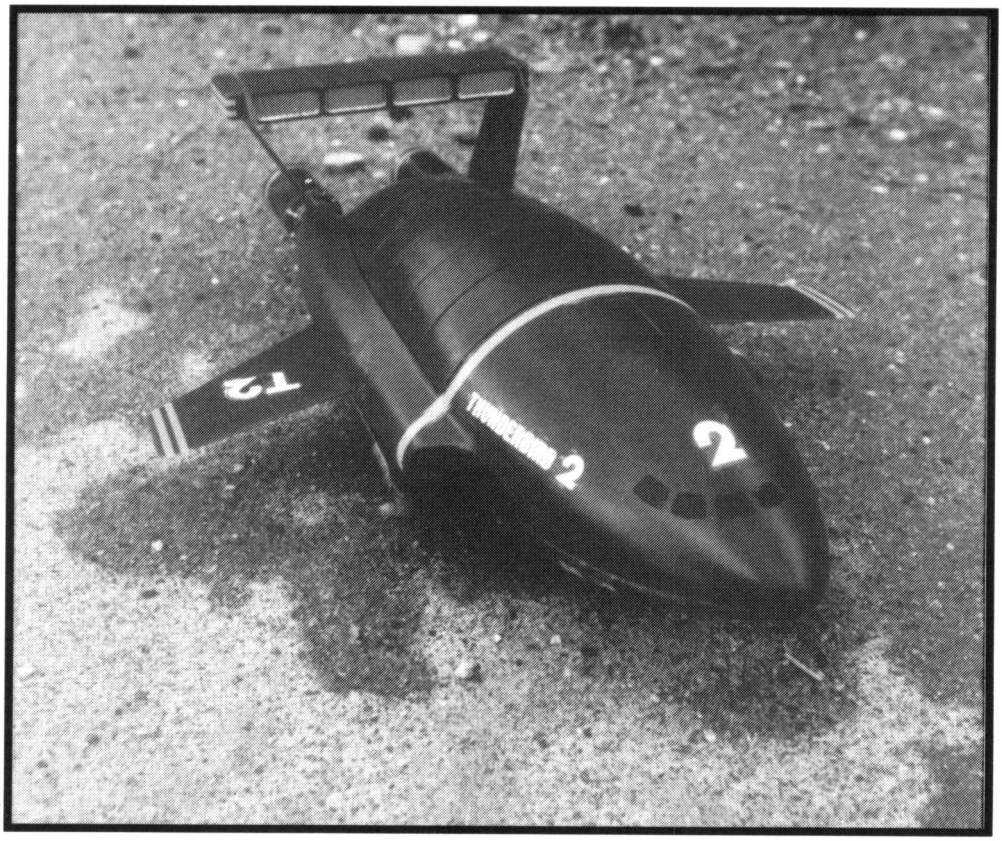
MOVING-GRID METHODS
FOR TIME-DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS

Paul Zegeling



Moving-Grid Methods for Time-Dependent Partial Differential Equations

Moving-Grid Methods for Time-Dependent Partial Differential Equations

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam,
op gezag van de Rector Magnificus,
prof. dr. P.W.M. de Meijer,
in het openbaar te verdedigen in de Aula der Universiteit
(Oude Lutherse Kerk, ingang Singel 411, hoek Spui),
op donderdag 8 oktober 1992 te 13.30 uur

door

Paul Andries Zegeling

geboren te Moskou

Centrum voor Wiskunde en Informatica
Amsterdam
1992

Bewegende roostermethoden voor tijdsafhankelijke partiële differentiaalvergelijkingen

Promotor: Prof. Dr. P.J. van der Houwen

Co-promotor: Dr. J.G. Verwer

Faculteit: Wiskunde en Informatica

Het onderzoek voor dit proefschrift werd uitgevoerd in het kader van het CWI/Shell-project 'Adaptive-Grid Techniques in Software for Evolutionary Partial Differential Equations'. Dit project werd financieel ondersteund door de 'Nederlandse Organisatie voor Wetenschappelijk Onderzoek' (NWO) via de 'Stichting voor de Technische Wetenschappen' (STW).

Dedicated to:

Mo

Tea & Goejp

Evert Kaplan (his brother)

Lokomotiv TNT

Acknowledgements

The subject of this thesis is moving-grid methods for time-dependent partial differential equations in one and two space dimensions. The research for this thesis was carried out at the Department of Numerical Mathematics of the CWI (Centre for Mathematics and Computer Science). It was part of a joint CWI/Shell project titled 'Adaptive-Grid Techniques in Software for Evolutionary Partial Differential Equations'. This project was financially supported by the 'Netherlands Organization for Scientific Research' (NWO) via the 'Netherlands Foundation for the Technical Sciences' (STW).

I want to thank all those who contributed somehow to the realization of this work. First of all, I'd like to express my particular gratitude to dr. J.G. Verwer for his guidance during this project and for his many contributions to this thesis. I am also grateful to Prof. dr. P.J. van der Houwen for acting as a promotor and, as head of the department, for creating a pleasant and stimulating environment for doing scientific research. My special thanks are due to Joke Blom. As 'wandelende manual' she was always ready to provide assistance whenever it was needed. Her contributions to many chapters of this thesis are also indispensible. Further, I thank Joke Blom and Ron Trompert for the "fruitful" discussions on mathematics and other subjects, that we had in M026. Finally, I'd like to thank the CWI, STW, and NWO, respectively, for giving me the opportunity to prepare the thesis.

Contents

Chapter 1

General Introduction

"FAB, Lady Penelope"

Standard numerical methods to solve time-dependent partial differential equations (PDEs) integrate on a uniform grid that is kept fixed on the entire time interval. If the solutions have regions of high spatial activity, a standard fixed-grid method is computationally inefficient, since to afford an accurate approximation, it should contain a very large number of nodes. The grid then needs to be locally refined. If the regions of high spatial activity are, moreover, moving in time, like for steep moving fronts, then methods are needed that also adapt the grid in time.

Roughly spoken, one may distinguish two classes of time-dependent adaptive methods: 1. (class I) dynamic-regridding (moving-grid) methods and 2. (class II) static-regridding methods. In the latter class of methods, for which the adaptivity is also denoted by terms like 'local refinement' or 'h-refinement', the grid is only adapted at discrete time levels. Methods from class I, sometimes characterized by the term 'r-refinement', have the special feature to move the spatial grid continuously in the space-time domain while the discretization of the PDE and the grid selection procedure are intrinsically coupled. Both approaches have their advantages and disadvantages, depending, e.g., on the PDE model to be solved, the hardware used, the spatial domain in the model, etc..

The main advantage of class II methods is their conceptual simplicity and robustness in the sense, that they permit the tracking of a varying number of wave fronts. A drawback, however, is that interpolation must be used to transfer numerical quantities from the old grid to new grids. Also, numerical dispersion, appearing, e.g., when hyperbolic PDEs are numerically approximated, is not fully annihilated. Another disadvantage of static-regridding methods compared with methods using moving-grids is the fact that they produce no 'smoothing' in the time direction. For these methods the time-stepping accuracy therefore will demand, in general, smaller time steps than for moving-grid methods. Examples of methods belonging to class

II are found in [5, 10, 25, 33].

Class I methods use a fixed number of grid points, without need of interpolation, and let them move with whatever fronts are present. In the case of several steep fronts acting in different regions of the spatial domain, this could give problems in the numerical computation, if the grid is following one wave front and another one arises somewhere else. Since the number of grid points is fixed throughout the entire course of the computation, no 'new' grid is created for the new wave, but rather the 'old' grid has to adjust itself abruptly. Another difficulty is of a topological nature, viz., the so-called 'mesh-tangling'. Moving-grid methods, therefore, often need a kind of regularization to cope with this phenomenon. This, unfortunately, involves, more or less, tuning of the extra regularization parameters. On the other hand, even though more computations per grid point are needed, the use of moving-grid methods may work out very efficiently, since, in general, fewer spatial grid points will be necessary. Some characteristic members in this class of methods can be found in [11, 13, 15, 17].

This thesis deals with moving-grid methods for time-dependent PDEs in one and two space dimensions. Below we will give a short description of the essential components of which such methods consist.

1.1. THE METHOD OF LINES

The discretization of time-dependent PDEs is often performed in two basic stages. First, the spatial variables are discretized on a selected space mesh, e.g., using finite-difference or finite-element approximations, so as to convert the PDE problem into a system of ordinary differential equations (ODEs) with time as independent variable. These ODEs are usually stiff. The discretization in time of this stiff ODE system then yields the required fully discretized scheme. This two-stage approach is often referred to as the method of lines (MOL). With this approach in mind, several sophisticated PDE software-packages have been developed in recent years. These MOL packages greatly benefit from the very successful developments of automatic stiff ODE solvers. A key factor here is the development of implicit BDF codes, such as the ones described by Hindmarsh [21], Petzold [26], and Berzins and Furzeland [6, 7]. Although most BDF codes have been designed to solve stiff ODE systems in an accurate and efficient way, experiences with MOL packages have revealed that this is also true for PDE problems. However, for solutions possessing large space-time gradients, like travelling wave fronts, a grid held fixed for the entire calculation can be computationally inefficient, since the grid will have to contain a very large number of nodes and the time steps still have to be small. 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. Since the grid movement, in general, will permit larger time stepsizes, it is attractive to automatize the time integration by combining the MOL procedure on a moving grid with a BDF code, just as for the fixed-grid case.

1.2. MOVING-GRID METHODS

Consider the scalar PDE in one space dimension

$$\frac{\partial u}{\partial t} = \mathcal{L}(u), \quad x \in \Omega, \quad t > 0, \tag{1.1}$$

with initial and boundary conditions

$$u\,|_{t=0} = u^0(x), \quad x \in \Omega,$$

$$\mathcal{B}(u, \frac{\partial u}{\partial x})\,|_{\partial\Omega} = g(t), \quad t > 0,$$

where u^0 and g are given functions, and $\mathcal{L}$ represents a differential operator involving only spatial derivatives up to second order. In the following, the domain $\Omega := (x_L, x_R)$ is supposed to be fixed for all times $t > 0$ under consideration. In general, the solution $u(x,t)$ of (1.1) may have a very complex behaviour. Even for a restricted situation (a scalar linear PDE with simple boundary conditions), one can have severely varying u-values in space x and time t.

A common approach handling these phenomena is to introduce a transformation which maps the variables x and t into new variables ξ and τ. Such a transformation can be defined as, e.g.,

$$\begin{aligned} x &= x(\xi, \tau) \\ t &= \tau \\ u(x,t) &= v(\xi, \tau). \end{aligned} \tag{1.2}$$

The effect of the transformation ought to be to stretch the co-ordinate in boundary or internal layers so that $\partial v/\partial \xi$ is small when $\partial u/\partial x$ is not. More generally, transformations are required to map strongly varying behaviour of u to a more moderate behaviour of v. An attractive side-effect of the time-dependent transformation may be to obtain values for $\partial v/\partial \tau$ essentially smaller than the $\partial u/\partial t$ values, a phenomenon also appearing in the method of characteristics. If we take for example $\mathcal{L}(u) = -\gamma\, \partial u/\partial x$, the 'optimal' mapping is defined by the characteristic equation $\partial x/\partial \tau = \gamma$, and the grid points merely follow the characteristics of the PDE, in which case we have the inverse transformation $\xi(x,t) = x - \gamma t$. Of course, when using a transformation, most difficulties are shifted to the problem of how to define and carry out the mapping.

The Jacobian of the transformation (1.2) is given by

$$\mathcal{J} = \frac{\partial(x,t)}{\partial(\xi,\tau)}$$

and, in order to preserve invertibility of the transformation, its determinant ($= x_\xi$) is supposed to be non-zero for all points of time. This is, in the discrete case, equivalent to demanding that the grid points do not cross. Using the chain rule and the inverse of $\mathcal{J}$, equation (1.1) can be rewritten as

$$\frac{\partial u}{\partial t} = \frac{\partial v}{\partial \tau} - u_x \frac{\partial x}{\partial \tau} = \mathcal{L}, \tag{1.3}$$

where $u_x = v_\xi / x_\xi$, and an additional equation for x_τ must be defined. Equation (1.3) is usually called the Lagrangian form of the PDE.

Several possibilities are on hand to proceed further. A well-known choice is to define the mapping implicitly as

$$x_\xi \, \mathcal{M} = \text{constant}, \quad \text{for all } t>0, \tag{1.4}$$

where $\mathcal{M}$ is a positive function, the so-called weight or monitor function, e.g., depending on first and/or second order derivatives of the solution. Equation (1.4) reflects the basic principle of equidistribution. In discretized form, this becomes more obvious:

$$(X_i - X_{i-1}) \, \mathcal{M}_i = \text{constant}, \tag{1.5}$$

where X_i $(i=1, \cdots, N)$ are the time-dependent grid points subdividing the spatial domain Ω into $N+1$ parts, and $\mathcal{M}_i$ is a discrete formulation for $\mathcal{M}$ on the interval (X_{i-1}, X_i). Note, that we have assumed a uniform grid distribution in ξ-space. With this condition, the grid interval in x-space will, of course, be small where the weight function is large, and vice versa. The term equidistribution stems from the observation that (1.5) can be rewritten as

$$\int_{X_{i-1}}^{X_i} \mathcal{M}(\phi,t) \, d\phi = \text{constant},$$

thereby showing that the weight function $\mathcal{M}$ is equally distributed over the spatial domain. The inverse co-ordinate transformation belonging to (1.5) then reads

$$\xi(x,t) = \int_{x_L}^{x} \mathcal{M}(\phi,t) \, d\phi \Big/ \int_{\Omega} \mathcal{M}(\phi,t) \, d\phi.$$

Equidistribution principles have been used in many different ways to numerically solve one-dimensional PDEs having solutions with steep transitions. One of the earliest attempts to apply equidistribution to 1D time-dependent PDEs can be found in [34]. A more sophisticated application of the equidistribution idea is described by Dorfi and Drury [16]. They produce an adaptive method based on (1.4) with an arc-length monitor function $\mathcal{M}$. Additionally, the equidistribution principle is supplemented by two regularization procedures to cater for smooth grid trajectories in both space and time. Other examples, related to equidistribution being applied to 1D PDEs, are described in, e.g., [1, 8, 29].

Beside equidistribution, a well-known approach to define a moving-grid method is to use a minimization of a functional (or integral) depending on measures of the error of the solution and/or grid structure properties. One approach is described by Petzold [27]. Using the transformed PDE (1.3), she defines the grid movement by minimizing a measure consisting of a combination of node velocities and time-derivatives of solution values. Two regularization terms serve to keep the transformation non-singular and to smooth the grid movement. A regridding strategy is added to insert or delete or move nodes to resolve the spatial gradients.

Perhaps the most important representative in this class of methods is the moving-

finite-element method of Miller et al. [19, 23, 24]. In this method the error measure may be interpreted as being the square of the residual of the PDE written in finite-element form. Semi-discrete ODEs for the solution and the grid points are obtained by minimizing the integral of this error measure over these unknown quantities. Regularization of the minimization is needed to prevent it from becoming degenerate and to keep control over the grid movement in time. An additional regularization must be used for second order PDE operators if the approximation space consists of piecewise linear trial functions.

In two space dimensions application of moving-grid methods is much more difficult than in 1D. For instance, there are many possibilities to treat the one-dimensional boundary and to discretize the spatial domain, each having their own difficulties for specific PDEs. However, the essential concepts underlying the co-ordinate transformation in 1D also apply to higher dimensions. Important references in this respect are, e.g., [11, 13, 15, 17]. It must be noted, that the application of equidistribution principles in 2D is less straightforward than in 1D. This is not the case for methods that are derived from a functional minimization, which are, in theory, easily extendable to PDE operators in higher space dimensions. It is also interesting to note, that the 1D equidistribution principle (1.4) can be derived from the minimization method in [11] by restricting the method to one dimension. In fact, equation (1.4) can be obtained by applying the minimization, i.e., working out the Euler equations for the functional, and then integrating the so-obtained elliptic PDE for the grid.

Other moving-grid methods in 2D based on minimization or equidistribution principles are described in [2, 12, 14, 22, 28, 30, 35]. For more information on adaptive methods, in general, and on moving grids, in particular, the interested reader is referred to [3, 4, 18, 32] or the review papers [20] and [31].

1.3. Contents of the thesis

Chapter 2 describes a numerical study of three sample moving-grid methods for 1D time-dependent partial differential equations. The three methods are: 1. a moving-finite-difference method (MFD) proposed by Petzold [27], 2. a moving-finite-difference method originally due to Dorfi & Drury [16], and 3. a moving-finite-element method (MFE) introduced by Miller et al. [23, 24]. The paper emphasizes the performance of the methods with respect to efficiency, accuracy and robustness (effect of regularization parameters and reliability of the numerical solution as for the fixed-grid case). The methods are tested on a set of three test problems each having their own characteristic solution behaviour. Chapter 2 has also been published in the 'Journal of Computational Physics' (Volume 89, pp.349-388, 1990). The paper is joint work with J.G. Verwer (CWI) and R.M. Furzeland (Shell Research).

Chapter 3 deals with some theoretical and analytical properties of the moving-finite-difference method due to Dorfi & Drury. Especially, the effects of the appearing regularization parameters, which serve to smooth the grid distribution in space and the grid movement in time, are examined. This paper (together with J.G. Verwer, J.G. Blom (CWI) and R.M. Furzeland) has been published in the

proceedings [18] of the workshop on 'Adaptive Computational Methods for Partial Differential Equations', held at Rensselaer Polytechnic Institute, October 13-15, 1988.

In Chapter 4 the 1D gradient-weighted MFE (GWMFE) method of Miller is evaluated. GWMFE is an extension of MFE based on adding a weighting term, that depends on the first spatial derivative of the solution, to the minimization procedure that determines the movement of the grid points. A brief comparison is made between this method, its predecessor MFE and the MFD method of Chapter 3. This paper, which is joint work with J.G. Blom, is accepted for publication in the 'Journal of Computational Physics'.

Chapter 5 describes the grid movement induced by the MFE method, when applied to hyperbolic and parabolic PDEs in one and two space dimensions. Numerical examples show the relation between MFE and both equidistribution principles and the method of characteristics. This paper is joint work with J.G. Blom and is accepted for publication in the 'International Journal for Numerical Methods in Engineering'.

Chapter 6 combines the CWI-report [9], describing a software interface in which the moving-grid method of Chapter 3 is incorporated, and a paper (co-authors: J.G. Verwer and J.C.H. v. Eijkeren (RIVM)), which is accepted for publication in the 'International Journal for Numerical Methods in Engineering'. It shows an application of the software interface to a class of 1D brine transport problems in a porous medium, stemming from hydrology.

Chapter 7 presents numerical testresults obtained by applying the MFE method in 2D to different classes of PDEs. Among others, MFE is applied to the so-called 'Molenkamp test', a standard test problem from meteorology, to a 'flame problem' from combustion theory, and to a 2D brine transport model from hydrology (a 2D version of the model used in Chapter 6). Some of the examined aspects are: the effect of the regularization parameters on the 2D grid movement, the quality of the solution, and the numerical time-stepping procedure. Also, the effect of a small diffusion coefficient in a convection-diffusion equation on the semi-discrete MFE system is studied. This paper is submitted to the journal 'Computational Fluid Dynamics'.

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. D.C. Arney and J.E. Flaherty (1986). A Two-Dimensional Mesh Moving Technique for Time-Dependent Partial Differential Equations, *J. Comput. Phys.*, 67, 124-144.
3. I. Babuška, J. Chandra, and J.E. Flaherty, editors. (1983). *Adaptive Computational Methods for Partial Differential Equations*, SIAM, Philadelphia.
4. I. Babuška, O.C. Zienkiewicz, J. Gago, and E.R. de A. Oliveira, editors. (1986). *Accuracy estimates and adaptive refinements in finite element computations*, John Wiley & Sons Ltd..

5. M.J. BERGER and J. OLIGER (1984). Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations, *J. Comput. Phys.*, 53, 484-512.
6. 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..
7. M. BERZINS and R.M. FURZELAND (1986). *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.
8. J.G. BLOM, J.M. SANZ-SERNA, and J.G. VERWER (1989). A Lagrangian Moving Grid Scheme for One-Dimensional Evolutionary Partial Differential Equations, in *Annals on Computing and Applied Mathematics, Numerical and Applied Mathematics, Vol 1.1*, 247-255, ed. W.F. AMES, IMACS.
9. 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 (CWI), Amsterdam (submitted to ACM Trans. Math. Software).
10. F.A. BORNEMANN (1991). *An Adaptive Multilevel Approach to Parabolic Equations in Two Space Dimensions,* Techn. Report TR 91-7, Konrad-Zuse-Zentrum für Informationstechnik Berlin.
11. J.U. BRACKBILL and J.S. SALTZMAN (1982). Adaptive Zoning for Singular Problems in Two Dimensions, *J. Comput. Phys.*, 46, 342-368.
12. R.C. CARCAILLET, G.S. DULIKRAVICH, and S.R. KENNON (1986). Generation of Solution-Adaptive Computational Grids Using Optimization, *Comp. Meth. in Appl. Mech. and Eng.*, 57, 279-295.
13. N. CARLSON and K. MILLER (1988). The Gradient Weighted Moving Finite Element Method in Two Dimensions, in *Finite Elements Theory and Application*, 152-164, ed. D.L. DWOYER, M.Y. HUSSAINI AND R.G. VOIGHT, Springer Verlag.
14. T.K. DELILLO and K.E. JORDAN (1987). Some Experiments with a Dynamic Grid Technique for Fluid Flow Codes, in *Advances in Comp. Meth. for Part. Diff. Eq.*, 495-500, ed. R. VICHNEVETSKY AND R.S. STEPLEMAN.
15. M.J. DJOMEHRI and K. MILLER (1981). *A Moving Finite Element Code for General Systems of PDE's in 2-D,* Report PAM-57, Center for Pure and Applied Mathematics, University of California, Berkeley.
16. E.A. DORFI and L. O'DRURY (1987). Simple Adaptive Grids for 1-D Initial Value Problems, *J. Comput. Phys.*, 69, 175-195.
17. H.A. DWYER, B.R. SANDERS, and F. RAISZADEK (1983). Ignition and Flame Propagation Studies with Adaptive Numerical Grids, *Combustion and Flame*, 52, 11-23.
18. J.E. FLAHERTY, P.J. PASLOW, M.S. SHEPHARD, and J.D. VASILAKIS, EDITORS. (1989). *Adaptive Methods for Partial Differential Equations*, SIAM, Philadelphia.
19. R.J. GELINAS, S.K. DOSS, and K. MILLER (1981). The Moving Finite Element

Method: Application to General Equations with Multiple Large Gradients, *J. Comput. Phys.*, 40, 202-249.

20. D.F. HAWKEN, J.J. GOTTLIEB, and J.S. HANSEN (1991). Review of Some Adaptive Node-Movement Techniques in Finite-Element and Finite-Difference Solutions of Partial Differential Equations, *J. Comput. Phys.*, 95, 254-302.
21. A.C. HINDMARSH (1981). ODE Solvers for Use with the Method of Lines, in *Advances in Computer Methods for Partial Differential Equations-IV*, 312-316, ed. R. VICHNEVETSKY AND R.S. STEPLEMAN, IMACS.
22. D. LEE and Y.M. TSUEI (1992). A Modified Adaptive Grid Method for Recirculating Flows, *Int. J. Numer. Methods Fluids*, 14, 775-791.
23. K. MILLER (1981). Moving Finite Elements II, *SIAM J. Numer. Anal.*, 18, 1033-1057.
24. K. MILLER and R.N. MILLER (1981). Moving Finite Elements I, *SIAM J. Numer. Anal.*, 18, 1019-1032.
25. P.K. MOORE and J.E. FLAHERTY (1992). Adaptive Local Overlapping Grid Methods for Parabolic Systems in Two Space Dimensions, *J. Comput. Phys.*, 98, 54-63.
26. L.R. PETZOLD (1983). A Description of DASSL: A Differential/Algebraic System Solver, in *IMACS Trans. on Scientific Computation*, ed. R.S. STEPLEMAN.
27. L.R. PETZOLD (1987). Observations on an Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations, *Appl. Numer. Math.*, 3, 347-360.
28. M.M. RAI and D.A. ANDERSON (1981). Grid Evolution in Time Asymptotic Problems, *J. Comput. Phys.*, 43, 327-344.
29. M.D. SMOOKE and M.L. KOSZYKOWSKI (1986). Fully Adaptive Solutions of One Dimensional Mixed Initial-Boundary Value Problems with Applications to Unstable Problems in Combustion, *SIAM J. Sci. Stat. Comput.*, 7, 301-321.
30. M.D. SMOOKE and M.L. KOSZYKOWSKI (1986). Two-Dimensional Fully Adaptive Solutions of Solid-Solid Alloying Reactions, *J. Comput. Phys.*, 62, 1-25.
31. J.F. THOMPSON (1985). A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations, *Applied Numer. Maths.*, 1, 3-27.
32. J.F. THOMPSON, Z.U.A. WARSI, and C.W. MASTIN (1985). *Numerical Grid Generation, Foundations and Applications*, North Holland.
33. R.A. TROMPERT and J.G. VERWER (1991). A Static-Regridding Method for Two-Dimensional Parabolic Partial Differential Equations, *Applied Numer. Maths.*, 8, 65-90.
34. A.B. WHITE (1982). On the Numerical Solution of Initial/Boundary-Value Problems in One Space Dimension, *SIAM J. Numer. Anal.*, 19, 683-697.
35. N.N. YANENKO, E.A. KROSHKO, V.V. LISEIKIN, V.M. FOMIN, V.P. SHAPEEV, and YU. A. SHITOV (1976). Methods for the Construction of Moving Grids for Problems of Fluid Dynamics with Big Deformations, in *Lecture Notes in Physics*, 454-459, Springer Verlag.

Chapter 2

A Numerical Study of Three Moving-Grid Methods for One-Dimensional Partial Differential Equations Which Are Based on the Method of Lines

"Het is weer tijd om dubbel 6 te gooien"

2.1. Introduction

It is well known that many discretizations of time-dependent problems in partial differential equations (PDEs) can be derived by means of the following two-stage procedure. First, the space variables are discretized on a selected space mesh, mainly using finite-difference or finite-element approximations, so as to convert the PDE problem into a system of, usually stiff, ordinary differential equations (ODEs) with time as an independent variable. The discretization in time of this stiff ODE system then yields the required fully discretized scheme. In the literature this two-stage approach is often referred to as the method of lines (MOL). With this approach in mind, several sophisticated PDE packages have been developed in recent years, notably for one-space-dimensional problems [3, 4, 11, 15, 25, 26]. These MOL packages greatly benefit from the very successful developments of automatic stiff ODE solvers. Needless to say, the development of implicit BDF codes, initiated by Gear and further improved by, among others, Hindmarsh and Petzold, is a key factor here (see [11, 23] and the references therein). Indeed, certainly for intelligent users who know their problem, Gear-type solvers have proved to be highly 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 for semi-discrete PDE problems.

However, from the PDE point of view, conventional MOL packages integrate in a semi-automatic way in the sense that they adjust the time step sizes automatically, but use a fixed space grid, chosen a priori, for the entire calculation. Depending on the degree of spatial activity, such a space grid is usually equispaced or mildly nonuniform. In many cases this semi-automatic approach works very satisfactorily, notably for problems in which the solution does not exhibit a high degree of spatial

activity, but also for problems where regions of rapid variation in space do not move when time evolves (stationary layers). However, for solutions possessing sharp moving spatial transitions, 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, methods which attempt to adjust automatically both the space and the time step sizes are likely to be more successful in efficiently resolving critical regions of high spatial and temporal activity. Methods and codes which operate this way belong to the realm of adaptive or moving-grid methods.

Over the past several years the interest in moving-grid methods has rapidly increased. Unfortunately, very few, if any, moving-grid software packages, generally applicable up to nearly the same level of efficiency, robustness and reliability as conventional packages, are available yet, even for the relatively simple 1-D case. Admittedly, for an interesting variety of difficult example problems, various adaptive techniques have been shown to be potentially very efficient, a prominent example being the moving-finite-element method invented by Miller and his co-workers [6, 10, 16-18, 20]. However, most of the techniques, including the moving-finite-element method, require some form of tuning to ensure that the automatic choice of the changing space nodes is safely governed. This additional tuning is to the detriment of reliability. Experience so far has made clear that, in general, the automatic space node selection is intrinsically difficult, in the sense that the tuning, being rather problem-dependent, does not lend itself to automation. Hence, algorithms employing moving-grid techniques usually require considerably more expertise of the user than most of the common fixed-grid algorithms in order for the best possible results in terms of efficiency, robustness and reliability to be obtained. Noteworthy, in this connection, is that the moving-grid construction, with the accompanying tuning, is often a determining factor for the computational effort spent in the time integration. Traditionally, this point has been neglected in most of the work on time-dependent problems, probably because the greater part of the development effort is spent in doing a good job in the spatial direction.

Following the philosophy of the MOL approach, this paper is devoted to an evaluation and comparison, mainly based on extensive numerical tests, of three moving-grid methods for 1-D problems, viz., the finite-element method of Miller et al., the method published by Petzold [24], and a method based on ideas adopted from Dorfi and Drury [8]. The two latter ones are finite-difference methods. Concerning the time integration, all these three moving-grid methods can be straightforwardly combined with a stiff solver, just as in the conventional MOL approach. In the referenced papers, interesting results have been shown already using such a type of time integrator. Our examination of the three methods, presented in this paper, is principally aimed at assessing which of the three methods is most suitable from the point of view of retaining the acknowledged features of reliability, robustness and efficiency of the conventional MOL approach. As already indicated by the remark made above, in such an examination the moving-grid determination should be considered not only in relation to spatial solvability properties, but also in relation to the time-stepping process. Hence we shall pay considerable attention to the question of

efficiency of the time-stepping process.

Briefly, the paper is arranged as follows. In Section 2.2 we present an outline of the three methods under consideration, preceded by some general observations on the Lagrangian approach. This approach underlies the two finite-difference methods, while also the finite-element method can be interpreted this way. This section concentrates on the semi-discretization. Section 2.3 deals with the numerical time integration by means of stiff BDF solvers. In Section 2.4 we discuss results of extensive numerical testing on a set of three test models from existing moving-grid literature. This test set includes a reaction-diffusion equation which models a problem from combustion theory, the well-known convection-diffusion equation of Burgers', and a system of two quasi-nonlinear hyperbolic equations, which may be considered as a prototype of an opposite travelling waves problem. It is worth emphasizing at the outset that these three problems show different solution behaviour. This is of importance with respect to our aim, which is to assess which of the three methods under consideration best enables the acknowledged features of reliability, robustness and efficiency of the conventional MOL approach to be realized. We are aware, of course, that experience based on a test set containing three example problems is necessarily limited. By choosing problems differing in solution behaviour, however, we are confident that our conclusions and recommendations have a much wider scope. This holds particularly true for the time integration aspect. Our conclusions and recommendations are summarized in Section 2.5.

To conclude this introduction we wish to emphasize that in the present paper we do not consider the extension of the methods to higher space dimensional problems. It should be acknowledged, however, that work reported by Miller, Baines, Wathen and others contains interesting results in this direction for the moving-finite-element method (see, e.g., [6] and the references therein). We do not know of higher space dimensional applications of the two finite-difference methods examined here.

2.2. Outline of the Moving-Grid Techniques

In order for this article to be read independently, we present in this section a brief outline of the main principles on which the three moving-grid methods are based. In view of the need for brevity, as well as for simplicity of presentation, this outline concentrates on the scalar form. This restriction is not essential. Concerning the automatic grid generation, the principles behind the three methods are the same as for systems and none of the three methods really distinguishes between scalar problems and systems (the necessary changes for systems are always at the implementation level, see [8,10,23] where applications to systems are discussed). For clarity, Section 2.2 deals only with the semi-discretization. We begin our outline with some general observations on the Lagrangian approach.

2.2.1. The Lagrangian approach

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 thus is 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. Lest we miss the obvious, 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. This is inevitable, because when on a non-moving mesh the moving front passes a grid point, the solution at this grid point will change very rapidly. Small time steps are then necessary to retain accuracy.

