Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.G. Blom, J.G. Verwer

On the use of the arclength and curvature monitor in a moving-grid method which is based on the method of lines

Department of Numerical Mathematics Note NM-N8902 July

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

On the Use of the Arclength and Curvature Monitor in a Moving-Grid Method which is Based on the Method of Lines

J.G. Blom, J.G. Verwer
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

This note deals with the numerical solution of one-space dimensional, time-dependent partial differential equations by means of a moving-grid method that is based on the Method of Lines. In this moving-grid method, the partial differential equation is discretized on grid lines that move continuously in the space-time domain. The grid movement is dictated by a space monitor through the standard equidistribution principle. Two monitors are discussed, viz. the arclength and the curvature monitor. The Method of Lines approach works well with the arclength monitor. This monitor, however, concentrates nodal points in steep fronts and neglects, to some extent, sharp transition regions. In this respect, the curvature monitor is to be preferred. Unfortunately, the curvature monitor turns out to be less attractive with regard to the time-stepping process. Specifically, the time-stepping process appears to be much more expensive than with the arclength monitor. This discrepancy is discussed through a number of numerical experiments.

1980 Mathematics subject classification: Primary:65M50. Secondary:65M20.
1987 CR Categories: G.1.8.
Key Words & Phrases: partial differential equations, time-dependent problems, Method of Lines, Lagrangian methods, moving grids.

Note: This work has been carried out in connection with a joint CWI/Shell project on 'Adaptive Grids'. Financial support is obtained 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. CWI 55.092).

1. Introduction

In [5] three different moving-grid methods were evaluated to solve systems of one-space dimensional partial differential equations that have solutions with large space-time gradients. These three methods are all based on the Method of Lines approach. One method, first suggested by Dorfi & Drury[4], has been investigated further in [3, 6]. In this method, which we will describe in Section 2, the PDE is discretized on grid lines that are continuous in time and an extra set of equations to move the grid is simultaneously solved. These grid equations strive to equidistribute in space a monitor function that typically signals large spatial gradients. In [3, 5, 6] we successfully used the so-called arclength monitor but we suggested there that a grid equation equidistributing the curvature would be more appropriate. This was based on the observation that, generally spoken, the arclength monitor places too many points in steep fronts where they are not needed and neglects transition regions. Curvature monitors perform significantly better in this respect, as can be seen when comparing the results in [7], with those obtained with the arclength monitor in [5, 6].

Experiments reveal however, that solving the ODE system arising from the semi-discretization of the PDE and the moving-grid equations with the curvature monitor, is not only less robust, but also considerably more expensive than with the arclength monitor. The aim of this note is to illustrate this by discussing some of these experiments. In Section 3 we will show for two scalar problems that although the distribution of the grid points and the accuracy of the result is in general better for the

curvature monitor, the moving-grid method behaves with this monitor less satisfactorily with regard to efficiency and robustness. We therefore consider the curvature monitor less appropriate than the arclength monitor for use in the present straightforward Method of Lines approach.

2. The method used

In this section we give a brief description of the moving-grid method and of the monitors used. We also discuss how we determine a starting grid in case the initial solution can not be represented well enough on a uniform grid. We formulate the method for a scalar PDE, but the extension to systems of equations is obvious and can be found in the cited literature.

2.1. The moving-grid method

The moving-grid technique that controls the grid movement in space and time is due to Dorfi & Drury[4] and is analyzed further in [6]. Here we will only enumerate the formulae involved.

Given a PDE

$$u_t = f(u,x,t), \qquad x_L < x < x_R, \quad t > t_0, \tag{2.1}$$

it is transformed to its Lagrangian form

$$\dot{u} - u_x \dot{x} = f(u,x,t), \quad \text{with } \dot{u} \text{ the total time-derivative,} \tag{2.2}$$

and discretized in space on N, time-dependent, grid points:

$$x_L = X_0 < \cdots < X_i(t) < X_{i+1}(t) < \cdots < X_{N+1} = x_R. \tag{2.3}$$

We then have

$$\dot{U}_i - \frac{(U_{i+1} - U_{i-1})}{(X_{i+1} - X_{i-1})} \dot{X}_i = F_i, \quad t > t_0, \quad 1 \leqslant i \leqslant N, \tag{2.4}$$

where U_i and F_i represent the semi-discrete approximation to the exact PDE solution u, resp. the righthand-side function $f(u,x,t)$, at the point $(x,t) = (X_i(t),t)$. In the examples discussed in this note, F_i is based on the same standard 3-point difference formulas that were used in [5-7]. Also boundary conditions for (2.1) are implemented as in [5-7].

The additional equations for the time-dependent grid points X_i are formulated in terms of the so-called point concentration of the grid,

$$n_i := (\Delta X_i)^{-1}, \quad \Delta X_i := X_{i+1} - X_i,$$

and read

$$\frac{\tilde{n}_{i-1} + \tau \dot{\tilde{n}}_{i-1}}{M_{i-1}} - \frac{\tilde{n}_i + \tau \dot{\tilde{n}}_i}{M_i} = 0, \quad 1 \leqslant i \leqslant N, \tag{2.5}$$

with

$$\tilde{n}_i := n_i - \kappa(\kappa+1)(n_{i+1} - 2n_i + n_{i-1}), \quad 0 \leqslant i \leqslant N, \tag{2.6}$$

$$\text{(assuming } n_{-1} := n_0, \; n_N =: n_{N+1}).$$

M_i is the monitor function which will be specified below, $\tau \geqslant 0$ is a temporal grid-smoothing parameter and $\kappa \geqslant 0$ denotes a spatial grid-smoothing parameter. For the precise meaning of these parameters and the grid equation itself we refer to [6]. Here we note that the spatial grid smoothing, as introduced in (2.6), guarantees that the ratios of adjacent grid intervals satisfy the inequality $\kappa/(\kappa+1) \leqslant \Delta X_i / \Delta X_{i+1} \leqslant (\kappa+1)/\kappa$. In applications, κ is normally set equal to 1 or 2.

The temporal grid-smoothing introduces the derivative of the point concentration in the grid equation. This serves to prevent the grid movement from adjusting solely to new monitor values. Instead, the introduction of $\tau\dot{n}$ 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. 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. In actual application τ is chosen close to zero, depending on the time scale of the problem, such that, over one or a few time levels, the influence of past monitor values is felt.

