The method described in this paper is of Lagrangian type and, at the semi-discrete level, automatically moves continuous-time grid lines to regions of high spatial activity. The grid movement underlies the principle of spatial equidistribution of nodes and employs regularization techniques borrowed from Dorfi and Drury [4]. The spatial discretization is based on standard central differencing since we aim at a large problem class. For the numerical integration in time we use a sophisticated BDF code [2, 3, 11]. From the users point of view it is of interest to note that this stiff solver can be used in a similar easy way as in the conventional (non-moving) approach. Some parameter tuning is required to govern the regularization of the grid movement as well as to optimise the efficiency. Needless to say, tuning is an important issue since the need for tuning is in conflict with robustness and ease of use. The numerical study of [7], where a comparison is presented between our current method, the adaptive moving-grid method of Petzold [12], and the moving-finite-element method (MFE) of Miller, shows that in this respect the current method compares favourably with the MFE method.

In Section 2 we introduce the semi-discretization in a moving reference frame, completely in line with the common MOL approach. In Section 3 we give the moving-grid equation that determines the continuous-time grid trajectories implicitly in terms of the semi-discrete solution on this grid. Section 4 is devoted to a discussion of the two grid-smoothing procedures that are used to regularize the grid movement. In Section 5 we discuss the complete semi-discrete system and its numerical integration. Section 6 presents results of numerical experiments with three different example problems and the final Section 7 is devoted to a brief conclusion.

2. The semi-discrete PDE. Virtually all of the space mesh adapting techniques for time-dependent problems attempt to move the nodes in such a way that, in regions of high spatial activity, there is enough spatial resolution. In other words, the construction of these methods is aimed at minimizing the number of space nodes relative to a certain level of spatial accuracy. On the other hand, in most time-dependent applications large spatial gradients are accompanied by large temporal gradients, the standard example being provided by the simple running wave form $u(x,t) = w(x - ct)$. It is thus natural not only to minimize the computational effort put into the

spatial discretization, but also to attempt to minimize the computational effort put into the time integration. Note that on a non-moving mesh a steep wave form such as $u(x,t) = w(x-ct)$ will require standard time-stepping techniques, including the sophisticated Gear methods, to use small time-steps. The reason for this is that as the moving front passes a grid point, the solution at this grid point will change very rapidly and so small time steps are then necessary to retain accuracy. The above observation naturally leads one to consider the Lagrangian discretization approach where the grid is moved continuously along with the solution with the aim of reducing these rapid transitions. Note, however, that it is not always possible to reduce them simultaneously in space and time (see [7, 16] for a more comprehensive discussion).

We start our derivation at the semi-discrete level. Thus, completely in line with the common MOL approach, consider smooth, continuous-time trajectories

$$x_L = X_0 < \cdots < X_i(t) < X_{i+1}(t) < \cdots < X_{N+1} = x_R \quad \text{for } t \geqslant 0, \tag{2.1}$$

which are, as yet, unknown. Introduce, along $x(t) = X_i(t)$, the total derivative

$$u' = x'u_x + u_t = X_i' u_x + f(u,X_i(t),t), \quad 1 \leqslant i \leqslant N, \tag{2.2}$$

and spatially discretize, for each fixed t, the space operators $\partial/\partial x$ and f so as to obtain the semi-discrete system

$$U_i' = X_i' [(U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1})] + F_i, \quad t > 0, \quad 1 \leqslant i \leqslant N. \tag{2.3}$$

As usual, $U_i(t)$ represents the semi-discrete approximation to the exact PDE solution u at the point $(x,t) = (X_i(t),t)$ and F_i is the finite difference replacement for $f(u,x,t)$ at this point. Note that the standard, central difference approximation for u_x is used. It is supposed that F_i is also based on standard, 3-point, central differencing. Further it is of interest to observe that at this stage of development the only errors introduced are the space discretization errors. With the associated grid functions

$$X = [X_1, \cdots, X_N]^T, \quad U = [U_1^T, \cdots, U_N^T]^T, \quad F = [F_1^T, \cdots, F_N^T]^T,$$
$$D_i = (U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1}), \quad D = [D_1^T, \cdots, D_N^T]^T,$$

we reformulate (2.3) in the more compact form

$$U' = X' \circ D + F, \quad t > 0, \quad U(0) \text{ given}, \tag{2.4}$$

which represents the semi-discrete system to be numerically integrated in time. The notation $X' \circ D$ means that X_i' is to be multiplied with all components of the vector D_i.

In the discussion to follow, we neglect the treatment of boundary conditions, since these are dealt with in the usual way. We also wish to emphasize that for convection-diffusion problems with steep gradient or near-shock behaviour, the use of central differencing of first order terms is not ideal and one would probably consider stable upwind or flux-corrected approximations, since otherwise any deviation from an ideal Lagrangian grid movement, assuming this exists, readily results in unphysical oscillatory solutions. It is emphasized that the actual generation of the moving grid is the central issue here and that other spatial discretizations can be easily implemented.

3. The moving-grid equation.

3.1. Spatial equidistribution. We shall construct an equation that defines the time-dependent grid X implicitly in terms of the continuous-time solution U. This grid equation underlies the familiar notion of spatial equidistribution. Introduce the point concentration values

$$n_i = (\Delta X_i)^{-1}, \quad \Delta X_i = X_{i+1} - X_i, \quad 0 \leqslant i \leqslant N, \tag{3.1}$$

and the spatial equidistribution equation

$$n_{i-1}/M_{i-1} = n_i/M_i, \quad 1 \leqslant i \leqslant N, \tag{3.2}$$

where $M_i \geqslant \alpha > 0$ represents a monitor value that reflects spatial variation over the i-th subinterval $[X_i, X_{i+1}]$. Typically, M_i is a semi-discrete replacement of a solution functional $m(u)$ containing one or more spatial derivatives. For example, the 1-st derivative functional (in scalar form; the change for systems is obvious)

$$m(u) = (\alpha + (u_x)^2)^{1/2} \tag{3.3}$$

yields, employing central differencing,

$$M_i = (\alpha + (U_{i+1} - U_i)^2/(X_{i+1} - X_i)^2)^{1/2} \tag{3.4}$$

The parameter $\alpha > 0$ serves to ensure that M_i is strictly positive. Unless noted otherwise $\alpha = 1$, which leads to the well-known arc-length monitor which has the property of placing points along uniform arc-length intervals. All numerical results reported in this paper have been obtained with the monitor (3.4) or its modification for systems. Of course, other choices for the monitor (e.g. solution curvature) could be used.

3.2. The grid-smoothing procedures. Equation (3.2) prescribes X in an implicit way in terms of U. However, as well known, for practical application the grid movement dictated by such an equidistribution equation needs to be regularized in order to avoid an oscillatory, distorted grid. For this purpose we now introduce two grid-smoothing procedures (borrowed from [4]), one for generating a spatially smooth grid, and the other for avoiding oscillations for evolving time. Use of the two grid-smoothing procedures amounts to modifying (3.2). We will first briefly describe these modifications and delay a more comprehensive discussion of the grid-smoothing to Section 4.