The above observation naturally leads one to consider the Lagrangian approach, which is best introduced via a co-ordinate transformation. Consider the PDE problem

$$\partial u/\partial t = \mathcal{L}(u), \quad x_L < x < x_R, \ t>0, \tag{2.1}$$

where $\mathcal{L}$ represents a differential operator involving only spatial derivatives, e.g.,

$$\partial u/\partial t = \mathcal{L}(u) := -\partial c(u)/\partial x + \varepsilon\, \partial^2 u/\partial x^2 + g(u), \tag{2.1a}$$

$$x_L < x < x_R, \ t>0, \ \varepsilon>0.$$

The space interval is supposed to be fixed for all times $t>0$ under consideration. Let (s,t) be new independent variables linked with the old independent variables (x,t) through a co-ordinate transformation $x = x(s,t)$. Denote $v(s,t) = u(x,t)$. Then the total derivative of u is $\partial v/\partial t = \partial u/\partial x\, \partial x/\partial t + \partial u/\partial t$ and the Lagrangian form of (2.1) reads

$$\partial v/\partial t = \partial u/\partial x\, \partial x/\partial t + \mathcal{L}(u), \quad s_L < s < s_R, \ t>0, \tag{2.2}$$

and that of (2.1a),

$$\partial v/\partial t = \partial u/\partial x\, \partial x/\partial t - \partial c(u)/\partial x + \varepsilon\, \partial^2 u/\partial x^2 + g(u), \tag{2.2a}$$

$$s_L < s < s_R, \ t>0.$$

Note that $\partial u/\partial t$ measures the changes of u as a function of t at a fixed x value (Eulerian description) and $\partial v/\partial t$ at a fixed s value (Lagrangian description). Thus the basic idea of the Lagrangian approach is that in the variables (s,t) the problem should be easier to handle numerically than in the original pair (x,t). Ideally, in the new variables any rapid transition should be absent; we can then take acceptable step sizes in the time direction while using a coarse uniform s-grid in space. A suitable nonuniform x-grid then exists according to the change of variables $x = x(s,t)$.

In classical Lagrangian methods, as are being applied successfully to some types of fluid-flow problems, the movement of the nodes is attached, in an a priori manner, to a physically motivated, specific flow quantity. For example, for a problem like (2.1a) it makes sense to attach the movement of the nodes to the convection term $\partial c(u)/\partial x$, i.e., to choose $\partial x/\partial t = dc(u)/du$ so as to obtain the parabolic equation $\partial v/\partial t = \varepsilon\, \partial^2 u/\partial x^2 + g(u)$ (in a moving reference frame). The rationale behind

this choice is that parabolic problems without large first order terms usually possess smoother solutions and thus are less difficult to solve numerically. Of course, the numerical realization of the prescription $\partial x/\partial t = dc(u)/du$ involves its own difficulties, but these are usually surmountable.

Because we aim at application to a wide variety of problems we require that the transformation be based on a general 'systematic rule', e.g., spatial equidistribution. In fact, the choice of this 'systematic rule' determines to a great extent the moving-grid method under consideration. This will be illustrated quite clearly in the remaining sections. Here we wish to point out that it is not always possible to smooth the solution, through the co-ordinate transformation, in space and in time simultaneously. This obviously depends on the nature of the solution sought, which can be nicely illustrated by examining Problem I of Section 2.4 (cf. [27], Section 2.5.3). Let us consider its solution near the left boundary, while the steep front is forming (the ignition phase). Assuming a uniform grid at the initial line (a choice suggested by the constant initial solution $u(x, 0) = 1$), the derivatives $\partial x/\partial t$ of many of the trajectories should be negative in order for the required refinement in the region of the steep front to be obtained, which is in accordance with the objective of smoothing the problem in space. However, during the formation of the front, $\partial u/\partial x < 0$ and $\partial u/\partial t > 0$. It then follows immediately that $\partial v/\partial t > \partial u/\partial t$, violating the objective of getting a smoother problem in time. Most Lagrangian-type methods do underly the first objective through a co-ordinate transformation based on spatial equidistribution properties. Spatial equidistribution forces nodes to migrate to regions of high spatial activity. So, during the formation of the front, for the present combustion problem these methods offer no benefit as far as the time stepping is concerned. Once the front is formed and starts to propagate, both smoothing objectives are fulfilled if the transformation underlies spatial equidistribution, because then $\partial x/\partial t > 0$ and still $\partial u/\partial x < 0$ and $\partial u/\partial t > 0$. Any simple travelling wave form $u(x,t) = f(x - ct)$ is a trivial solution, in this respect, provided the grid trajectories satisfy $\partial x/\partial t = c$. Interestingly, the Lagrangian approach followed by Petzold [23] underlies the second objective. This approach, originally due to Hyman [13], is basically aimed at finding those trajectories along which the time rate of solution change is minimized, that is, $\partial v/\partial t < \partial u/\partial t$. However, during the formation of the front in the present combustion example, this must imply that, in the front region, $\partial x/\partial t > 0$, which means that points are moved away from the front, and thus the first objective is violated. This is contrary to the desired aim; however, Petzold's algorithm has a built-in regridding step which corrects this deficiency (see the next section). For this method it also holds that, once the front is formed and starts to propagate, both smoothing objectives are fulfilled.

The two finite-difference methods we examine are based on the standard, central semi-discretization of the above Lagrangian form (2.2). More precisely, completely in line with the common MOL approach, consider numerical, continuous-time trajectories

$$x_L = X_0 < \ldots < X_i(t) < X_{i+1}(t) < \ldots < X_{N+1} = x_R \quad \text{for } 0 \le t \le t_{end}, \tag{2.3}$$

with the associated grid functions

$$U = [U_1, \cdots, U_N]^T, \quad X = [X_1, \cdots, X_N]^T.$$

Thus, U represents the semi-discrete approximation to the PDE solution u restricted to the moving space grid X and is the solution of the ODE system

$$\dot{U}_i = \dot{X}_i[(U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1})] + L_i(U, X), \quad 1 \le i \le N, \tag{2.4}$$

where the symbol $\dot{}$ denotes differentiation to time, $\dot{U}_i$ denotes the semi-discrete total derivative and the operator L stands for the difference operator replacing the differential operator $\mathcal{L}$ on the grid. For example, the right-hand side function of (2.1a) is approximated at grid point i, $1 \le i \le N$, by

$$\begin{aligned} L_i(U,X) = & -\{[c(U_{i+1}) - c(U_{i-1})]/[X_{i+1} - X_{i-1}]\} \\ & + \varepsilon\{((U_{i+1} - U_i)/(X_{i+1} - X_i) \\ & - (U_i - U_{i-1})/(X_i - X_{i-1}))/(0.5(X_{i+1} - X_{i-1}))\} + g_i(U). \end{aligned}$$

In the discussion to follow, we neglect the treatment of boundary conditions, since these are dealt with in the usual way. We recall 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. In this paper, the central approximation is used since it facilitates comparisons between the three methods (the finite-element method uses standard piecewise linear basis functions) and because it represents a severe test for a good moving grid $X(t)$. Any deviation from an ideal Lagrangian grid movement, assuming this exists, will soon result in unphysical, oscillatory solutions. As already indicated above, the definition of $X(t)$ is highly important and determines to a great extent the complete moving-grid method.

2.2.2. Method I

Method I is the finite-difference moving-grid method proposed by Petzold [23] (version A). Each time step consists of two computational stages: a moving Lagrangian step, involving the application of a stiff ODE solver to an augmented semi-discrete system, followed by a second (regridding) stage in which a redistribution of points at the forward time level is carried out through a De Boor-type equidistribution algorithm. Both are equally important for the application of the method. However, in contrast to most methods, grid points are not necessarily moved in the desired direction of high spatial activity. Loosely speaking, one of the purposes of the regridding stage is to correct this deficiency.

The semi-discrete system

We begin our outline with the derivation of the (augmented) semi-discrete system, which consists of the equations of system (2.4) together with grid equations for the implicit determination of the unknown grid X. Consider the Lagrangian form (2.2) where, for convenience of notation, $\dot{u}$ and $\dot{x}$ now denote the derivatives $\partial v/\partial t$ and $\partial x/\partial t$, respectively. The underlying transformation, which is originally due to Hyman [13], is chosen to minimize in a certain sense the total derivative $\dot{u}$. This is done by selecting $\dot{x}$ such that

$$(\dot{u})^2 + \alpha\,(\dot{x})^2 = (\partial u/\partial t + \dot{x}\;\partial u/\partial x)^2 + \alpha(\dot{x})^2$$

is minimized, where $\alpha \geq 0$ is a real number. Differentiation to $\dot{x}$ and equating to zero yields the differential expression $\dot{u}\;\partial u/\partial x + \alpha\dot{x} = 0$, which can be written as the single ODE

$$\dot{x} = (-\partial u/\partial t\;\partial u/\partial x)/(\alpha + (\partial u/\partial x)^2),$$

with x as a dependent and t as an independent variable, provided u, $\partial u/\partial t$ and $\partial u/\partial x$ are known functions of x. For given $x(0)$ on the initial space interval, the solution of this ODE defines the trajectory $x(t)$ $(t \geq 0)$ along which the rate of change of u, that is

$$\dot{u} = (\alpha\;\partial u/\partial t)/(\alpha + (\partial u/\partial x)^2),$$

is minimized in the above sense. It is hereby tacitly assumed that the above ODE is uniquely solvable. The parameter α serves to regularize the transformation. For $\alpha = 0$ we have $\dot{u} = 0$, which in general cannot be a solution. Observe that, in regions where $(\partial u/\partial x)^2$ is negligible relative to α, the transformation has no effect. The travelling wave form $u(x,t) = w(x - ct)$ nicely shows the idea behind this transformation. For this solution we have

$$\dot{x} = (c\;(\partial w/\partial x)^2)/(\alpha + (\partial w/\partial x)^2),$$

$$\dot{u} = (-\alpha\, c(\partial w/\partial x))/(\alpha + (\partial w/\partial x)^2)$$

and for $\alpha = 0$ the grid point $x(t)$ moves with the wave with speed c. Recall, however, that grid points are not always moved in the desired direction.

Hence, the transformation employed leads to the grid equation

$$\dot{u}\;\partial u/\partial x + \alpha\,\dot{x} = 0, \qquad s_L < s < s_R,\; t > 0.$$

When combined with (2.2), it can be solved for the unknowns u and x. The grid equation is written in this form to avoid ill-conditioning problems in the numerical solution process [23]. At this stage it is pointed out that in actual application the new variable s is not used explicitly, that is, computations will always be performed in terms of the original variables (x,t). Note that explicit use of s would require that its bounds be properly defined, which we have not done. Like (2.2), this grid equation is spatially discretized on the grid (2.3) so that we obtain

$$\dot{U}_i[(U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1})] + \alpha\,\dot{X}_i = 0, \qquad 1 \leq i \leq N. \tag{2.5}$$

Equations (2.4), (2.5) form the augmented, semi-discrete system and define the unknown variables U and X.

In addition to the regularization term $\alpha\dot{x}$, the grid computation needs an extra regularization to prevent neighbouring grid points from crossing. Note that, even when the single ODE for the exact grid trajectory is uniquely solvable, the grid trajectories for a set of given initial points may approach each other arbitrarily closely. Petzold [23] has suggested that, instead of (2.5),

$$\begin{aligned}&\dot{U}_i[(U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1})] + \alpha\,\dot{X}_i \\ &+ \lambda\,[(\dot{X}_i - \dot{X}_{i-1})/(X_i - X_{i-1})^2) - (\dot{X}_{i+1} - \dot{X}_i)/(X_{i+1} - X_i)^2] \;=\; 0\end{aligned} \tag{2.6}$$

should be used, where $\lambda > 0$ is the second regularization parameter. This form results when

$$(\dot{u}_i)^2 + \alpha\,(\dot{x}_i)^2 \qquad (2.7)$$
$$+ \lambda\,[(\dot{x}_i - \dot{x}_{i-1})^2/(x_i - x_{i-1})^2 + (\dot{x}_{i+1} - \dot{x}_i)^2/(x_{i+1} - x_i)^2]$$

is minimized with respect to $\dot{X}_i$. This regularization term is related to the 'internodal viscosity' term of the moving-finite-element method (see Section 2.2.4). The use of this type of regularization is based on heuristic considerations. If neighbouring points tend to approach each other very closely, the denominators in (2.6) will eventually decrease beyond the level needed to let the regularization term dominate the entire expression. If this happens, the minimization procedure will result in nearly equal neighbouring grid velocities, with the effect that, when time evolves, neighbouring points are prevented from approaching further.

Necessarily, the regularization is problem-dependent and in actual application there is no guarantee that points will not cross. On the other hand, at sufficiently large λ the grid becomes non-moving. Hence, if λ is chosen too large, it may happen that points are forced to stay apart too much so that locally the grid is not fine enough to resolve anticipated small-scale structures. Following [23], we have used throughout the values $\alpha = 1$, $\lambda = 0.2$. Needless to say other choices of regularization terms are conceivable. It should be emphasized, though, that it is not easy, if possible at all, to find an optimal regularization. It is noteworthy that regularization always has some smoothing effect on the grid trajectories, which is desirable for the time integration. Hence, regularization not only influences the spatial solvability performance of the moving-grid method, but also the performance of the stiff solver.

In order to bring the augmented semi-discrete system (2.4), (2.6) into a more compact form, we introduce the vector $Y = [U_1, X_1, \ldots, U_i, X_i, \ldots, U_N, X_N]^T$. The semi-discrete system then takes the linearly implicit form

$$\mathcal{A}(Y)\dot{Y} = G(Y) \quad \text{for } t > 0 \text{ and } Y(0) \text{ given}, \qquad (2.8)$$

where $\mathcal{A}(Y)$ is block tridiagonal and the $(2i-1)$th and the $(2i)$th element of the vector-valued function G are given by

$$G_{2i-1}(Y) = L_i(U,X), \quad G_{2i}(Y) = 0, \quad (1 \le i \le N). \qquad (2.9)$$

Inspection of the matrix $\mathcal{M} = -\mathcal{A}$ reveals that for any vector Y its symmetric part $(\mathcal{M} + \mathcal{M}^T)/2$ is negative definite, so that, according to the known property that the real part of any eigenvalue is smaller than or equal to the maximum eigenvalue of $(\mathcal{M} + \mathcal{M}^T)/2$, the matrix $\mathcal{A}$ is non-singular. This means that system (2.8) is a genuine, stiff ODE system. Even when grid points cross, the matrix remains non-singular. This is handy because it means that crossing need not be fatal. More precisely, after each (modified) Newton iteration within an implicit moving integration step with the stiff solver, a check on crossing is made. If crossing is detected, the current step is interrupted and redone with a smaller step size.

The regridding step

The above transformation is interesting in itself, because it provides a smoother

problem in time. This will be beneficial for the numerical integration process. A disadvantage is that this transformation may not necessarily move the grid points in the direction of high spatial activity. To overcome this deficiency, an intermediate regridding is carried out, in principle after every successful moving integration step.

Suppose the moving step with the stiff solver has delivered the numerical (vector) values U^n, X^n at the n-th forward time level. Then, by application of a De Boor-type regridding algorithm, which uses U^n, X^n for input, a second level n grid is determined. This new grid, Z^n, say, satisfies (approximately)

$$\Delta z \, |\partial u/\partial x| + (\Delta z)^2 \, |\partial^2 u/\partial x^2| = \text{constant}. \tag{2.10}$$

It would lead us too far here to discuss the implemented De Boor algorithm in detail. Here we only remark that we keep the number of moving points fixed, whereas Petzold [23] adapts the number of moving points so that (approximately)

$$\Delta z \, |\partial u/\partial x| + (\Delta z)^2 \, |\partial^2 u/\partial x^2| \leq \text{specified tolerance},$$

while the number of points is the smallest number needed to satisfy this inequality. We have decided to work with a fixed, given number of moving points for comparison with the other two methods. In conclusion, Z^n equidistributes (2.10), which has the effect that points are concentrated in regions of high spatial activity. This alleviates the deficiency mentioned above. Because the grid Z^n will normally differ from X^n, it is necessary to interpolate from X^n onto Z^n prior to the next moving integration step. This is done via the 'dual reconnecting grid' approach, which is a compromise between choosing the best grid and avoiding needless interpolations. Briefly, the idea is as follows. Z^n divides the space interval into zones. Each zone is allowed to contain one point from X^n. If a zone contains just one point, no interpolation takes place. If a zone is empty, a point is added and a (monotone) interpolation is carried out. If there are more points from X^n in a zone, points are deleted. Grid points at the edge of zones which are too close to other points are moved apart. In this way the final grid to be used for the next moving step is created. Hence, on most time steps only a few interpolations are carried out (and eventually none). This is of importance, since interpolation usually damages the accuracy a little. Another attractive feature of the dual reconnecting grid approach is that points can be added and deleted locally. This is advantageous when locally the solution undergoes sudden rapid changes (birth of new layers).

When considered on its own, the idea of intermediate regridding is interesting because, as a sort of added bonus, it provides the possibility of more direct control on the placement of nodes through equidistribution (and connected herewith heuristic spatial error monitoring based on equation (2.16)). One could say that the intermediate regridding step makes the regularization less critical, though regularization should not be omitted. A considerable disadvantage of regridding is that it necessitates interpolation and that it interrupts the time-stepping process. Frequent interpolation may damage the accuracy considerably, while the interruption of the time-stepping process causes a restart situation for the stiff solver (in our case a BDF solver). In other words, after a regridding the Jacobian matrix is updated and the integration is continued with the implicit Euler method on the newly chosen grid

(with that step size that would have been used on the next step had there been no restart). The inevitable consequence is that, when there are many genuine regriddings, the solver does not get the chance of switching to a higher order formula, which no doubt is detrimental when high accuracy in time is needed. It is clear that this situation is somewhat in contradiction with the MOL approach and that there is room for some improvement here [23].

2.2.3. Method II

Method II is also a finite-difference method based on the semi-discrete Lagrangian form (2.4). The main ideas of moving the grid are derived from Dorfi and Drury [8]. An implicit equation for $X(t)$ is used which underlies a spatial equidistribution transformation based on an arc-length monitor function. An important feature of Method II is that the grid movement is regularized by employing a smoothing technique in both 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 ensures a smooth progression of $X(t)$ by preventing the points from responding too quickly to current solution gradients. This is highly desirable for efficient numerical time stepping.

The semi-discrete system

We shall derive the semi-discrete grid equations for the implicit determination of the moving grid $X(t)$. Let us first recall the idea of the spatial equidistribution transformation which is used in Method II. Hence, the theoretical co-ordinate transformation supposed in equation (2.2) is now of the form

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

where $M(x,t)$ is a chosen monitor function which should reflect space dependence of the PDE solution. The spatial equidistribution of this monitor function is enforced by dividing the interval $0 \le s \le 1$ into $N+1$ equal parts. Through the inverse transformation, the N theoretical grid trajectories $x_i(t) = x(i/N,t)$, $t \ge 0$, $(1 \le i \le N)$ where x_0, x_{N+1} are the given boundaries x_L and x_R, respectively, then satisfy the equidistribution relation

$$\int_{x_i}^{x_{i+1}} M(\xi,t)\, d\xi = \eta(t)/N \quad (0 \le i \le N).$$

Consequently, in regions where M will be large, the grid trajectories will become close and vice versa. By applying the midpoint quadrature rule and inserting semi-discrete variables, at the semi-discrete level this equidistribution relation is taken to be

$$(X_{i+1} - X_i)\, M_i = \text{constant} \quad (0 \le i \le N), \tag{2.11}$$

where M_i now represents the semi-discrete monitor value at the midpoint of the i-th sub-interval $[X_i, X_{i+1}]$. Following Dorfi and Drury, we use the arc-length monitor

$$M_i = (1 + (U_{i+1} - U_i)^2/(X_{i+1} - X_i)^2)^{1/2},$$

which has the property of placing grid points along uniform arc-length intervals and gives good point placement at the 'lip' of a shock. Of course, other choices of M are conceivable. Because M is positive, solution values X_i of (2.11) cannot cross. For a discussion of monitor functions and equidistribution, see for example Pereyra and Sewell [22], Furzeland [9], Carey and Dinh [5].

By elimination of the constant in (2.11), a set of N semi-discrete grid equations for the implicit determination of the moving grid X is obtained

$$(X_i - X_{i-1})\, M_{i-1} = (X_{i+1} - X_i)\, M_i \quad (1 \le i \le N). \tag{2.12}$$

Combining these with the semi-discrete PDE equations (2.4) yields the (augmented) semi-discrete problem for the unknown grid functions U and X. However, as mentioned previously, Dorfi and Drury regularize the grid movement by performing a smoothing technique both in space and time. This amounts to modifying the grid equation system (2.12). We shall first describe their modification for the spatial grid smoothing.

For this purpose we introduce the point concentrations

$$n_i = 1/(X_{i+1} - X_i) \quad (0 \le i \le N). \tag{2.13}$$

Using these variables, the grid equation system (2.12) is written in the form

$$n_{i-1}/M_{i-1} = n_i/M_i \quad (1 \le i \le N) \tag{2.14}$$

and the spatial grid smoothing is then carried out by replacing the point concentrations in this system by their smoothed (numerically diffused) counterparts

$$\tilde{n}_i = n_i - \kappa(\kappa+1)(n_{i+1} - 2n_i + n_{i-1}), \quad \kappa > 0, \tag{2.15}$$

to obtain the new grid equation system

$$\tilde{n}_{i-1}/M_{i-1} = \tilde{n}_i/M_i \quad (2 \le i \le N-1). \tag{2.16}$$

Neglecting the influence of the boundaries, it can be shown that this filtering procedure is equivalent to a certain smoothing procedure for the monitor function [8], thus ensuring that the adjacent point concentrations are restricted such that

$$\kappa/(\kappa+1) \le n_{i-1}/n_i \le (\kappa+1)/\kappa. \tag{2.17}$$

This spatial smoothing can also be achieved by 'padding' the monitor function [9, 14], but this approach is not recommended here since within a MOL framework the implicit coupling between X and U then varies at each time step. For a given N and a given monitor function distribution, the choice of κ determines the minimum and maximum interval lengths. The monitor function determines the relative shape of the X_i distribution, κ and N determine the absolute level of clustering [8]. In actual application, a value of κ of about 1 or 2 is recommended. This yields modestly graded space grids. In our experiments we have used $\kappa = 2$. The value of κ plays an important role in controlling space discretization errors on non-uniform grids (see, for example, [9]).

System (2.16) must be completed with boundary conditions. Following [8], at the

boundaries the 'concentration gradients' are set to zero,

$$n_0 = n_1 \quad \text{and} \quad n_{N-1} = n_N. \tag{2.18}$$

Note that the use of the grid equations (2.16), (2.18) introduces a 5-point coupling in X. Needless to say, this slightly increases the computational costs of the method (per step).

The temporal grid smoothing described next replaces the set of algebraic equations (2.16) by the following set of differential equations

$$(\tilde{n}_{i-1} + \tau\, d\tilde{n}_{i-1}/dt)/M_{i-1} = (\tilde{n}_i + \tau\, d\tilde{n}_i/dt)/M_i, \tag{2.19}$$
$$\tau \geq 0 \quad (2 \leq i \leq N-1),$$

again with boundary conditions (2.18). This system is constructed as follows. Consider the monitor function values $M(t)$ occurring in equation (2.16) (for convenience of notation we suppress the lower index i). The temporal grid smoothing hinges on the replacement of $M(t)$ by

$$R(t) = \int_0^\infty M(t-\sigma\tau)\, e^{-\sigma} d\sigma, \quad \tau \geq 0,$$

where $M(t)$ is now thought of as being defined on the semi-infinite interval $[-\infty, t]$. In actual application, the extension to the interval $[-\infty, 0]$ is neglected. This is allowed due to the presence of the exponential damping factor and the fact that the parameter τ is supposed to be rather small (the choice $\tau = 0$ yields $R(t) = M(t)$). By partial integration the differential form

$$M(t) = R(t) + \tau\, dR(t)/dt$$

can be recovered, which is used to construct (2.19). More precisely, the numerically diffused point concentration values $\tilde{n}(t)$ of equation (2.16) are now taken proportional to $R(t)$, rather than to $M(t)$. Let $c(t)$ be the proportionality constant, that is, $R(t) = c(t)\,\tilde{n}(t)$. Substitution into this differential form gives

$$M(t) = c(t)(\tilde{n}(t) + \tau\, d\tilde{n}(t)/dt) + \tau\, \tilde{n}(t) dc(t)/dt.$$

If we then neglect the time dependence of the proportionality constant $c(t)$, and subsequently eliminate it, the grid equation system (2.19) is recovered.

The motivation behind the use of the monitor function $R(t)$, which is 'averaged in time', is that, when the grid movement is attached to $R(t)$ rather than to $M(t)$, it is prevented from adjusting immediately to the new monitor values. Instead, the use of $R(t)$ 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 of this approach is to avoid temporal oscillations in the original grid trajectories defined by (2.16). These oscillations are typical for grids generated via numerical spatial equidistribution techniques. When applied to solutions with very large gradients, relatively large errors occur with these techniques. Needless to say, for the numerical time integration a smooth grid $X(t)$ is highly desirable, otherwise too many Jacobian evaluations are needed when an implicit solver is applied.

Albeit heuristic in nature, there is no doubt that the temporal grid smoothing procedure is of importance. The choice of the delay factor τ requires some expertise but, in our experience, this is not too critical. Increasing τ too much results in a grid that lags too far behind any propagating wave or shock. Note that, for sufficiently large values of τ, a non-moving grid results. Trivially, too small values for τ render no effect. In practice it makes sense 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 stabilizing effect of τ is similar to that of the damping factor λ introduced in Coyle, Flaherty and Ludwig [7].

To sum up, the semi-discrete grid equations (2.19) with the boundary conditions (2.18) determine the continuous-time moving grid $X(t)$. Of importance to note is that we work with the $2N$ unknowns $U_i, X_i (1 \leq i \leq N)$ and that in our implementation the point concentration derivatives that occur in (2.15), (2.19) are replaced by

$$dn_i/dt = -(\dot{X}_{i+1} - \dot{X}_i)/(X_{i+1} - X_i)^2.$$

More specifically, in the numerical integration $U_i(t)$ and $X_i(t)$ are computed with the same integration formulas, which is different from the implementation in [8] (see formula (10)). Consequently, in our case the i-th equation of system (2.19) couples the nodal points

$$X_{i+2}, \quad X_{i+1}, \quad X_i, \quad X_{i-1}, \quad X_{i-2},$$

with the nodal point velocities

$$\dot{X}_{i+2}, \quad \dot{X}_{i+1}, \quad \dot{X}_i, \quad \dot{X}_{i-1}, \quad \dot{X}_{i-2},$$

and the solution values

$$U_{i-1}, \quad U_i, \quad U_{i+1}.$$

A little inspection reveals that the vector version of the final augmented semi-discretized system of ODEs can be brought to a linearly implicit ODE form with a known bandwidth in a similar way to that used for Method I (cf. (2.8)),

$$\mathcal{A}(Y)\dot{Y} = G(Y) \quad \text{for } t > 0 \text{ and } Y(0) \text{ given.} \tag{2.20}$$

2.2.4. Method III

Method III is the moving-finite-element method introduced by Miller et al. [16, 20]. This method also generates a system of continuous-time ODEs for mesh points and numerical approximations in these moving points. The grid movement is regularized by using penalty functions.

The semi-discrete system

Consider the continuous-time grid X introduced in (2.3) with unknown components. On such a grid, the moving-finite-element method approximates the solution $u(x,t)$ of problem (2.1) by an expansion (summation from 1 to N)

$$U(x,t) = \sum_i U_i(t)\, \alpha_i(x, X(t)), \tag{2.21}$$

where α_i are the standard piecewise linear basis functions that depend on the nodal positions X_i and U_i are the amplitudes of the approximate solution $U(x,t)$ at the corresponding nodal positions. Differentiating this expression with respect to t gives, after some elementary calculations

$$U_t = \sum_i \dot{U}_i \alpha_i + \dot{X}_i \beta_i, \tag{2.22}$$

where the β_i are piecewise linear discontinuous basis functions with the same support as α_i. We have

$$\beta_i(x) = \begin{cases} -m_i \alpha_i & \text{for } X_{i-1} \le x \le X_i, \\ -m_{i+1} \alpha_i & \text{for } X_i \le x \le X_{i+1}, \\ 0 & \text{elsewhere,} \end{cases}$$

where $m_i = (U_i - U_{i-1})/(X_i - X_{i-1})$ is the slope of the semi-discrete approximation $U(x,t)$ on $[X_{i-1}, X_i]$. It is of interest to note that (2.22) is akin to the Lagrangian form (2.2); $\dot{U}_i$ plays the role of the Lagrangian derivative $\partial v/\partial t$ and the nodal velocity $\dot{X}_i$ that of $\partial x/\partial t$ (see [2, 9, 21] for a discussion of the Lagragian nature of Method III).

The equations determining the semi-discrete unknowns U_i and X_i are now obtained in the standard Galerkin way by minimizing the square of the L_2-norm of the residual $R(U) = U_t - \mathcal{L}(U)$ with respect to $\dot{U}_i$ and $\dot{X}_i$. This gives a system of $2N$ equations in the $2N$ unknowns U_i, X_i (boundary conditions are incorporated in the standard way):

$$\sum_j < \alpha_i, \alpha_j > \dot{U}_j + < \alpha_i, \beta_j > \dot{X}_j = < \alpha_i, \mathcal{L}(U) > \quad 1 \le i \le N, \tag{2.23a}$$

$$\sum_j < \beta_i, \alpha_j > \dot{U}_j + < \beta_i, \beta_j > \dot{X}_j = < \beta_i, \mathcal{L}(U) > \quad 1 \le i \le N, \tag{2.23b}$$

where <.,.> denotes the usual inner product. Assuming zero velocities, the first equation is readily recognized as the standard, semi-discrete Galerkin equation. The second equation originates from the additional minimization with respect to the nodal velocities. Using the linear forms for α_i and β_i, the inner products on the left-hand side may be evaluated to give, respectively,

$$\frac{1}{6}[\Delta X_i \dot{U}_{i-1} + 2(\Delta X_i + \Delta X_{i+1})\dot{U}_i + \Delta X_{i+1} \dot{U}_{i+1}] \tag{2.24a}$$

$$- \frac{1}{6}[\Delta U_i \dot{X}_{i-1} + 2(\Delta U_i + \Delta U_{i+1})\dot{X}_i + \Delta U_{i+1} \dot{X}_{i+1}]$$

$$= < \alpha_i, \mathcal{L}(U) >, \quad 1 \le i \le N,$$

$$- \frac{1}{6}[\Delta U_i \dot{U}_{i-1} + 2(\Delta U_i + \Delta U_{i+1})\dot{U}_i + \Delta U_{i+1} \dot{U}_{i+1}] \tag{2.24b}$$

$$+ \frac{1}{6}[m_i \Delta U_i \dot{X}_{i-1} + 2(m_i \Delta U_i + m_{i+1} \Delta U_{i+1})\dot{X}_i + m_{i+1} \Delta U_{i+1} \dot{X}_{i+1}]$$

$$= < \beta_i, \mathcal{L}(U) >, \quad 1 \le i \le N,$$

where $\Delta X_i = X_i - X_{i-1}$, etc. Using the vector notation $Y = [U_1, X_1, \ldots, U_i, X_i, \ldots, U_N, X_N]^T$, we thus arrive at the continuous-time, semi-discrete moving-finite-element system

$$\mathcal{A}(Y)\dot{Y} = G(Y) \quad \text{for } t > 0 \text{ and } Y(0) \text{ given,} \tag{2.25}$$

where $\mathcal{A}(Y)$ is a block tridiagonal matrix and $G(Y)$ is given by

$$G(Y) = (<\alpha_1, \mathcal{L}(U)>, <\beta_1, \mathcal{L}(U)>, \ldots, <\alpha_N, \mathcal{L}(U)>, <\beta_N, \mathcal{L}(U)>)^T.$$

The matrix $\mathcal{A}(Y)$ contains only quantities from the left-hand sides of (2.24), which are related to the discretization of $\partial u/\partial t$ on the moving mesh (cf. (2.8), (2.20)). What remains now is to integrate this ODE system numerically to obtain the required fully discretized solution.