Equations (2.4) and (2.5) are combined into the system of ODEs

$$\begin{aligned} \dot{U} - D\dot{X} &= F \\ \tau B\dot{X} &= g \end{aligned} \quad , \tag{2.7}$$

where D is a diagonal matrix and B a penta-diagonal matrix, both of dimension $N \times N$. D and B are solution-dependent matrices, and F and g solution-dependent vectors, containing all information about the monitor function and the righthand-side of the PDE itself, respectively. We note in passing that the grid equation (2.5) has now been formulated in terms of the nodal values. By way of illustration we give its i-th component ($2 \leqslant i \leqslant N-1$) where $\mu = \kappa(\kappa+1)$:

$$\begin{aligned} &- \tau \left[\frac{\mu}{M_{i-1}(\Delta X_{i-2})^2} \right] \dot{X}_{i-2} + \\ &+ \tau \left[\frac{\mu}{M_i(\Delta X_{i-1})^2} + \frac{1+2\mu}{M_{i-1}(\Delta X_{i-1})^2} + \frac{\mu}{M_{i-1}(\Delta X_{i-2})^2} \right] \dot{X}_{i-1} + \\ &- \tau \left[\frac{\mu}{M_i(\Delta X_{i-1})^2} + \frac{1+2\mu}{M_i(\Delta X_i)^2} + \frac{1+2\mu}{M_{i-1}(\Delta X_{i-1})^2} + \frac{\mu}{M_{i-1}(\Delta X_i)^2} \right] \dot{X}_i + \\ &+ \tau \left[\frac{\mu}{M_i(\Delta X_{i+1})^2} + \frac{1+2\mu}{M_i(\Delta X_i)^2} + \frac{\mu}{M_{i-1}(\Delta X_i)^2} \right] \dot{X}_{i+1} + \\ &- \tau \left[\frac{\mu}{M_i(\Delta X_{i+1})^2} \right] \dot{X}_{i+2} = \\ &= \left[-\frac{\mu}{\Delta X_{i+1}} + \frac{1+2\mu}{\Delta X_i} - \frac{\mu}{\Delta X_{i-1}} \right] / M_i - \left[-\frac{\mu}{\Delta X_i} + \frac{1+2\mu}{\Delta X_{i-1}} - \frac{\mu}{\Delta X_{i-2}} \right] / M_{i-1}. \end{aligned} \tag{2.8}$$

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 $\dot{X}_{i+2}$, $\dot{X}_{i+1}$, $\dot{X}_i$, $\dot{X}_{i-1}$, $\dot{X}_{i-2}$ and the monitor values M_{i-1}, M_i. Needless to say that the grid equation $\tau B\dot{X} = g$ is very nonlinear.

The linearly implicit ODE system (2.7) is solved by the BDF integrator SPGEAR of the SPRINT package[2] in the usual way. Note that for $\tau = 0$ this ODE system reduces to a DAE system.

2.2. The monitors

The monitor function used in [3, 6],

$$M_i := \left[\alpha + (U_{i+1} - U_i)^2/(X_{i+1} - X_i)^2\right]^{1/2}, \tag{2.9}$$

is the semi-discretization of the first derivative solution functional

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

In case one wants to equidistribute the curvature a second derivative solution functional like

$$m(u) = (\alpha + (u_{xx})^2)^{1/4}$$

has to be discretized. Here, we use the following,

$$M_i := \left[\left[\alpha + (U_{xx_i})^2\right]^{1/4} + \left[\alpha + (U_{xx_{i+1}})^2\right]^{1/4}\right]/2, \tag{2.10}$$

where U_{xx_i} is the finite difference approximation of $u_{xx}(X_i,t)$, i.e.,

$$U_{xx_i} = \frac{\dfrac{U_{i+1} - U_i}{X_{i+1} - X_i} - \dfrac{U_i - U_{i-1}}{X_i - X_{i-1}}}{(X_{i+1} - X_{i-1})/2}, \quad \text{for } i = 1, \cdots, N, \tag{2.11}$$

and

$$U_{xx_0} = U_{xx_1}, \quad U_{xx_{N+1}} = U_{xx_N}.$$

This curvature monitor naturally arises if the trapezium rule is applied for discretizing the equidistribution rule

$$\int_{X_i}^{X_{i+1}} m(u)dx = \text{constant}. \tag{2.12}$$

Instead of using the trapezium rule for discretizing (2.12) one can of course also apply the midpoint rule. It is our experience however, that (2.10) performs slightly better than its midpoint-rule counterpart. We have also considered a few other curvature type monitors (cf. Section 4). Among the curvature monitors we have experimented with, (2.10) seems to be the best possible. The monitors (2.9) and (2.10) will be compared in the next section.

2.3. The initial grid

In [3, 5, 6] a uniform distribution of the grid points has been used at the initial time, even when the solution is not flat. This means that the grid equation (2.5) with $\tau = 0$ is normally not satisfied at the initial time. It can be seen from the pictures of the grid trajectories in the above mentioned papers that in these cases the grid is adapting very fast to the solution (hence the use of a uniform start grid is allowed). To avoid eventual undesirable influences from a wrong initial grid on the later performance of the method, we decided to use here an adapted initial grid. To compute this grid we have used the same technique as Dorfi & Drury[4]. We start with a uniform grid and integrate the grid equation (2.5) forward in time with **SPGEAR**, using exact values $U_i = u(X_i, t_0)$, until a stationary grid distribution is reached (marching to steady state).

3. Numerical experiments

We tested the moving-grid method with the monitors (2.9) and (2.10) on two scalar problems. The second problem was used before as a test case in a.o. [6, 7].

3.1. Problem I: Burgers' equation

The first problem is the well-known Burgers' equation

$$\partial u/\partial t = -\partial f(u)/\partial x + \epsilon \partial^2 u/\partial x^2, \quad 0 < x < 1, \quad t > 0, \quad f(u) = u^2/2, \tag{3.1}$$

with Dirichlet boundary conditions and solution

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

where

$$r_1 = \exp\left[-\frac{x-0.5}{20\epsilon} - \frac{99t}{400\epsilon}\right], \quad r_2 = \exp\left[-\frac{x-0.5}{4\epsilon} - \frac{3t}{16\epsilon}\right],$$

$$r_3 = \exp\left[-\frac{x-3/8}{2\epsilon}\right].$$

For ϵ small this solution initially contains two layers which merge in the subsequent evolution into one layer. For $\epsilon = 0.001$ this merging occurs at $t \approx 0.55$.