The spatial grid-smoothing is effected by replacing the point concentrations in (3.2) by their numerically 'anti-diffused' counterparts

$$\begin{aligned} \tilde{n}_0 &= n_0 - \kappa(\kappa+1)(n_1 - n_0), \\ \tilde{n}_i &= n_i - \kappa(\kappa+1)(n_{i+1} - 2n_i + n_{i-1}), \quad \kappa > 0, \quad 1 \leqslant i \leqslant N-1, \\ \tilde{n}_N &= n_N - \kappa(\kappa+1)(n_{N-1} - n_N), \end{aligned} \tag{3.5}$$

which results in the now 5-point coupled (in X) system

$$\tilde{n}_{i-1}/M_{i-1} = \tilde{n}_i/M_i, \qquad 1 \leqslant i \leqslant N. \tag{3.6}$$

The first and last equation in (3.5) involve the 'zero concentration gradient' boundary conditions

$$n_0 = n_{-1}, \quad n_{N+1} = n_N,$$

where n_{-1} and n_{N+1} correspond to the artificial points X_{-1} and X_{N+2}, respectively. In [7], and also in [4], the similar conditions $n_0 = n_1$, $n_{N-1} = n_N$ have been used. However, these imply that the first and last monitor values, M_0 and M_N, respectively, are removed from the moving grid equation (in (3.6) the index i then runs from 2 to $N-1$). This is not appropriate in cases where the boundary monitor values are much larger than the interior ones, like, e.g., in Problem I of Section 6 during the generation of the steep flame front at the right boundary. The present boundary conditions overcome this deficiency.

The introduction of the 'anti-diffused' point concentrations is equivalent to a certain smoothing procedure for the monitor function (see Section 4), thus ensuring that the adjacent point concentrations are restricted such that

$$\kappa/(\kappa+1) \leqslant n_{i-1}/n_i \leqslant (\kappa+1)/\kappa. \tag{3.7}$$

This condition implies that the grid we compute is locally bounded and, most importantly, provides a natural way to control clustering and grid expansion. While the monitor function determines the relative shape of X, the value of κ and N determine the level of clustering. Further, for a given N and a given monitor function distribution, the choice of κ determines the minimum and maximum interval lengths. In actual application, a value of κ of about 1 or 2 is recommended so that modestly graded space grids are obtained. In all our experiments we have used the (rather conservative) default value $\kappa = 2$. Recall that the grading of the space grid plays an important role in controlling space discretization errors (see, for example, [6]).

When combined with the spatial grid-smoothing, the temporal grid-smoothing is effected by replacing the system of algebraic equations (3.6) by the following system of differential equations

$$(\tilde{n}_{i-1} + \tau\tilde{n}'_{i-1})/M_{i-1} = (\tilde{n}_i + \tau\tilde{n}'_i)/M_i, \qquad \tau > 0, \quad 1 \leqslant i \leqslant N. \tag{3.8}$$

The introduction of the derivatives of the point concentrations serves to prevent the grid movement

from adjusting solely to new monitor values. Instead, the use of (3.8) forces the grid to adjust over a time interval of length τ from old to new monitor values, i.e. the parameter τ acts as a delay factor (see Section 4). The aim here is to avoid temporal oscillations and hence to obtain a smoother progression of $X(t)$. These oscillations can arise in grids generated via spatial equidistribution techniques, because when applied to solutions with extremely large gradients, the numerical monitor values are very sensitive to small perturbations in the grid and vice versa. With oscillatory trajectories it is certain that near steep fronts one or more components in the ODE system rapidly vary for evolving time. This is detrimental for the numerical time stepping and also causes difficulty in the Newton solution of the sets of nonlinear algebraic equations that arise in the implicit time integration with the stiff solver.

In contrast to the choice of κ, the choice of a good value for τ is less simple. Increasing τ too much results in a grid that lags too far behind any moving spatial transition. In fact, for sufficiently large values of τ a non-moving grid results. Fortunately, our numerical experience (see Section 6) indicates that in many situations temporal grid-smoothing is actually redundant. We owe this to the spatial grid-smoothing which also helps to prevent the grid from oscillating. However, in situations where smoothing in time is advisable, it makes sense to attempt to choose τ close to the anticipated temporal step size value such that, over one or a few time levels, the influence of past monitor values is felt. The discussion of the next section is aimed at providing more insight in this matter.

4. Discussion of the smoothing procedures.

4.1. Preliminaries. Equations (3.8) are based on the relation

$$\tau\tilde{n}_i' + \tilde{n}_i = cM_i, \qquad t > 0, \quad 0 \leqslant i \leqslant N, \tag{4.1a}$$

where $c = c(t)$ is the proportionality constant involved. This proportionality constant is solution dependent and in fact also depends on the parameters τ and κ. This dependence is suppressed in our notation and we shall use $c(t)$ as a generic notation for, possibly, different constants of proportionality. Using $\mu = \kappa(\kappa+1)$, we first rewrite $\tilde{n}_i$ in (3.5) as

$$\begin{aligned}
\tilde{n}_0 &= -\mu n_1 + (1+\mu)n_0, \\
\tilde{n}_i &= -\mu n_{i+1} + (1+2\mu)n_i - \mu n_{i-1}, \qquad 1 \leqslant i \leqslant N-1, \\
\tilde{n}_N &= -\mu n_{N-1} + (1+\mu)n_N.
\end{aligned} \tag{4.1b}$$

For initial conditions we suppose a given concentration distribution $n_i(0)$, $0 \leqslant i \leqslant N$, that has been subjected already to the spatial grid-smoothing procedure, i.e. the initial grid satisfies (3.6) at $t = 0$. For the actual practice this is a natural assumption because the space smoothing is also applied at later times. Violation of this assumption makes it likely that already within the first time-step the grid is forced to undergo a large change. However, in principle, an initial grid not satisfying (3.6) can be used.

We have $N+1$ equations for the $N+1$ unknowns n_i, $0 \leqslant i \leqslant N$, if we consider the proportionality constant $c(t)$ and the monitor values $M_i(t)$ as being given. In fact, for the analysis presented in the remainder of this section it is convenient to uniquely represent the $N+1$ concentrations $n_i(t)$ for $t \geqslant 0$ in terms of the initial concentrations $n_i(0)$ and the values $c(t)$, $M_i(t)$ as described below. First, solving (4.1a) yields the nonlinear Volterra integral equation system

$$\tilde{n}_i(t) = e^{-t/\tau}[\tilde{n}_i(0) + \int_0^t \tau^{-1} e^{s/\tau} c(s) M_i(s) ds], \qquad t \geqslant 0, \quad 0 \leqslant i \leqslant N, \tag{4.2}$$

where $\tilde{n}_i(0)$ is determined by $n_i(0)$ through (4.1b). We have a system of nonlinear Volterra integral equations because the monitor function values M_i depend on all concentrations in a nonlinear way. Second, the matrix $\mathbf{M}$ associated to the system of linear equations (4.1b), i.e.

$$\mathbf{M}\mathbf{n} = \tilde{\mathbf{n}}, \quad \mathbf{n} = [n_0, \cdots, n_N]^T, \tilde{\mathbf{n}} = [\tilde{n}_0, \cdots, \tilde{n}_N]^T, \tag{4.3}$$

is symmetric, positive definite. Hence, $\mathbf{M}$ is non-singular and the point concentrations n_i are uniquely expressed into $\tilde{n}_i$ by

$$\mathbf{n} = \mathbf{M}^{-1}\tilde{\mathbf{n}}. \tag{4.4}$$

Equations (4.2)-(4.4) define the moving grid $X(t)$ in an implicit way. Although this definition is not of much practical use, it is useful for a qualitative study of the smoothing procedures.

4.2. Spatial grid-smoothing. Let us first discuss the spatial grid-smoothing in isolation from the temporal smoothing ($\tau = 0$, $\kappa > 0$). As outlined above, given $\tilde{\mathbf{n}}$, the spatial grid-smoothing amounts to solving for the point concentrations n_i from system (4.3). We present a rather technical lemma that gives the precise form of the solution of this system.