The moving-finite-element method has aroused considerable interest yet at the same time has been subject to criticism because of its complexity and the inherent problems of parallelism and points drifting extremely close together. Parallelism occurs when the gradients of U on adjacent cells, say m_i and m_{i+1}, become equal. The $(2i+1)$–*th* column of $\mathcal{A}$ is then equal to m_i times the $2i$-th column, so that the mass matrix $\mathcal{A}$ becomes singular. When nodes drift extremely close together, the mesh may become tangled or nodes may even cross in the numerical integration process. Miller [17] suggests that these two problems can be overcome by introducing regularization terms (penalty functions) in the residual minimization. Instead of using $R(U)$ alone the minimization is thus carried out for

$$< R(U), R(U) > + \sum_j (\varepsilon_j \Delta\dot{X}_j - S_j)^2,$$

where

$$\varepsilon_i^2 = C_1^2/(\Delta X_i - d), \quad \varepsilon_i S_i = C_2^2/(\Delta X_i - d)^2, \tag{2.26}$$

with C_1, C_2 and d small, user-chosen constants. In particular, d serves as a user-defined minimal node distance. The modifications involved are only made to the mesh point equations (2.23b) and the combined effect is to add

$$\varepsilon_i^2 \Delta\dot{X}_i - \varepsilon_{i+1}^2 \Delta\dot{X}_{i+1} \quad \text{and} \quad \varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}$$

to the left- and right-hand side, respectively. The ε-terms serve to avoid parallelism. It can be shown that the addition of these terms renders the mass matrix $\mathcal{A}$ diagonally dominant [18], and thus regular. They represent a form of 'internodal' viscosity, since they penalize relative motion between nodes and, provided the penalty is sufficiently large to take over before the mass matrix becomes numerically singular, result in the degenerate nodes being carried along with the rest of the solution. The ε-terms do prevent node overtaking in a dynamic way since the internodal viscosities become infinite as Δx tends to zero; however over longer time intervals degenerate nodes (those caught in straight line segments where they are unneeded) may still slowly drift together. The S-terms, sometimes called internodal spring forces, serve to prevent this long term numerical drift.

As for any other method, the regularization is somewhat heuristic and necessarily problem-dependent. For example, if C_1 is chosen too large, the grid movement is

restricted ($C_1 \approx \infty$ gives a non-moving grid) with the result that there may not be sufficient refinement in regions of large spatial activity (a typical phenomenon is then that the grid moves slower than a front region). On the other hand, if C_1 is too small, the mass matrix $\mathcal{A}$ may become numerically singular. Also of great importance is that the minimal node distance d be small enough in relation to the anticipated small-scale structure. However, too small values of d and C_2 may allow numerical errors to lead to near node overtaking (or even worse), which is a source of severe numerical difficulties in the time integration, even for the most robust stiff solver. When nodes drift extremely close together, the sets of nonlinear algebraic equations to be solved at each time step are likely to become badly conditioned. This hampers the Newton iterative process and results in a higher number of iterations and Jacobian updates than in the conventional MOL application. It is our experience that Method III is rather sensitive in this respect. We shall illustrate this extensively in the discussion of the numerical experiments.

At this place we should also mention that the explicit time stepping approach advocated by Baines, Wathen and their co-workers (see [2] and the references contained in [6]) is aimed at avoiding the necessity of regularization with the accompanying difficulties. However, while these explicit techniques work very successfully on purely first order hyperbolic problems, they obviously suffer from the explicit time-step restriction when applied to parabolic problems, including those of the diffusion-convection type, even with little diffusion. Therefore we consider explicit techniques as less feasible for use in a general-purpose MOL algorithm.

Finally, we list some of the inner products that are needed to handle our test problems (g represents a nonlinear source function; cf. [10]):

$$
\begin{aligned}
&<\alpha_i, U_{xx}> = m_{i+1} - m_i, \quad <\beta_i, U_{xx}> = -(m_{i+1} - m_i)(m_{i+1} + m_i)/2,\\
&<\alpha_i, -UU_x> = -\Delta U_i(U_i/9 + U_{i-1}/18) - \Delta U_{i+1}(U_i/9 + U_{i+1}/18),\\
&<\beta_i, -UU_x> = m_i \Delta U_i(U_i/3 + U_{i-1}/6) + m_{i+1}\Delta U_{i+1}(U_i/3 + U_{i+1}/6),\\
&<\alpha_i, g(U)> = [g((U_{i-1} + U_i)/2)\Delta X_i + g((U_{i+1} + U_i)/2)\Delta X_{i+1}]/2,\\
&<\beta_i, g(U)> = -[m_i g((U_{i-1} + U_i)/2)\Delta X_i\\
&\qquad\qquad + m_{i+1} g((U_{i+1} + U_i)/2)\Delta X_{i+1}]/2.
\end{aligned}
$$

2.3. The Numerical Time Integration

For the numerical time integration of the three derived semi-discrete systems (2.8), (2.20) and (2.25), we have used two existing stiff Gear solvers. All results for Method I have been obtained with the original (version A) source code of Petzold [23], except that we have applied it with a fixed number of moving points. Consequently, Method I uses Petzold's own BDF code DASSL. The software implementing Methods II and III has been prepared by ourselves. Both these methods use the LSODI-based BDF code of the SPRINT package [3, 4] for the time integration. Because the Gear codes of SPRINT and DASSL are very much alike, the choice between the two should be of minor importance for the performances observed. For all three methods use of the banded form of the equations is exploited in the Jacobian formation and numerical linear algebra computations.

From the user's point of view it is of interest to note that the stiff solvers can be used in the same way as in the conventional approach. Apart from providing a subroutine for the ODE system (numerical differencing for Jacobians was used) and specifying the initial vector $Y(0)$ and required output times, one must define the familiar local error tolerances *atol* and *rtol* , the desired local error norm, and, optionally, an initial time-step value. Throughout we have used $atol = rtol = TOL$ (to be specified) and the common L_2-norm. For the automatic grid determination one must specify N, the number of moving space nodes, and the various regularization parameters. Recall that for Method I their values have been specified already in Section 2.2.2. For Method II we still must specify τ (see Section 2.2.3) and for Method III the parameters C_1, C_2 and d (see Section 2.2.4).

We emphasize that the choice of the regularization parameters is of importance, not only to obtain a good positioning of grid points, but also to obtain an efficient time-stepping process. This will be illustrated quite clearly in the next section, which deals with the numerical experiments. In other words, we wish to pay considerable attention to the efficiency (number of time steps, Jacobian updates and back solves) of the time-stepping process, a point which has been neglected in most of the moving-grid work on time-dependent problems.

2.4. Numerical Comparisons

We shall present results from extensive numerical testing with three example problems, viz., (I) a scalar reaction-diffusion equation that models a 'hot spot' problem from combustion theory, (II) Burgers' equation, a scalar prototype for modelling nonlinear convection-diffusion phenomena, and (III) a system of two quasi-nonlinear hyperbolic equations modelling the interaction of two waves travelling in opposite directions. It is worth emphasizing at the outset that these three problems have different solution behaviours. We recall that our main aim is to assess which of the three moving-grid methods is most suitable for retaining the acknowledged features of reliability, robustness and efficiency of the conventional MOL approach. For this reason, our first problem was chosen such that a comparison with results obtained on a non-moving grid is still feasible. This will enable us to compare the mutual efficiency of time-stepping on moving and non-moving grids, a point which has received insufficient attention in the moving-grid literature.

2.4.1. Problem I: A scalar reaction-diffusion problem from combustion theory

This problem is described in Adjerid and Flaherty [1] as a model of a single-step reaction with diffusion and reads

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

where $D = Re^{\delta}/(a\delta)$ 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 rapidly to $1 + a$. A flame front then forms and propagates towards $x = 1$ at a very high speed. The degree of difficulty of the problem is very much determined by the value of δ. Following [1], we have selected the problem parameters $a = 1$, $\delta = 20$, $R = 5$. Petzold [23] also used this problem as a test example, but with the more difficult parameter choice $a = 1$, $\delta = 30$, $R = 5$. 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. The problem has also been used as a test example in [27], whence we have copied the plotted reference solution (solid lines in the plots). We use times $t = 0.26$, 0.27, 0.28, 0.29 for output.

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 hereby is that the start of the ignition must be detected accurately and without overshoot by the local error control mechanism of the stiff solver, so that the step size can be rapidly reduced to a level small enough to simulate the ignition accurately. Small errors at this time point result in significantly larger global errors later on. Some trial and error tests have revealed that the BDF codes need a time tolerance value *TOL* of 1.E-5, using an initial step size of 1.E-5. For methods which are able to step in time with higher order formulas, such a small tolerance should cause no problems. It is certainly detrimental to a method which is forced to use a low order time-stepping formula, like Method I. For clarity we emphasize that due to the sensitivity of estimating the ignition point, the errors resulting from the time-integration are more important than the errors resulting from the spatial discretization.

Because the flame is not very thin, this problem can also be satisfactorily solved in the conventional way on a uniform, non-moving mesh consisting of, say, about 40 to 100 nodes, at least for the current choice of $\delta = 20$. The problem is of interest for moving-grid methods of the Lagrangian type, since these methods should be able to reduce significantly the number of time steps needed to complete the propagation phase. Finally, in all the experiments described below, including those done on a uniform non-moving mesh, we have used 40 moving points and in all cases the start

grid was taken to be uniform.

In the plots the solid or dashed lines represent accurate reference solutions while the marks represent the PDE approximations generated in the experiment discussed. Integration information, which serves to compare the mutual time-stepping efficiency of the three methods, 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.

Results for Method I

For the present problem Method I (version (A)) is indeed not competitive because the low order time-stepping method turns out to be too expensive. Only during the formation of the 'hot spot' can the advantage of using higher order in time formulas be really employed. At the start of the ignition and during the whole of the propagation phase, the method keeps regridding, which means that very many restarts are made with the first order implicit Euler rule, for which the local accuracy demand of TOL = 1.E-5 is simply too high. Fig. 2.1 shows the PDE solution generated at the four output times (specified above) and gives the values for STEPS, JACS and BS. Observe that the numerical front is ahead of the true one. Concerning the quality of the reference solution we note that in [27] it is claimed that the reference solution is 'exact up to plotting accuracy', except perhaps in the neighbourhood of $x = 0$ at the first output time $t = 0.26$. All experiments with the present flame problem, including those with Methods II and III, show a deviation here. It should be remarked that we counted a large number of 259 error test failures. The greater part of these occur directly after a genuine regridding, indicating that the step-size selection of the restart mechanism is not well tuned. To test this we have repeated the integration using a maximal order of one, so that then at all integration steps the implicit Euler method is used. We now counted 960 successful steps and only 28 error test failures, which is normal. The results of this experiment are also shown in Fig. 2.1. One sees that the results of the backward Euler run are less accurate, in spite of the fact that more time steps are used. This shows nicely that, during the formation of the 'hot spot', version (A) benefits from the use of the higher order formulas. It should also be realized that a large number of step rejections will considerably increase BS and, most likely, also JACS (compare the given quantities of the two experiments). No attempt has been made to repair this failure because, even without these many rejections, the method would not be competitive with Method II and III.

We recall that we have applied the method with a fixed number of nodes, whereas in [23] the number of nodes is variable. This, however, is of minor importance. Even with a variable number of nodes many regriddings are performed, which is the main shortcoming. Admittedly, when a fixed number of nodes is used the dual reconnection strategy will probably lead to somewhat more interpolations, as it is then not possible to truly delete points.

A natural question is how Method I would perform if the regridding were carried out not every time step, but every k-th time step (k to be prescribed) or at prescribed times. If the chosen time intervals are large enough, so that DASSL can enlarge the

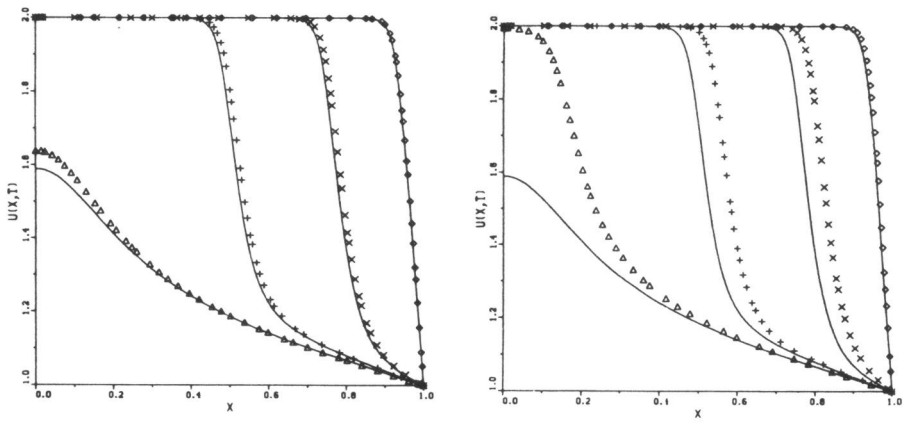


FIGURE 2.1. Results for Problem I obtained with Method I. We have used $t = 0.26, 0.27, 0.28, 0.29$ for output. The left-hand plot corresponds to the version (A) run with STEPS = 663, JACS = 709, BS = 1845, and the right-hand plot to the implicit Euler run with STEPS = 960, JACS = 663, BS = 1974.

order and use the same Jacobian, the drawback of the intermediate regriddings should then be alleviated considerably. In addition, the co-ordinate transformation governing the grid movement softens the solution behaviour in time, which in itself is beneficial for the time-stepping process. A word of warning is in order, of course. During the moving-integration process, the nodes may be sent away from the evolving front (cf. Section 2.2.1), which makes this alternative mode of operation a bit risky. Yet we do believe that this approach of intermediate regridding is much more promising and that it deserves further attention. By way of illustration, we have again solved the current flame problem with regridding at step points nearest to the prescribed times $t = 100k/0.29$ for $k = 1(1)100$. This gives a solution of comparable accuracy to that observed in the version (A) run, but with a significant reduction in computational costs. The data are STEPS = 331, JACS = 98, BS = 739 and we counted 35 error test failures. As anticipated, DASSL now also uses higher order formulas (mostly order 3) over the entire time interval.

Results for Method II

An important parameter of Method II is the grid parameter τ, which has been introduced to govern the temporal grid smoothing. Figure 2.2 shows typical results for four decreasing values of τ, of which the largest value has been chosen such that a non-moving grid results. This enables us to compare the mutual efficiency of time-stepping on a moving and a non-moving grid. We see that, as the values of τ decrease, the grid follows the flame better and better and STEPS is steadily reduced, which nicely reflects the Lagrangian nature of the method in the propagation phase. Further, and this is most important for efficiency reasons, the method keeps JACS and BS at the same low level, which is the desired MOL behaviour. Needless to say, compared to the first method Method II performs much more efficiently. This is largely due to the fact that in all runs BDF orders up to three (occasionally four and five) were used over the entire time interval. The accuracy is also much better, though it should be observed that the numerical flame front is a little too fast over the entire solution interval because the scheme is taking too large time steps. The accuracy improves notably by reducing *TOL*, but at the cost of more computational work. The experiments indicate that it suffices to work with a fairly small value for τ. In fact, for the present problem temporal grid smoothing turns out to have little effect. The choice $\tau = 1.E{-}8$ yields STEPS = 162, JACS = 41, BS = 511 without a noticeable change in accuracy.

Finally, we wish to point out that the 'non-moving, uniform grid computation' of Fig. 2.2 (the case $\tau = 1.0$) should not be interpreted as the conventional uniform grid computation, because the semi-discrete systems differ. Although this should have no influence on the PDE solutions generated, it obviously may influence the solution process through the Newton iteration.

Results for Method III

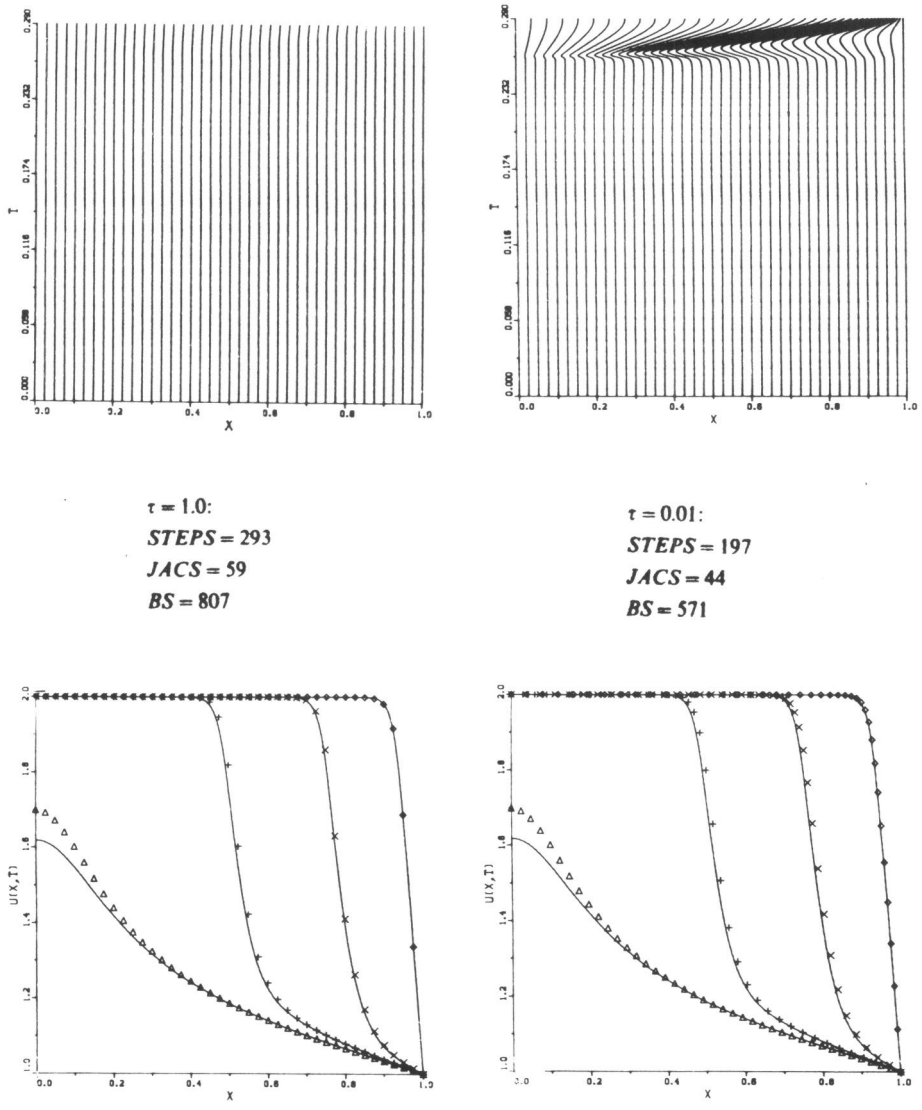
Let us inspect Fig. 2.3, which shows plots of grids and PDE approximations for four decreasing values of the regularization parameter C_1, beginning with $C_1 = 10$ (in all four cases $C_2 = d = 0$). This largest value for C_1 yields a virtually non-moving grid. The aim of this experiment, as mentioned with respect to Method II, is to illustrate the dependence of the time-stepping process on the grid movement. It is our experience that in this respect the finite-element method behaves less satisfactorily than Method II. The approximations on the uniform non-moving grid are very accurate, except perhaps within the vicinity of $x = 0$ during ignition. The attractive, conventional MOL behaviour is nicely visible. JACS is only a small fraction of STEPS and also BS is rather low. This is just why the conventional MOL approach is often so efficient. To avoid a possible misunderstanding, it should again be realized here that the 'uniform grid computation' (the case $C_1 = 10$) differs from the conventional one, in the sense that the semi-discrete systems, and thus the Jacobian matrices encountered, are different. This may have some influence on the solution process but the PDE solutions generated should be identical.

Let us now consider the remaining cases. As to be expected, we see that the grid follows the flame better and better for decreasing C_1, with the result that an even better resolution is obtained during the propagation phase. We also see that, in spite

of the fact that STEPS slightly decreases with C_1, JACS and BS steadily grow. For example, the increase in JACS and BS when C_1 changes from 10 to 0.1 is significant, while the grid movement is still rather modest and also the nodes are well separated (hence, it here suffices to put $C_2 = d = 0$). Disappointingly, for the smaller C_1 values quite a lot of computational effort must be spent in order to solve the nonlinear systems which arise. It will be clear that, in such a situation, anticipated savings in total computational effort, due to a reduction of the number of space nodes, may well be largely annihilated owing to the much larger costs for the time-stepping. It should further be observed that, in contrast to Method II, the Lagrangian nature of the moving-finite-element method during the propagation phase does not lead to a considerable decrease of time steps.

In a sense, the application of the moving-finite-element method places us in a dilemma. A near-'optimal' value for the regularization parameters would yield a near-'optimal' grid movement and an excellent approximation. On the other hand, the current experiment indicates that the grid may move at the expense of considerably higher computational costs. One might argue here that for C_1 small the points come too close, since a further decrease of C_1 would yield node overtaking. However, in all four cases illustrated, the grid points are still sufficiently separated. We conjecture that the problems associated with the iterative Newton solution of the nonlinear algebraic moving-finite-element system are probably due to some kind of ill-conditionedness which is inherent to the moving-finite-element construction. This conjecture is supported by the observation that in all four runs BDF orders of three, and occasionally four, have been used over the entire time interval, which indicates that the semi-discrete solutions cannot be very unsmooth. Moreover, the number of time steps is not markedly large. In other words, the observed difficulty of the high frequency of Jacobian evaluations is probably hidden somewhere in the nonlinear equation system itself.

Another point of concern is the choice of the regularization parameters C_1, C_2 and d. In spite of the fact that the meaning of these parameters is sufficiently clear, it is not clear at the outset how to select them. Loosely speaking, the control offered by them is in a sense not direct enough. By way of illustration, consider the two choices $C_1 = 0.025, 0.05$. For $C_1 = 0.025$ the grid is positioned rather well, which can be seen by taking a closer look at the steady state solution. Most of the points are concentrated where the curvature is largest and also the distribution within the layer is good. On the other hand, one might still argue that the ratio of adjacent points left of the front is rather large, which, as is well known, may be detrimental to spatial accuracy. Doubling C_1 yields better ratios, but then the grid is somewhat too slow, with the result that now too many points are wasted in the flat part. We admit that these observations are rather subtle and that similar observations can be made for Method II concerning the choice of the parameter τ. Still, it is our experience that fine-tuning Method III can be rather troublesome, which brings us in direct conflict with the important issues of robustness and reliability. For example, decreasing C_1 further to 0.0125 results in a totally wrong steady state solution, whereas the generated transient solution is perfectly all right (with $C_2 = d = 0$; this failure can be overcome by adjusting C_2 and d).



$\tau = 1.0$:
STEPS = 293
JACS = 59
BS = 807

$\tau = 0.01$:
STEPS = 197
JACS = 44
BS = 571

FIGURE 2.2. Solutions and trajectories for Problem I generated by Method II. The output times are as in Fig. 2.1.

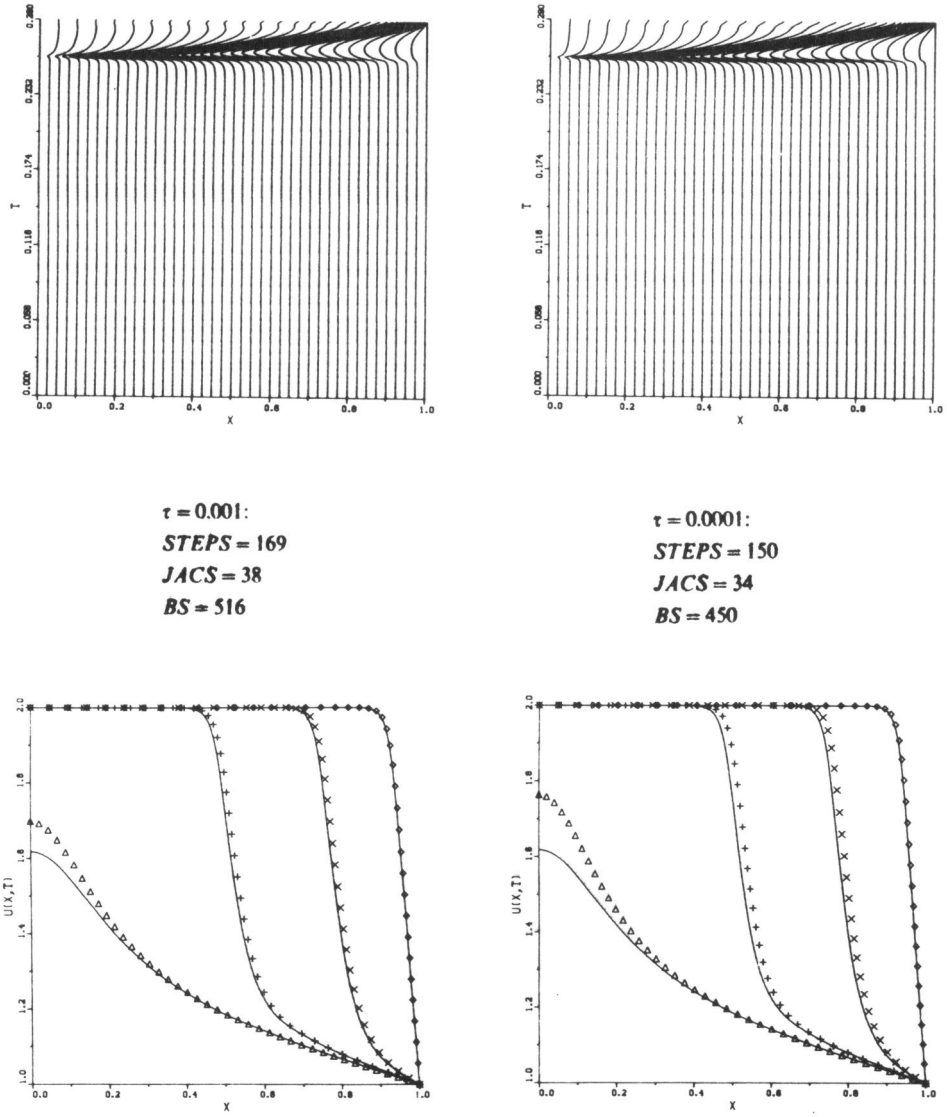


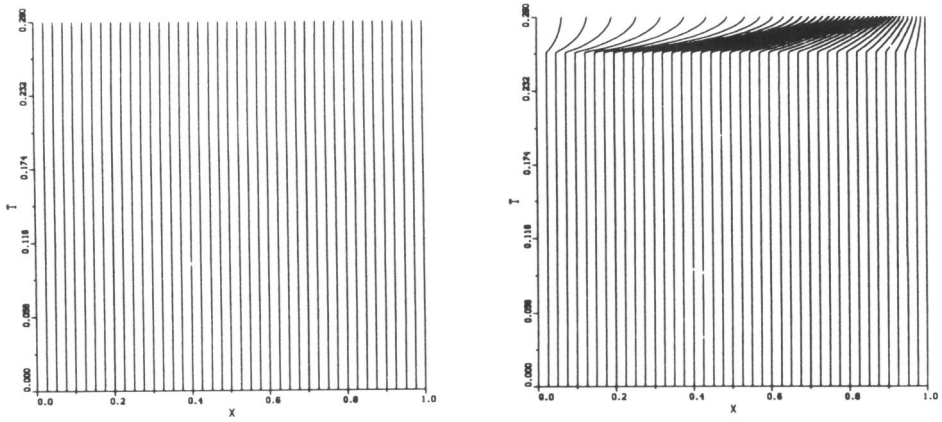
Fig. 2.2.-*Continued.*

2.4.2. Problem II: Burgers' equation

Our second example is the well known Burgers' equation

$$\partial u/\partial t = -\partial f(u)/\partial x + \varepsilon\, \partial^2 u/\partial x^2, \quad 0 < x < 1,\ t > 0,$$

$$f(u) = u^2/2,\ \varepsilon = 10^{-4},$$



$C_1 = 10.0$:
$STEPS = 230$
$JACS = 33$
$BS = 340$

$C_1 = 0.1$:
$STEPS = 220$
$JACS = 104$
$BS = 631$

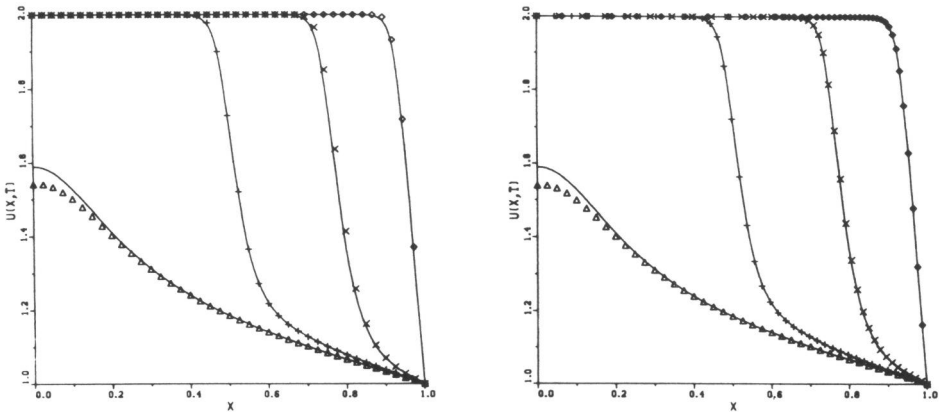


FIGURE 2.3. Solutions and trajectories for Problem I generated by Method III. The output times are as in Fig. 2.1.

supplemented with the smooth initial function $u(x, 0) = sin(2\pi x) + 0.5sin(\pi x)$ and homogeneous Dirichlet boundary conditions. This problem also served as a test example in [10,12]. The solution is a wave that first develops a very steep gradient and subsequently moves towards $x = 1$. Because of the zero boundary values, the wave amplitude diminishes with increasing time. We consider the time interval [0,2]

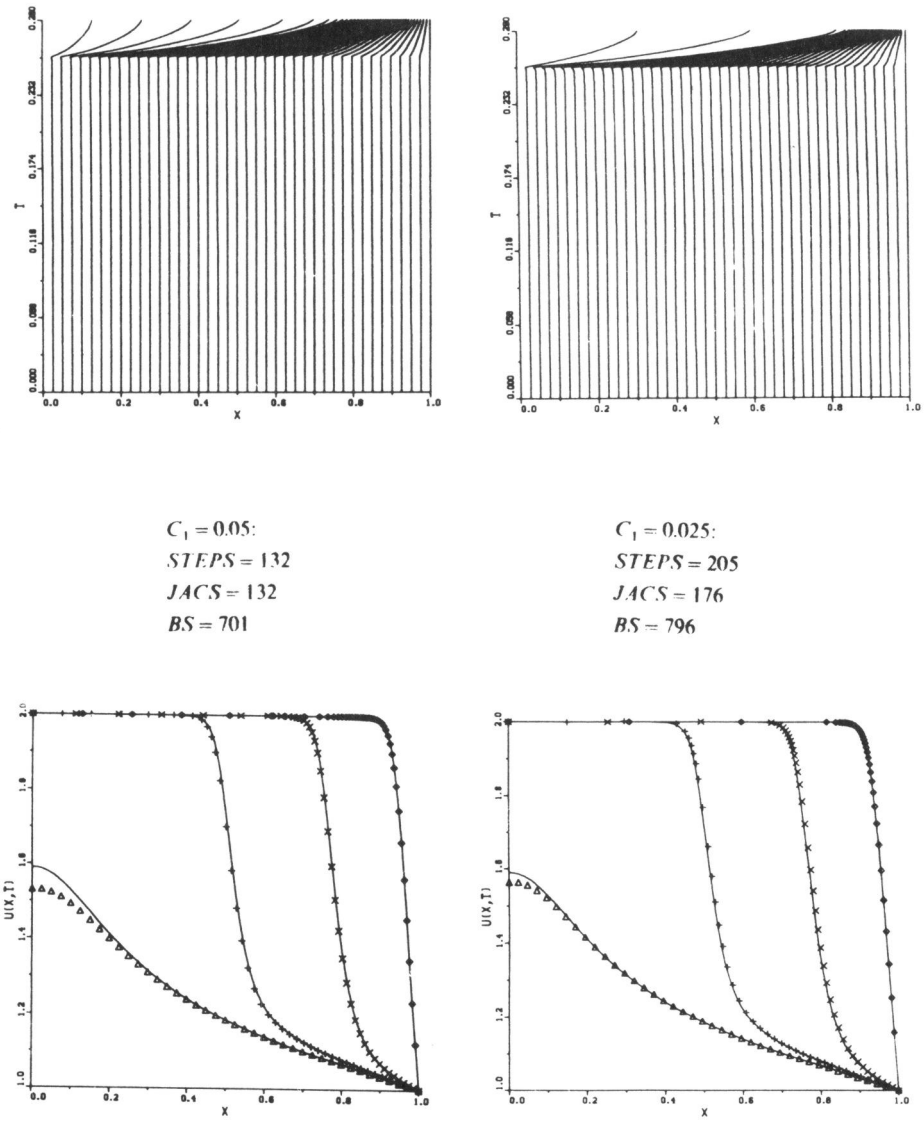


Fig. 2.3.-*Continued.*

and use times $t = 0.2$, 0.6, 1.0, 1.4, 2.0 for output.