3.1.1. The initial grid.

For the monitors (2.9) and (2.10) the initial grid has been determined using the technique described in Section 2.3 with $\tau = 10^{-3}$ and a local error tolerance in SPGEAR of 10^{-3}. To decide when the grid distribution was stationary, we used the criterion $\dot{X}_i < 10^{-8}$ for all internal points $i = 1, \cdots, N$.

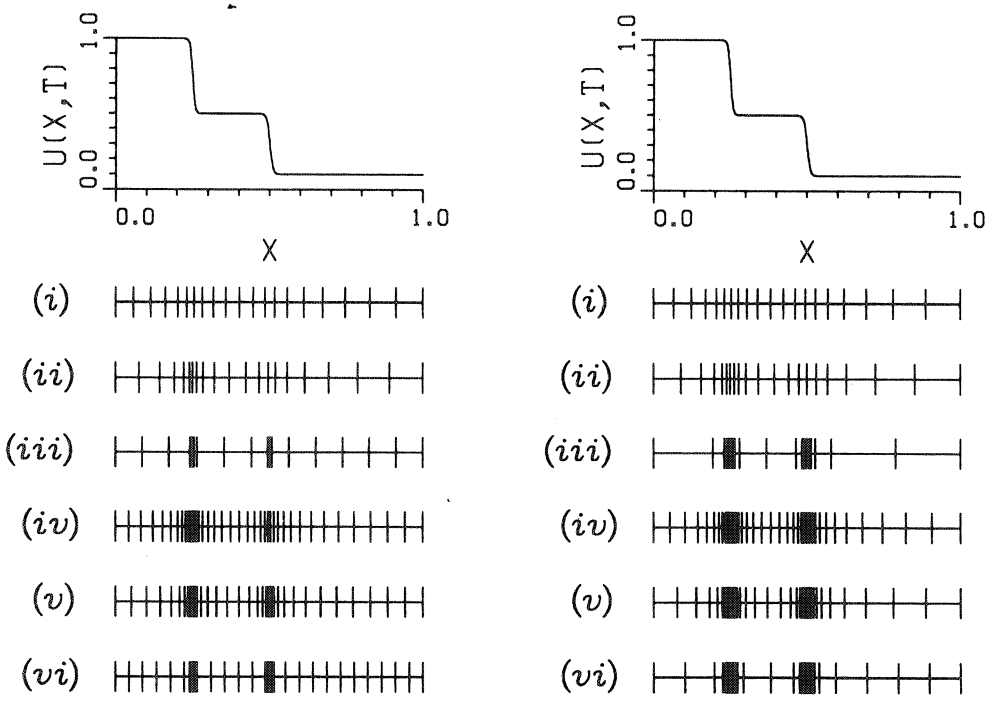

FIGURE 3.1. Problem I. Initial grid distribution for different κ and N values. Arclength monitor (left) and curvature monitor (right).

Fig. 3.1 shows how the grids differ for both monitors and for the parameter choices $\alpha = 1$ and

(*i*) $\kappa = 2$, $N = 19$, (*iv*) $\kappa = 2$, $N = 39$,
(*ii*) $\kappa = 1$, $N = 19$, (*v*) $\kappa = 1$, $N = 39$,
(*iii*) $\kappa = 0$, $N = 19$, (*vi*) $\kappa = 0$. $N = 39$.

One can see that distributing the grid points with the curvature monitor is more adequate, especially with a small number of points and $\kappa = 0$. The arclength monitor concentrates too much points within the two layers and neglects, to some extent, the transition regions. For $\kappa = 1$ or 2 and $N = 19$ there is not much difference between the arclength and curvature grid, due to the grid ratio inequality. For $N = 39$, the grids differ also for $\kappa = 1$ and 2 while the curvature grid is better.

In Table 3.1 the integration history is shown, given by:

STEPS: total number of successful time steps
JACS: total number of Jacobian evaluations
BS: total number of backsolves
ETF: total number of time error test failures
CTF: total number of correction iteration convergence failures; i.e., the Newton process did not converge after 3 iterations, if necessary followed by 3 iterations with an updated Jacobian, or node crossing was detected during the Newton process.

	Arclength monitor					Curvature monitor				
Case	STEPS	JACS	BS	ETF	CTF	STEPS	JACS	BS	ETF	CTF
(*i*)	254	44	663	14	0	472	94	1209	49	0
(*ii*)	299	53	817	21	0	569	114	1503	65	0
(*iii*)	403	52	980	16	0	657	104	1722	60	0
(*iv*)	344	77	985	28	0	864	178	2293	121	0
(*v*)	346	68	937	25	0	864	182	2344	120	0
(*vi*)	490	64	1265	26	0	902	186	2439	113	0

TABLE 3.1. Problem I. Integration history initial grid computation.

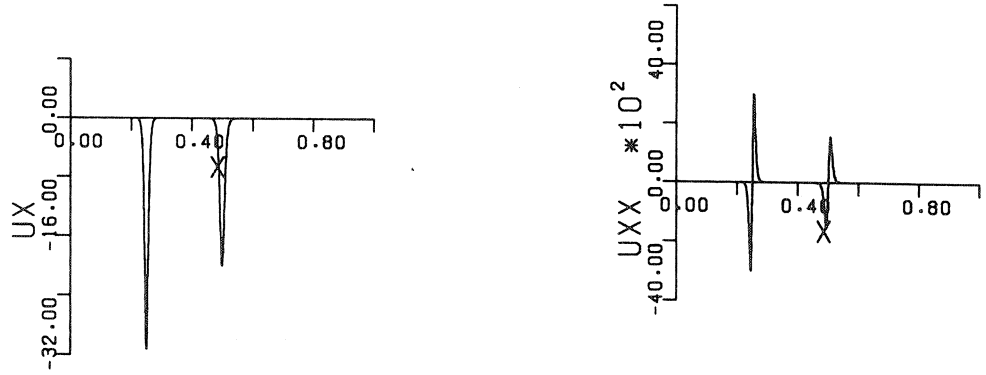

FIGURE 3.2. Problem I. Plot of u_x (left) and u_{xx} (right) functions at $t = 0$.