LEMMA 4.1. Let $\nu = \kappa/(\kappa+1)$. The solution of the linear system (4.3) can be represented in the form

$$n_i = (1+2\kappa)^{-1} \sum_{j=0}^{N} \nu^{|i-j|} V_j, \qquad 0 \leqslant i \leqslant N, \tag{4.5}$$

where

$$V_0 = (1+2\kappa)C_2, \quad V_j = \tilde{n}_j, \quad 1 \leqslant j \leqslant N-1, \quad V_N = (1+2\kappa)\nu^{-N}C_1,$$

with, for $k = 1$ and 2,

$$C_k = a_{k1}\tilde{n}_0 + a_{k2}\tilde{n}_N + \kappa(1+2\kappa)^{-1} \sum_{j=1}^{N-1} [a_{k1}\nu^j + a_{k2}\nu^{N-j}]\tilde{n}_j,$$

$$a_{22} = -\kappa/D, \quad a_{11} = \nu^N a_{22}, \quad a_{12} = -(1+\kappa)/D, \quad a_{21} = \nu^{-N} a_{12},$$

$$D = \kappa^2 \nu^N - (1+\kappa)^2 \nu^{-N}.$$

PROOF The characteristic equation of the homogeneous recursion associated with (4.1b) has the roots ν and ν^{-1}, so that the associated homogeneous solution is given by

$$n_{i,hom} = C_1 \nu^{-i} + C_2 \nu^i, \qquad 0 \leqslant i \leqslant N,$$

where C_1, C_2 are arbitrary constants. A particular solution of the inhomogeneous recursion is easily checked to be

$$n_{i,par} = (1+2\kappa)^{-1} \sum_{j=1}^{N-1} \nu^{|i-j|} \tilde{n}_j, \qquad 0 \leqslant i \leqslant N,$$

which is just (4.5) with the first and last term omitted. Hence, the general solution of (4.1b) reads

$$n_i = C_1 \nu^{-i} + C_2 \nu^i + (1+2\kappa)^{-1} \sum_{j=1}^{N-1} \nu^{|i-j|} \tilde{n}_j, \qquad 0 \leqslant i \leqslant N,$$

where the two constants C_1, C_2 serve to match the boundary conditions, i.e. the first and last equation of (4.1b). An elementary calculation leads to (4.5). The introduction of the auxiliary quantities V_j only serves to express the solution in this specific form. □

At first sight expression (4.5) is a bit complicated by the incorporation of the boundary conditions. Neglecting these leads to the more transparent expression

$$n_i = (1+2\kappa)^{-1} \sum_{j=1}^{N-1} \nu^{|i-j|} \tilde{n}_j \tag{4.6}$$

given in [4]. The relevant point in all this is the appearance of the 'smoothing kernel' $\nu^{|i-j|}$. Note that $0 < \nu < 1$.

Next the equidistribution equation (4.1a) is taken into account, i.e. we now simply substitute $\tilde{n}_j = cM_j$ into (4.5) to obtain

$$n_i = c(1+2\kappa)^{-1} \sum_{j=0}^{N} \nu^{|i-j|} \tilde{M}_j, \qquad 0 \leqslant i \leqslant N, \tag{4.7}$$

where $\tilde{M}_j = M_j$ for $1 \leqslant j \leqslant N-1$ and $\tilde{M}_0$ and $\tilde{M}_N$ are defined in exactly the same way as V_0 and V_N in (4.5). Likewise, (4.6) then reads

$$n_i = c(1+2\kappa)^{-1} \sum_{j=1}^{N-1} \nu^{|i-j|} M_j. \tag{4.7'}$$

The following important corollary can thus be made:

COROLLARY 4.1. Taking the anti-diffused concentrations $\tilde{n}_i$ proportional to M_i is equivalent to taking the concentrations n_i proportional to the smoothed monitor value

$$A_i = \sum_{j=0}^{N} \nu^{|i-j|} \tilde{M}_j. \quad \square$$

REMARK 4.1. A trivial consequence of the proportionality of n_i to the positive 'monitor' values A_i, is that all concentrations n_i remain positive which means that the spatial grid-smoothing cannot lead to node crossing. Of course, this is also a direct consequence of the grid ratio condition (3.7). Further it is of interest to note that all values $\tilde{n}_i$ are positive too, which can be concluded from the two following observations. First, all $\tilde{n}_i$ are either positive or negative, as they are proportional to M_i. Second, if all $\tilde{n}_i < 0$, then all n_i must be negative which is a contradiction. $\square$

The motivation behind the spatial grid-smoothing lies in the desirable grid condition (3.7) which serves to control clustering and grid expansion:

THEOREM 4.1. The spatial grid-smoothing restricts the concentrations n_i such that (3.7) is satisfied.
PROOF Consider (4.7). From the inequalities $|i-j-1| \leqslant |i-j|+1$ and $0 < \nu < 1$ we directly deduce

$$n_i / n_{i+1} = \nu^{-1}[\sum_{j=0}^{N} \tilde{M}_j \nu^{|i-j|+1}]/[\sum_{j=0}^{N} \tilde{M}_j \nu^{|i-j-1|}] \leqslant \nu^{-1},$$

because all terms in the numerator are smaller than or equal to the corresponding terms in the denominator. In a similar simple way the left-hand side inequality of (3.7) is proved. $\square$

In the proof, the size of the 'monitor values' M_i plays no role whatsoever, only the fact that they are positive is used. As a matter of fact, for any randomly chosen set of positive values M_i, condition (3.7) is satisfied. This is an attractive feature with respect to robustness, but also makes it difficult to precisely quantify the effect of the space smoothing on the original equidistributing grid. An additional complicating factor, in this respect, is the effect of the 'zero concentration gradient' boundary conditions, although having

$$X_1 - X_0 = X_0 - X_{-1}, \quad X_{N+1} - X_N = X_{N+2} - X_{N+1} \qquad (t \geqslant 0)$$

is a natural restriction and certainly advantageous with respect to spatial accuracy near the boundary. Further, while neglecting the boundaries, the averaged expression (4.7') looks very natural. Our practical experience is that the spatial grid-smoothing procedure leads to a point distribution where the monitor function will determine the relative shape of the distribution and the value of κ and N the level of clustering. We refer to Dorfi and Drury [4] for a numerical illustration.

It is of interest to observe that, for a given N, the choice of κ determines the minimum and maximum interval lengths. In actual application, the minimum should be related to the expected small scale features in the solution to be computed. Suppose that in a transition from small to large space gradients and back, a solution requires a local refinement in a grid with a factor of 10^m. Let N_{loc} be the number of points in this transition region. Then, if the point concentration variation is bounded by $1+1/\kappa$, it follows from

$$(1+1/\kappa)^{0.5N_{loc}} = 10^m,$$

that N_{loc} is at least

$$N_{loc} = 2m\ln(10)/\ln(1+1/\kappa) \approx 4.6m/\ln(1+1/\kappa). \tag{4.8}$$

For example, for $m = 3$ and $\kappa = 1, 2, 3$, we have, respectively, $N_{loc} \approx 20$, 34 and 48. Note that the factor of 0.5 above accounts for the fact that a local grid refinement is supposed to be followed by a local grid expansion. Using the 'rule of thumb' (4.8), one can make a quick (but somewhat crude) estimate of the number of points needed for a particular problem by summing the minimum number required to solve each small scale feature [4].