In contrast with the previous problem, the location of the fine grid region is very critical, since all three methods are known to generate spurious oscillations readily if the grid in the layer region is too coarse, just as with standard central differences on a non-moving grid. Concomitant with this form of space instability is the danger of having non-smooth continuous-time, semi-discrete solutions. In other words, despite

the fact that we move the grid, these solutions still have a tendency to oscillate, even for small grid deviations. There is no doubt that this non-smoothness is detrimental to any ODE solver and therefore the present problem provides a difficult test for any moving-grid method. In all experiments we have worked with 40 moving nodal points and a uniform start grid.

Results for Method I

For the above Burgers' equation problem, Method I, at least the (A) version, falls dramatically behind when compared with Methods II and III. Using an initial step size of 1.E-5, we have run the method for three values of *TOL*, viz., 1.E-2, 1.E-3 and 1.E-4. In all three cases the method generates the correct spatial profile; however, the numerical wave runs much too fast, in particular for the two lower tolerances. This must be attributed to the inaccuracy of the implicit Euler scheme, which is used in almost all steps, and to the very frequent interpolations. Taking into account the computational effort needed for *TOL* = 1.E-4, a further reduction of *TOL* was not considered worthwhile. Again we must conclude that the disappointing performance is due to the regridding at virtually all steps, forcing the method to use the first order Euler formula. In passing we note that for this problem the number of step failures, which in an experiment with Problem I turned out to be uncommonly large, is here virtually negligible.

Again the question arises as to what extent the less frequent regridding approach mentioned in the discussion of results for Problem I would be more promising. By way of illustration we have rerun the method for *TOL* = 1.E-3, 1.E-4 while regridding only at step points nearest to the prescribed times $t = k/50$ for $k = 1(1)100$. As for Problem I, this gives a considerable improvement, both in accuracy and with respect to computational costs. The results are shown in Fig. 2.4. These results, while not yet competitive with those of Method II and III, do, however, indicate clearly that the approach of occasional regridding in time is to be preferred to the approach of regridding at (nearly) every step, which underlies version (A). It is likely that here is room for considerable improvement. For example, the number of time steps for $TOL = 1.E{-}4$ is about $\sqrt{10}$ times larger than for *TOL* = 1.E-3, which indicates that the (locally second order) implicit Euler method is still used very frequently. No doubt, had higher order formulas been used, better performance would have been observed.

Results for Method II

Figure 2.5 depicts the grids and solutions for Method II for τ = 1.E-1 and 1.E-3 (*TOL* = 1.E-3 and the initial time step is 1.E-5). The corresponding integration data are listed in Table 2.1, together with the results obtained for τ = 1.E-2 and 1.E-4. The (plotting) accuracy for the three smaller τ values is the same and without doubt can be called excellent. Recall that the solid lines represent a highly accurate reference solution and that the marks correspond to the numerical solutions generated in the present experiment.

As already observed in the problem description, due to the small amount of diffusion the semi-discrete solutions have a tendency to oscillate as soon as the grid

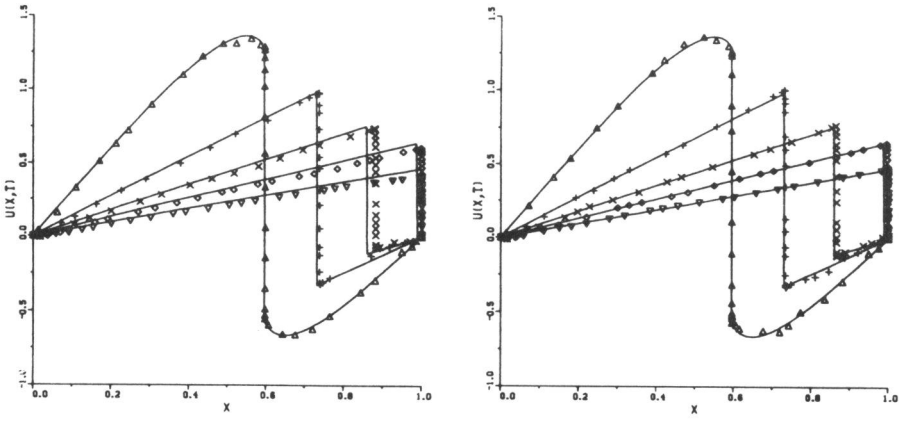


FIGURE 2.4. Results for Problem II obtained with Method I using the intermediate regridding approach. The output times are $t = 0.2$, 0.6, 1.0, 1.4, 2.0. The left-hand plot corresponds to TOL = 1.E-3 (STEPS = 561, JACS = 482, BS = 1486) and the right-hand one to TOL = 1.E-4 (STEPS = 1650, JACS = 919, BS = 3986).

becomes a little bit too coarse in the layer region. This makes the problem difficult to solve and, in fact, is the main cause for the relatively large number of Jacobian updates. It also explains the much larger effort and wiggles for $\tau = 0.1$, for which value the grid is a little bit too slow. It is obvious that, for a problem like this, the choice of τ, which dictates the grid movement, is more critical than for Problem I. On the other hand, as for Problem I, a rather small value for τ (of the order of the averaged time step used) turns out to be most appropriate. Most of the time steps used were for the shock formation and collision with $x = 1$; very few steps were needed to propagate the shock from $x = 0.6$ to $x = 0.95$. Finally, the cusps in the (τ = 1.E-3) grid near t = 1.4 are due to the change of shape in the solution when the shock reaches the right-hand boundary. The fact that these are virtually absent in the ($\tau = 1.E-1$) grid nicely illustrates that here the temporal grid smoothing is too large.

Results for Method III

In all the experiments we have used the time tolerance value TOL = 1.E-3 with initial step size 1.E-5. As a first experiment we tried the method using regularization parameter values copied from Hrymak, McRae and Westerberg [12]. Their values are: $C_1 = 0.01$, C_2 = 1.E-4 and $d = 5.E-5$. Hrymak et al. integrate Problem II only until $t = 1$ and on this time interval the integration is successful. However, upon continuing the integration to the end point $t = 2$, we experienced node crossing near approximately $t = 1.4$. Increasing C_1, for example, overcomes the crossing. For C_1

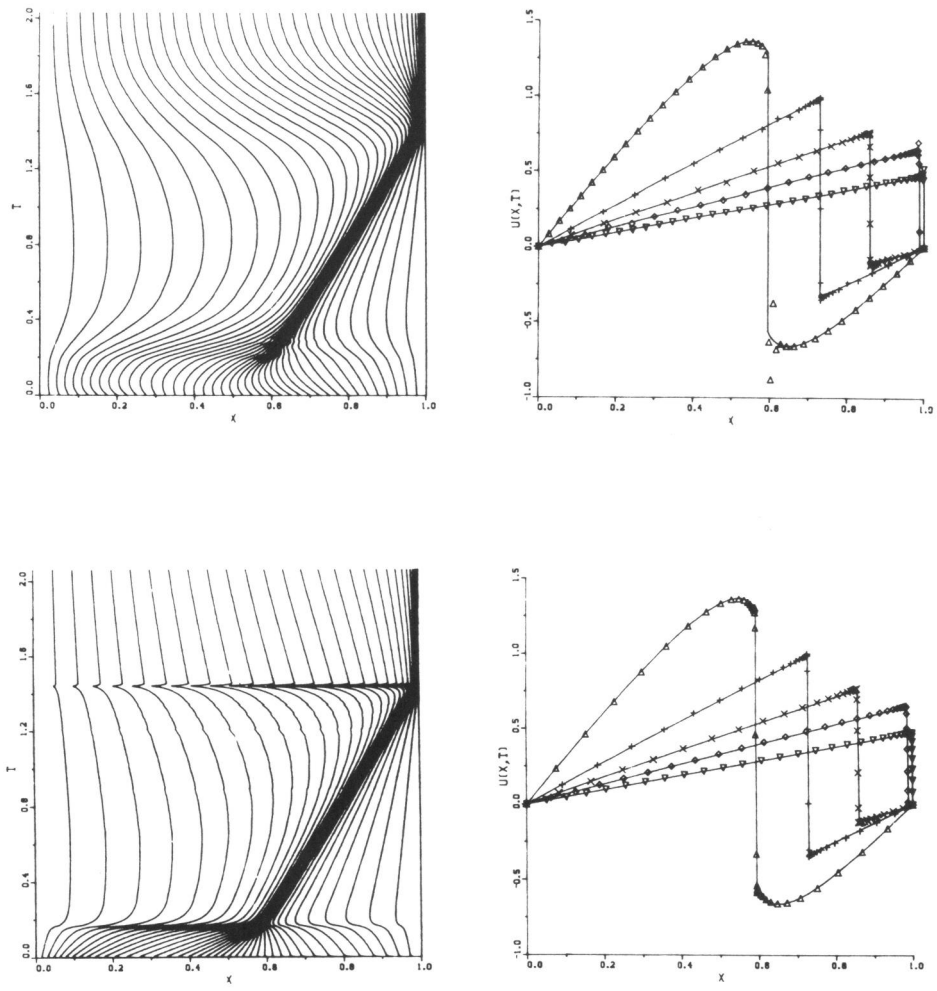


FIGURE 2.5. Results for Problem II obtained with Method II. The output times are the same as in Figure 2.4. The two upper plots correspond to $\tau = 0.1$ and the two lower ones to $\tau = 0.001$.

$= 0.025$ the integration is successful over the entire time interval $0 \leq t \leq 2$ and leads to a very accurate solution, but at rather large costs, viz., STEPS = 364, JACS = 270 and BS = 941. This, in turn, can be improved by enlarging the minimal node distance parameter d, for example, d = 1.E-4 yields an equally accurate solution. Figure

τ	STEPS	JACS	BS
1.E-1	747	428	2595
1.E-2	293	164	910
1.E-3	212	120	708
1.E-4	224	134	749

TABLE 2.1. Variation of τ.

2.6 shows this solution, obtained using $C_1 = 0.025$, $C_2 = d =$ 1.E-4, for which the costs are STEPS = 271, JACS = 190, BS = 719.

A further increase of d, to about $5.E{-}4$, is not possible since then the grid in the layer becomes too coarse and thus the familiar oscillations arise. In the present experiment these also end in node crossing. In view of the oscillations, we recall that there should be an upper limit on d and that this upper limit is related to the size of the viscosity parameter ε in Burgers' equation, since it is this parameter which determines the width of the layer region. Some further trial and error runs, with the earlier values $C_1 = 0.025$ and $C_2 =$ 1.E-4, revealed that the admissible range for d is not very large. We observed node crossings for $d =$ 1.E-5 and $d = 5.E{-}4$.

In conclusion, Method III is able to solve the present difficult Burgers' equation problem with high accuracy, but not without considerable tuning. It should also be noted that the costs of the successful, accurate computation of Fig. 2.6 are larger than those of the successful runs with Method II for the smaller τ values. We attribute this to the fact that here SPRINT starts to integrate with the first order implicit Euler scheme as soon as the wave develops the steep gradient, and hence does not exploit the higher order BDF formulas. This, in turn, indicates that, for the convection dominated problem, the continuous-time, semi-discrete solution generated by the moving-finite-element method will be rather non-smooth in time, a situation we already anticipated in the problem description.

2.4.3. Problem III: Waves travelling in opposite directions

Our third example problem is a two-component, semi-linear hyperbolic system, the solution of which is constituted by two waves travelling in opposite directions (copied from Madsen [15], see also [27]). The system is given by

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

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

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

$u(x, 0) = (1 + cos(10\pi x))/2$ for $x \in [-0.3,-0.1]$ and $u(x, 0) = 0$ otherwise,

$v(x, 0) = (1 + cos(10\pi x))/2$ for $x \in [0.1,0.3]$ and $v(x, 0) = 0$ otherwise.

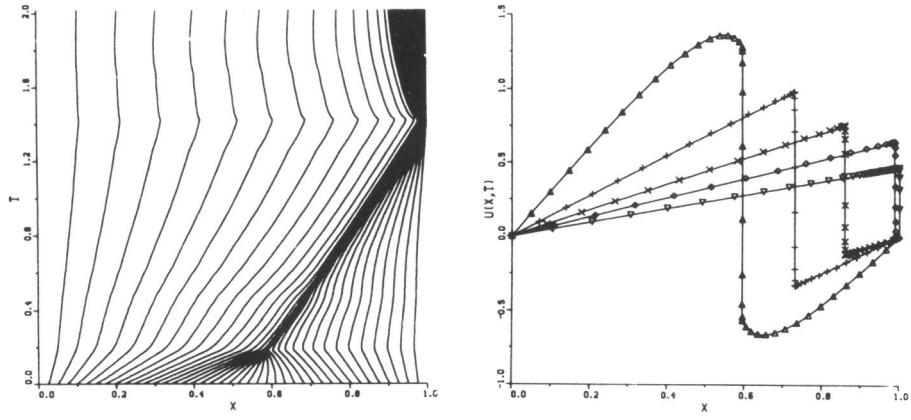


FIGURE 2.6. Solutions for Problem II computed with Method III. The output times are the same as in Figures 2.4 and 2.5. The parameter values are $C_1 = 0.025$, $C_2 = 1.E\text{-}4$, $d = 1.E\text{-}4$.

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, the nonlinear term $100uv$ vanishes, so that for $t > 0$ these waves start to move without change of shape and with speed 1, u to the right and v to the left. At $t = 0.1$ they collide at $x = 0$ and the nonlinear term becomes positive, 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 terms. At the nonlinear interaction, the pulses lose their symmetry and experience a decrease in amplitude.

To save space, in this section we restrict ourselves to presenting results for Method II and III (Method I was applied, but with rather inaccurate results). As output times we have selected the values $t = 0.1$, 0.2, 0.25, 0.3, 0.5 and in all experiments the integration has been started at $t = 0$ on a non-uniform, solution-adapted grid consisting of 41 points. For both methods we have used the time step tolerance value $TOL = 1.E\text{-}3$ and an initial step size of 1.E-5.

Results for Method II

Figure 2.7 shows the grid and the numerical approximations at the specified output times, obtained with a value of 1.E-3 for the grid delay parameter τ. We see that the solutions are fairly accurate and point out that the visible inaccuracies are only due to a somewhat optimistic choice for TOL and the number of points. These inaccuracies will vanish if more points and a smaller tolerance are used. Also the grid

positioning is good over the entire time interval, i.e., there is sufficient refinement near the travelling waves before and after the interaction. In the present experiment we have replaced the (regularization) constant 1 of the arc-length monitor

$$(1+(\partial u/\partial x)^2+(\partial v/\partial x)^2)^{1/2}$$

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 when only 41 points are used. With this number of points, it is also necessary that, after the separation, the grid refines properly in the vicinity of the waves, since otherwise spurious oscillations will 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 monitor function be made solution-dependent, in some way or another. Finally, the costs of the run are STEPS = 105, JACS = 58 and BS = 332.

Results for Method III

A typical result obtained with the parameter values $C_1 = 0.05$, C_2 = 1.E-4 and d = 1.E-5 is shown in Fig. 2.8. The costs of the run are STEPS = 71, JACS = 38 and BS = 177. We see that up to approximately $t = 0.25$ the grid moves in the right way and the two numerical waves follow the exact ones quite accurately. As for Method II, the small, visible inaccuracies are due to a somewhat optimistic choice of *TOL* and the number of points. Unfortunately, the method fails to track accurately the separation of the waves, which can be seen by inspecting the grid. Although after the separation the solution is quite accurate, except for the wiggle at the tails (see $t = 0.5$), the grid positioning is not in accordance with the location of the two waves, in contrast with the positioning for $0 \le t \le 0.25$. For $t > 0.25$ the grid tends to become more or less uniform over the greater part of the space interval and does not refine in the vicinity of the travelling waves.

It is noted that this grid deficiency does not vanish upon increasing the number of points and the temporal accuracy level, at least for 60 moving points and *TOL* = 1.E-4 (the right upper plot of Fig. 2.8 depicts the corresponding grid). Attempts to overcome it by changing the penalty parameters were not successful either; nor was the addition of a small amount of viscosity (1.E-4) to suppress spurious oscillations. The addition of a small amount of artificial viscosity, which was suggested by Keith Miller (personal communication), does reduce the oscillations in the solution, but does not have a visible impact on the grid. It is conjectured that the observed difficulty has to do with the property that, for the hyperbolic model problem we are actually solving after the separation, the moving-finite-element method moves the grid at speed one and returns the exact solution, but does not adjust the grid to the new separated pulse profile. Nodes are dragged out of the pulses as they separate because of the large, internodal viscosity coefficient C_1^2 which we found necessary to use to get the code to work. This value of C_1^2 is 100 to 600 times the standard choice of Miller [18].

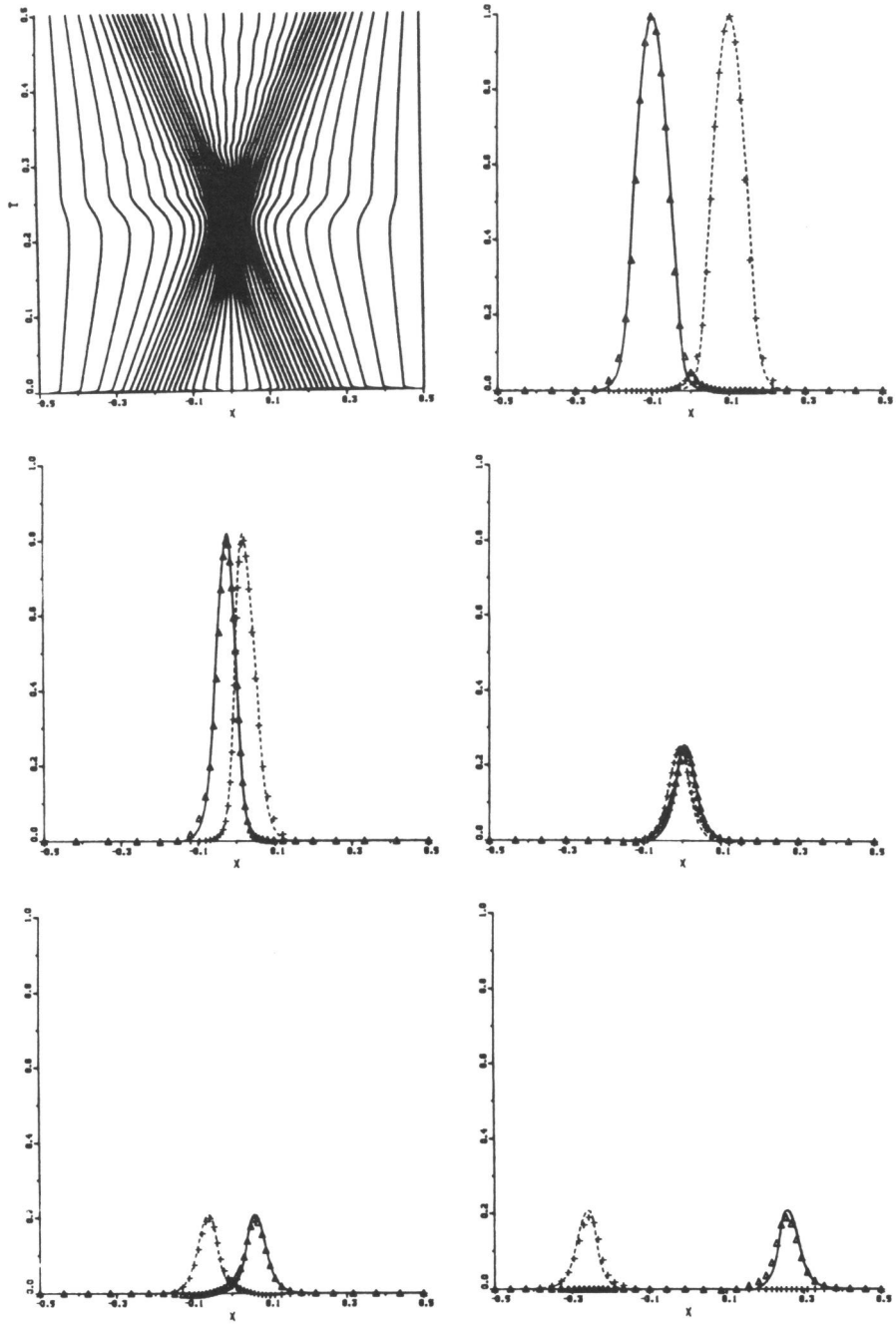


FIGURE 2.7. Grid trajectories and solutions for Problem III computed with Method II. The output times are $t = 0.1, 0.2, 0.25, 0.3, 0.5$.

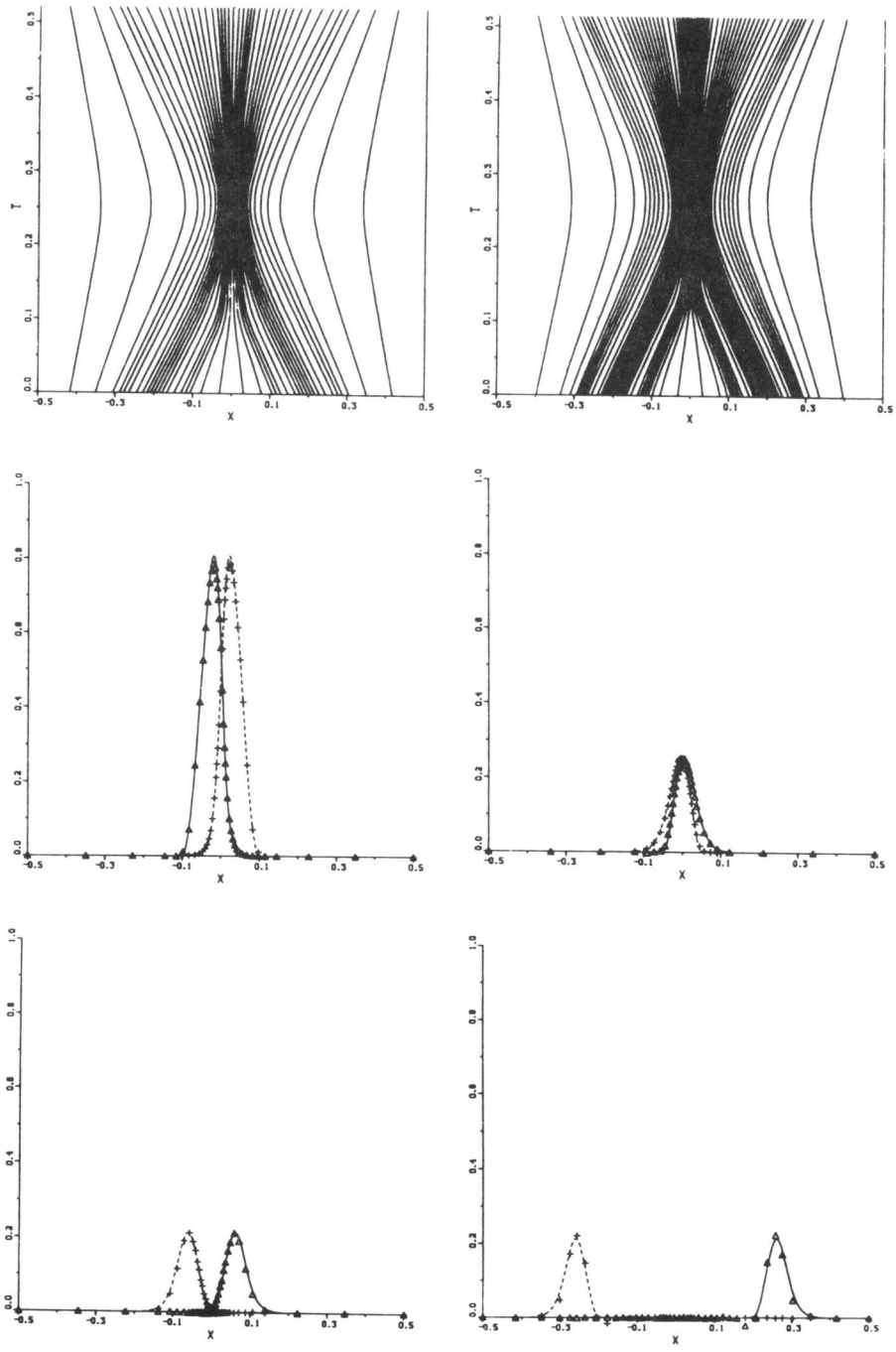


FIGURE 2.8. Grid trajectories and solutions for Problem III computed with Method III. The left-hand grid plot and the four lower plots belong to the run with 40 moving points and *TOL* = 1.E-3. Output times are the same as in Figure 2.7, except that here $t = 0.1$ has been omitted. The right-hand grid corresponds to 60 points and *TOL* = 1.E-4.

2.5. CONCLUSIONS

We have examined three Lagrangian-based moving-grid methods for systems of 1-D time-dependent partial differential equations. Our aim has been to assess which of these methods offers the best prospects for reliable, efficient and robust method-of-lines application, preferably with as little user intervention as possible. For this purpose we have carried out a numerical comparison with three different test examples. For the time integration we have used two existing, closely related stiff ODE codes, both of which are based on the acknowledged BDF formulas. We formulate the following conclusions:

(i) We cannot recommend Method I for general use, although it is quite reliable and robust; for example, we never found it necessary to use values for the parameters α and λ different from the specified default values. The very frequent regriddings mean that the method has to integrate almost always with the first order implicit Euler rule, thus preventing the Lagrangian procedure from exploiting the attractive, higher order BDF formulas. In many situations this will be detrimental to efficiency, apart from incurring the extra cost of a Jacobian update after regridding. A second drawback of regridding is the need to interpolate. In spite of the fact that accurate monotone interpolation is combined with the dual reconnection strategy, which implies that after a regridding the number of point interpolations is not very numerous, many successive interpolations can still cause a perceptible loss of spatial accuracy. In this connection it is worthwhile to note that one of the recognized advantages of Lagrangian schemes, when operating with a fixed number of moving points, is that they do not require interpolation.

Our experiments indicate that a significant improvement can be obtained when the number of regriddings is limited in some way or another (the intermediate regridding approach) because then the time-stepping can benefit more from the Lagrangian nature of the method. When considered on its own, the underlying Lagrangian transformation is of interest since the aim is to achieve smoothness in time, which is of course attractive, certainly when the higher order BDF formulas are available for the time integration.

(ii) We do not wish to conceal the fact that we have mixed feelings about the moving-finite-element approach underlying Method III, at least as far as our application is concerned. This is based on the following observations. In this approach the movement of the grid is basically governed by a minimization procedure, akin to the procedure for standard non-moving-grid Galerkin schemes. For practical application within an implicit method-of-lines procedure it is necessary, through the use of penalty terms, to regularize this minimization so as to avoid node overtaking and singular mass matrices. Inevitably, the choice of the parameters involved is problem-dependent and experience has revealed clearly that this often leads to troublesome application. Quite some tuning may be needed to make the grid move in a satisfactory way. In a sense, the effect of the regularization on the minimization does not seem to provide a sufficiently clear and unique set of rules for moving the grid. In this respect the spatial equidistribution approach which underlies Method II is more transparent.

The need for tuning is obviously in conflict with the aim of robustness. Another

point of concern we should like to bring forward here is that the time-stepping behaviour of Method III is rather sensitive with respect to the grid movement. If the grid does not move in the right way, the time-stepping can easily become rather expensive. Furthermore, even if the grid does move satisfactorily, it may still happen that the time-stepping costs are rather large compared with the costs of time-stepping in the conventional way on a non-moving grid (assuming of course that a non-moving grid is feasible); see, for example, the experiment carried out with Problem I. We admit that this comment will apply to any moving-grid procedure, including Method II. It is our experience, however, that in this respect the latter method behaves better.

(iii) We believe that, for the application we have in mind, the approach of the finite-difference Method II is to be preferred above the moving-finite-element approach of Method III. We have found Method II easier to work with and implement than Method III and also more efficient. The grid movement of Method II is directly attached to equidistribution in space of a chosen monitor function whereas that of Method III has no underlying equidistribution principle and so there is no improvement mechanism for an incorrect initial node distribution. As already indicated under (ii), it is our experience that this approach provides a better and more unique way of automatically adjusting the grid to large spatial gradients.

However, Method II may easily encounter difficulties in tracking sharp corners of a solution where, nearby, the first derivative is not very large. A simple example of such a situation is provided by the model convection equation $u_t + u_x = 0$ with a triangular pulse as initial value. Computing the moving triangular pulse solution with Method II will result in a numerical solution showing the familiar spurious oscillations. This does not happen with genuine shocklike structures because these have an arclength associated with it. Very large spatial derivatives attract enough points to prevent the oscillations to arise but the triangular pulse form does not lead to sufficient refinement near the sharp moving corners. We have experienced numerically that this sort of difficulty will also arise when solving the Burgers' equation with a trapezoidal pulse as an initial value instead of the sinusoidal one, a test example suggested by Keith Miller [19]. In this connection it should be emphasized that the MFE method does not suffer from this particular deficiency and can handle this sort of initial values in the Burgers' equation with great accuracy using relatively few points [10,20].

An important feature of the approach of Method II is the grid smoothing capability. Despite involving two method parameters, viz., κ and τ, the choice of these parameters has not proved to be troublesome. The meaning of κ is very clear and for general use κ can be taken equal to, say, 1 or 2. Admittedly, the actual choice to be made for τ is less clear. Our experience in the experiments is that it is best to keep τ small so that the grid movement is almost exclusively dictated by the spatial equidistribution at the forward time level, as long as this does not lead to oscillatory grids. However, for general use it is not recommended to set $\tau = 0$. The temporal grid smoothing property deserves some more study.

(iv) In conclusion, we consider the approach of Method II as most promising for a general method-of-lines application. In the near future we therefore plan to study

this specific approach in more detail, with the aim of extending the current (ad hoc) implementation of Method II into a reliable, efficient and robust, user-oriented piece of software which can be easily linked to existing PDE packages like SPRINT.

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.J. BAINES (1987). *Moving Finite Envelopes,* Numerical Analysis Report no. 12/87, University of Reading.
3. 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..
4. M. BERZINS and R.M. FURZELAND (1986). *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.
5. G.F. CAREY and H.T. DINH (1985). Grading Functions and Mesh Redistribution, *SIAM J. Numer. Anal.*, 22, 1028-1040.
6. N. CARLSON and K. MILLER (1988). The Gradient Weighted Moving Finite Element Method in Two Dimensions, in *Finite Elements Theory and Application*, 152-164, ed. D.L. DWOYER, M.Y. HUSSAINI AND R.G. VOIGHT, Springer Verlag.
7. J.M. COYLE, J.E. FLAHERTY, and R. LUDWIG (1986). On the Stability of Mesh Equidistribution Strategies for Time-Dependent Partial Differential Equations, *J. Comput. Phys.*, 62, 26-39.
8. E.A. DORFI and L. O'DRURY (1987). Simple Adaptive Grids for 1-D Initial Value Problems, *J. Comput. Phys.*, 69, 175-195.
9. R.M. FURZELAND (1985). *The Construction of Adaptive Space Meshes for the Discretization of Ordinary and Partial Differential Equations,* Report TNER.85.022, Thornton Research Centre, Shell Research Limited.
10. R.J. GELINAS, S.K. DOSS, and K. MILLER (1981). The Moving Finite Element Method: Application to General Equations with Multiple Large Gradients, *J. Comput. Phys.*, 40, 202-249.
11. A.C. HINDMARSH (1981). ODE Solvers for Use with the Method of Lines, in *Advances in Computer Methods for Partial Differential Equations-IV*, 312-316, ed. R. VICHNEVETSKY AND R.S. STEPLEMAN, IMACS.
12. A.N. HRYMAK, G.J. MCRAE, and A.W. WESTERBERG (1986). An Implementation of a Moving Finite Element Method, *J. Comput. Phys.*, 63, 168-190.
13. J.M. HYMAN (1982). *Adaptive Moving Meshes for Partial Differential Equations,* Report LA-UR-82-3690, Los Alamos National Laboratory.
14. J. KAUTSKY and N.K. NICHOLS (1980). Equidistributing Meshes with Constraints, *SIAM J. Sci. Stat. Comput.*, 1, 499-511.
15. N.K. MADSEN (1984). MOLAG, A Method of Lines Adaptive Grid Interface

for Nonlinear Partial Differential Equations, in *PDE Software: Modules, Interfaces and Systems*, ed. B. ENGQUIST AND T. SMEDSAAS, North Holland.