It is obvious that solving the grid equation (2.5) with the curvature monitor is much more expensive

than with the arclength monitor. This could be explained by the fact that the second derivative of the initial solution function is considerably more varying in x than the first derivative and therefore the ODE system $\tau B\dot{X} = g$ will be harder to solve. (See Fig. 3.2 for a plot of u_x and u_{xx}.) Specifically, for the curvature monitor the entries of B and g will vary much stronger with X_i than for the arclength monitor.

3.1.2. Time-stepping.

We have solved (3.1) starting on the obtained initial grid and using a time-tolerance value in SPGEAR of 10^{-3}. The moving-grid parameters were $\tau = 10^{-3}$ and $\alpha = 1.0$ together with the choices $(i) - (vi)$ for κ and N.

Monitor	Case	$t=0.25$ STEPS	$t=0.55$ STEPS	$t=1.0$ STEPS	JACS	BS	ETF	CTF	error(1.0) $\|.\|_\infty$	$\|.\|_2$
(2.9)	(i)	24	41	76	34	259	6	4	1.1E-1	2.0E-2
	(ii)	16	33	48	26	173	9	2	7.1E-2	7.0E-3
	(iii)	26	42	90	56	254	0	20	4.6E-1	8.5E-2
(2.10)	(i)	85	191	338	228	1157	15	59	1.6E-1	2.4E-2
	(ii)	60	192	209	154	710	3	48	5.9E-2	6.5E-3
	(iii)	11	86	136	94	459	6	26	4.7E-2	4.2E-3
(2.9)	(iv)	16	30	42	23	158	6	2	1.5E-1	1.2E-2
	(v)	16	31	45	28	153	4	4	2.3E-1	1.8E-2
	(vi)	31	87	141	87	392	0	36	5.8E-2	7.6E-3
(2.10)	(iv)	17	81	95	75	335	1	22	1.3E-2	8.9E-4
	(v)	34	78	104	81	357	0	25	2.7E-2	2.1E-3
	(vi)	26	128	177	135	594	4	42	6.9E-3	5.8E-4

TABLE 3.2. Problem I. Integration history and accuracy.

With regard to accuracy, the advantage of the curvature monitor can be clearly seen in Table 3.2 and Fig. 3.3. With 19 internal points and for $\kappa = 2$, and to a less extent for $\kappa = 1$, the grid is more determined by the spatial smoothness demand than by the monitor values. Therefore with both monitors oscillations occur in the solution. For $\kappa = 0$ the arclength monitor concentrates too much of the grid points in the steep parts and apart from the oscillations a phase error can be seen. With the curvature monitor the solution is very accurate but the grid lines are wobbling a bit. In Fig. 3.3 we also see, for the same choices, how the method functions with a larger number of points ($N = 39$). Here the arclength monitor gives both smooth grid lines and a reasonably accurate solution, although for $\kappa = 2$ and 1 the wave-velocity is not quite correct and for $\kappa = 0$ again oscillations in the solution occur. Note that for $\kappa > 0$ the spatial grid smoothing prevents the arclength monitor from clustering a great deal of points inside the layers. The curvature monitor gives for every κ very good accuracy but the grid lines are not smooth functions in time. Non-smooth grid lines $X_i(t)$ are of course detrimental for the numerical integration process. It is conjectured that the non-smoothness of the grid lines is a consequence of the sharp peaks in u_{xx}.

From the results in Table 3.2 it can be seen that, using 39 internal points, the curvature monitor gives much better accuracy than the arclength monitor, but at considerably higher costs. For $N = 19$ and $\kappa = 1$, 2 (case (i), (ii)) there is no difference in accuracy but for the curvature monitor the costs are dramatically larger. This is caused by the oscillations in both the solution and the grid (see Fig. 3.3). The computational costs moreover are rather dependent on the parameter choices. We can see that the number of Newton failures is lower in the integration with (2.9) as monitor than with (2.10). It occurs in the latter integration process more than once that solving the nonlinear system