REMARK 4.2. The range of summation in (4.7) may be changed without violating the grid ratio condition. For example, if only the direct neighbouring monitor values are used, n_i becomes proportional to

$$A_i = \sum_{j=i-1}^{i+1} \nu^{|i-j|} M_j = \nu M_{i-1} + M_i + \nu M_{i+1}, \qquad 1 \leqslant i \leqslant N-1,$$

while condition (3.7) remains valid. This suggests, for example, to realize the grid smoothing directly via the rule

$$(X_i - X_{i-1})A_{i-1} = (X_{i+1} - X_i)A_i, \qquad 2 \leqslant i \leqslant N-1. \tag{4.9}$$

We have not tested this alternative. Note that this technique preserves the 3-point coupling in X, but a drawback is that M_i becomes coupled to M_{i-2}, M_{i-1} and M_{i+1}. Another obvious alternative which comes to mind is to perform the smoothing on the ΔX_i values rather than on the point concentrations. The ΔX_i values are then replaced by

$$G_i = \Delta X_i - \kappa(\kappa+1)(\Delta X_{i+1} - 2\Delta X_i + \Delta X_{i-1}), \qquad \kappa > 0,$$

so as to obtain the grid equation system

$$G_{i-1}M_{i-1} = G_i M_i. \tag{4.10}$$

This smoothing procedure also leads to a grid X satisfying condition (3.7) and to slightly simpler equations (certainly so after the temporal grid-smoothing). As yet we don't know whether this particular choice of smoothing is better or worse than that based on the point concentrations. □

4.3. Temporal grid-smoothing. In terms of equidistribution, temporal grid-smoothing means that $\tau\tilde{n}_i' + \tilde{n}_i$ is taken proportional to the monitor values M_i, as can be seen in equation (4.1a). The introduction of the derivative of the point concentration implies that the grid movement is no longer dictated by solution values at the current time level t, but also depends on past solution values. By preventing the grid movement from adjusting solely to new monitor values at time t, we hope to introduce a smoothing effect so as to avoid oscillatory trajectories $X_i(t)$, $t \geqslant 0$.

Let us examine the solution for $\tilde{n}_i(t)$ in the following form (cf. (4.2)), where Δt represents a typical stepsize that is taken in a numerical time integration:

$$\tilde{n}_i(t) = e^{-\Delta t/\tau}\tilde{n}_i(t-\Delta t) + \int_{t-\Delta t}^{t} \tau^{-1} e^{(s-t)/\tau} c(s) M_i(s)\,ds, \qquad t \geqslant \Delta t, \quad 0 \leqslant i \leqslant N. \tag{4.11}$$

We see that $\tilde{n}_i(t)$ is determined by the sum of $e^{-\Delta t/\tau}\tilde{n}_i(t-\Delta t)$ and a weighted average of values $c(s)M_i(s)$ over the interval $[t-\Delta t, t]$. The weighting is determined by the size of τ and is exponentially decaying for backward time values. One can see that τ acts as a delay factor for the grid movement and that the influence of past solution values is exponentially decaying.

For $\tau \to 0$, $\tilde{n}_i(t) \to c(t)M_i(t)$ whereas $\tilde{n}_i(t) \to \tilde{n}_i(t-\Delta t)$ as $\tau \to \infty$. It follows that for sufficiently large values of τ a non-moving grid results. This means that increasing τ too much will result in a grid that lags too far behind any moving steep spatial transition. On the other hand, too small values for τ render no smoothing effect. In situations where temporal grid-smoothing is advisable, it makes sense to choose τ close to the anticipated Δt-values, so that over one or a few time levels the influence of past monitor values is felt. This suggests allowing τ vary with Δt. Note that so far we have assumed that τ is constant over the whole range of integration.

For an alternative interpretation of the smoothing in time procedure, it is illustrative to examine the implicit Euler discretization (1-st order BDF formula) of the equation

$$-\tau\Delta X_i' (\Delta X_i)^{-2} + (\Delta X_i)^{-1} = cM_i, \qquad t > 0, \quad 0 \leqslant i \leqslant N, \tag{4.12}$$

which arises from (4.1a) by putting $\kappa = 0$ and by substituting

$$dn_i/dt = -\Delta X_i'/(\Delta X_i)^2.$$

Spatial grid-smoothing is omitted here to simplify the presentation. Observe that, apart from the spatial smoothing, it is just this semi-discrete equation which is numerically integrated in time after elimination of the constant of proportionality (see Section 5). Let $\gamma = \tau/\Delta t$. Then the implicit Euler replacement of (4.12) is given by

$$-\gamma(\Delta X_{i,k} - \Delta X_{i,k-1})(\Delta X_{i,k})^{-2} + (\Delta X_{i,k})^{-1} = c_k M_{i,k}, \qquad k \geqslant 1, \quad 0 \leqslant i \leqslant N, \tag{4.13}$$

where $\Delta X_{i,k}$ is the approximation to ΔX_i at time $t = t_k$, $t_k = t_{k-1} + \Delta t$ and $t_0 = 0$. This fully discrete relation shows that, instead of taking $(\Delta X_{i,k})^{-1}$ proportional to $M_{i,k}$, with numerical temporal grid-smoothing we take the entire grid point expression at the left-hand side of (4.13) proportional to $M_{i,k}$. This term contains only grid values. The contribution from the previous time-level should introduce the desired smoothing effect. For the special choice $\tau = \Delta t$, the simple equidistribution relation

$$(\Delta X_{i,k})^{-1}(\Delta X_{i,k-1} / \Delta X_{i,k}) = c_k M_{i,k} \tag{4.14}$$

results. Observe that for the higher order BDF formulas, similar equidistribution relations are found, the only difference being that then $\Delta X_{i,k-1}$ is replaced by a linear combination of such differences over more previous time-levels.

Finally, the following result shows that smoothing in time does not interfere with the grid-ratio condition (3.7):

LEMMA 4.2. The combined space-time grid-smoothing restricts the concentrations n_i such that (3.7) is satisfied.

PROOF For condition (3.7) to hold, the actual size of the values $\tilde{M}_j$ is irrelevant, according to the proof of Theorem 4.1. It is sufficient that all $\tilde{M}_j > 0$. It thus suffices to prove that the solutions $\tilde{n}_i$ of the differential equations (4.1a), as given in (4.2), remain positive for all $t \geqslant 0$, since this implies that all $\tilde{M}_j > 0$ (see Lemma 4.1). First we recall that $\tilde{n}_i(0) > 0$, as shown in Remark 4.1. Now suppose that at a certain time t' the constant of proportionality $c(t)$ becomes negative (if $c(t) > 0$ for all t, the proof is complete). Then, since $M_i > 0$, a right neighbourhood of $t = t'$ exists where all $\tilde{n}_i(t)$ will decrease. Because all entries of the matrix $\mathbf{M}^{-1}$ arising in equation (4.3) are positive (see again Lemma 4.1 or observe that $\mathbf{M}$ is a Stieltjes matrix), all point concentrations $n_i(t)$ will also decrease in this right neighbourhood. This is impossible since the interval $[x_L, x_R]$ is fixed. Hence we have a contradiction for the assumption that $c(t)$ can be negative and the proof is complete. □

REMARK 4.3. The temporal grid-smoothing discussed here is closely related to that suggested in [1,9]. The main difference lies in the fact that in [1,9] the derivative of X_i is introduced directly into an equidistribution equation based on nodal values X_i, whereas here the equation for the concentration values n_i is modified. This leads to a different system of grid equations when written in terms of X_i and X_i'. □