16. K. MILLER (1981). Moving Finite Elements II, *SIAM J. Numer. Anal.*, 18, 1033-1057.
17. K. MILLER (1983). Alternate Modes to Control the Nodes in the Moving Finite Element Method, in *Adaptive Computational Methods for PDEs*, 165-182, ed. I. BABUŠKA, J. CHANDRA AND J.E. FLAHERTY, SIAM, Philadelphia.
18. K. MILLER (1986). Recent Results on Finite Element Methods with Moving Nodes, in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, 325-338, ed. I. BABUŠKA, O.C. ZIENKIEWICZ, J. GAGO AND E.R. DE A. OLIVEIRA, John Wiley & Sons Ltd..
19. K. MILLER (1988-1989). *Private Communications.*
20. K. MILLER and R.N. MILLER (1981). Moving Finite Elements I, *SIAM J. Numer. Anal.*, 18, 1019-1032.
21. A.C. MUELLER and G.F. CAREY (1985). Continuously Deforming Finite Elements, *Int. J. Numer. Methods Eng.*, 21, 2099-2126.
22. V. PEREYRA and E.G. SEWELL (1975). Mesh Selection for Discrete Solution of Boundary Problems in Ordinary Differential Equations, *Numer. Math.*, 23, 261-268.
23. L.R. PETZOLD (1983). A Description of DASSL: A Differential/Algebraic System Solver, in *IMACS Trans. on Scientific Computation*, ed. R.S. STEPLEMAN.
24. L.R. PETZOLD (1987). Observations on an Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations, *Appl. Numer. Math.*, 3, 347-360.
25. R.F. SINCOVEC and N.K. MADSEN (1975). Software for Nonlinear Partial Differential Equations, *ACM Trans. Math. Software 1*, 232-260.
26. R.F. SINCOVEC and N.K. MADSEN (1975). Algorithm 494, PDEONE, Solutions of Systems of Partial Differential Equations, *ACM Trans. Math. Software 1*, 261-263.
27. J.G. VERWER, J.G. BLOM, and J.M. SANZ-SERNA (1989). An Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations, *J. Comput. Phys.*, 82, 454-486.

Chapter 3

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

"We were surprised at the simplicity of this result and the almost trivial nature of its proof"

3.1. Introduction

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

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

with the initial and boundary conditions

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

Here $\mathcal{L}$ 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 $\mathcal{L}$ 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 3.2 we introduce the semi-discretization in a moving reference frame, completely in line with the common MOL approach. In Section 3.3 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 3.4 is devoted to a discussion of the two grid-smoothing procedures that are used to regularize the grid movement. In Section 3.5 we discuss the complete semi-discrete system and its numerical integration. Section 3.6 presents results of numerical experiments with three different example problems and the final Section 3.7 is devoted to a brief conclusion.

3.2. 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

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

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

$$\dot{u} = \dot{x}u_x + u_t = \dot{X}_i\, u_x + \mathcal{L}(u,X_i(t),t), \qquad 1 \leq i \leq N, \tag{2.2}$$

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

$$\dot{U}_i = \dot{X}_i\, [(U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1})] + L_i, \quad t > 0, \quad 1 \leq i \leq N. \tag{2.3}$$

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 L_i is the finite difference replacement for $\mathcal{L}(u,x,t)$ at this point. Note that the standard, central difference approximation for u_x is used. It is supposed that L_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

$$X = [X_1, \cdots, X_N]^T, \quad U = [U_1^T, \cdots, U_N^T]^T, \quad L = [L_1^T, \cdots, L_N^T]^T,$$
$$D_i = (U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1}), \quad D = [D_1^T, \cdots, D_N^T]^T,$$

we reformulate (2.3) in the more compact form

$$\dot{U} = \dot{X} \circ D + L, \quad t > 0, \quad U(0) \text{ given}, \tag{2.4}$$

which represents the semi-discrete system to be numerically integrated in time. The notation $\dot{X} \circ D$ means that $\dot{X}_i$ 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.

3.3. THE MOVING-GRID EQUATION

3.3.1. 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

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

and the spatial equidistribution equation

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

where $M_i \ge \sqrt{\alpha} > 0$ 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.

3.3.2. 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 3.4.

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

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

$$\tilde{n}_N = n_N - \kappa(\kappa+1)(n_{N-1} - n_N),$$

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

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

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 3.6 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 3.4), thus ensuring that the adjacent point concentrations are restricted such that

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

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

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

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 3.4). 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 3.6) 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.

3.4. Discussion of the smoothing procedures

3.4.1. Preliminaries

Equations (3.8) are based on the relation

$$\tau\dot{\tilde{n}}_i + \tilde{n}_i = cM_i, \qquad t > 0, \quad 0 \le i \le N, \tag{4.1a}$$

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

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

For initial conditions we suppose a given concentration distribution $n_i(0)$, $0 \le i \le N$, 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, $0 \le i \le N$, 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 $t \ge 0$ 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

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

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.

3.4.2. 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 3.1. Let $\nu = \kappa/(\kappa+1)$. The solution of the linear system (4.3) can be represented in the form

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

where

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

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

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

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

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

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

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

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

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

where $\tilde{M}_j = M_j$ for $1 \le j \le N-1$ 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

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

The following important corollary can thus be made:

COROLLARY 3.1. 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

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

REMARK 3.1. 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. □

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

THEOREM 3.1. The spatial grid-smoothing restricts the concentrations n_i such that (3.7) is satisfied.

PROOF Consider (4.7). From the inequalities $|i-j-1| \leq |i-j|+1$ and $0 < \nu < 1$ we directly deduce

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

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. □

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

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

is a natural restriction and certainly advantageous with respect to spatial accuracy near the boundary. Further, while neglecting the boundaries, the averaged expression (4.7a) 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 3.2. 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

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

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

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

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. □

3.4.3. Temporal grid-smoothing

In terms of equidistribution, temporal grid-smoothing means that $\tau\dot{\tilde{n}}_i + \tilde{n}_i$ 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 $X_i(t)$, $t \ge 0$.

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:

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

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

$$-\tau \Delta \dot{X}_i \, (\Delta X_i)^{-2} + (\Delta X_i)^{-1} = cM_i, \qquad t > 0, \quad 0 \le i \le N, \tag{4.12}$$

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

$$dn_i/dt = -\Delta \dot{X}_i/(\Delta X_i)^2.$$

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 3.5). Let $\gamma = \tau/\Delta t$. Then the implicit Euler replacement of (4.12) is given by

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

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 3.2. 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 3.1. 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 $t \ge 0$, since this implies that all $\tilde{M}_j > 0$ (see Lemma 3.1). First we recall that $\tilde{n}_i(0) > 0$, as shown in Remark 3.1. 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 3.1 or observe that **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 3.3. 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 $\dot{X}_i$. □

3.5. THE COMPLETE SEMI-DISCRETE SYSTEM

3.5.1. The moving-grid equation in terms of nodal values

Inserting

$$n_i = (\Delta X_i)^{-1}, \quad \dot{n}_i = -\Delta \dot{X}_i/(\Delta X_i)^2 \tag{5.1}$$

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

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

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.

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

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

where $\mathcal{B}$ is the $N \times N$ 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 $\mathcal{B}(X,U)$ 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 3.4. 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 $\mathcal{B}$ tri-diagonal and symmetric positive definite. In terms of point concentrations, the resulting system reads $\tau\dot{n}_i + \tilde{n}_i = cM_i$ (cf. (4.1a)), which shows that the temporal grid-smoothing is carried out on the concentration values n_i rather than on $\tilde{n}_i$. □

3.5.2. 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,

$$\tau\mathcal{B}\dot{X} = g, \quad t > 0, \quad X(0) \text{ given}, \tag{5.5a}$$

$$\dot{U} - \dot{X} \circ D = L, \quad t > 0, \quad U(0) \text{ given}. \tag{5.5b}$$

In case of Dirichlet boundary conditions, the total number of equations and unknowns is $(NPDE+1)\times N$, 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 $\mathcal{B}$ 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 $\dot{X}$, $\dot{U}$ 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 L 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 $4\times(NPDE+1) + 1$. 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))

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

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.

3.6. 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.

3.6.1. 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

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

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

where $f(v) = 3.52\times10^6\exp(-4/v)$. The initial functions are $u(x, 0) = 1$, $v(x, 0) = 0.2$ $(0 \le x \le 1)$ and the boundary conditions are given by

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

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

$$v(1,t) = 1.2\ (t \ge 0.0002).$$

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.

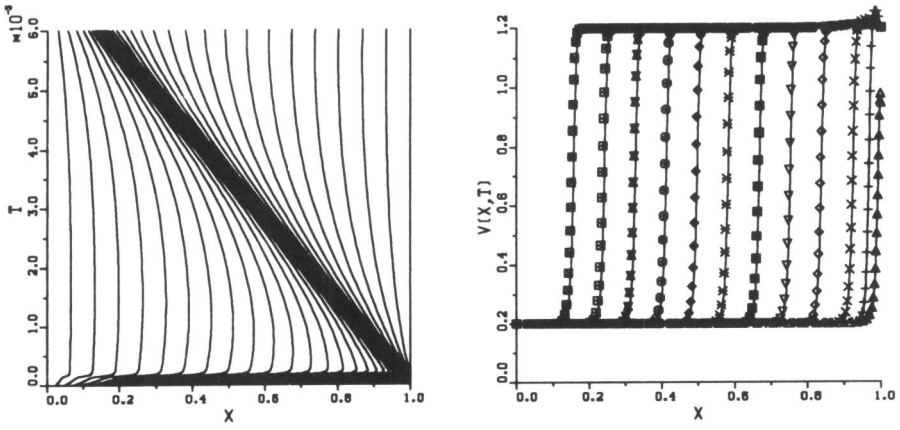


FIGURE 3.1. Problem I ($N = 40$). Grid and temperature front at times $t = .15\times10^{-3}$, $.3\times10^{-3}$, $.6\times10^{-3}$ $(.6\times10^{-3})$ $.6\times10^{-2}$.

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 $.1276\times10^{-6}$.

Fig. 3.1 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 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.

3.6.2. Problem II: A 'hot spot' problem from combustion theory

This problem is described in Adjerid and Flaherty [1] as a model of a single-step reaction with diffusion and reads

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

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

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

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 the high-order integrators.

Figure 3.2 shows a plot of the computed grid and the flame front on this grid for

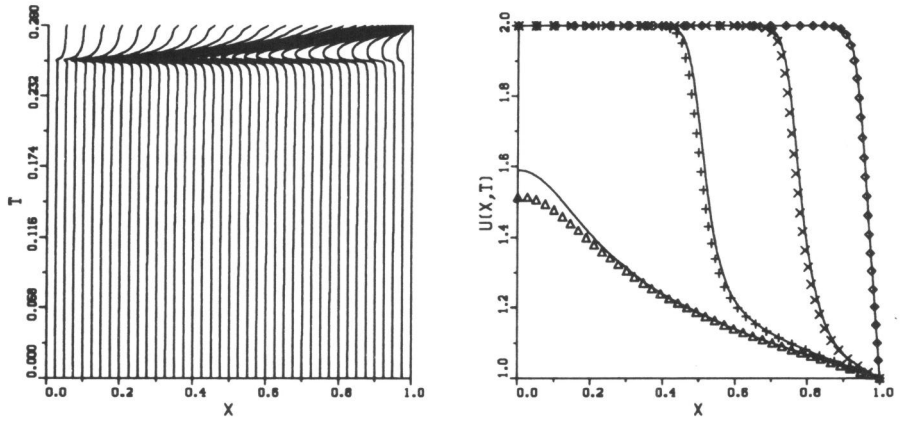


FIGURE 3.2. Problem II ($N = 40$). Grid and flame front at times $t = .26, .27, .28, .29$.

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.

3.6.3. 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

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

$v(x, 0) = 0.5(1 + cos(10\pi x))$ for $x \in [+0.1,+0.3]$ and $v(x, 0) = 0$ 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, 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 $.3\times10^{-10}$ 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. 3.3 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.

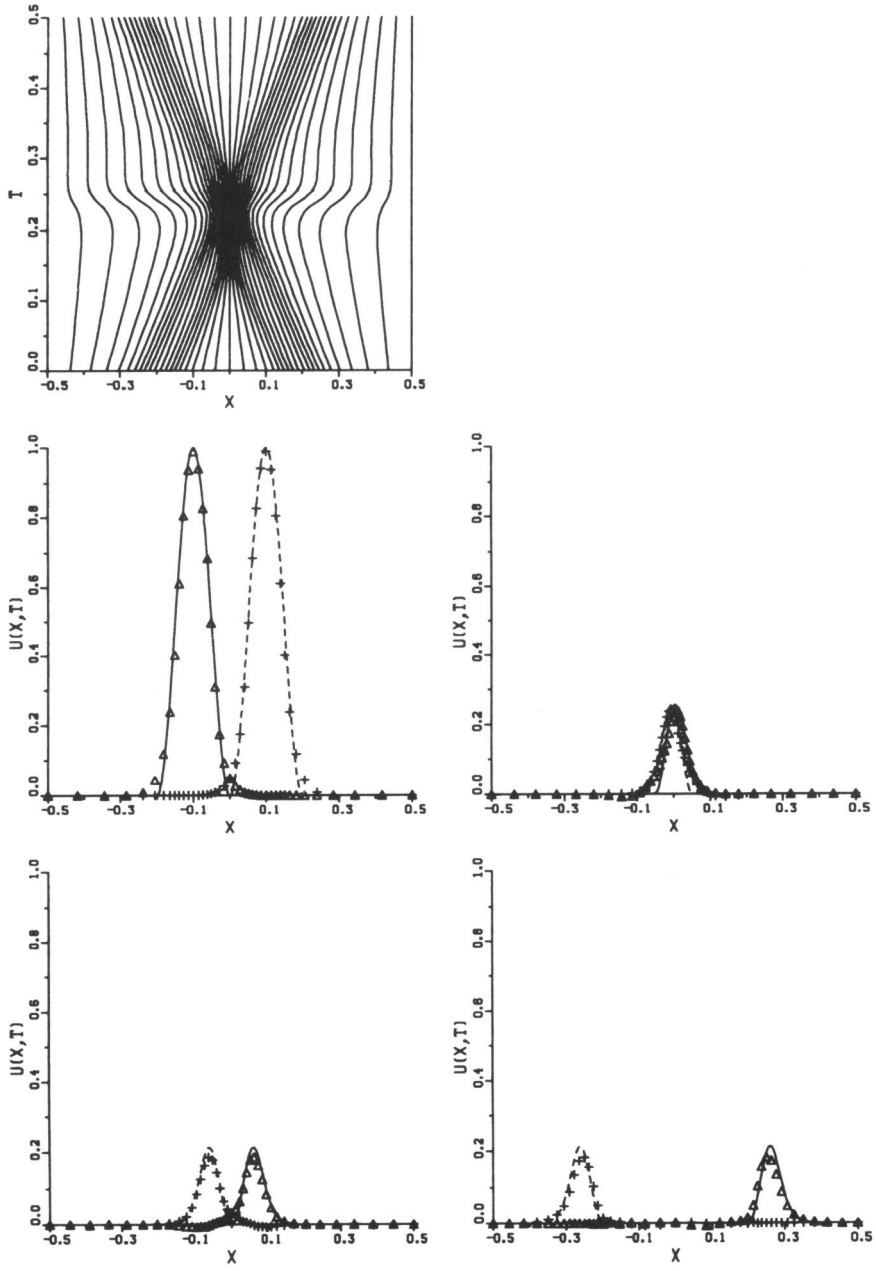


FIGURE 3.3. Problem III ($N = 40$). Grid and solution at times
$t = 0.1, 0.25, 0.3, 0.5$.

3.7. 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 (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. M. Berzins and R.M. Furzeland (1986). *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.
4. E.A. Dorfi and L. O'Drury (1987). Simple Adaptive Grids for 1-D Initial Value Problems, *J. Comput. Phys.*, 69, 175-195.
5. H.A. Dwyer and B.R. Sanders (1978). *Numerical Modeling of Unsteady Flame Propagation,* Report SAND77-8275, Sandia National Laboratories, Livermore, USA.
6. R.M. Furzeland (1985). *The Construction of Adaptive Space Meshes for the Discretization of Ordinary and Partial Differential Equations,* Report

TNER.85.022,
Thornton Research Centre, Shell Research Limited.

7. R.M. FURZELAND, J.G. VERWER, and P.A. ZEGELING (1990). A Numerical Study of Three Moving Grid Methods for One-Dimensional Partial Differential Equations which are based on the Method of Lines, *J. Comput. Phys.*, 89, 349-388.
8. A.C. HINDMARSH (1981). ODE Solvers for Use with the Method of Lines, in *Advances in Computer Methods for Partial Differential Equations-IV*, 312-316, ed. R. VICHNEVETSKY AND R.S. STEPLEMAN, IMACS.
9. J.M. HYMAN and M.J. NAUGHTON (1984). Static Rezone Methods for Tensor-product grids, in *Proc. SIAM-AMS Conference on Large Scale Computation in Fluid mechanics*, SIAM, Philadelphia, PA.
10. N.K. MADSEN (1984). MOLAG, A Method of Lines Adaptive Grid Interface for Nonlinear Partial Differential Equations, in *PDE Software: Modules, Interfaces and Systems*, ed. B. ENGQUIST AND T. SMEDSAAS, North Holland.
11. L.R. PETZOLD (1983). A Description of DASSL: A Differential/Algebraic System Solver, in *IMACS Trans. on Scientific Computation*, ed. R.S. STEPLEMAN.
12. L.R. PETZOLD (1987). Observations on an Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations, *Appl. Numer. Math.*, 3, 347-360.
13. L.R. PETZOLD and P. LÖTSTEDT (1986). Numerical Solution of Nonlinear Differential Equations with Algebraic Constraints II: Practical Implications, *SIAM J. Sci. Stat. Comput.*, 7, 720-733.
14. R.F. SINCOVEC and N.K. MADSEN (1975). Software for Nonlinear Partial Differential Equations, *ACM Trans. Math. Software 1*, 232-260.
15. R.F. SINCOVEC and N.K. MADSEN (1975). Algorithm 494, PDEONE, Solutions of Systems of Partial Differential Equations, *ACM Trans. Math. Software 1*, 261-263.
16. J.G. VERWER, J.G. BLOM, and J.M. SANZ-SERNA (1989). An Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations, *J. Comput. Phys.*, 82, 454-486.

Chapter 4

An Evaluation of the Gradient-Weighted Moving-Finite-Element Method in One Space Dimension

"Dit is geen houtje-touwtje methode!"

4.1. Introduction

Moving-grid methods are becoming increasingly popular for several kinds of parabolic and hyperbolic partial differential equations (PDEs) involving fine scale structures such as steep moving fronts, emerging steep layers, pulses, shocks, etc.. Moving-grid methods use nonuniform space grids and, like Lagrangian methods, move the grid continuously in the space-time domain while the discretization of the PDE and the grid selection are intrinsically coupled. Well-known examples are provided by the moving-finite-element (MFE) method originally proposed by Miller and Miller [16] and Miller [11], and by the moving-finite-difference (MFD) method discussed in Verwer, Blom, Furzeland and Zegeling [20] (see also references therein). The MFD method is restricted to problems in one space dimension and is strongly based on ideas due to Dorfi and Drury [6].

Because of the intrinsic coupling between the discretization of the PDE and the grid selection, the application of moving-grid methods is not without difficulties, not even in the relatively simple case of one space dimension. The main difficulty we are referring to is the threat of grid distortion which can only be avoided by using penalty terms which, to some extent, are artificial and invariably involve parameter tuning. The parameter tuning is known to be very important, not only to provide for a safe automatic grid-point selection, but also for efficiency in the time-stepping process. Another difficulty is that the automatic grid-point selection introduces nonlinear equations which may appear troublesome if handled with standard Newton solvers as commonly in use in implicit, stiff ODE solvers.

Due to these specific difficulties, the question arises as to how moving-grid methods combined with implicit, stiff ODE solvers (method-of-lines (MOL) approach) do compare with common fixed-grid MOL procedures concerning the important issues of efficiency and, in particular, robustness, reliability and ease of

use. This is a natural question because, on the one hand, fixed-grid MOL procedures are known to become more and more popular, but, on the other hand, their use is limited when steep moving transitions must be resolved, since in such situations too many points in space and time may be needed.

In a previous evaluation report, see Furzeland, Verwer and Zegeling [8], we have attempted to provide insight in this question. There we have tested three moving-grid methods for time-dependent PDE problems in one space dimension, including the MFE and the above mentioned MFD method. On account of this investigation a moving-grid interface was developed meant for automatic use in combination with the MFD method and a stiff ODE integrator (see Blom and Zegeling [5]). The interface provides the possibility of letting grid points move in time and performs the spatial discretization of the PDE problem under consideration without additional programming effort for the user, completely similar as in standard, fixed-grid interfaces like those of the SPRINT package [3, 4] and of Sincovec and Madsen [17, 18].

In [8] we have also reported rather severe difficulties in applying MFE. The current evaluation report is to a great extent devoted to the gradient-weighted MFE (GWMFE) method, again for the one-dimensional case. The gradient-weighting amounts to the use of weighting functions in the finite-element formulation that depend on the gradient u_x of the solution. This treatment results in a more robust process in that the parameter tuning becomes easier and also less critical. A second improvement, specifically concerning the implicit solution of the nonlinear system required in the time-stepping process, results from a particular block-diagonal preconditioning of the fully discretized equations (Miller [15]). One of the goals of the current examination therefore, is to find out to which extent GWMFE is a general purpose method. While most tests in the literature of (GW)MFE refer to strongly convection dominated convection-diffusion problems, in this paper we test GWMFE also on true parabolic equations.

The paper is divided into five sections. In Section 4.2 we describe the main ideas of MFE and GWMFE and the implementation of the latter. Section 4.3 contains the results of extensive numerical experiments on a set of five test models. In this test set are included Burgers' equation with a small diffusion coefficient, a scalar diffusion problem describing a shifting pulse, a system of two nonlinear convection-reaction equations, a flame-propagation model with a heat source at the boundary, and a problem from gasdynamics with a small diffusion term. Section 4.4 is devoted to a concise comparison between GWMFE and the MFD method from [5, 20]. In Section 4.5 our conclusions and recommendations are summarized.

4.2. Description of the method

In this section an outline is given of GWMFE. Miller derived the method from his own moving-finite-element method (MFE). Since many basic properties of GWMFE are related very naturally to MFE properties, we first give a description of the MFE method.

4.2.1. MFE

Consider the scalar PDE problem

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

where L represents a differential operator involving only spatial derivatives up to second order. The space interval is supposed to be fixed for all times $t > 0$ under consideration. Corresponding to the common method-of-lines approach, we consider N time-dependent grid points

$$x_L = X_0 < \cdots < X_i(t) < X_{i+1}(t) < \cdots < X_{N+1} = x_R \tag{2.2}$$

On such a grid, MFE approximates the solution $u(x,t)$ of (2.1) by

$$\begin{aligned} u \approx U &= \sum_{j=1}^{N} U_j(t)\, \alpha_j(x, \{X_i(t)\}) \\ &= \sum_{j=1}^{N} U_j(t)\, \alpha_j(x, X_{j-1}(t), X_j(t), X_{j+1}(t)), \end{aligned} \tag{2.3}$$

where α_j is the standard piecewise linear basis function which is 1 at the jth node and 0 at the other nodes. Differentiating U with respect to t and applying the chain rule gives

$$U_t = \sum_{j=1}^{N} \dot{U}_j \alpha_j + \dot{X}_j \beta_j, \tag{2.4}$$

where $\beta_j = -U_x \alpha_j$. It must be noted that β_j is piecewise linear discontinuous. The equations determining the semi-discrete unknowns U_j and X_j are now obtained in the standard Galerkin way by minimizing the L_2-norm $\|R(U)\|_2^2$ with respect to $\dot{U}_i$ and $\dot{X}_i$, where

$$R(U) := U_t - \mathcal{L}(U) \tag{2.5}$$

is the PDE residual. This minimization gives a system of $2N$ ordinary differential equations in the $2N$ unknowns U_i and X_i:

$$\sum_{j=1}^{N} < \alpha_i, \alpha_j > \dot{U}_j + < \alpha_i, \beta_j > \dot{X}_j = < \alpha_i, \mathcal{L}(U) >, \; i = 1,...,N, \tag{2.6a}$$

$$\sum_{j=1}^{N} < \beta_i, \alpha_j > \dot{U}_j + < \beta_i, \beta_j > \dot{X}_j = < \beta_i, \mathcal{L}(U) >, \; i = 1,...,N, \tag{2.6b}$$

where <,> denotes the usual L_2-innerproduct. It is clear that (2.6a) without the $\dot{X}$-innerproducts is just the standard Galerkin method applied to (2.1) using piecewise linear basis and test functions on a nonuniform grid. The time dependency of the grid is reflected in the $\dot{X}$-innerproducts in (2.6a) and the complete equation (2.6b).

Working out the innerproducts and defining the vector

$$Y := (U_1, X_1, \cdots, U_i, X_i, \cdots, U_N, X_N)^T,$$

we arrive at the semi-discrete MFE system

$$\mathcal{A}(Y)\,\dot{Y} = G(Y), \quad t > 0, \quad Y(0) \text{ given}, \tag{2.7}$$

where $\mathcal{A}(Y)$ is a block-tridiagonal matrix, the so-called mass-matrix, containing the innerproducts of the basisfunctions $\{\alpha_j\}$ and $\{\beta_j\}$, whereas the only problem-specific terms are contained in the vector $G(Y)$. Note that the boundary conditions are assumed to be incorporated in (2.7).

This ODE-system must be integrated numerically to obtain the required fully discretized solution. Before starting to integrate in time, we must ask ourselves whether (2.7) represents a well-defined system. The minimization of $\|R(U)\|_2^2$ (cf. (2.5)) has a unique solution if and only if the basis functions $\{\alpha_j\}$ and $\{\beta_j\}$ are linearly independent. This is only the case as long as $m_j \neq m_{j+1}$ at every node, where m_j is the slope of the semi-discrete approximation U on $[X_{j-1}, X_j]$. But even if the solution exists and is unique the question remains whether (2.7) is 'easily' solvable. A natural requirement for that is regularity of the mass-matrix $\mathcal{A}(Y)$ to avoid the problem of solving a DAE system of index 1 or higher. Concerning this, it can be shown that $\mathcal{A}(Y)$ is singular in exactly two situations (cf. Wathen [22]).

The first singularity is caused by the same reason as above and is called parallelism, which means that the approximation U has zero second differences at some node ($m_k = m_{k+1}$ for some $k \in \{1, \cdots, N\}$). This implies that the determinant of $\mathcal{A}$ is zero. In other words, system (2.7) becomes singular whenever a straight line can be drawn through the three neighboring points (X_{i-1}, U_{i-1}), (X_i, U_i) and (X_{i+1}, U_{i+1}). In physical terms this means that, in absence of curvature ($u_{xx} = 0$ locally), the method has no way to determine in which direction the grid points should be moved.

The second degeneracy of $\mathcal{A}$ arises whenever two nodes are coming too close together. $\mathcal{A}$ will then become very ill-conditioned and numerically singular. Hence one will need some mechanism to control the grid-point motion.

Furthermore, the nonlinear steady-state system $G(Y) = 0$ may exhibit degeneracies as well, for instance, in the case of parallelism.

To overcome these problems, Miller [12] introduces the following regularization terms (penalty functions) in the residual minimization. Instead of $\|R(U)\|_2^2$ the minimization is carried out for

$$\|R(U)\|_2^2 + \sum_{j=1}^{N+1} (\varepsilon_j \Delta\dot{X}_j - S_j)^2, \tag{2.8}$$

where

$$\varepsilon_j^2 = \frac{C_1^2}{\Delta X_j - \delta}, \quad \varepsilon_j S_j = \frac{C_2^2}{(\Delta X_j - \delta)^2}, \quad \Delta X_j := X_j - X_{j-1}, \tag{2.9}$$

with C_1, C_2 and δ small, user-chosen, constants. In particular, δ serves as a user-defined minimum node distance. The modifications involved are only made to the grid-point equations (2.6b) and the combined effect is to add

$$\varepsilon_i^2 \Delta\dot{X}_i - \varepsilon_{i+1}^2 \Delta\dot{X}_{i+1} \quad \text{and} \quad \varepsilon_i S_i - \varepsilon_{i+1} S_{i+1}$$

to the left- and right-hand side, respectively. The ε-terms serve to avoid the degeneracy caused by parallelism. It can be shown that the addition of these terms renders

the mass-matrix $\mathcal{A}$ positive definite [16], and thus regular. They represent a form of 'internodal' viscosity, since they penalize relative motion between the nodes and result in the degenerate nodes being carried along with the rest of the solution, provided the penalty is sufficiently large to take over before the mass-matrix becomes numerically singular. The ε-terms do prevent node overtaking in a dynamic way since the internodal viscosities become infinite as ΔX tends to δ; however over longer time intervals degenerate nodes (those caught in straight line segments where they are unneeded) may still slowly drift together. The S-terms, sometimes called internodal spring forces, serve to prevent this long term numerical drift. For a clarification of the effect of the internodal spring forces, we refer to Herbst et al. [9].

As for any other method, the regularization is somewhat heuristic and necessarily problem-dependent. For example, if C_1 is chosen too large, the grid movement is restricted ($C_1 \rightarrow \infty$ gives a non-moving grid) with the result that there may not be sufficient refinement in regions of large spatial activity (a typical phenomenon is then that the grid moves slower than a front region). On the other hand, if C_1 is too small, the mass-matrix $\mathcal{A}$ may become numerically singular. Also of great importance is that the minimum node distance δ be small enough in relation to the anticipated small-scale structure. However, too small values of δ and C_2 may allow numerical errors to lead to near node overtaking (or even worse), which is a source of severe numerical difficulties in the time integration, even for the most robust stiff solver. When nodes drift extremely close together, the sets of nonlinear algebraic equations to be solved at each time step are likely to become badly conditioned.

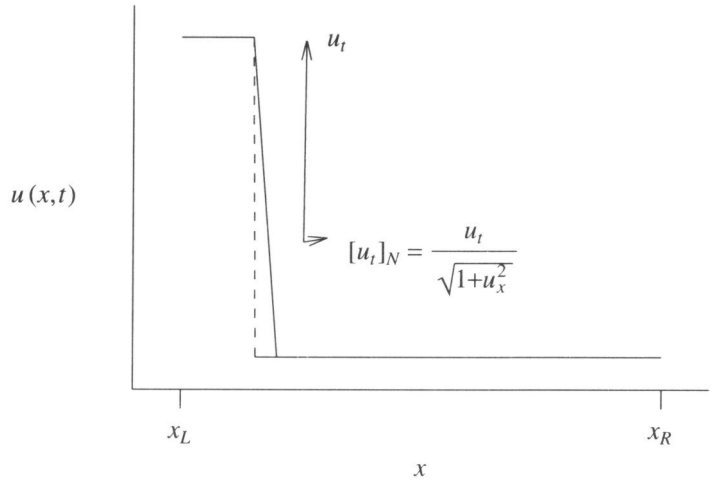
As can be seen in the numerical experiments in [8] (using a straightforward implementation without the features mentioned in Section 4.2.3), it is not possible to give a problem-independent interval for the parameters C_1, C_2 and δ, for which the MFE method solves the PDE properly, proper in the sense of reliability of the obtained solution with respect to the user-chosen penalty parameters, and time-integrational aspects, respectively. Among others, for this reason the gradient-weighted MFE method has been developed.

4.2.2. GWMFE

An important class of PDE problems may be represented by the well-known Burgers' equation

$$u_t = \varepsilon u_{xx} - u u_x \tag{2.10}$$

with a steep moving front solution $u(x,t)$ as pictured below for two given points in time.