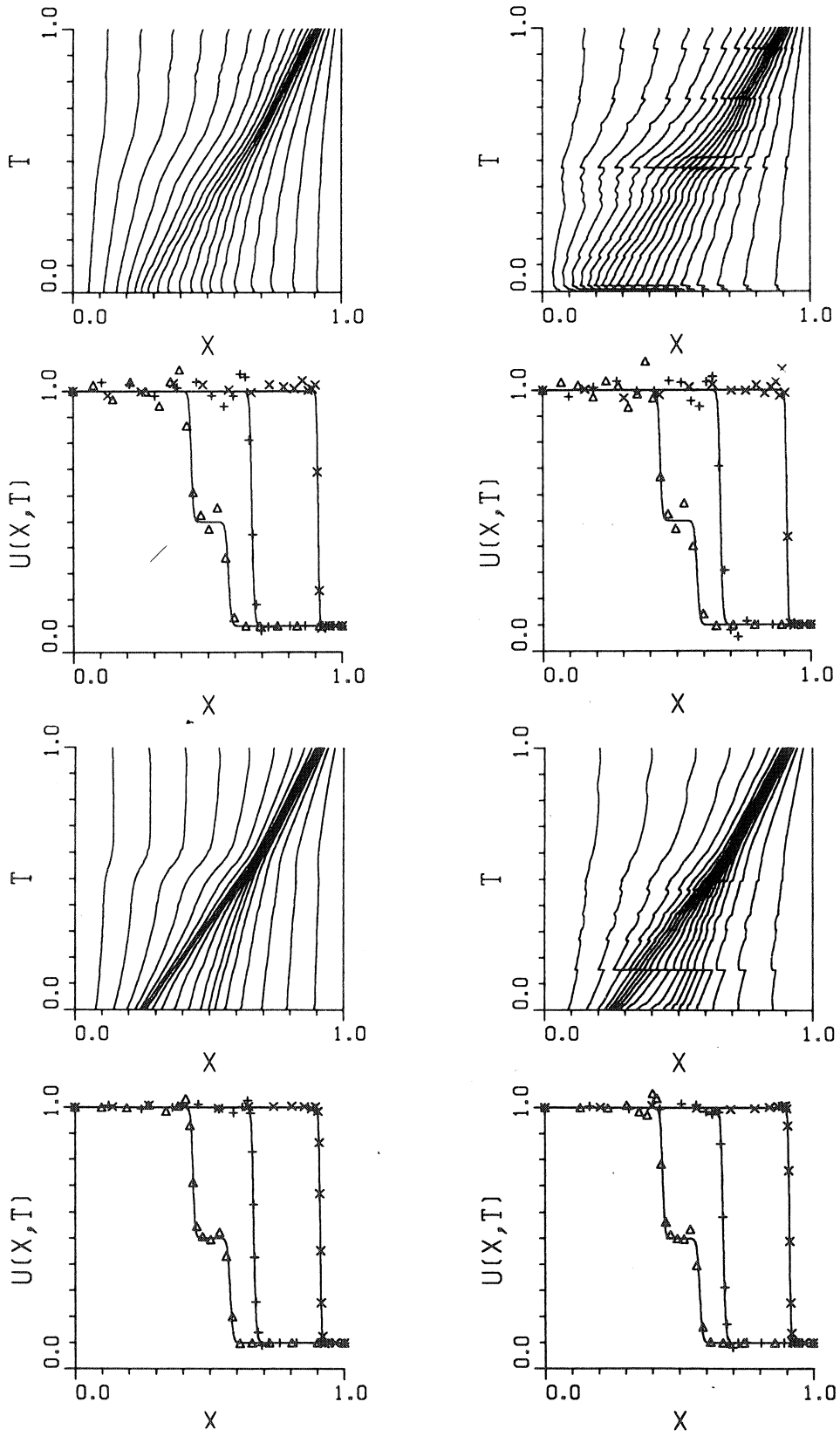

FIGURE 3.3. Problem I. Cases (*i*) and (*ii*).
Grid and solution at times $t=0.25$, 0.55, 1.0 (Δ, $+$, $\times$).
Arclength monitor (left) and curvature monitor (right).

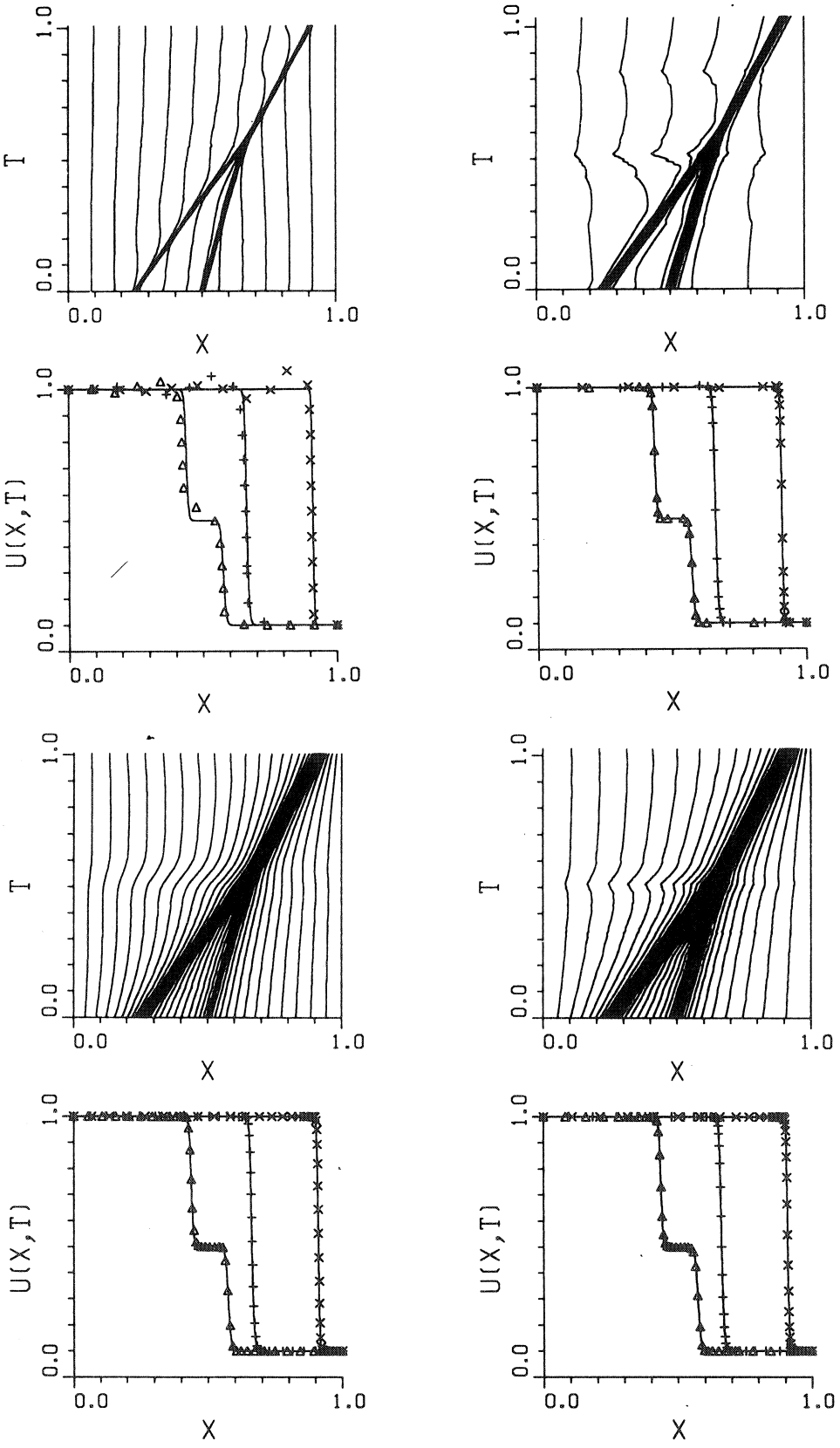

FIGURE 3.3 (cont.). Problem I. Cases (*iii*) and (*iv*).
Grid and solution at times $t=0.25, 0.55, 1.0$ (Δ, $+$, $\times$).
Arclength monitor (left) and curvature monitor (right).