5. The complete semi-discrete system.

5.1. The moving-grid equation in terms of nodal values. Inserting

$$n_i = (\Delta X_i)^{-1}, \quad n_i' = -\Delta X_i' / (\Delta X_i)^2 \tag{5.1}$$

into (3.8) leads to the moving-grid equation system that is actually used. Its i-th equation reads, $2 \leqslant i \leqslant N-1$,

$$\begin{aligned}
&-\tau\left[\frac{\mu}{M_{i-1}(\Delta X_{i-2})^2}\right] X_{i-2}' + \\
&+\tau\left[\frac{\mu}{M_i(\Delta X_{i-1})^2} + \frac{1+2\mu}{M_{i-1}(\Delta X_{i-1})^2} + \frac{\mu}{M_{i-1}(\Delta X_{i-2})^2}\right] X_{i-1}' + \\
&-\tau\left[\frac{\mu}{M_i(\Delta X_{i-1})^2} + \frac{1+2\mu}{M_i(\Delta X_i)^2} + \frac{1+2\mu}{M_{i-1}(\Delta X_{i-1})^2} + \frac{\mu}{M_{i-1}(\Delta X_i)^2}\right] X_i' + \\
&+\tau\left[\frac{\mu}{M_i(\Delta X_{i+1})^2} + \frac{1+2\mu}{M_i(\Delta X_i)^2} + \frac{\mu}{M_{i-1}(\Delta X_i)^2}\right] X_{i+1}' + \\
&-\tau\left[\frac{\mu}{M_i(\Delta X_{i+1})^2}\right] X_{i+2}' =
\end{aligned} \tag{5.2}$$

$$= \left[-\frac{\mu}{\Delta X_{i+1}} + \frac{1+2\mu}{\Delta X_i} - \frac{\mu}{\Delta X_{i-1}} \right] / M_i - \left[-\frac{\mu}{\Delta X_i} + \frac{1+2\mu}{\Delta X_{i-1}} - \frac{\mu}{\Delta X_{i-2}} \right] / M_{i-1}.$$

The 1-st and N-th equation slightly differ due to the boundary conditions and are easily found. Note that, away from the boundary, the nodal points X_{i+2}, X_{i+1}, X_i, X_{i-1}, X_{i-2} are coupled with the nodal point velocities X'_{i+2}, X'_{i+1}, X'_i, X'_{i-1}, X'_{i-2} and the monitor values M_{i-1}, M_i.

For future reference, system (5.2), together with the 1-st and N-th equation, is represented in the form of the nonlinear ODE system

$$\tau B(X,U)X' = g(X,U) \tag{5.3}$$

where B is the N x N penta-diagonal matrix associated to the left-hand side part of (5.2). In order that we have a genuine ODE system, it is required that $B(X,U)$ is non-singular for any X, U. If no time smoothing is carried out, i.e. $\tau = 0$, we are left with the algebraic system

$$g(X,U) = 0, \tag{5.4}$$

which represents the equidistribution relation combined with spatial grid-smoothing.

REMARK 5.1. An alternative and somewhat simpler moving-grid equation system that has essentially the same smoothing properties as (5.2) is obtained by putting $\mu = 0$ in its left-hand side. This renders B tri-diagonal and symmetric positive definite. In terms of point concentrations, the resulting system reads $\tau n'_i + \tilde{n}_i = cM_i$ (cf. (4.1a)), which shows that the temporal grid-smoothing is carried out on the concentration values n_i rather than on $\tilde{n}_i$. □

5.2. The complete semi-discrete system and its numerical integration. Systems (2.4) and (5.3) together form the complete semi-discrete system that is numerically integrated in time,

$$\tau BX' = g, \quad t > 0, \quad X(0) \text{ given}, \tag{5.5a}$$

$$U' - X' \circ D = F, \quad t > 0, \quad U(0) \text{ given}. \tag{5.5b}$$

In case of Dirichlet boundary conditions, the total number of equations and unknowns is $(NPDE+1)*N$, where $NPDE$ is the number of components of the original PDE problem (1.1). For other types of boundary conditions, the number of equations and unknowns slightly differs. The supposed non-singularity of the matrix B trivially implies that for $\tau > 0$ we have a genuine ODE system; for $\tau = 0$ we have a DAE system of index one. The large matrix that multiplies the derivatives X', U' in (5.5) has a rather simple, lower block-triangular structure. We cannot exploit this advantage since the system is numerically integrated with an implicit method. The Newton iteration matrix involved contains the partial derivative matrices of g and F with respect to X and U, or approximations thereof, and hence the lower block-triangular structure is lost. It is therefore computationally more attractive to change the order of unknowns so as to obtain a band-matrix. When using the order $\cdots, U_{i-1}, X_{i-1}, U_i, X_i, U_{i+1}, X_{i+1}, \cdots$, the band-width for the Newton matrix becomes $4*(NPDE+1)+1$. This is based on the fact that we work with standard 3-point central differences for the spatial operators, that X is 5-point coupled, and that the monitor M_i is given by (3.4).

For the numerical integration of the above semi-discrete system, one can use, in principle, any stiff method designed to solve linearly implicit systems of the present type. The results of the next section have been obtained with the BDF code DASSL (version of 830315) [11]. A similar code is the LSODI-based BDF code of the SPRINT package [2, 3]. We have experimented with both these codes (see also [7]) and since they are very much alike, the choice between the two should be of minor influence to the performances observed. This indeed turns out to be true in the case of successful runs. However, in some cases we have experienced a rather different performance. With both codes and for different problems runs were interrupted due to fatal Newton errors, especially so when using extremely fine grids. This could be due to the fact that in our experiments the local error and Newton convergence test has been applied to X_i and not to ΔX_i. Also, with moving grid methods a poor prediction of X_i can be generated in the preparation of the actual BDF step, thus causing convergence problems for the Newton solver. These aspects need further attention (e.g. in a study along the lines of Petzold and Lötstedt [13]).

From the user's point of view it is of interest to note that DASSL, and likewise the stiff solver

of SPRINT, are used in the same way as in the conventional, non-moving MOL approach. Apart from providing a subroutine for the semi-discrete system (numerical differencing for Jacobians was used) and specifying the initial values and required output times, one must define only the local absolute and relative error tolerances, *atol* and *rtol*, the desired local error norm, and an optional initial time-step value Δt_0. Throughout we have used $atol = rtol := TOL$ and the standard weighted Euclidean norm; TOL and Δt_0 will be specified with the three example problems in the next section.

The method parameters for the grid are N, the number of moving points, the grid-smoothing parameters κ and τ, and the constant α of the monitor (cf. (3.4))

$$M_i = \left[\alpha + NPDE^{-1}\sum_{j=1}^{NPDE}(U_{i+1,j} - U_{i,j})^2/(X_{i+1} - X_i)^2\right]^{\frac{1}{2}} \tag{5.6}$$

The choice $\alpha = 1$ yields the common arc-length monitor; this we have used throughout, unless noted otherwise. For κ the default value 2 was selected, while τ was simply put equal to zero. Additional tests have shown that for the three example problems below the temporal grid-smoothing is redundant, which is of course a favourable situation. We wish to emphasize, however, that for other problems a positive value for τ may lead to a better performance. As observed previously, this aspect deserves more attention.

6. Numerical examples. We present numerical results for three different example problems. In the plots the solid or dashed lines represent accurate reference solutions (obtained from [16]) while the marks represent the generated PDE approximations. Integration information, which serves to show the time-stepping efficiency of the process, is presented in terms of $STEPS$ = total number of successful time steps, $JACS$ = total number of Jacobian evaluations, and BS = total number of back solves. The two latter quantities determine, to a great extent, the CPU time needed to complete the integration over the specified time interval.