In such a front u_t is a near delta function and in case of a true shock not an L_2-function. To use the L_2-norm in the minimization of the residual $u_t - L(u)$ is therefore for such problems not appropriate. Since the normal component of u_t, $[u_t]_N$, remains bounded even in an arbitrarily steep front, it is preferable to minimize the residual of the PDE for the normal motion of the solution. So, instead of using the L_2-norm, GWMFE uses the weighted L_2-norm

$$|||\, R(U)\, |||^2 = \int [U_t - \mathcal{L}(U)]_N^2 ds = \int (R(U))^2 w dx, \tag{2.11}$$

where the weighting function $w = w(U_x)$ is defined by

$$w(U_x) = \frac{1}{\sqrt{1+U_x^2}}. \tag{2.12}$$

Baines [2] has proved that for (2.10), with $\varepsilon = 0$, MFE is identical to the method of characteristics and therefore will gradually concentrate most of the grid points into the front. It is likely that such a grid movement will also occur for (2.10) with $0 < \varepsilon \ll 1$. Apart from the fact that points are then wasted in the steepest part of the front, this leads to numerical problems since the grid points may come very close to one another. The penalty terms introduced in (2.8) will partly remedy the situation, but this may require subtle tuning and, as already mentioned in the previous section, the practical experience with MFE is that tuning alone is not always sufficient for a good performance. The gradient-weighting, as incorporated in GWMFE, aims at de-emphasizing the steep parts of the solution and, as a positive side result, at reducing the need for tuning. The grid points will be concentrated more near the corners of the front (but still *in* the front). For scalar truly hyperbolic PDEs, however, both MFE and GWMFE will be (mathematically) equivalent to the method of characteristics, so in this case the gradient weighting will not provide a remedy.

For GWMFE the minimization of (2.11) with respect to $\dot{U}_i$ and $\dot{X}_i$ gives, as

before, a system of $2N$ ODEs in the $2N$ unknowns U_i and X_i

$$\sum_{j=1}^{N} < \alpha_i, \alpha_j w > \dot{U}_j + < \alpha_i, \beta_j w > \dot{X}_j = < \alpha_i, \mathcal{L}(U) w >, \tag{2.13a}$$

$$\sum_{j=1}^{N} < \beta_i, \alpha_j w > \dot{U}_j + < \beta_i, \beta_j w > \dot{X}_j = < \beta_i, \mathcal{L}(U) w >, \tag{2.13b}$$

$$i = 1, \ldots, N,$$

where the weighting function $w = w(U_x)$ is defined by (2.12). The only difference with (2.6) is that the inner products are replaced by weighted inner products. A nice property of w, due to the piecewise linear approximation (2.3), is the fact that it is constant on each cell. Like before, insertion of all innerproducts yields the semi-discrete GWMFE system of the form

$$\mathcal{A}_g(Y)\, \dot{Y} = G_g(Y). \tag{2.14}$$

Also in this case, the mass-matrix $\mathcal{A}_g$ may become singular. It is known that singularity occurs if we have parallelism. It is also known that in case of parallelism the steady-state system $G_g(Y) = 0$ has at least two linearly dependent equations. In order to prevent these singularities, Miller [13] has suggested to carry out the minimization for the penalized expression

$$\| R(U) \|^2 + \sum_{i=1}^{N+1} (\varepsilon_i \dot{l}_i - S_i)^2, \tag{2.15}$$

where $\varepsilon_i^2 := A^2/l_i$, $\varepsilon_i S_i := B^2/l_i^2$, with A and B user-chosen constants, and l_i is the length of the ith segment. In contrast with MFE, the modifications involved induce changes to both equations (2.13a) and (2.13b). The combined effect is that each ith segment adds a 'viscous' penalty force of magnitude $\varepsilon_i^2 \dot{l}_i = A^2 \dot{l}_i / l_i$, and a 'spring' penalty force of magnitude $\varepsilon_i S_i = B^2/l_i^2$ to the two nodes at its ends, both penalty forces working in the tangential direction. It is clear that, with these modifications, GWMFE produces equations that are even more complicated and nonlinear than the penalized MFE equations (2.6) (see also Section 4.2.3).

As for MFE, the 'segment viscosity' terms ε_i^2 serve to avoid parallelism. This means that the parameter A provides for the regularity of the mass-matrix $\mathcal{A}_g$ in the near degenerate situation of an almost flat solution. Likewise, the 'internodal spring' terms $\varepsilon_i S_i$ take over to regularize the semi-discrete system in the steady-state case $G_g = 0$ whenever parallelism occurs. In applications, it is often possible to put B equal to zero so that only the parameter A remains. A third penalty parameter, such as the δ in MFE, is not considered in the present form of ε_i or $\varepsilon_i S_i$. The direct analogue $l_i - \delta$ is redundant: it is unlikely that l_i tends to zero because this would require that both $\Delta X_i \to 0$ and $\Delta U_i \to 0$. Leaving out the penalty parameter to refrain ΔX_i from becoming zero might be defended by noting that GWMFE is supposed to send considerably less points in the steep parts of the solution.

It must be noted that we derived MFE and GWMFE for scalar PDEs. However, the foregoing can be generalized very naturally to a system of PDEs by replacing the residual (2.15) by

$$\sum_{k=1}^{NPDE} \{W_k \,|||\, U_t^k - \mathcal{L}^k(U) \,|||_k^2 + \sum_{j=1}^{N+1} (\varepsilon_{jk} \dot{l}_{jk} - S_{jk})^2\}, \tag{2.16}$$

where k denotes the k-th PDE component, *NPDE* equals the total number of PDEs, and

$$|||\, \phi \,|||_k^2 := \int_{x_L}^{x_R} \frac{\phi^2(x)}{\sqrt{1+(U_x^k)^2}} dx,$$

$$(\varepsilon_{jk})^2 := \frac{(A_k)^2}{l_{jk}}, \quad \varepsilon_{jk} S_{jk} := \frac{(B_k)^2}{(l_{jk})^2}, \quad l_{jk} := \sqrt{(\Delta U_j^k)^2 + (\Delta X_j)^2}.$$

Here W_k represents a weighting factor to emphasize, if wanted, a particular PDE component. In our tests we have taken $W_k = 1$ for all components. Likewise, A_k and B_k have been chosen to be independent of k.

Carlson and Miller use in their code GWMFE1DS a shared set of x positions for the nodes of all the approximating functions U^k. Although it is possible to use more than one grid this seems only advisable for very specific systems of PDEs, since the number of equations would be increased and it would complicate the implementation considerably.

4.2.3. Implementation

The test results with GWMFE in [13] were obtained with the GWMFE1DS code developed by N. Carlson and K. Miller. In that code a second order Diagonally Implicit Runge-Kutta method (DIRK2) has been used as time integrator for the ODE system (2.14). Miller conjectured [15] that it would be profitable to use a higher order stiff ODE solver like the SPGEAR module in SPRINT. We therefore disconnected the modules of GWMFE1DS which compute the residual and coupled them directly to SPRINT, using the stiff BDF code SPGEAR as time integrator.

In this subsection we will discuss some of the 'implementation tricks' in GWMFE1DS which we feel to contribute significantly to the performance of the code and which are not previously described in the open literature by the authors Carlson and Miller.

But firstly we would like to give the reader an idea of the complexity of the ODE system (2.14). To that intent we work out equation (2.13) + penalty terms for the scalar PDE

$$u_t = \varepsilon u_{xx} + (f(t,x,u))_x + g(t,x,u).$$

Let w_i be defined by $w_i := 1/\sqrt{1+m_i^2}$ and $\Delta U_i := U_i - U_{i-1}$ for $X_{i-1} \le X \le X_i$. Then (2.13a) plus penalties yield for $i = 1, \cdots, N$,

$$(\frac{w_i}{6}\Delta X_i - \frac{A^2}{l_i^3}(\Delta U_i)^2)\dot{U}_{i-1} + (-\frac{w_i}{6}\Delta U_i - \frac{A^2}{l_i^3}\Delta X_i \Delta U_i)\dot{X}_{i-1} +$$

$$(\frac{w_i}{3}\Delta X_i + \frac{A^2}{l_i^3}(\Delta U_i)^2 + \frac{w_{i+1}}{3}\Delta X_{i+1} + \frac{A^2}{l_{i+1}^3}(\Delta U_{i+1})^2)\dot{U}_i +$$

$$
\begin{aligned}
&(-\frac{w_i}{3}\Delta U_i + \frac{A^2}{l_i^3}\Delta X_i \Delta U_i - \frac{w_{i+1}}{3}\Delta U_{i+1} + \frac{A^2}{l_{i+1}^3}\Delta X_{i+1}\Delta U_{i+1})\dot{X}_i + \\
&(\frac{w_{i+1}}{6}\Delta X_{i+1} - \frac{A^2}{l_{i+1}^3}(\Delta U_{i+1})^2)\dot{U}_{i+1} + (-\frac{w_{i+1}}{6}\Delta U_{i+1} - \frac{A^2}{l_{i+1}^3}\Delta X_{i+1}\Delta U_{i+1})\dot{X}_{i+1} = \\
&\frac{B^2}{l_i^3}\Delta U_i - \frac{B^2}{l_{i+1}^3}\Delta U_{i+1} \\
&+ \varepsilon(-\ln(m_i + \sqrt{m_i^2+1}) + \ln(m_{i+1} + \sqrt{m_{i+1}^2+1})) \\
&+ w_i \left[f(t,X_i,U_i) - \frac{1}{\Delta X_i}\int_{X_{i-1}}^{X_i} f(t,x,U(x))dx \right] \\
&\quad + w_{i+1}\left[-f(t,X_i,U_i) + \frac{1}{\Delta X_{i+1}}\int_{X_i}^{X_{i+1}} f(t,x,U(x))dx \right] \\
&+ w_i \int_{X_{i-1}}^{X_i} \alpha_i g(t,x,U(x))dx + w_{i+1}\int_{X_i}^{X_{i+1}} \alpha_i g(t,x,U(x))dx. \qquad (2.17a)
\end{aligned}
$$

(2.13b) together with the penalties gives a similar expression as (2.17a) except for the diffusion term, which reads

$$\varepsilon(\sqrt{m_i^2+1} - \sqrt{m_{i+1}^2+1}). \qquad (2.17b)$$

It is obvious that the resulting system is extremely nonlinear.

Note, that for the gradient-weighted MFE method the evaluation of *both* the innerproducts $<\alpha_i, u_{xx}w>$ and $<\beta_i, u_{xx}w>$ has to be interpreted in the sense of 'mollification', i.e., the piecewise linear function U is smoothed at the nodal points (cf. Miller [12, 16]). The ε-terms in (2.17a) and (2.17b) are the limits obtained for the 'mollified' innerproducts if the mollification parameter tends to zero.

The implementation of the 'u_{xx}-terms' has to be done carefully because both the formulae $-\ln(m_i + \sqrt{m_i^2+1}) + \ln(m_{i+1} + \sqrt{m_{i+1}^2+1})$ and $\sqrt{m_i^2+1} - \sqrt{m_{i+1}^2+1}$ are susceptible to loss of accuracy by roundoff error if m_i and m_{i+1} are small and the first formula also if m_i or m_{i+1} is large and negative. In GWMFE1DS, $<\beta_i, u_{xx}w>$ is evaluated as

$$\sqrt{m_i^2+1} - \sqrt{m_{i+1}^2+1} = \frac{m_i^2}{1+\sqrt{m_i^2+1}} - \frac{m_{i+1}^2}{1+\sqrt{m_{i+1}^2+1}}$$

which gives automatically the correct expression even for small values of m_i. In $<\alpha_i, u_{xx}w>$, $\ln(m_i + \sqrt{m_i^2+1})$ is evaluated as

$$\operatorname{sign}(m_i)\ln(|m_i| + \sqrt{m_i^2+1})$$

to avoid the problems for large and negative m_i, and in case $\eta = m_i/\sqrt{m_i^2+1}$ is small

as a truncated Taylor series, viz.,

$$\ln(m_i + \sqrt{m_i^2+1}) = \tfrac{1}{2}\ln(\frac{1+\eta}{1-\eta}) \approx \eta + \tfrac{1}{3}\eta^3 + \tfrac{1}{5}\eta^5 + \tfrac{1}{7}\eta^7.$$

A second problem which arises if one would implement the method straightforwardly within the method-of-lines context is that a tolerance of, say, 10^{-4} for both the time error and the convergence to the solution of the nonlinear system is quite insufficient if the horizontal distance between two nodes is also of order 10^{-4}. Therefore we have, following the GWMFE1DS implementation, used as acceptance criterion for both the time error and the convergence of the Newton process that as well

$$\|v/tol\| < 1$$

should hold as

$$\max_i \frac{|v(X_{i+1}) - v(X_i)|}{\Delta X_{i+1}\rho} < 1, \tag{2.18}$$

where v is a vector either containing an estimate of the time error or the last correction in the Newton process, and ρ a user-defined parameter to indicate what weight should be given to the relative error tolerance on node distance. This implies that for $0 < \rho \leq 1$ the 'uncertainty' in ΔX_i will not be larger than ΔX_i itself.

Another feature that is implemented in GWMFE1DS and which we also adopted is the block-diagonal preconditioning of the highly nonlinear implicit BDF equations

$$\tilde{R}_g(Y) := \mathcal{A}_g(Y)\frac{Y - Z}{\Delta t\, d} - G_g(Y) = 0, \tag{2.19}$$

where $(Y - Z)/(\Delta t\, d)$ is in our case the BDF substitute for $\dot{Y}$, with Z a vector depending on information from previous time steps and d a parameter that depends on the integration formula in use.

This preconditioning is prompted by the results of Wathen [22] for the MFE mass-matrix $\mathcal{A}$ in (2.7). He proved that premultiplying A by the inverse of its block-diagonal $\mathcal{D}$ results in a matrix $\mathcal{D}^{-1}(Y)\,\mathcal{A}(Y)$ which is very well-conditioned. In fact, the condition number is even independent of the grid and the solution. Miller [14] showed that this holds also for $\mathcal{D}_g^{-1}(Y)\,\mathcal{A}_g(Y)$ (the analogue of $\mathcal{D}^{-1}\,\mathcal{A}$ in case of gradient-weighting). Although the effects of preconditioning system (2.19) with $\mathcal{D}_g^{-1}(Y)$ has not yet been analytically shown, numerical results suggest that it has a considerable influence on the condition number of the Jacobian of the nonlinear system (2.19) too. Therefore we solve not (2.19) but instead

$$\mathcal{D}_g^{-1}(Y)\,\tilde{R}_g(Y) = 0. \tag{2.19'}$$

Note that the $\mathcal{D}_g^{-1}$ in (2.19') includes also that part of the penalty functions that occurs in the left-hand side of (2.17).

4.3. NUMERICAL EXPERIMENTS

In this section we discuss test results obtained with our implementation of the GWMFE method for five example problems, viz., (I) Burgers' equation, a scalar model for nonlinear convection-diffusion phenomena; for this PDE we took two different initial solutions, (II) a linear heat conduction problem with a shifting and oscillating pulse as solution, (III) a system of two nonlinear convection-reaction equations representing two opposite traveling pulses, (IV) a flame-propagation model with a heat source at the boundary, and (V) Sod's problem from gasdynamics with a small diffusion term. With these five problems we test the performance of the GWMFE method on a wide variety of solutions having a high degree of spatial activity, ranging from steep moving wave fronts to pulses and emerging and dying layers.

In [12] Miller gives a rationale of the penalty choice, based on a remedy of the degeneracies in both $\mathcal{A}$ and the residual system (see also Section 4.2.2). This results in a 'standard choice' coupled with the time-tolerance TOL. The parameter in the viscous penalty force should be $A^2 \gtrsim \mathrm{TOL}^2$, say $\mathrm{TOL} < A < 10\,\mathrm{TOL}$. The standard choice for the B in the spring penalty force is $B^2 = 0$, unless it concerns a problem approaching steady-state with possible geometrical parallelism degeneracies. In this case the balancing of penalty contributions and true terms lead to $B^2 \approx 0.1\,\varepsilon\,\mathrm{TOL}^2$, where ε is the coefficient of the diffusion term, cf. (2.10). To get an impression of the dependency of the GWMFE method on the penalty parameters, the first Burgers' problem was tested for a large set of penalty parameter values A^2 and B^2. Moreover, for this problem the robustness of GWMFE was compared with respect to that of MFE as tested in [8]. All other problems were run with a smaller range of penalty parameter values based on the standard choice.

For all runs the 'cell-width' relative error tolerance parameter ρ from (2.18) was taken 0.1 and block-diagonal preconditioning was used in solving the nonlinear system with Newton. For a few cases we evaluated the effects of these 'implementation tricks'. The relative error tolerance on cell widths was, as can be expected, especially effective for the problems with a steep moving wave as solution; e.g., without this feature Burgers' problem often broke down at the point where the shock reaches the boundary due to node crossing. Block-diagonal preconditioning was of great benefit for the condition number of the Jacobian of the nonlinear system. Without preconditioning the condition number was frequently of the order of the inverse of the machine precision (say 10^{14} with a machine precision of $\approx 10^{-16}$). Preconditioning reduced it to $\approx 10^{7}$. The actual speedup was, in view of these numbers, not so large, but it is clear that preconditioning makes the method much more robust.

In some cases, for example in problems I, III and V, vertical rescaling of the PDE-system could result also in a better performance of GWMFE [15]. Such a vertical rescaling, say by a factor M (replacing 'u' by 'Mu' everywhere in the PDE), could allow a larger range of successful values of A^2. However, to choose the value of M, some insight into the solution behavior is needed, which makes it difficult to incorporate this parameter in an automatic code. For this reason we do not present results for rescaled PDE-systems.

In [23] we have given a catalogue of worked-out innerproducts. The integrals

resulting from the innerproducts $< \alpha_i, \mathcal{L}(U)w >$ and $< \beta_i, \mathcal{L}(U)w >$ were evaluated exactly unless indicated otherwise. If numerical quadrature was used, Boole's rule was applied (closed Newton-Cotes with error $O(h^7)$). We have also tried Simpson's rule ($O(h^5)$) but this gave, for problem IV, far worse results. This difficulty with numerical quadrature on certain types of problems has already been mentioned by Miller [11] in his extensive testing of the MFE method.

The results will be presented in tables and for a few parameter choices in plots wherein marks will indicate the GWMFE approximation and the solid line the exact solution. If no exact solution was available, we used a very accurate numerical reference solution.

In the description of the experiments the following notation has been used:

Δt_0	initial step size,
TOL	time-tolerance value (absolute and relative) for the SPGEAR integrator,
NPTS	number of grid points,
STEPS	number of successful time steps,
JACS	number of Jacobian evaluations,
CTF	number of correction time failures, i.e., no convergence of the Newton process after 3 iterations with a new Jacobian, or node crossing,
ETF	number of times the ODE integrator rejected a step,
CPU	normalized CPU-time, i.e., CPU := CPU-secs/CPU-secs$_{\min}$, where CPU-secs$_{\min}$ is the minimum number of CPU seconds used for the problem under consideration,
ORD	average order used by the time integrator measured over the whole time range.

Finally, we give marks for the quality of the computed solution (compared to either the exact solution or (in plots) to the numerical reference solution) and the quality of the grid (distribution and the smoothness of the motion in time): ++ (very good), + (good), □ (reasonable), − (bad), and −− (very bad). × indicates that GWMFE broke down during the run.

4.3.1. Problem I: Burgers' equation

This model, which can be considered as the simplest, non-trivial 1-D analogue to the Navier-Stokes equations, possesses a nonlinear convection term combined with a very small diffusion term,

$$u_t = \varepsilon u_{xx} - u u_x, \qquad 0 < x < 1, \quad t > 0, \qquad 0 < \varepsilon \ll 1. \tag{3.1}$$

We make a distinction between two specific problems (both stemming from Miller [15]):

a) the initial condition is the smooth function

$$u\,|_{t=0} = \sin(2\pi x) + 0.5\sin(\pi x), \qquad 0 \le x \le 1,$$

accompanied by homogeneous Dirichlet boundary conditions.

In this case the solution is a wave that first develops a very steep gradient, with a shock width proportional to ε, and subsequently moves towards the right boundary $x = 1$. It then collides with the boundary and forms a very thin boundary layer. This

collision is a difficult part of the computation. Next, for increasing time t the amplitude u decreases due to the Dirichlet boundary conditions. Finally, for $t \to \infty$ the solution dies out towards the steady-state solution $u = 0$. While the choice $\varepsilon = 10^{-3}$ yields a problem having all properties for testing a moving-grid method, we take the even smaller value $\varepsilon = 10^{-4}$ as a more severe test case. The problem is solved on the time interval [0,2]. (See also [8].)

b) the initial condition is the trapezoid

$$u|_{t=0} = \begin{cases} 0.2 & 0 \le x \le 0.1 \\ 8x - 0.6 & 0.1 \le x \le 0.2 \\ 1 & 0.2 \le x \le 0.5 \\ -10x + 6 & 0.5 \le x \le 0.6 \\ 0 & 0.6 \le x \le 1 \end{cases},$$

with the boundary conditions

$$u|_{x=0} = 0.2, \quad u|_{x=1} = 0, \quad t > 0.$$

For this case the course of the amplitude u is roughly the same as for case a), with the understanding that the solution now possesses several sharp features unlike the sinusoidal pulse which is very smooth outside the shock region. Again we consider the case $\varepsilon = 10^{-4}$ and the time interval [0,2].

Numerical results for Problem Ia

Starting on a uniform grid with the number of grid points NPTS = 21, and as time-integration parameters TOL = 10^{-3} and $\Delta t_0 = 10^{-5}$, we obtain a series of test results by choosing $B^2 = 10^{-8}$, 10^{-11} (standard choice), resp., 0 and by letting A^2 increase from 10^{-9} to 10^{-3}. The results are given in Table 4.1.

It can be seen that except for the largest value of A^2 the results are very satisfying. For $A^2 = 10^{-3}$ the speed of the shock was much too slow. There was not much difference between the grids and the solutions for the other values of A^2, but for extreme values of A^2 the ODE system and the resulting nonlinear system sometimes were harder to solve, which made the computation more expensive. For $B^2 = 0$ and to some extent also for the standard choice $B^2 = 10^{-11}$ the grid points were concentrated in the shock and no grid points were lying in the curvature. This makes the behavior of GWMFE more precarious. Choosing a spring penalty value of $B^2 = 10^{-8}$, which is too large from the view of a reasonable balance of penalty contributions and terms of the system (2.14), results in a case like this in a very efficient performance, especially with the standard choice for A^2. This efficiency is likely to be caused by the fact that the grid points are pushed out of the front into the curvature by the large spring forces (see also Fig. 4.2). The computation broke down only twice, for $B^2 = 0$, both times because of (near) node crossing. This robustness is strikingly compared with the MFE method as tested in [8], as can be seen from the plots in Fig. 4.1 where the acceptable range of penalty parameter values is graphically represented (by a shaded area) for both the GWMFE and the MFE method. One should remember, however, that in the MFE implementation neither some form

A^2	B^2	$t = 1.4$ STEPS	$t = 1.4$ JACS	$t = 2.0$ STEPS	$t = 2.0$ JACS	CTF	ETF	CPU	ORD	qual. sol.	qual. grid
1E-9	1E-8	278	205	429	326	91	16	2.3	1.27	++	+
1E-8	1E-8	274	183	396	280	79	27	2.1	1.18	++	+
1E-7	1E-8	197	144	266	197	56	13	1.4	1.37	++	+
1E-6	1E-8	174	140	232	186	56	7	1.3	1.41	++	++
1E-5	1E-8	140	105	191	146	49	2	1.0	1.40	++	++
1E-4	1E-8	135	111	179	147	55	1	1.0	1.40	++	++
1E-3	1E-8	162	138	201	171	62	3	1.2	1.37	–	+
1E-9	1E-11	303	214	377	261	80	18	1.9	1.28	++	+
1E-8	1E-11	276	200	356	250	70	23	1.9	1.24	++	+
1E-7	1E-11	243	184	300	229	69	16	1.6	1.31	++	+
1E-6	1E-11	255	227	315	279	98	5	1.9	1.38	++	++
1E-5	1E-11	266	253	334	315	106	11	2.1	1.35	++	++
1E-4	1E-11	150	126	400	386	146	0	2.6	1.38	++	++
1E-3	1E-11	163	135	206	175	64	3	1.2	1.42	–	+
1E-9	0	305	224	424	306	84	31	2.2	1.26	++	+
1E-8	0	275	195	351	253	70	16	1.8	1.23	++	+
1E-7	0	222	161	292	215	70	11	1.5	1.41	++	+
1E-6	0	225	196	291	255	87	8	1.7	1.36	++	++
1E-5	0	206	189	×		301	6		1.34	++	++
1E-4	0	156	136	×		266	1		1.36	++	++
1E-3	0	170	138	208	167	63	3	1.2	1.45	–	+

TABLE 4.1. Problem Ia. Integration history.

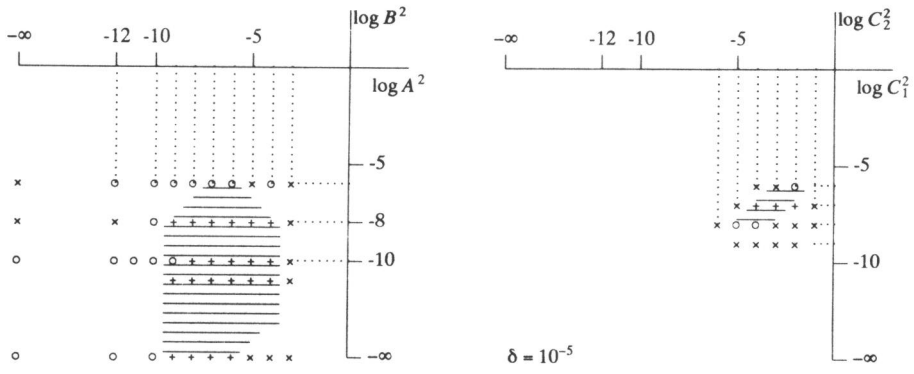


FIGURE 4.1. Problem Ia. Penalty parameter dependence for GWMFE (left) and MFE (right). Results at $t = 1.4$ are:
+: good, ○: dubious, ×: unacceptable.

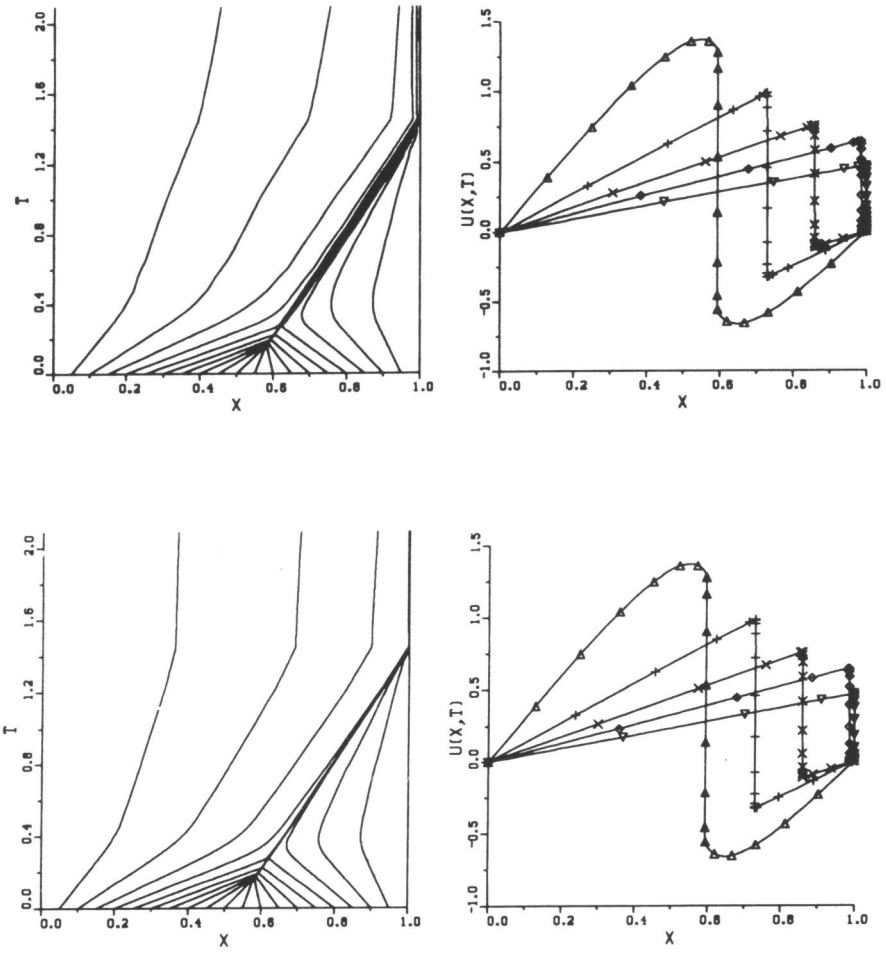


FIGURE 4.2. Problem Ia. Grid and solution at times $t = 0.2, 0.6, 1.0, 1.4, 2.0$ (Δ,+,$\times$,$\diamond$,∇) for $A^2 =$ 1E-5 and $B^2 =$ 1E-8 (above), 1E-11 below.

of preconditioning nor relative error tolerance on cell widths was available. We therefore incorporate only the results up to time $t = 1.4$ in these plots.

In Fig. 4.2 we give plots of the typical grid behavior and solution. One can see that in both cases illustrated the solution is accurate up to plot resolution.

The time-integration process is not really satisfying for this problem. The number of Jacobians almost equals the number of (successful) steps. Even if we take into

account the number of step failures (ranging from $\approx 50 - 100$) the number is still quite large. Also the observed average order turns out to be rather low. In fact SPGEAR almost never uses a third order method, not even in the time intervals where the problem is smooth and no step rejection or convergence failure occurs ($t \in [0.4, 1.2]$ and $t \in [1.5, 2.0]$). The fact that the order is not increased in these regions is somewhat amazing since plots of the $X_i(t)$ and the $U_i(t)$ show that both are reasonably smooth curves. However, in these areas the step size is drastically increased (only 10% of the steps is used in the smooth parts) and it could be that this is more efficient than an increase of the order for this coarse time tolerance. Most of the computational work is done where the shock is formed (at $t \approx 0.2$) and when the shock reaches the boundary (at $t \approx 1.3$). In these regions no high order method will be used because of the continual (true or near) node crossings within the iterative Newton process which result in convergence problems.

		$t = 0.9$		$t = 2.0$						qual.	qual.
A^2	B^2	STEPS	JACS	STEPS	JACS	CTF	ETF	CPU	ORD	sol.	grid
1E-6	1E-8	200	168	248	203	64	8	1.1	1.44	++	+
1E-5	1E-8	178	160	223	191	68	1	1.0	1.38	++	++
1E-4	1E-8	189	161	228	189	65	7	1.0	1.36	+	++
1E-6	1E-11	177	144	227	184	70	3	1.0	1.39	++	+
1E-5	1E-11	191	176	276	247	91	3	1.3	1.37	++	++
1E-4	1E-11	206	179	403	373	138	9	2.0	1.37	+	++

TABLE 4.2. Problem Ib. Integration history.

Numerical results for Problem Ib

For this problem, which is of the same nature as the above, we used only a small range of penalty parameter values, viz., $A^2 = 10^{-6}$, 10^{-5}, 10^{-4} and $B^2 = 10^{-8}$, 10^{-11}. The integration parameters were chosen the same, i.e., NPTS = 21, TOL = 10^{-3}, and $\Delta t_0 = 10^{-5}$. The results are given in Table 4.2 and Fig. 4.3. The performance is comparable with that of problem Ia. We have also run this problem with $B^2 = 0$ and the same A^2-values. Again the results are comparable *if* the method does not break down, but it seems advisable to take B^2 slightly larger than zero for this problem to handle the degeneracies in the near steady-state situation. Although Miller's standard choice results in $B^2 = 10^{-11}$, the larger value of $B^2 = 10^{-8}$ seems both for this and for the previous problem to lead to more efficiency.

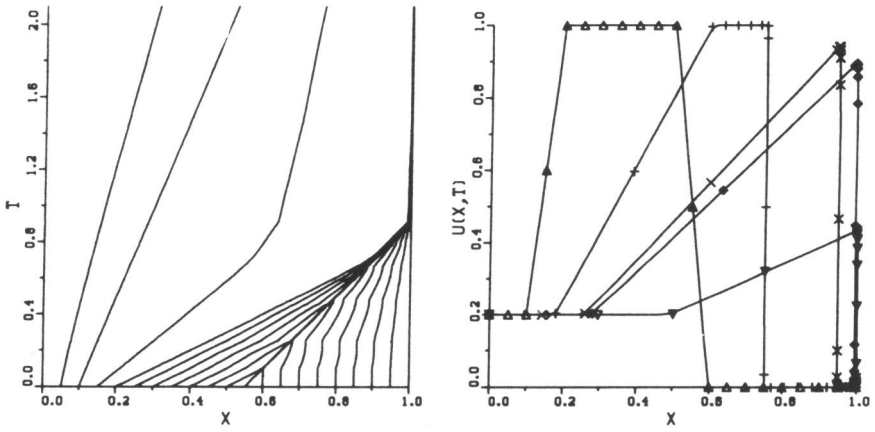


FIGURE 4.3. Problem Ib. Grid and solution at times $t = 0.0, 0.4, 0.8, 0.9, 2.0$ (Δ,+,$\times$,$\diamond$,∇) for $A^2 = 1\text{E-}5$ and $B^2 = 1\text{E-}11$.