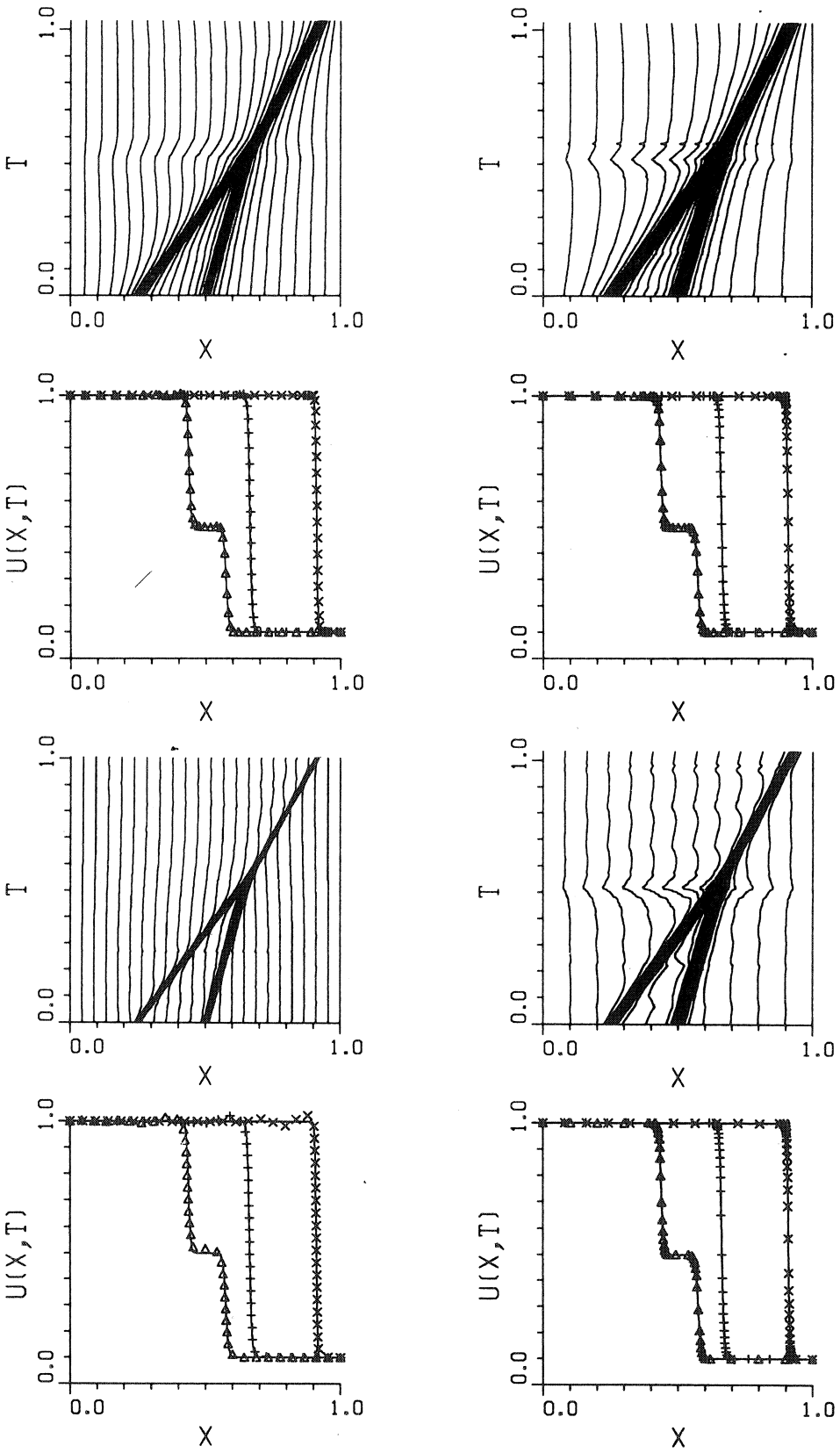

FIGURE 3.3 (cont.). Problem I. Cases (*v*) and (*vi*).
Grid and solution at times $t=0.25$, 0.55, 1.0 (Δ, $+$, $\times$).
Arclength monitor (left) and curvature monitor (right).

repeatedly fails to converge, resulting in a sharp decrease in stepsize. Moreover, if we take $\tau = 0$, which is possible since (2.5) is approximately satisfied at the initial time, then the results for the arclength monitor do not change much but the process with the curvature monitor again and again breaks down due to repeated Newton convergence failures. It is then impossible to restart the integration process, although both the solution and the grid seem to be reasonable well. Note that for $\tau = 0$ a DAE system results. For monitor (2.10) this DAE system thus appears to be much more difficult to solve with SPGEAR.

3.2. 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. As before we selected the problem parameters $a = 1$, $\delta = 20$, $R = 5$. For small times the temperature gradually increases from unity with a 'hot spot' forming at $x = 0$. At a finite time, around $t = 0.26$, ignition occurs, causing the temperature at $x = 0$ to increase very rapidly to 2. A flame front then forms and propagates towards $x = 1$ at high speed to a steady state ($t \approx .29$).

Since the solution at $t = 0$ is a constant, we start on a uniform grid using 41 grid points ($N = 39$) and a time-tolerance in SPGEAR of 10^{-5}. The moving-grid parameters are $\alpha = 1.0$ and

(*i*) $\kappa = 1$, $\tau = 0.001$, (*ii*) $\kappa = 1$, $\tau = 0$.

Monitor	Case	$t=0.25$ STEPS	$t=0.26$ STEPS	$t=0.27$ STEPS	$t=0.29$ STEPS	JACS	BS	ETF	CTF
(2.9)	(*i*)	34	49	105	136	36	453	23	0
	(*ii*)	34	51	106	133	35	445	22	0
(2.10)	(*i*)	42	59	197	416	131	1240	86	1
	(*ii*)	43	62	198	*				

TABLE 3.3. Problem II. Integration history;
* means that the integration has been interrupted due to repeated Newton failures.

In Fig. 3.4 plots of the grid movement and of the PDE solution at times $t = .26$, $.27$, and $.29$ are shown for both monitors. The grid lines are only shown for $t > 0.25$; before this time point the grid is uniform. The solid lines in the plot of the PDE solution represent a very accurate reference solution.