6.1. Problem I: The Dwyer-Sanders flame propagation model. This model, first proposed as a test example in [5], simulates several basic features of flame propagation. It has two components, a mass density u and a temperature v. The PDE system is given by

$$\partial u/\partial t = \partial^2 u/\partial x^2 - uf(v), \qquad 0 < x < 1, \quad 0 < t \leqslant .006,$$

$$\partial v/\partial t = \partial^2 v/\partial x^2 + uf(v), \qquad 0 < x < 1, \quad 0 < t \leqslant .006,$$

where $f(v) = 3.52*10^6\exp(-4/v)$. The initial functions are $u(x,0) = 1$, $v(x,0) = 0.2$ $(0 \leqslant x \leqslant 1)$ and the boundary conditions are given by

$$\partial u/\partial x\ (0,t) = \partial v/\partial x\ (0,t) = 0,$$

$$\partial u/\partial x\ (1,t) = 0 \text{ and } v(1,t) = 0.2 + t/0.0002\ (t \leqslant 0.0002),\ v(1,t) = 1.2\ (t \geqslant 0.0002).$$

The given function for v at the right boundary represents a heat source that generates a steep flame front. When v reaches its maximum, this front starts to propagate from right to left at a relatively high speed. The speed of propagation of the front is almost constant. At the final time $t = 0.006$, the front has come close to the left boundary.

The initial grid $X(0)$ was taken uniform with $N = 40$. A uniform start grid provides a difficult test since the method rapidly must refine near $x = 1$ in order to accurately simulate the fast generation of the front. The remaining parameters to be specified are $\Delta t_0 = 10^{-6}$ and $TOL = 10^{-4}$. In passing we note that the error control mechanism of DASSL may reduce the specified initial stepsize Δt_0. In the present experiment Δt_0 was reduced to $.1276*10^{-6}$.

Fig. 6.1 shows plots of the grid and the computed temperature front for a range of output times. The costs of the run amount to $STEPS = 148$, $BS = 410$, $JACS = 52$. Inspection of the plots justifies the conclusion that the grid movement and the accuracy of the approximation are very satisfactory over the entire time interval (also for the density which is not shown here). The small lump for early times is genuine and is contaminated with only very little overshoot (not visible here). For later times the numerical front is slightly too fast. These small errors are spatial, i.e., they remain if many more time steps are spent and disappear if more space points are used. For example, for $N = 80$ and $TOL = 10^{-4}$, which costs $STEPS = 164$, $BS = 492$, $JACS = 66$, the

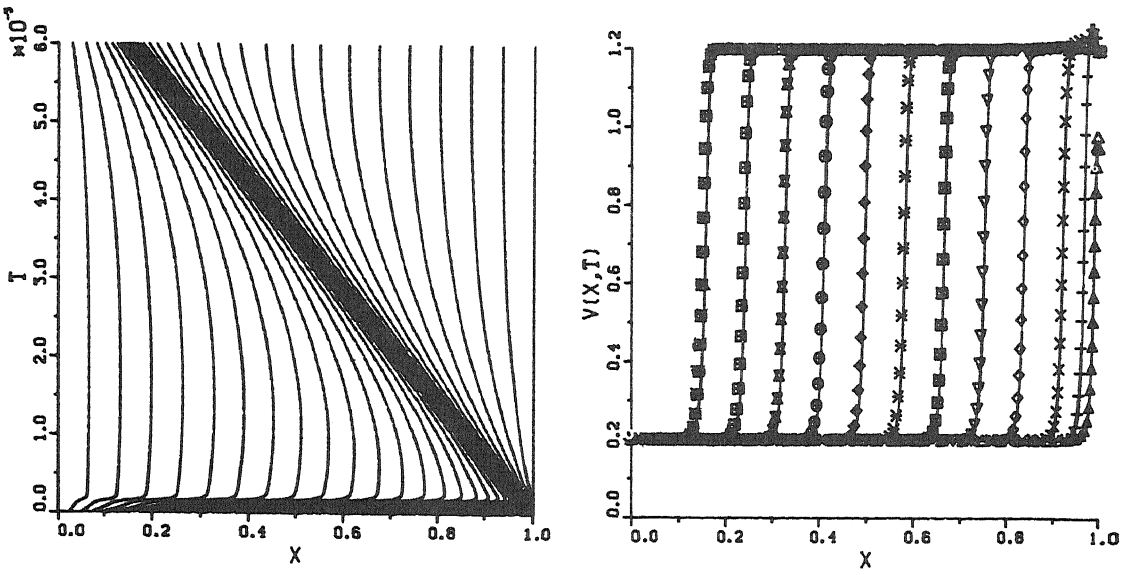

FIGURE 6.1. Problem I ($N = 40$). Grid and temperature front at times $t = .15*10^{-3}$, $.3*10^{-3}$, $.6*10^{-3}$ $(.6*10^{-3)}$ $.6*10^{-2}$.

approximations are exact up to plotting accuracy. Admittedly, 80 moving points for this problem is quite a lot. It turns out that a relatively large number of points are wasted in the front, especially for $N = 80$, while there are not too many near the foot and the top. We owe this to the arc-length monitor. A comparison with results shown in [16], where a second derivative monitor is used that deemphasizes the front and places more points where the curvature is largest, suggests that implementation of a second derivative monitor in the current algorithm would improve the spatial accuracy.

6.2. Problem II: A 'hot spot' problem from combustion theory. This problem is described in Adjerid & Flaherty [1] as a model of a single-step reaction with diffusion and reads

$$\partial u/\partial t = \partial^2 u/\partial x^2 + D(1 + a - u)\exp(-\delta/u), \quad 0 < x < 1, \quad t > 0,$$

$$\partial u/\partial x\ (0,t) = 0, \quad u(1,t) = 1, \quad t > 0,$$

$$u(x,0) = 1, \quad 0 \leqslant x \leqslant 1,$$

where $D = \mathrm{Re}^{\delta}/(a\delta)$ and R, δ, a are constant numbers. The solution represents a temperature of a reactant in a chemical system. For small times the temperature gradually increases from unity with a 'hot spot' forming at $x = 0$. At a finite time, ignition occurs, causing the temperature at $x = 0$ to increase very rapidly to $1 + a$. A flame front then forms and propagates towards $x = 1$ at high speed. The degree of difficulty of the problem is very much determined by the value of δ. Following [1, 7, 16], we have selected the problem parameters $a = 1$, $\delta = 20$, $R = 5$. The problem reaches a steady state once the flame propagates to $x = 1$. For the current choice of parameters, the steady state is reached slightly before time $t = 0.29$, which we take as the end point. We use times $t = 0.26$, 0.27, 0.28, 0.29 for output. It is noted that for $t = 0.26$ the reference solution is not sufficiently accurate near $x = 0$, but it is very accurate for the remaining output times [16].

For the numerical process, two solution phases should be distinguished, viz. the formation of the 'hot spot' with the flame front (the ignition phase) and the propagation of this front to the right end point $x = 1$ (the propagation phase). Accurate handling of the formation of the 'hot spot' and the ignition is of importance. The ignition proceeds very rapidly, causing a widely different time scale, so that variable steps in time are a necessity. A difficulty is that the code must detect the start of the ignition very accurately at the right time, so that the step size can be rapidly reduced to a level small enough to simulate this ignition in a sufficiently accurate way. Small errors at this time point result in significantly larger global errors later on. Some trial and error tests have revealed that the BDF code needs at least a time tolerance value TOL of 10^{-5}, while using an initial step size of 10^{-5} [7]. These are the values we have used. The small tolerance does not cause any problems with