4.3.2. Problem II: A shifting pulse

The ideas for this problem stem from Adjerid and Flaherty [1], who constructed a model (in 2-D) of a rotating cone using an exact solution. The PDE reads as follows:

$$u_t = u_{xx} + f(x,t), \quad 0 < x < 1, \quad t > 0, \tag{3.2}$$

where f is chosen in such a way that

$$u_{\text{exact}} := e^{-\alpha(x - r_\beta(t))^2}(1 - \sin(\gamma\pi t)), \qquad r_\beta(t) := \tfrac{1}{4}(2 + \sin(\beta\pi t))$$

satisfies (3.2). The boundary conditions at $x = 0$ and $x = 1$, being of Dirichlet type, and the initial condition, being a Gaussian pulse, are derived from the exact solution u_{exact}. The three parameters $\alpha > 0$, $\beta > 0$ and $\gamma \geq 0$ each have their own meaning in the model. Choosing $\gamma > 0$ means that the pulse will decrease and rise again with a period of $2/\gamma$. The steepness of the solution is controlled by the parameter α in the exponential function and β represents the speed of the pulse which moves periodically from the left to the right boundary and back again in a period of $2/\beta$. We have chosen the values $\alpha = 320$, $\beta = 1$ and $\gamma = 2$. The PDE is integrated over one period, i.e., until $t = 2.0$.

The integrals stemming from $< \alpha_i, fw >$ and $< \beta_i, fw >$ were evaluated by numerical quadrature using Boole's rule.

Numerical results for Problem II

For this problem we start on a nonuniform grid with NPTS = 41 and all but the two boundary points concentrated around the pulse, uniformly distributed between

A^2	B^2	$t = 2.0$ STEPS	JACS	CTF	ETF	$\|err\|_\infty$	CPU	ORD	qual. sol.	qual. grid
1E-6	0	310	215	52	9	5.1E-2	1.0	1.33	□	+
1E-5	0	608	504	173	0	4.2E-2	2.2	1.34	□	++
1E-4	0	988	912	348	2	5.3E-2	3.8	1.25	□	++

TABLE 4.3. Problem II. Integration history.

0.35 and 0.65. If one starts with a uniform grid the results are slightly worse. The time-integration parameters were again TOL = 10^{-3} and $\Delta t_0 = 10^{-5}$. Since there is no steady-state involved in this problem we use the standard choice for the spring force penalty, $B^2 = 0$ and for A^2 the range 10^{-6}, 10^{-5}, 10^{-4}.

The performance of the GWMFE method for this problem is significantly less satisfying than for the convection dominated Burgers' problem of the previous section. The oscillating character of the solution makes that GWMFE loses track of the movement of the pulse when the amplitude goes to zero and picks it up again only if the pulse is already at some height, thereby losing accuracy. The grid plots show that after the solution has become zero (at $t = 0.25$ and 1.25) the grid points do not return fast enough to their position around the pulses to get a correct approximation of the right-hand side of the PDE. The fact that GWMFE does not adjust itself fast enough to an emerging pulse can also be shown by starting the problem at $t_0 = 0.25$ and on a uniform grid (since $u \equiv 0$) (cf. Fig. 4.4).

The efficiency of the GWMFE method is, for this problem, strongly dependent on the penalty parameter choice; for approximately the same accuracy the amount of work varies rather capriciously with a factor 3 to 4 for different choices of A^2 (cf. Table 4.3).

The time integrator reacts on this problem in a similar way as on the previous one. Again we see that for all parameter choices the number of Jacobian updates is large relative to the number of time steps even if we add the number of rejected steps. Also the order behavior is more or less the same. In regions which are supposed to be easy for GWMFE, i.e., a moving pulse which is significantly larger than zero, SPGEAR rather increases the step size than the order.

4.3.3. Problem III: Pulses traveling in opposite directions

Our third example problem is a two-component, semi-linear hyperbolic system, the solution of which is given by two pulses traveling in opposite directions (copied from [10], see also [8, 20, 21]). The system is given by

$$\begin{aligned} u_t &= -u_x - 100uv \\ v_t &= v_x - 100uv \end{aligned} \qquad -0.5 < x < 0.5, \quad t > 0, \tag{3.3}$$

and the solution is subjected to homogeneous Dirichlet boundary conditions and the

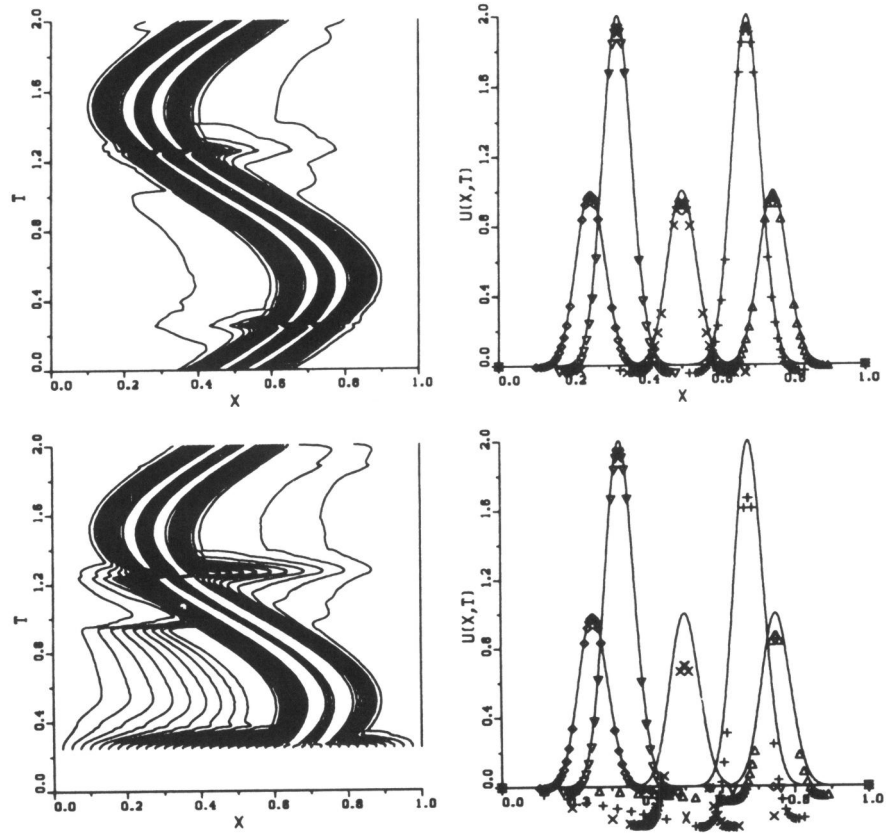


FIGURE 4.4. Problem II. Grid and solution at $t = 0.5, 0.75, 1.0, 1.5, 1.75$ (Δ,+,$\times$,$\diamond$,∇) for $A^2 = 1\text{E-}6$ and $B^2 = 0$ starting at $t_0 = 0.0$ (above) and $t_0 = 0.25$ (below).

initial condition

$$u\,|_{t=0} = \begin{cases} 0.5\,(1 + \cos(10\pi x)), & -0.3 \le x \le 0.1 \\ 0, & \text{elsewhere} \end{cases},$$

$$v\,|_{t=0} = \begin{cases} 0.5\,(1 + \cos(10\pi x)), & 0.1 \le x \le 0.3 \\ 0, & \text{elsewhere} \end{cases}.$$

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 pulses 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 pulses. Specifically, the crests of the pulses collide a little beyond $t = 0.25$ and they have separated again at $t \approx 0.3$, so that from this time on the solution behavior is again dictated by the linear advection terms. At the nonlinear interaction, the pulses lose their symmetry and experience a decrease in amplitude.

		$t = 0.5$						qual.	qual.
A^2	B^2	STEPS	JACS	CTF	ETF	CPU	ORD	sol.	grid
1E-6	0	210	166	38	14	1.0	1.30	+	□
1E-5	0	267	221	60	10	1.3	1.39	+	+
1E-4	0	345	297	98	12	1.8	1.35	—	–

TABLE 4.4. Problem III. Integration history.

Numerical results for Problem III

In contrast with our experience with the MFE method, GWMFE is not able to solve this problem without addition of (artificial) diffusion. Therefore, we added to both equations a diffusion term εu_{xx}, resp. εv_{xx}. The tests as described below are done with $\varepsilon = 10^{-4}$; we also have tried $\varepsilon = 10^{-5}$ but then GWMFE broke down.

Again we start on a nonuniform grid with NPTS = 41 and all but the two boundary points concentrated uniformly around the pulses. In this case too a uniform initial grid led to slightly worse results. The time-integration parameters TOL and Δt_0, and the GWMFE penalty parameters A^2 and B^2 have the same values as in Problem II. The results are given in Table 4.4 and Figs. 4.5 and 4.6. Note that the solid (u), resp. dashed (v), line in the plots represents an accurate reference solution of the original problem *without* diffusion term.

For this problem a correct choice of A^2 is of importance. $A^2 = 10^{-6}$ or 10^{-5} yields a satisfactory approximation, but $A^2 = 10^{-4}$ results in a very bad performance after the pulses have collided, as is illustrated in Fig. 4.6. For the other values of A^2 the approximation is much better (cf. Fig. 4.5), but the computation is still quite expensive.

Miller showed [15] that for this problem a 'vertical rescaling' of the PDEs by a large factor (say 1000) could help to improve the performance of GWMFE. Rescaling the PDEs gives the method much lower traveling pulses to deal with. In fact it means that GWMFE is replaced by the original MFE method, which is profitable since for this problem no gradient weighting is needed.

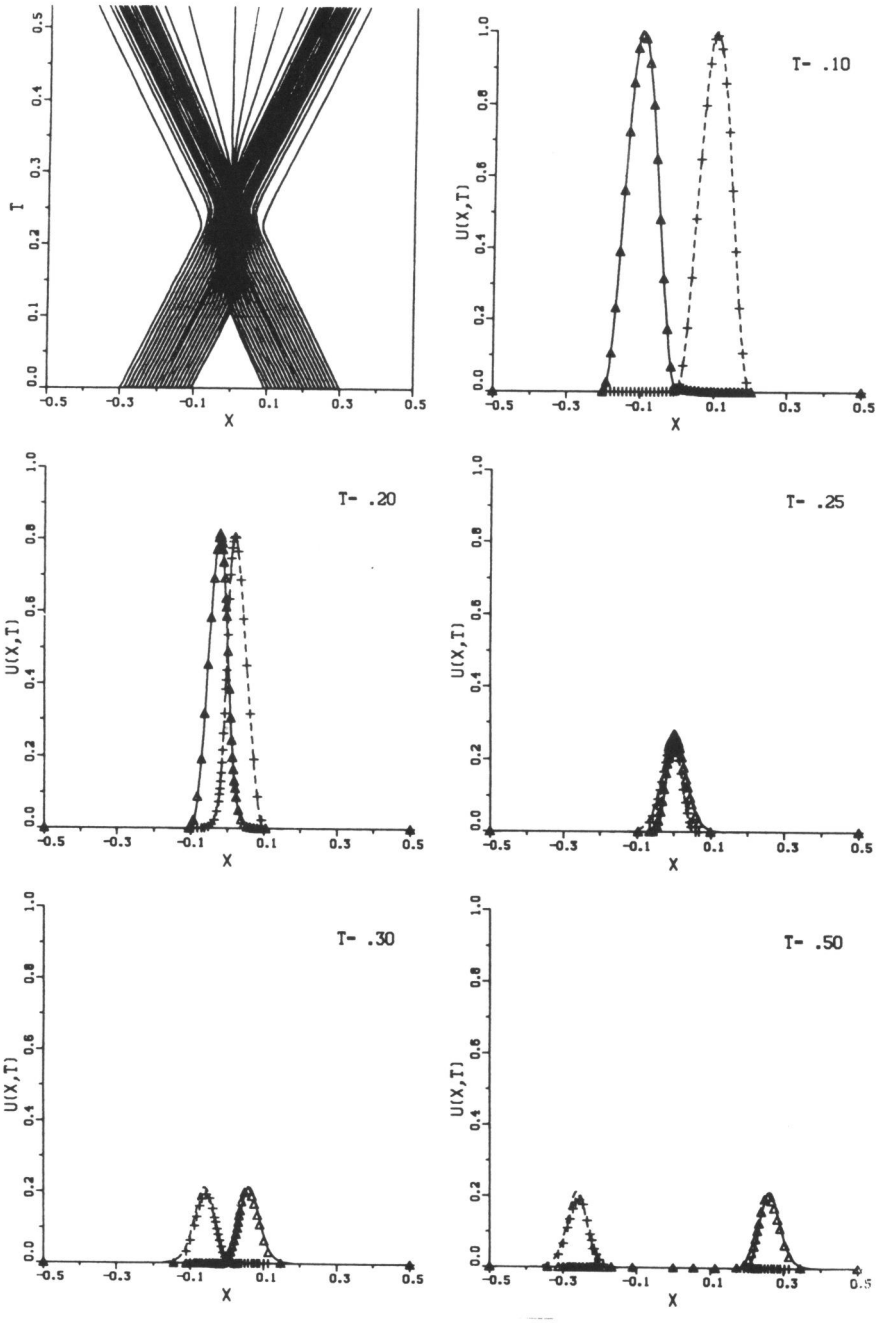


FIGURE 4.5. Problem III. Grid and solution at times $t =$ 0.1, 0.2, 0.25, 0.3, 0.5 for $A^2 =$ 1E-5 and $B^2 = 0$.

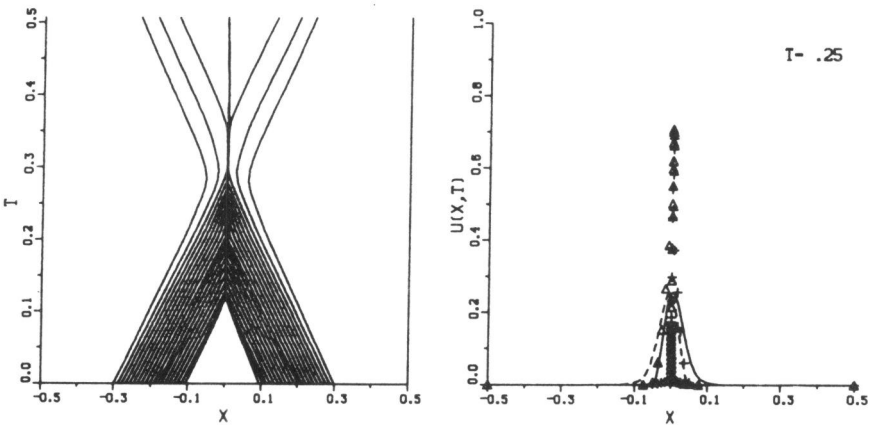


FIGURE 4.6. Problem III. Grid and solution at $t = 0.25$ for $A^2 = 1\text{E-}4$ and $B^2 = 0$.

4.3.4. Problem IV: The Dwyer-Sanders flame-propagation model

Our fourth problem (see [7] for more details and also [20, 21]) serves as a useful test example for the simulation of several basic features which occur in physical flame models. The two PDEs for mass density u and temperature v are given by

$$\begin{aligned} u_t &= u_{xx} - u\, f(v) \\ v_t &= v_{xx} + u\, f(v) \end{aligned}, \qquad 0 < x < 1, \quad 0 < t \le 0.006, \tag{3.4}$$

where $f(v) = 3.52\, 10^6\, e^{-4/v}$. The initial functions are

$$\begin{aligned} u\,|_{t=0} &= 1.0 \\ v\,|_{t=0} &= 0.2 \end{aligned} \qquad 0 \le x \le 1,$$

and the boundary conditions read

$$u_x\,|_{x=0} = 0, \quad v_x\,|_{x=0} = 0, \quad t > 0,$$

and

$$u_x\,|_{x=1} = 0, \quad v\,|_{x=1} = \begin{cases} 0.2 + \dfrac{t}{0.0002}, & 0 < t \le 0.0002 \\ 1.2, & 0.0002 \le t \le 0.006 \end{cases}.$$

The time-dependent forcing function for the temperature at the right boundary represents a heat source which generates a flame front. As soon as the temperature $v\,|_{x=1}$ reaches its maximum value 1.2 at $t = 0.0002$, this flame front starts propagating to the left at a relatively high (almost constant) speed ≈ 150. For $t = 0.006$ the front has nearly reached the left boundary.

The integrals stemming from $< \alpha_i, ufw >$ and $< \beta_i, ufw >$ were evaluated by numerical quadrature using Boole's rule.

A^2	B^2	$t = 0.006$ STEPS	JACS	CTF	ETF	CPU	ORD	qual. sol.	qual. grid
1E-8	0	1566	627	82	166	3.9	2.07	–	□
1E-7	0	400	181	19	55	1.1	1.71	□	+
1E-6	0	361	164	25	27	1.0	1.79	+	+

TABLE 4.5. Problem IV. Integration history.

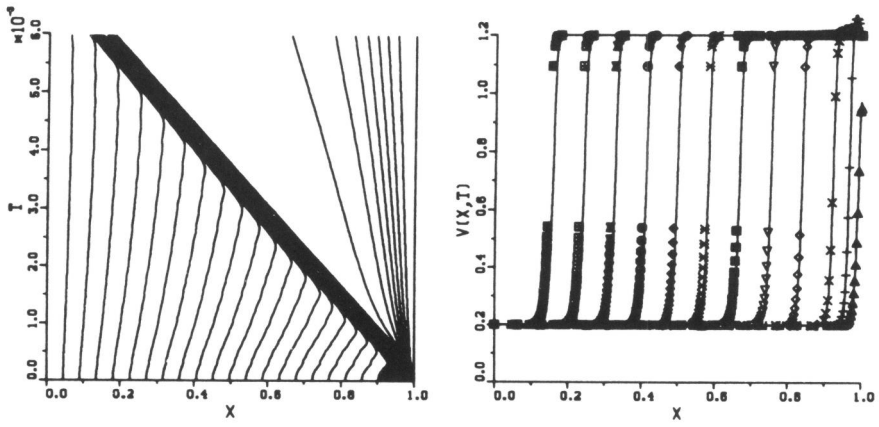


FIGURE 4.7. Problem IV. Grid and temperature component for $A^2 = 1E\text{-}6$ and $B^2 = 0$ at $t =$ .15E-3, .3E-3, .6E-3, 1.2 E-3, ... , 6E-3 (from right to left).

Numerical results for Problem IV

For this problem a strongly nonuniform initial grid was needed with NPTS = 41: 20 uniformly distributed grid points in [0.0,0.9], 10 in [0.9,0.99] and 10 in [0.99,1.0]. The time-integration parameters were TOL = 10^{-4} and $\Delta t_0 = 10^{-5}$. We only present data for the standard penalty parameter choices $A^2 = 10^{-8}$, 10^{-7}, 10^{-6}; $B^2 = 0$.

If we start on a uniform grid the flame front at the right boundary starts at the wrong time, but the solution has more or less the correct speed. This behavior is conform our observation in Problem II that GWMFE can not detect and resolve an emerging pulse. If one approximates the innerproducts with Simpson quadrature instead of Boole's rule (with a nonuniform starting grid) the solution is initially the same, but the flame propagates much too fast. It is possible that even the seventh order quadrature rule is not accurate enough to approximate the integral over the

source term and that this causes the flame to propagate slightly too fast as can be seen in Fig. 4.7 (plot marks are centered). This problem would probably benefit from an appropriate adaptive quadrature method. The grid behavior can be explained for the lower band that points are absorbed in the front and can not pass through a zero curvature (cf. Baines [2]). Note that the gap in the X-T diagram above the upper band is desirable because the solution is nearly constant there.

The obtained average order is higher than for the previous problems (probably because of the tighter tolerance), but unfortunately here the step size behaves very erratically. A plot of the step sizes shows a saw-tooth: the step size is increased, say 4 times in a row, then a convergence error occurs whereupon the step size is decreased by a factor of 4. Then the time error is found to be very small, so the step size is increased, etc., etc.. It is possible however, that this behavior results from the fact that SPGEAR is not tuned to the strongly nonlinear problems arising from PDEs discretized on a grid which moves continuously in time. Another explanation, given by Miller [15], is that each time a new node runs into the front this results in small residual oscillations set up in the nodes just outside the lip of the shock as the nodes readjust. The 20 widely spaced nodes placed in the initial grid ahead of the front are not really needed in the present problem and it would probably be more efficient to use fewer of them.

4.3.5. Problem V: A gasdynamics problem with a small diffusion term

The system of equations for this problem are the one-dimensional Euler equations of gasdynamics in conservative form supplemented by a small diffusion term

$$\begin{aligned} u_t &= -v_x + \varepsilon u_{xx} \\ v_t &= -\frac{\partial}{\partial x}\{\,(\gamma-1)w - 0.5(\gamma-3)\frac{v^2}{u}\,\} + \varepsilon v_{xx}\,, \qquad 0<x<1,\quad t>0, \qquad (3.5) \\ w_t &= -\frac{\partial}{\partial x}\{\,(\gamma w - 0.5(\gamma-1)\frac{v^2}{u})\frac{v}{u}\,\} + \varepsilon w_{xx} \end{aligned}$$

where u, v and w are the density, momentum and total energy per unit volume, respectively, and γ is the ratio of specific heats ($\gamma = 1.4$ in the case of a perfect gas). The initial conditions are linear ramps

$$\begin{aligned} u\,|_{t=0} &= \begin{cases} 1, & 0 \le x \le 0.5-5\varepsilon \\ \text{linear}, & 0.5-5\varepsilon \le x \le 0.5+5\varepsilon \\ 0.125, & 0.5+5\varepsilon \le x \le 1 \end{cases} \\ v\,|_{t=0} &= \quad 0, \qquad\qquad 0 \le x \le 1\,. \\ w\,|_{t=0} &= \begin{cases} 2.5, & 0 \le x \le 0.5-5\varepsilon \\ \text{linear}, & 0.5-5\varepsilon \le x \le 0.5+5\varepsilon \\ 0.25, & 0.5+5\varepsilon \le x \le 1 \end{cases} \end{aligned}$$

The boundary conditions for u and w are of Neumann-type

$$u_x\,|_{x=0} = w_x\,|_{x=0} = 0, \quad \text{resp.,} \quad u_x\,|_{x=1} = w_x\,|_{x=1} = 0;$$

v is subjected to homogeneous Dirichlet boundary conditions. For $\varepsilon = 0$ there is no classical solution for this problem, but we are interested in the weak solution, which is the limiting solution as $\varepsilon \downarrow 0$. This is the so-called shocktube problem, cf. Sod [19], and the problem and its weak solution are briefly described as follows. Consider a

long thin cylindrical tube containing a gas separated by a thin membrane, and assume the gas is at rest on both sides of the membrane, but with different constant pressures and densities on each side. At time $t = 0$, the membrane is broken, for example by a laser beam, and the problem is to determine the ensuing motion of the gas. The course of the solution is as follows: at $t = 0$ the membrane in the tube bursts, with the consequence that the initial discontinuity breaks up into two discontinuities, a contact-discontinuity and a shock wave, which move to the right boundary, and a rarefaction wave moving to the left. If the shock wave has reached the right boundary, it reflects from the wall. For $0 < \varepsilon \ll 1$ the course of the solution is expected to be approximately the same, but now without true discontinuities. In fact the contact discontinuity will be rather smeared in comparison with the inviscid case. Of course, the shock wave and the rarefaction wave will also be smoothed depending on the size of ε.

The integrals resulting from the innerproducts in the right-hand sides of the second and third PDE were evaluated by numerical quadrature using Boole's rule.

Numerical results for Problem V

In the experiment described below we used a diffusion coefficient $\varepsilon = 10^{-3}$. We have also tried $\varepsilon = 10^{-4}$. This resulted in a failure of GWMFE because the stepsizes taken by the integrator were much too small to reach the endpoint due to convergence problems.

We started on a nonuniform grid with NPTS = 41 with 33 points on the linear ramp between [0.5-5ε,0.5+5ε], 3 points in an interval of length 10ε on both sides of the ramp and the boundary points. The time-integration parameters were TOL = 10^{-3} and $\Delta t_0 = 10^{-5}$ and the (standard) penalty parameter values $A^2 = 10^{-6}$ and $B^2 = 0$. The time-integration interval was [0,1].

The integration statistics at the endpoint were STEPS = 698, JACS = 522, CTF = 120, ETF = 53, and ORD = 1.48. To give some insight where GWMFE experienced most trouble: 33 steps were needed to reach $t = 0.01$, 60 for $t = 0.1$ and only 35 to go from $t = 0.1$ to the wall at $t = 0.28$. The reflection phase, $t = 0.28$ until $t = 0.29$, took 102 (successful) steps. Until $t = 0.41$, when the rarefaction wave has reached the left boundary and the contact discontinuity has crossed the reflected shock, another 102 steps were needed. The last phase from $t = 0.41$ until 1.0 took surprisingly many steps, 399. This can be only explained by the oscillations both in the grid movement and in the solution itself, the latter caused by a too coarse grid around the reflected shock. In another run we used the true discontinuities for u and w as initial conditions and all but the two boundary points uniformly distributed over the interval [0.45,0.55]. In this case the initial phase gave, as could be expected, more difficulties. On a total of 586 steps 170 were used to reach $t = 0.1$, but from $t = 0.41$ until 1.0 only 134 steps were needed, the solution remained without oscillations and the grid points stayed in the shock band. This difference in behavior shows that for this problem GWMFE should be applied with care. For the graphical representation of the results we refer to Figs. 4.8 and 4.9. The reference solution in Fig. 4.8 was obtained by using 81 grid points and a time tolerance of 10^{-5}. A comparison with the solution for $\varepsilon = 0$ shows that the added diffusion induces considerable smearing, but on the other hand the speed of the shock is approximated satisfactorily. It should be noted however that more carefully chosen diffusion terms could be used to decrease the smearing of the contact discontinuity in particular. The grid movement is not really optimal. The grid follows the shock wave quite

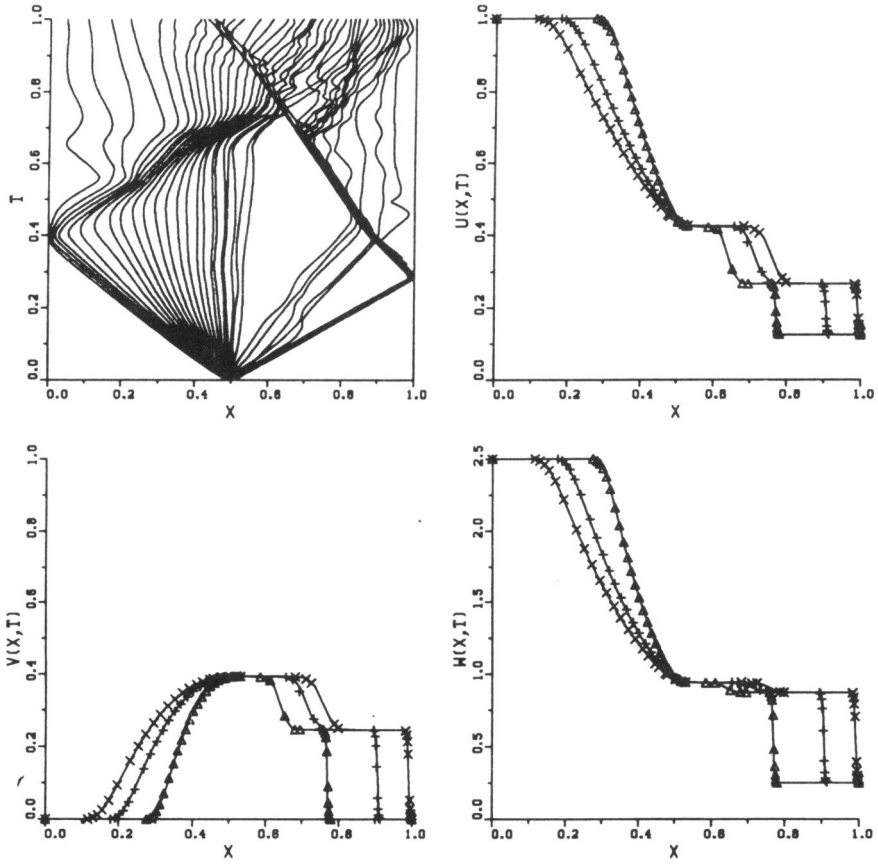


FIGURE 4.8. Problem V. Grid and PDE components at $t = .15, .23, .28$ (Δ,+,$\times$) for $A^2 = 1\text{E-}6$ and $B^2 = 0$.

well and also the rarefaction wave can be clearly seen in Fig. 4.8; but there are very few points in the region of the contact discontinuity. And on the whole the grid movement is not very smooth, although for the last part of the integration this is probably due to the inaccurate and oscillating approximation of the solution.

This is a very hard problem for a nonspecialized code and we therefore consider the result as satisfying although GWMFE showed itself more sensitive to the choice of the penalty parameters and the initial grid or solution than for the previous

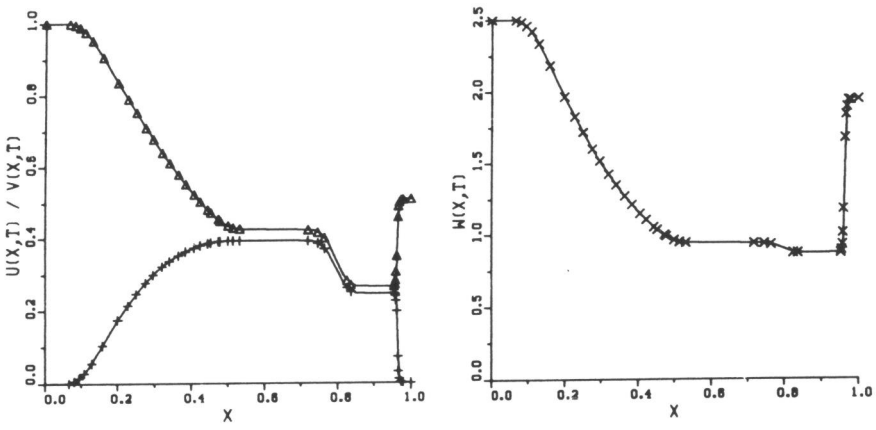


FIGURE 4.9. Problem V. PDE components at $t = .32$ for $A^2 = 1\text{E-}6$ and $B^2 = 0$
u (Δ) and v (+) left, and w right.

problems; small changes in A^2 (say 10^{-5}, with $B^2 = 0$ or 10^{-11}) resulted in a failure and changes in the initial grid more than once caused strongly oscillating solutions at a later time.

4.4. A COMPARISON WITH A MOVING-FINITE-DIFFERENCE METHOD

In [8] a numerical comparison was made, a.o., between MFE (i.e., without gradient-weighting) and a moving-finite-difference method MFD (see also [5, 20]). The methods were tested extensively on three test problems. One of the conclusions was that MFD performed favorably with respect to efficiency and robustness compared to MFE. In this section we will update that test work with the comparison of MFD versus GWMFE on the current set of test problems which has more variety (e.g., sharp moving corners) than the previous. For the sake of completeness we will first give a short description of the MFD method.