For this problem the time integration is of more weight than the precise placement of the grid points and therefore it shows even better that using the curvature monitor in the moving-grid procedure is expensive and not very robust, the costs being unacceptably high and varying for different parameter settings. Note in particular for the curvature monitor the oscillatory grid lines for $t > .27$ if $\tau = 0$, resulting in very small time steps. In Table 3.3 we see that for $\tau = 0$ the moving-grid method with (2.10) does not reach steady state, due to repeated Newton failures (at $t \approx .287$) even if the time step has already been reduced to approximately zero. The reason for this, is that the condition number of the Jacobian matrix is very large, say $> 10^9$, and therefore the numerical

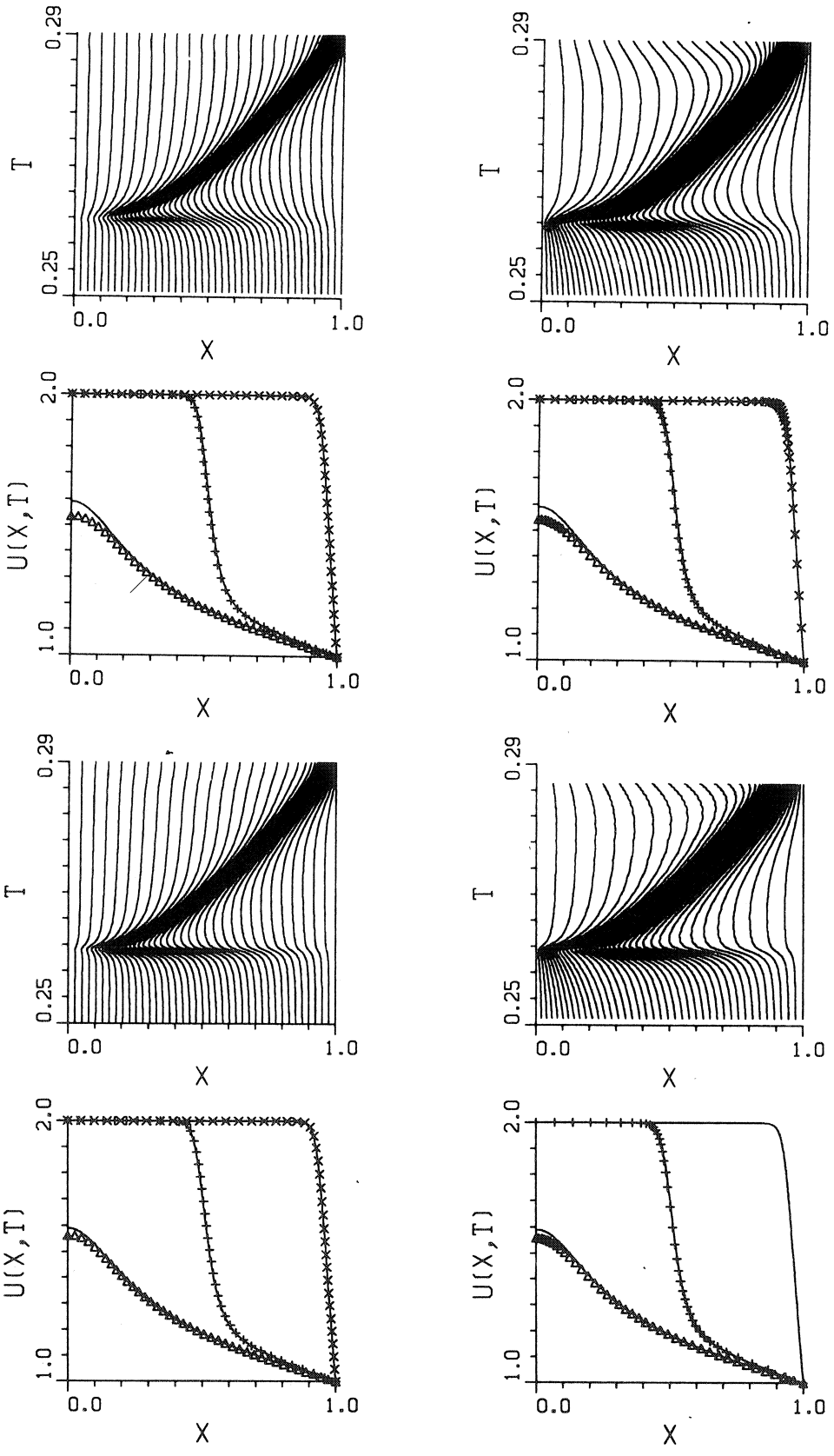

FIGURE 3.4. Problem II. Cases (*i*) and (*ii*).
Grid and solution at times $t = .26, .27, .29$ (Δ, $+$, $\times$).
Arclength monitor (left) and curvature monitor (right).

perturbation will be larger than the tolerance for the Newton process. With $\tau = 10^{-3}$ the steady state is reached without any difficulties in solving the nonlinear system, but again the moving-grid method with (2.10) uses much more time steps than with (2.9) as monitor. This test indicates again that the curvature monitor gives rise to a much more difficult semi-discrete problem. Observe that for the arclength monitor there is not much difference in performance for both choices of parameters.

4. Some further observations on monitors and on the grid equation

In [7] we have used the curvature solution functional

$$m(u) = \alpha + \left[(u_{xx})^2\right]^{1/4}, \tag{4.1}$$

instead of the $m(u)$ underlying (2.10), i.e.,

$$m(u) = \left[\alpha + (u_{xx})^2\right]^{1/4}. \tag{4.2}$$

At first sight the difference between the two is minor as this merely concerns the regularization constant α. However, in a numerical calculation the regularization in (4.1) may lead to erroneous results.

To see this, we introduce the related functions

$$g_1(y) = \alpha + (y^2)^{1/4}, \tag{4.3}$$

$$g_2(y) = (\alpha + y^2)^{1/4}, \tag{4.4}$$

and compute

$$\frac{dg_1(y)}{dy} = \frac{\frac{1}{2}\mathrm{sign}(y)}{(y^2)^{1/4}}, \tag{4.5}$$

$$\frac{dg_2(y)}{dy} = \frac{\frac{1}{2}y}{(\alpha + y^2)^{3/4}}. \tag{4.6}$$

The variable y represents the true second derivative u_{xx} in the above analytical expressions and in an actual numerical implementation y represents the numerical replacement for u_{xx}, e.g., expression (2.11). Clearly, for small values of u_{xx} or its numerical replacement the above derivatives behave in an entirely different way. While $dg_2(y)/dy$ vanishes for $y \to 0$, $dg_1(y)/dy$ blows up. Furthermore, if the numerical replacement for u_{xx} has a wrong sign, which is of course conceivable if $u_{xx} \approx 0$, then also $dg_1(y)/dy$ has a wrong sign. This results in entirely wrong derivative values $\partial M_i/\partial U_j$, $\partial M_i/\partial X_j$ that appear in the Jacobian matrix which we encounter in implicitly solving the grid equation (2.8). In particular, it can imply that during a Newton iteration entries in the true Jacobian vary extremely rapidly. This of course forms a serious obstacle. Numerical experiments with the moving-grid method using a monitor based on (4.1) instead of (4.2), have confirmed this.