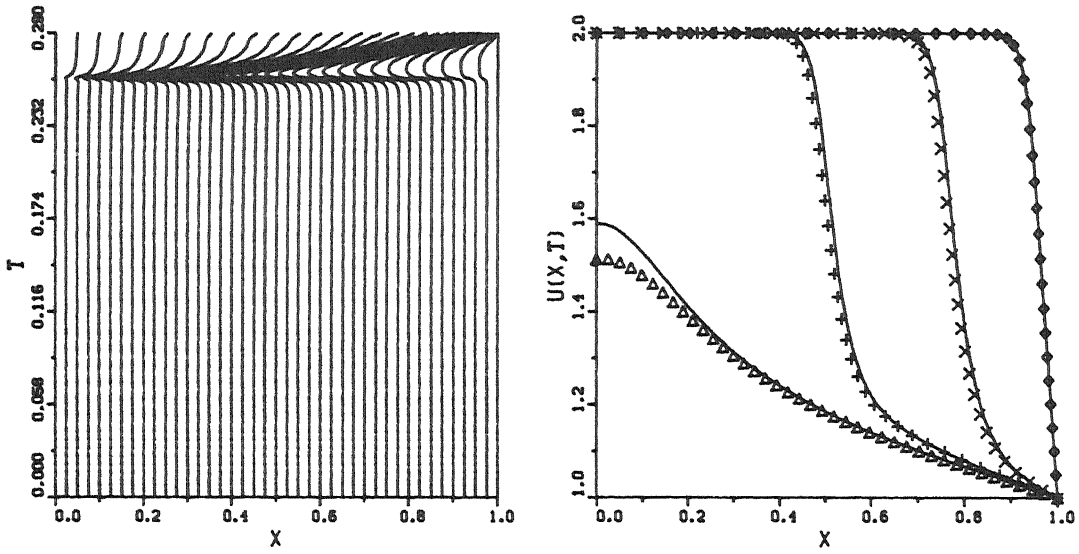

FIGURE 6.2. Problem II ($N = 40$). Grid and flame front at times $t = .26, .27, .28, .29$.

the high-order integrators.

Figure 6.2 shows a plot of the computed grid and the flame front on this grid for the four specified output times, using 40 moving nodes. The costs of this experiment amount to $STEPS = 136$, $BS = 382$, $JACS = 35$. The 'hot spot' nature is clearly visible from the grid. The numerical flame appears to be too slow, but is almost in the right position for $t = .27$ and .28 (the plot at $t = .29$ is the steady state solution). As for the previous problem, it is the spatial error that dominates and decreasing TOL gives no further improvement. Changing N to 80 yields a very accurate solution (up to plotting accuracy), while there is no great increase in the number of time steps, viz. $STEPS = 159$, $BS = 423$, $JACS = 37$. Inspection of the solution shows that, similar as for Problem I, there are quite a few points in the flame front, but not very many at the top. Also here a curvature monitor would improve the spatial accuracy, see [16] for comparison. Finally we refer to [7] where results for a range of values $\tau > 0$ are shown.

6.3. Problem III: Waves travelling in opposite directions. Our third example problem is a two-component, semi-linear hyperbolic system, the solution of which is given by two waves travelling in opposite directions (copied from [10], see also [7, 16]). The system is

$$\partial u/\partial t = -\partial u/\partial x - 100uv,$$

$$\partial v/\partial t = +\partial v/\partial x - 100uv,$$

for $t > 0$ and $-0.5 < x < 0.5$, and the solution is subjected to homogeneous Dirichlet boundary conditions and to the initial condition

$$u(x, 0) = 0.5(1 + cos(10\pi x)) \text{ for } x \in [-0.3, -0.1] \text{ and } u(x, 0) = 0 \text{ otherwise,}$$

$$v(x, 0) = 0.5(1 + cos(10\pi x)) \text{ for } x \in [+0.1, +0.3] \text{ and } v(x, 0) = 0 \text{ otherwise.}$$

Note that these are functions with a mere C^1 continuity, which represent wave pulses located at $x = -0.2$ and $x = 0.2$, respectively. Initially, while the pulses are separated, the nonlinear term $100uv$ vanishes, so that for $t > 0$ these waves start to move with speed 1 and without change of shape, u to the right and v to the left. At $t = 0.1$ they collide at $x = 0$ and the nonlinear term becomes nonzero, resulting in a nonlinear interaction leading to changes in the shapes and speeds of the waves. Specifically, the crests of the waves collide a little beyond $t = 0.25$ and they have separated again at approximately $t = 0.3$, so that from this time on the solution behaviour is again dictated by the linear advection terms. At the nonlinear interaction, the pulses lose their symmetry and experience a decrease in amplitude.

DASSL has been applied with $N = 40$, $TOL = 10^{-3}$ and $\Delta t_0 = 10^{-5}$. For convenience, we have again used a uniform start grid. However, unlike the two previous problems, this uniform grid does not satisfy the constraint (5.4) which it should if $\tau = 0$. To circumvent this start up difficulty,

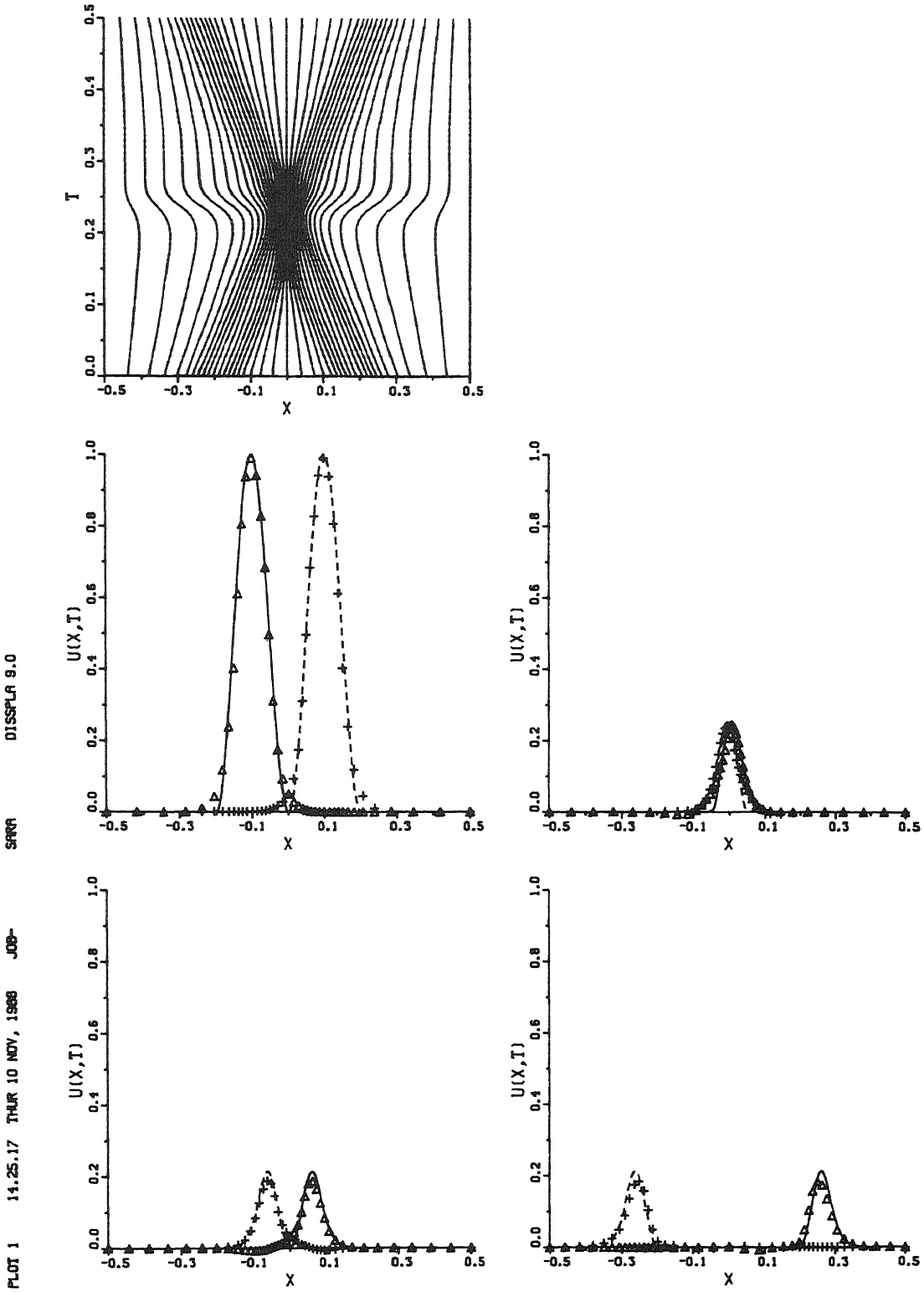

FIGURE 6.3. Problem III ($N = 40$). Grid and solution at times $t = 0.1, 0.25, 0.3, 0.5$.

we have simply put τ small (10^{-8}), so that we are in an ODE situation and any grid can be used to start up the time integrator. DASSL then lowers our guess of Δt_0 to $.3*10^{-10}$ and completes the integration using 111 successful steps (46 up to $t = 10^{-3}$), 327 back-solves and 78 Jacobian evaluations. The value $\tau = 10^{-8}$ is of course excessively small, so that, very soon after the start, we are very close to the $\tau = 0$ situation. It is emphasized that if $\tau = 0$ and we start on a grid satisfying (5.4), or choose τ larger than 10^{-8} in case of a uniform start grid, the number of required steps will be smaller (see also [7]).

Fig. 6.3 shows the grid and the numerical approximations at the specified output times. We see that the grid movement nicely mimics the interaction and point out that the visible inaccuracies are due to a somewhat optimistic choice for *TOL* and the number of points. These inaccuracies will vanish if more points are used and again we remark that a curvature monitor would probably lead to significantly more accuracy (see [16]). In the present experiment we have replaced the (regularization) constant $\alpha = 1$ of the arc-length monitor by 0.1. The reason is that when the waves have separated they are no longer very steep, with the result that the value 1.0 is somewhat too large for obtaining sufficient refinement in the vicinity of the two waves, at least for $N = 40$. With this number of points it is also necessary that, after the pulses separate, the grid refines properly in the vicinity of the waves, else spurious oscillations become visible. Recall that after the separation we are just solving the first order hyperbolic model problem using standard central differences. This experiment shows that it is desirable that the regularization constant of the arc-length monitor function be made solution-dependent, in some way or another. On the other hand, the results published in [16] indicate that with a curvature monitor this is less important.

7. Conclusions. This work has been carried out in connection with a joint CWI/Shell project on 'Adaptive Grids'. One of the aims of this project is to develop a reliable, robust and efficient 1D moving-grid method, based on the method of lines, which can be used in almost the same easy way as existing MOL packages that integrate on a non-moving grid. The demand of ease of use requires that, as far as possible, the user should be relieved from fine tuning the grid movement. The results obtained so far justify the conclusion that the technique discussed in this paper goes a long way towards fulfilling the above requirements.

An important feature is the grid-smoothing capability involving the two method parameters κ and τ. The meaning of κ is very clear and for general use κ can be taken equal to, say, 1 or 2. At the present stage of development, the actual choice to be made for τ is less clear. Fortunately, our numerical experience indicates that in many cases it is possible to simply put $\tau = 0$ or to select τ really small, so that the grid movement is almost exclusively dictated by the spatial equidistribution at the forward time level. The numerical results also suggest very clearly to implement a curvature monitor as in [16].

Finally we should mention that, in a few instances, the stiff solvers interrupted the integration due to a Newton convergence test failure, especially so when using extremely fine grids. This could be due to the fact that, in the experiments reported, the local error and Newton convergence test was applied to X_i and not to ΔX_i. Also poor prediction of the velocities may have caused difficulties for the Newton solver. These aspects need further attention (e.g. in a study along the lines of Petzold and Lötstedt [13]).

References

1. S. Adjerid and J.E. Flaherty, *A Moving Finite Element Method with Error Estimation and Refinement for One-Dimensional Time Dependent Partial Differential Equations*, SIAM J. Numer. Anal., 23 (1986), pp. 778-796.

2. M. Berzins and R.M. Furzeland, *A User's Manual for SPRINT - A Versatile Software Package for Solving Systems of Algebraic, Ordinary and Partial Differential Equations: Part 1 - Algebraic and Ordinary Differential Equations*, Report TNER.85.058, Thornton Research Centre, Shell Research Ltd., U.K., 1985.

3. M. Berzins and R.M. Furzeland, *A User's Manual for SPRINT - A Versatile Software Package for Solving Systems of Algebraic, Ordinary and Partial Differential Equations: Part 2 - Solving Partial Differential Equations*, Report No. 202, Department of Computer Studies, The University of Leeds, 1986.

4. E.A. Dorfi and L. O'C. Drury, *Simple Adaptive Grids for 1-D Initial Value Problems*, J. Comput.

Phys., 69 (1987), pp. 175-195.
5. H.A. Dwyer and B.R. Sanders, *Numerical Modeling of Unsteady Flame Propagation,* Report SAND77-8275, Sandia National Laboratories, Livermore, USA, 1978.
6. R.M. Furzeland, *The Construction of Adaptive Space Meshes for the Discretisation of Ordinary and Partial Differential Equations,* Report TNER.85.022, Thornton Research Centre, Shell Research Ltd., U.K., 1985.
7. R.M. Furzeland, J.G. Verwer and P.A. Zegeling, *A Numerical Study of Three Moving Grid Methods for One-Dimensional Partial Differential Equations which are based on the Method of Lines,* Report NM-R8806, Centre for Mathematics and Computer Science (CWI), Amsterdam, 1988. (Submitted to J. Comput. Physics.)
8. A.C. Hindmarsh, *ODE Solvers for Use with the Method of Lines,* in *Advances in Computer Methods for Partial Differential Equations-IV,* R. Vichnevetsky and R.S. Stepleman, ed., IMACS, 1981, pp. 312-316.
9. J.M. Hyman and M.J. Naughton, *Static Rezone Methods for Tensor-product grids,* in *Proc. SIAM-AMS Conference on Large Scale Computation in Fluid mechanics,* SIAM, Philadelphia, PA, 1984.
10. N.K. Madsen, *MOLAG: A Method of Lines Adaptive Grid Interface for Nonlinear Partial Differential Equations,* in *PDE Software: Modules, Interfaces and Systems,* B. Engquist and T. Smedsaas, ed., North-Holland, 1984.
11. L.R. Petzold, *A description of DASSL: A Differential/Algebraic System Solver,* in *IMACS Trans. on Scientific Computation,* R.S. Stepleman, ed., 1983.
12. L.R. Petzold, *Observations on an Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations,* Applied Numerical Mathematics, 3 (1987), pp. 347-360.
13. L.R. Petzold and P. Lötstedt, *Numerical Solution of Nonlinear Differential Equations with Algebraic Constraints II: Practical Implications,* SIAM J. Sci. Stat. Comput., 7 (1986), pp. 720-733.
14. R.F. Sincovec and N.K. Madsen, *Software for Nonlinear Partial Differential Equations,* ACM Trans. Math. Software, 1 (1975), pp. 232-260.
15. R.F. Sincovec and N.K. Madsen, *Algorithm 494, PDEONE, Solutions of Systems of Partial Differential Equations,* ACM Trans. Math. Software, 1 (1975), pp. 261-263.
16. J.G. Verwer, J.G. Blom and J.M. Sanz-Serna, *An Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations,* Report NM-R8804, Centre for Mathematics and Computer Science (CWI), Amsterdam, 1988. (To appear in J. Comput. Phys. in 1989.)