In conclusion, the regularization parameter α should not be separated from the derivative expression as in (4.1). It is emphasized, that this observation also applies to the two arclength monitors

$$m_1(u) = \alpha + \left[(u_x)^2\right]^{1/2} \tag{4.7}$$

and

$$m_2(u) = \left[\alpha + (u_x)^2\right]^{1/2}. \tag{4.8}$$

Note that

$$\frac{dm_1(u)}{du_x} = \mathrm{sign}(u_x), \qquad \frac{dm_2(u)}{du_x} = u_x/(\alpha + (u_x)^2)^{1/2}.$$

Finally, when applying the method of [7] no difficulty originating from the above point was perceived.

We owe this to the fact that the method of [7] uses monitor information in an explicit way.

We also considered a monitor which places the grid points in the curvature without computing the large and unreliable values of U_{xx} in reality. This monitor function is

$$M_i := d_i \cdot \left[\alpha + (U_{xx_{i+1/2}})^2\right]^{1/2}, \tag{4.9}$$

where

$$U_{xx_{i+1/2}} = \begin{cases} U_{xx_1} \ (cf.\ 2.11) & i = 0 \\ \dfrac{U_{x_{i+1}} - U_{x_i}}{d_i} & i = 1, \cdots, N-1, \\ U_{xx_N} \ (cf.\ 2.11) & i = N \end{cases} \tag{4.10}$$

and

$$U_{x_i} = \frac{U_{i+1} - U_{i-1}}{X_{i+1} - X_{i-1}}, \qquad d_i = (\Delta X_{i+1} + \Delta X_{i-1})/2.$$

Note, that since $\Delta X_0 = \Delta X_{-1}$ and $\Delta X_N = \Delta X_{N+1}$, the multiplier of $U_{xx_{i+1/2}}$ in (4.9) is for all i the same as the denominator in (4.10) and therefore M_i can be computed by

$$M_i = \left[\alpha . d_i^2 + (U_{x_{i+1}} - U_{x_i})^2\right]^{1/2}. \tag{4.11}$$

Hence, the idea behind this monitor is to compute a difference of two difference quotients for u_x rather than the common divided difference. If we assume that $\Delta X_i \approx d_i$ then this monitor can be obtained by discretizing the equidistribution rule

$$\left[\int_{X_i}^{X_{i+1}} m(u)dx\right]^2 = \text{constant},$$

where $m(u)$ is defined by (4.2).

The difficulty with this monitor is that M_i can become arbitrarily small (dependent on κ and N). If we take, e.g., $\Delta X_{i+1} \to 0$ and $\Delta X_{i-1} \to 0$ then $U_{x_{i+1}} \to U_{x_i}$ and therefore $M_i \to 0$. This is a very undesirable property since M_i should be larger than zero to prevent infinite terms in the grid equation.

As suggested in [6], the grid equation can alternatively be formulated in terms of the grid distance ΔX_i rather than in the point concentration n_i. This yields, instead of (2.5)

$$(\tau\dot{\tilde{\Delta}}_i + \tilde{\Delta}_i)M_i - (\tau\dot{\tilde{\Delta}}_{i-1} + \tilde{\Delta}_{i-1})M_{i-1} = 0, \tag{4.12}$$

with $\tilde{\Delta}_i := \Delta X_i - \kappa(\kappa+1)(\Delta X_{i+1} - 2\Delta X_i + \Delta X_{i-1})$ analogously defined as $\tilde{n}_i$. This formulation is, especially for analytical reasons, much simpler than (2.5) but, unfortunately, (4.12) appeared to be harder to solve with a Newton process. We presume that this is caused by the fact that the Jacobian entries are larger, and thus more varying, for (4.12) than for (2.5) in case of a not well-placed grid. This can be easily seen if we take $\tau = \kappa = 0$. The entries of the Jacobian for solving (4.12) are then approximately those going with (2.5) multiplied by $(\Delta X_i M_i)^2$, which is for all i more or less a constant in case the grid equations are solved, but which can be rather large if M_i is large and the corresponding ΔX_i not (yet).

5. Conclusions

We have seen in both experiments in Section 3 that the use of the curvature monitor results in a good distribution of the grid points and accurate results, but that the time integration both in the initial grid computation and in the moving-grid method itself brings about large and sometimes even unsurmountable problems. The curvature monitor gives rise to a semi-discrete problem that is harder to solve for the stiff solver SPGEAR. The origin of the difficulties seems to emanate from the fact that the second derivative of the solution may vary rapidly over the space interval (especially in a wave front), resulting in widely different entries in the grid equation. Because the grid equation is severely nonlinear, this strong variation very much hinders the Newton process. For the arclength monitor this variation is much less and it is our experience that for this monitor the time integration with stiff solvers like SPGEAR is normally done in an efficient way. Also, for the arclength monitor the choice of the parameters κ and τ is not very critical.

References

1. S. Adjerid and J.E. Flaherty (1986). A Moving Finite Element Method with Error Estimation and Refinement for One-Dimensional Time Dependent Partial Differential Equations, *SIAM J. Numer. Anal.*, 23, 778-796.
2. M. Berzins and R.M. Furzeland (1985). *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..
3. J.G. Blom and P.A. Zegeling (1989). *A Moving-Grid Interface for Systems of One-Dimensional Time-Dependent Partial Differential Equations,* Report NM-R8904, Centre for Mathematics and Computer Science, Amsterdam.
4. E.A. Dorfi and L. O'Č. Drury (1987). Simple Adaptive Grids for 1-D Initial Value Problems, *J. Comput. Phys.*, 69, 175-195.
5. R.M. Furzeland, J.G. Verwer, and P.A. Zegeling (1988). *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 (submitted to J. Comput. Physics).
6. J.G. Verwer, J.G. Blom, R.M. Furzeland, and P.A. Zegeling (1988). *A Moving-Grid Method for One-Dimensional PDEs based on the Method of Lines,* Report NM-R8818, Centre for Mathematics and Computer Science (CWI), Amsterdam. (Paper presented at the 'Workshop on Adaptive Methods for Partial Differential Equations', Rensselaer Polytechnic Institute, Troy, New York, Oct.13-15, 1988. To appear in the proceedings.).
7. J.G. Verwer, J.G. Blom, and J.M. Sanz-Serna (1988). *An Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations,* Report NM-R8804, Centre for Mathematics and Computer Science (CWI), Amsterdam. (to appear in J. Comput. Phys. in 1989).