4.4.1. The moving-finite-difference method

The MFD method is based on the Lagrangian discretization approach where the grid is moved continuously along with the solution with the aim of reducing the rapid transitions in space and in time that occur when a moving front passes a (fixed) grid point. The PDE (2.1) is transformed to its Lagrangian form

$$\dot{u} - u_x \dot{x} = \mathcal{L}(u), \tag{4.1}$$

where $\dot{u}$ denotes the total time derivative. This PDE is discretized in space using N time-dependent grid points (cf. (2.2)) to obtain

$$\dot{U}_i - \frac{(U_{i+1} - U_{i-1})}{(X_{i+1} - X_{i-1})} \dot{X}_i = L_i, \quad t > 0, \quad 1 \le i \le N. \tag{4.2}$$

Here, U_i represents the semi-discrete approximation to the exact PDE solution u and L_i is the (centered) finite-difference replacement for the differential operator $\mathcal{L}$, both at the point $(x,t) = (X_i(t),t)$. To solve the ODE system (4.2) additional equations are required for the time-dependent grid points X_i. The moving-grid technique that controls the spatial grid-movement in time is due to Dorfi and Drury [6]. For the theoretical background and some analytical aspects of the method we refer to [20], whereas a description of a MOL interface using this technique can be found in [5]. The underlying idea behind this grid movement is the spatial equidistribution of some monitor function. The grid equation reads

$$\frac{\tilde{n}_{i-1} + \tau\dot{\tilde{n}}_{i-1}}{M_{i-1}} = \frac{\tilde{n}_i + \tau\dot{\tilde{n}}_i}{M_i}, \quad 1 \le i \le N, \tag{4.3}$$

where $\tilde{n}_i := n_i - \kappa(\kappa+1)(n_{i+1} - 2n_i + n_{i-1})$ and n_i stands for the, so-called, point concentration $n_i := (\Delta X_i)^{-1}$ of the grid. κ and τ are smoothing parameters; $\kappa \ge 0$ denotes a spatial smoothing parameter and $\tau \ge 0$ is a time-smoothing parameter. M_i is a monitor function, viz., the semi-discrete representation of the first derivative solution functional

$$m(u) = \sqrt{\alpha + \|u_x\|^2}\,.$$

The parameter α should regularize the transformation in regions where u is flat; its magnitude determines the number of grid points in flat regions. In a sophisticated implementation α could be related to the total integral over $m(u)$ with $\alpha = 0$, but until now we just chose a constant related to the average magnitude (over the time-integration interval) of the first spatial derivative of the solution.

In the grid equation the parameter κ determines the level of clustering of the grid points and the arclength monitor M_i determines the shape of the X_i-distribution. The parameter τ prevents the grid movement from adjusting immediately to new values of the monitor function M_i, therefore trying to avoid temporal oscillations in the grid which may cause relatively large errors, when applied to solutions with steep gradients. A standard choice for the spatial smoothing parameter is $\kappa = 2$ and a typical choice for the temporal smoothing parameter $\tau = 10^{-3}$. Equations (4.2) and (4.3) are combined to yield a (stiff) system of ODEs.

4.4.2. MFD versus GWMFE

In this section we will compare the performance of the MFD method and GWMFE. As far as the results have been published before, notably in [8, 20], we will refer to those papers for the precise results of MFD and restrict ourselves here to some remarks.

For Problem Ia, the Burgers' equation with the sinusoidal initial condition, both methods are comparable (cf. Table 4.1 and the results given in [8]). For Problem III the over-all performance of MFD is better. Although GWMFE, with $A^2 = 10^{-6}$ or 10^{-5} and $B^2 = 0$, gives a good solution, the computation is still quite expensive (STEPS $\approx$ 250 and JACS $\approx$ 200) in comparison with the data obtained with the MFD-method in [8], viz., STEPS = 105 and JACS = 58. Also for Problem IV GWMFE needed much more time steps and Jacobians (500 and 250) than the

MFD method which gave an accurate solution at the cost of STEPS = 148 and JACS = 52 (cf. [20]).

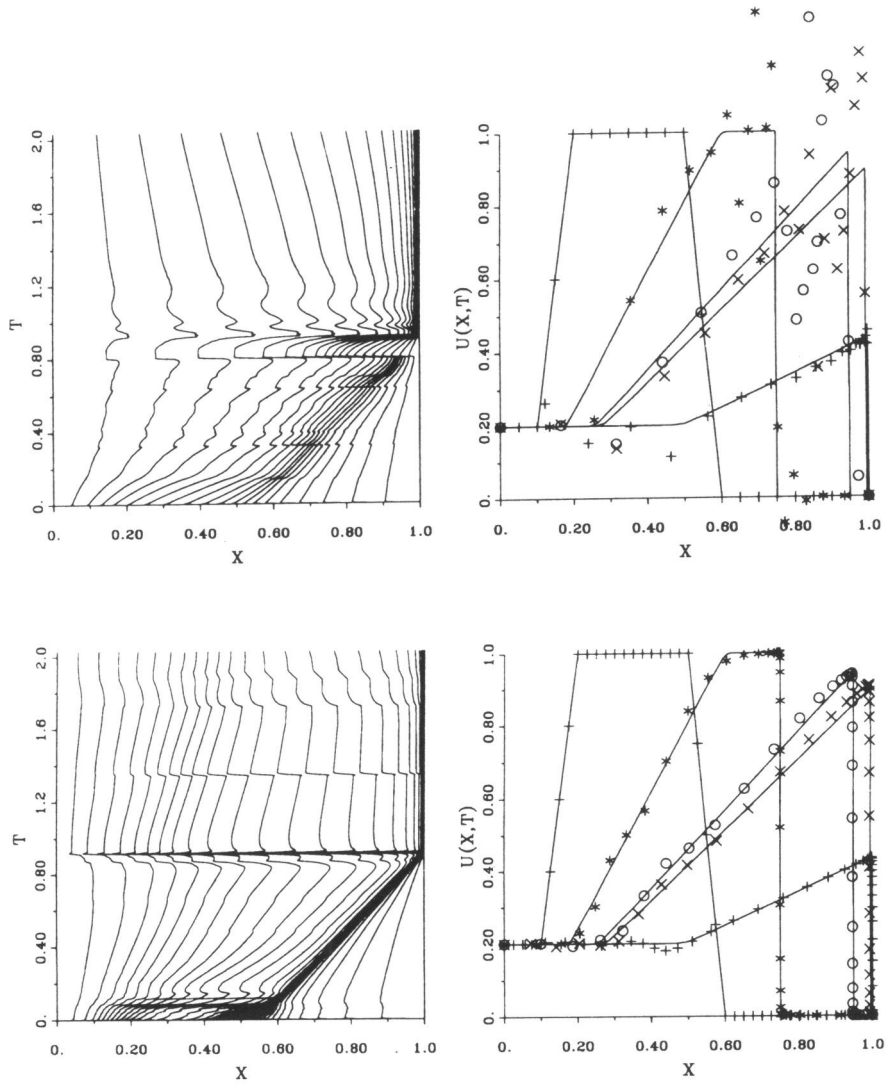


FIGURE 4.10. Problem Ib. Grid and solution at times t = 0.0, 0.4, 0.8, 0.9, 2.0 (+,*,○,×,+) MFD. 21 grid points (above) and 41 grid points (below).

However, MFD has considerable difficulties with Problem Ib (the trapezoid initial condition). We did two experiments, with 21, resp., 41 grid points. The first was roughly twice as expensive as the GWMFE run (STEPS = 447 and JACS = 224) and the results were extremely bad (cf. Fig. 4.10). With 41 grid points MFD performs much better (STEPS = 165 and JACS = 115), but the sharp corners at t = 0.4 are still

not very well resolved. This can be explained by the fact that MFD applies a grid-point movement based on the equidistribution of the arclength and accordingly puts most of the points in the shock ignoring the less steep slope at the left of it. As a consequence, the space derivatives in that region, using a total number of 21 grid points, cannot be approximated well enough by finite differences resulting in large oscillations.

As could be expected the difference in performance between both methods on problem II is similar to that on problem IV. With comparable results MFD (STEPS = 158, JACS = 80) is much cheaper than GWMFE. Even more important is the fact that there is no difference in performance of MFD if one starts at $t = 0.25$, indicating that MFD has less problems than GWMFE with emerging pulses.

Undoubtedly, GWMFE will perform better than MFD on problem V because of its resemblance to Problem Ib, although we did no actual experiments with the MFD method on this problem.

It is obvious from the data above that neither of the two methods is a general purpose method. MFD has problems with solutions having discontinuous derivatives (sharp corners) (resulting in smearing and/or oscillations), largely different monitor values in different parts (oscillations), or near-shocks (small time steps caused by (temporary) node-crossing). Adding more grid points improves almost always the total performance (including the time stepping), but this makes the method less efficient of course. GWMFE has its problems with solutions with emerging structures; it is, in contrast to MFD, dependent on the initial placement of the nodes. Moreover, it results in a strongly nonlinear ODE system which is difficult to solve and, in the framework of MOL-methods, most ODE solving packages, to our experience, will not efficiently solve the system.

4.5. Conclusions

In this paper we have tested the gradient-weighted MFE method in 1-D on five difficult problems with steep moving fronts from different areas of application. A first observation concerns the robustness of the preconditioned GWMFE method compared with the MFE method as used in [8]. Our experience has been, for one of the five problems at least, that for GWMFE the range of penalty parameters is much wider. Miller's rational choice for the values of the penalty parameters A^2 and B^2 has worked quite well for most problems, but there is some indication that for a tighter time tolerance the value of the viscous penalty parameter A^2 should be taken relatively larger than for a more coarse time tolerance (cf. Problem IV). The relative error tolerance on node distance (cf. (2.21)) meant an improvement especially when the nodes were concentrated in a small band; $\rho = 0.1$ appeared to be a good choice. We strongly advise to use the block-diagonal preconditioning of the residual. Although we as yet do not precisely understand why, it brings down the condition number of the Jacobian of the nonlinear system with several orders of magnitude.

We do not advocate to use GWMFE as a general purpose method for all kinds of evolutionary problems. The disadvantage is not only the much more complex nonlinear system resulting from the addition of the strongly nonlinear grid equation, but

also the fact that GWMFE does not get on with the method-of-lines approach. Compared to a fixed grid integration the number of Jacobians needed is much larger, say 1 Jacobian per 10 steps versus 2 every 3 steps, which means a factor 6. Although GWMFE solves Burgers' equation quite satisfactorily and the gasdynamics problem (with diffusion) reasonably, it has its difficulties with problems having an emerging solution. Our advise is to use GWMFE mainly when the solution is known to have steep moving fronts (not true shocks) over the whole time-integration interval.

References

1. S. Adjerid and J.E. Flaherty (1988). A Local Refinement Finite Element Method for Two-Dimensional Parabolic Systems, *SIAM J. Sci. Stat. Comput.*, 9, 792-811.
2. M.J. Baines (1991). An Analysis of the Moving Finite Element Procedure, *SIAM J. Numer. Anal.*, 28, 1323-1349.
3. 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..
4. M. Berzins and R.M. Furzeland (1986). *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.
5. 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 (CWI), Amsterdam.
6. E.A. Dorfi and L. O'C. Drury (1987). Simple Adaptive Grids for 1-D Initial Value Problems, *J. Comput. Phys.*, 69, 175-195.
7. H.A. Dwyer and B.R. Sanders (1978). *Numerical Modeling of Unsteady Flame Propagation,* Report SAND77-8275, Sandia National Laboratories, Livermore, USA.
8. R.M. Furzeland, J.G. Verwer, and P.A. Zegeling (1990). A Numerical Study of Three Moving Grid Methods for One-Dimensional Partial Differential Equations which are based on the Method of Lines, *J. Comput. Phys.*, 89, 349-388 (cf. Chapter 2).
9. B.M. Herbst, S.W. Schoombie, and A.R. Mitchell (1983). Equidistributing Principles in Moving Finite Element Methods, *J. Comp. Appl. Math.*, 9, 377-389.
10. N.K. Madsen (1984). MOLAG: A Method of Lines Adaptive Grid Interface for Nonlinear Partial Differential Equations, in *PDE Software: Modules, Interfaces and Systems*, 207-223, ed. B. Engquist and T. Smedsaas, North-Holland.
11. K. Miller (1981). Moving Finite Elements II, *SIAM J. Numer. Anal.*, 18, 1033-1057.

12. K. MILLER (1983). Alternate Modes to Control the Nodes in the Moving Finite Element Method, in *Adaptive Computational Methods for PDEs*, 165-182, ed. I. BABUŠKA, J. CHANDRA AND J.E. FLAHERTY, SIAM, Philadelphia.
13. K. MILLER (1986). Recent Results on Finite Element Methods with Moving Nodes, in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, 325-338, ed. I. BABUŠKA, O.C. ZIENKIEWICZ, J. GAGO AND E.R. DE A. OLIVEIRA, John Wiley & Sons Ltd..
14. K. MILLER (1988). *On the Mass Matrix Spectrum Bounds of Wathen and the Local Moving Finite Elements of Baines,* Report PAM-430, Center for Pure and Applied Mathematics, University of California, Berkeley (submitted to SIAM J. Numer. Anal.).
15. K. MILLER (1990). *Private Communication.*
16. K. MILLER and R.N. MILLER (1981). Moving Finite Elements I, *SIAM J. Numer. Anal.*, 18, 1019-1032.
17. R.F. SINCOVEC and N.K. MADSEN (1975). Algorithm 494, PDEONE, Solutions of Systems of Partial Differential Equations, *ACM Trans. Math. Software 1*, 261-263.
18. R.F. SINCOVEC and N.K. MADSEN (1975). Software for Nonlinear Partial Differential Equations, *ACM Trans. Math. Software 1*, 232-260.
19. G.A. SOD (1978). A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws, *J. Comput. Phys.*, 27, 1-31.
20. J.G. VERWER, J.G. BLOM, R.M. FURZELAND, and P.A. ZEGELING (1989). A Moving-Grid Method for One-Dimensional PDEs based on the Method of Lines, in *Adaptive Methods for Partial Differential Equations*, 160-175, ed. J.E. FLAHERTY, P.J. PASLOW, M.S. SHEPHARD AND J.D. VASILAKIS, SIAM, Philadelphia (cf. Chapter 3).
21. J.G. VERWER, J.G. BLOM, and J.M. SANZ-SERNA (1989). An Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations, *J. Comput. Phys.*, 82, 454-486.
22. A.J. WATHEN (1986). Mesh-Independent Spectra in the Moving Finite Element Equations, *SIAM J. Numer. Anal.*, 23, 797-814.
23. P.A. ZEGELING and J.G. BLOM (1990). *An Evaluation of the Gradient-Weighted Moving-Finite-Element Method in One Space Dimension,* Report NM-R9006 (Preprint), Centre for Mathematics and Computer Science (CWI), Amsterdam.

Chapter 5

A Note on the Grid Movement Induced by MFE

"Mr. G. drew one additional conclusion from his experiments, that he would like to work on some other kind of mathematics"

5.1. Introduction

During the last decade, moving-grid methods in one space dimension have become popular for solving several kinds of parabolic and hyperbolic Partial Differential Equations (PDEs) involving fine scale structures such as steep moving fronts, emerging steep layers, pulses, shocks, etc.. Moving-grid methods use nonuniform space grids, and move the grid continuously in the space-time domain while the discretization of the PDE and the grid selection procedure are intrinsically coupled. Examples are provided by the Moving-Finite-Element (MFE) method of Miller [11, 13], and by the Moving-Finite-Difference (MFD) method discussed in Verwer et al. [18] (see also references therein). The latter is, in contrast with MFE, restricted to problems in one space dimension.

In two space dimensions, however, application of moving-grid methods is less trivial than in 1D. For instance, there are many possibilities to treat the one-dimensional boundary and to discretize the spatial domain each having their own difficulties for specific PDEs. Therefore, 2D moving-grid methods have mostly been applied only to special types of PDEs. The MFE method ([7, 9, 12]), considering its general approach, allows in principle a large class of problems to be dealt with. However, because of the intrinsic coupling between the discretization of the PDE and the grid selection, the application of MFE, as for any other moving-grid method, is not without difficulties. The main difficulty we are referring to is the threat of grid distortion. Grid distortion can occur in many different ways due to the quite complex solution behaviour of 2D-evolution problems. For example, sharp layer regions could develop and propagate through the domain, or rotating pulses could emerge and die out again. The purpose of this note is to describe the node movement induced by MFE for various PDEs and to indicate some problems concerning the grid structure that can result from this movement.

A standard way of describing moving-grid methods, is the introduction of a transformation of the three dependent variables x, y (space), and t (time) into new variables ξ, η, and τ (usually one chooses $t=\tau$). The effect of the transformation may be to stretch the coordinates in a steep region, so that the transformed derivatives are small compared with the old ones. Of course, many of the difficulties that the spatial discretization yields are now shifted to the problem of how to define the mapping. After having applied the transformation, we obtain the so-called Lagrangian form of the PDE. Within this new formulation the time-derivatives of the spatial variables x and y appear. It is clear, that before using a numerical scheme to discretize the model, one has to define extra equations for these quantities. There are various approaches to take care of this. First, one can use a 2D extension of the equidistribution principle, see, e.g., Brackbill and Saltzman [6], or Dwyer [10]. This idea is either very difficult to work out and to implement, due to the complicated structure of the formulas, or, in a simpler form, it can only be applied to a small class of models. Second, one can use the method of characteristics. This method can, however, only be applied to certain scalar hyperbolic equations or systems having a common convective velocity. For general systems in 2D the use of this method is problematical if possible at all. We would like to focus our attention on the MFE method, which defines the transformation in terms of a residual minimization. For scalar hyperbolic equations MFE is related to the method of characteristics (see, e.g., Baines [2, 3]). This link with the characteristics of the PDE is very useful in one dimension. For in that case all 'disturbances', i.e., shocks, pulses, etc., can merely follow the characteristics. So, once the user has located the grid points at the right positions, the characteristics do the rest. This has the advantage that MFE needs very few points to follow such solutions. In two dimensions it may work properly as well, for the same reasons (see, e.g., Miller [12], or Carlson and Miller [7]). However, in some situations one has to be very careful in applying this method. We will illustrate this with some examples. For parabolic equations the node movement induced by MFE is less understood. For 1D scalar equations one can derive asymptotic relations for the node movement and for the node distribution, indicating that for parabolic equations MFE strives after an equidistribution of second and first order derivatives. An example gives some indication that these results possibly also hold in 2D.

The paper is divided into four sections. In Section 5.2 we briefly describe MFE in two space dimensions, its relation to the method of characteristics for hyperbolic equations and results on the grid movement that can be derived for the parabolic case. Section 5.3 contains two examples of hyperbolic PDEs with a typical solution behaviour. For these two examples it is shown that MFE yields a severely distorted grid, although the computed solution remains accurate. However, this distortion can lead to a breakdown of the numerical time-stepping procedure. Section 5.3 also contains an example of a parabolic equation for which MFE strives after a transformation equidistributing second order derivatives. Finally, Section 5.4 is devoted to some conclusions.

5.2. The movement of the nodes in MFE

Let us consider the scalar PDE

$$\frac{\partial u}{\partial t} = \mathcal{L}(u), \quad (x,y)\in\Omega, \quad t>0, \tag{2.1}$$

with initial and boundary conditions

$$u\,|_{t=0} = u^0(x,y), \quad (x,y)\in\Omega,$$

$$B(u,\nabla u)\,|_{\partial\Omega} = g(t), \quad t>0,$$

where u^0 and g are given functions, and $\mathcal{L}$ represents a differential operator involving only spatial derivatives up to second order. In general, the solution $u(x,y,t)$ of (2.1) may have a very complex behaviour. Even for a restricted situation (a scalar linear PDE with simple boundary conditions), one can have severely varying u-values in space (x,y) and time t. Some examples in this context are steep moving fronts and emerging and rotating pulses.

A common approach handling these phenomena is to introduce a transformation which maps the variables x, y, and t into new variables ξ, η, and τ. Such a transformation can be defined as, e.g.,

$$\begin{aligned} x &= x(\xi, \eta, \tau) \\ y &= y(\xi, \eta, \tau) \\ t &= \tau \\ u(x,y,t) &= v(\xi, \eta, \tau). \end{aligned} \tag{2.2}$$

Applied to the left-hand side of equation (2.1) this gives

$$\frac{\partial u}{\partial t} = \frac{\partial v}{\partial \tau} - u_x \frac{\partial x}{\partial \tau} - u_y \frac{\partial y}{\partial \tau}, \tag{2.3}$$

and additionally equations for x and y must be defined. The effect of the transformation may be to stretch the coordinates in a steep region in space so that, for example, u_ξ and u_η are small in contrast with u_x and u_y. This type of transformation is strived after by methods which equidistribute first or higher order derivatives of the solution. Another effect of the transformation may be to decrease the $\partial v/\partial\tau$ as is done by the method of characteristics and by the finite difference method of Petzold ([15], in 1D). Of course, when using a transformation, most difficulties are shifted to the problem of how to define and carry out the mapping. The Moving-Finite-Element (MFE) method can, in some cases, also be shown to underly a transformation of variables (Baines [3]). Below we will discuss this method and in particular the node movement induced.

5.2.1. Description of MFE

MFE restricts v, x, and y to U, X, and Y from a finite-dimensional subspace. The MFE-approximations are piecewise linear on a, in our case hexagonally connected, triangularization of Ω

$$v \approx U = \sum_{j\in J} U_j(\tau)\,\alpha_j(\xi,\eta),$$

$$x \approx X = \sum_{j \in J} X_j(\tau)\, \alpha_j(\xi, \eta), \tag{2.4}$$
$$y \approx Y = \sum_{j \in J} Y_j(\tau)\, \alpha_j(\xi, \eta),$$

where J is the set of indices of the grid points and α_j are the standard piecewise linear hat functions. Substituted in the PDE (2.1), (2.3), this approximation gives in general a non-zero residual R, defined by

$$R\left(\frac{\partial U}{\partial \tau}, \frac{\partial X}{\partial \tau}, \frac{\partial Y}{\partial \tau}\right) = \frac{\partial U}{\partial \tau} - U_x \frac{\partial X}{\partial \tau} - U_y \frac{\partial Y}{\partial \tau} - \mathcal{L}(U)\,. \tag{2.5}$$

A least-squares minimization is performed on R with respect to the unknowns $\partial U_i/\partial\tau$, $\partial X_i/\partial\tau$, and $\partial Y_i/\partial\tau$, yielding a system of implicit ODEs

$$\begin{aligned} &< R, \quad \alpha_i > = 0 \\ &< R, \quad -U_x \alpha_i > = 0 \\ &< R, \quad -U_y \alpha_i > = 0\,, \quad \forall i \in J, \end{aligned} \tag{2.6}$$

where $<,>$ is the usual L_2-innerproduct on Ω (for an elaboration of (2.6), see [3]). This ODE-system must be integrated numerically to obtain the required fully discretized solution. It is known, that this system may become very stiff. For integration in time, therefore, a suitable stiff ODE-solver must be used to cover all possibilities.

In practical applications, regularization terms (penalty functions) will be added before the minimization procedure is carried out. These penalties prevent the parametrization of $\dot{U}$, $\dot{X}$, and $\dot{Y}$ becoming degenerate (see [9]). Further, they produce forces on the grid movement to prevent the triangles from getting too thin or from losing their orientation. In our experiments we use the penalty functions as defined in [9], but in this section we will not discuss their influence on the grid movement, since the penalties are not the 'driving forces' behind the movement.

Although the Gradient Weighted version of MFE (see, e.g., [7, 20]) is more robust than MFE for steep solutions, the phenomena observed below will be essentially the same for GWMFE.

5.2.2. Relation of MFE with the method of characteristics

Only a few theoretical properties of the resulting ODE system (2.6) are known. One important property is the relation of MFE, in both 1D and 2D, with the method of characteristics for the scalar hyperbolic PDE with

$$\mathcal{L}(u) = -\beta_1(u,x,y,t)\frac{\partial u}{\partial x} - \beta_2(u,x,y,t)\frac{\partial u}{\partial y}. \tag{2.7}$$

It is easy to derive that for β_1 and β_2 linear in u, x, y, while setting aside boundary effects, the ODE system (2.6) is equivalent to

$$\begin{aligned} &\dot{X}_i = \beta_1(U_i, X_i, Y_i, t), \\ &\dot{Y}_i = \beta_2(U_i, X_i, Y_i, t), \\ &\dot{U}_i = 0, \quad i \in J. \end{aligned} \tag{2.8}$$

This simple formulation holds for nonlinear β_1 and β_2 in 1D as well (see, e.g., Baines [2, 3]). So, the ODE system is identical to the discretized system of characteristic ODEs for the PDE (2.1). In the case that

$$\mathcal{L}(u) = -\beta_1 \frac{\partial u}{\partial x} - \beta_2 \frac{\partial u}{\partial y} + \varepsilon \Delta u, \tag{2.9}$$

one can expect, that for small ε (and assuming that the parametrization of $\dot{U}$, $\dot{X}$, and $\dot{Y}$ is not degenerate), MFE results in a grid movement more or less the same as (2.8). (In one dimension one can even quantify the perturbation of the characteristics produced by the diffusion term (see below).)

In 1D this relation with the characteristics is very useful. For, in that case, shocks and pulses have only one degree of freedom to move: they propagate along the characteristic curves of the PDE. In many cases in two space dimensions, this characteristic behaviour is also very beneficial (cf. [8, 12]). However, there are some situations in 2D for which this behaviour will give problems. The main purpose of this note is to illustrate this. We will discuss some of these problems in Section 5.3.

5.2.3. Node movement for parabolic equations

Theoretically, little is known about the grid movement in 2D induced by MFE when applied to parabolic PDEs. In one space dimension, however, it is possible to get some insight by examining specific PDEs. Thrasher and Sepehrnoori [16] have derived expressions for the so-called asymptotic node velocity and density for the transport equation in 1D. These expressions are obtained by letting the number of grid points in an arbitrary subinterval tend to infinity using the concept of asymptotic grading functions (see below). Here, we will analyze in an analogous way the node distribution in an asymptotic sense for the scalar PDE

$$\frac{\partial u}{\partial t} = \mu \frac{\partial^2 u}{\partial x^2} + F(x,u,t), \tag{2.10}$$

where F can also contain spatial derivatives of u.

Let us first write down the MFE-equations (2.6) in one space dimension (without penalty terms) for the PDE (2.10). These are

$$\sum_j < \alpha_i, \alpha_j > \dot{U}_j + < \alpha_i, -U_x \alpha_j > \dot{X}_j = < \alpha_i, \mu u_{xx} + F >, \tag{2.11a}$$

$$\sum_j < -U_x \alpha_i, \alpha_j > \dot{U}_j + < -U_x \alpha_i, -U_x \alpha_j > \dot{X}_j = < -U_x \alpha_i, \mu u_{xx} + F >, \tag{2.11b}$$

$$i = 1, \cdots, N,$$

where N denotes the total number of moving-grid points. Before continuing, we must make some assumptions regarding smoothness of the variables and the rate of convergence. For more details we refer to [16].

Let $\mathcal{F}$ be defined by

$$\frac{d\mathcal{F}}{dx} = \frac{\partial \mathcal{F}}{\partial x} + \frac{\partial \mathcal{F}}{\partial u} \frac{\partial u}{\partial x} := F. \tag{2.12}$$

Let the spatial domain be defined as $[x_L, x_R]$ and let $[A, B]$ be an arbitrary subinterval, with time-dependent endpoints $A(t)$ and $B(t)$, not including any exceptional points, i.e., points which have a zero curvature or a zero asymptotic node density. Define $j=j(N)$ and $k=k(N)$ so that X_{j-1} is the first node and X_{k+1} the last node, respectively, in $[A, B]$. Define h to be $\max_{j \le i \le k+1}\{h_i\}$ and $h_i = X_i - X_{i-1}$. We then need the following assumptions:

(I) *convergence and smoothness of the transformation in* (2.2)

1) $x(\xi, \tau)$ is a continuous asymptotic grading function, i.e.,

$$\lim_{N\to\infty} X_{i(N)}(\tau) = \lim_{N\to\infty} x(i(N)/N, \tau).$$

2) x has continuous third derivatives (except possibly at a finite number of exceptional points).

(II) *smoothness of u and F*

the function $\mathcal{F}$ in (2.12) and the exact solution $u(x,t)$ have continuous third derivatives.

(III) *rate of convergence*

$$X_i = x(i/N) + O(1/N), \quad \text{for } j-1 \le i \le k+1$$

$$\dot{X}_i = x_\tau(i/N) + O(1/N), \quad \text{for } j-1 \le i \le k+1$$

$$X_i - X_{i-1} = x(i/N) - x((i-1)/N) + O(1/N^2), \quad \text{for } j \le i \le k+1$$

$$\dot{X}_i - \dot{X}_{i-1} = x_\tau(i/N) - x_\tau((i-1)/N) + O(1/N^2), \quad \text{for } j \le i \le k+1$$

$$U_i = u(X_i) + O(1/N), \quad \text{for } j-1 \le i \le k+1$$

$$\dot{U}_i = u_t(X_i) + u_x(X_i)\,\dot{X}_i + O(1/N), \quad \text{for } j-1 \le i \le k+1.$$

$$U_i - U_{i-1} = u(X_i) - u(X_{i-1}) + O(1/N^2) \quad \text{for } j \le i \le k+1$$

$$\dot{U}_i - \dot{U}_{i-1} = u_t(X_i) - u_t(X_{i-1}) + u_x(X_i)\dot{X}_i - u_x(X_{i-1})\dot{X}_{i-1} + O(1/N^2)$$
$$\text{for } j \le i \le k+1$$

From these assumptions the following relations can be derived

i) $h_i = x_\xi(i/N)/N + O(1/N^2) = O(1/N)$

ii) $m_i := (U_i - U_{i-1})/h_i = u_x(X_i) + O(h) = u_x(X_{i-1}) + O(h)$

iii) $\dot{U}_i^+ := \dot{U}_i - m_{i+1}\dot{X}_i = u_t(X_i) + O(h), \quad \dot{U}_i^- := \dot{U}_i - m_i\dot{X}_i = u_t(X_i) + O(h)$

iv) $\dot{m}_i = u_{x\tau}(X_i) + O(h)$

v) $\dot{X}_i$ is bounded

vi) $O(1/N)$ and $O(h)$ are interchangeable.

Working out the innerproducts and applying Taylor expansions of $\mathcal{F}$ around U_i and X_i, equations (2.11a) and (2.11b) can be combined to obtain

$$h_{i+1}(\dot{U}_{i+1}^- + 2\dot{U}_i^+) - h_i(2\dot{U}_i^- + \dot{U}_{i-1}^+) -$$
$$6\{\frac{h_{i+1} - h_i}{2}\mathcal{F}_{x_i} + \frac{m_{i+1}h_{i+1} - m_i h_i}{2}\mathcal{F}_{u_i} + \frac{m_{i+1}h_{i+1}^2 + m_i h_i^2}{3}\mathcal{F}_{xu_i} + \quad (2.13)$$

$$\frac{h_{i+1}^2 + h_i^2}{6} \mathcal{F}_{xx_i} + \frac{m_{i+1}^2 h_{i+1}^2 + m_i^2 h_i^2}{6} \mathcal{F}_{uu_i} \} = O(h^3).$$

The next step is summation of equation (2.13) on the interval $[A, B]$ from $i=j$ to k. This yields

$$\begin{aligned} & h_{k+1}(\dot{U}_{k+1}^- + 2\dot{U}_k^+) - h_j(2\dot{U}_j^- + \dot{U}_{j-1}^+) - \\ & 3\{h_{k+1}\mathcal{F}_{x_k} - h_j\mathcal{F}_{x_j} + m_{k+1}h_{k+1}\mathcal{F}_{u_k} - m_j h_j \mathcal{F}_{u_j}\} + \\ & \sum_{i=j+1}^{k} [h_i(-\dot{U}_i^- + \dot{U}_{i-1}^+) - 6\{ - \frac{h_i}{2}(\mathcal{F}_{x_i} - \mathcal{F}_{x_{i-1}}) - \frac{m_i h_i}{2}(\mathcal{F}_{u_i} - \mathcal{F}_{u_{i-1}}) + \\ & \frac{m_i h_i^2}{3}(\mathcal{F}_{xu_i} + \mathcal{F}_{xu_{i-1}}) + \frac{h_i^2}{6}(\mathcal{F}_{xx_i} + \mathcal{F}_{xx_{i-1}}) + \frac{m_i^2 h_i^2}{6}(\mathcal{F}_{uu_i} + \mathcal{F}_{uu_{i-1}})\}] = O(h^2). \end{aligned} \tag{2.14}$$

Using assumption II, the relations ii), iii), iv), and v), and substituting the PDE (2.10) we arrive at a discrete formulation of the distribution of the nodes

$$\begin{aligned} & 3\mu[h_{k+1}u_{xx}(X_k) - h_j u_{xx}(X_j)] - \\ & \sum_{i=j+1}^{k} \{h_i^2[\mu\, u_{xxx}(X_i) + u_{xx}(X_i)(\dot{X}_i + \mathcal{F}_{u_i})]\} = O(h^2). \end{aligned} \tag{2.15}$$

Applying i), multiplying by N, and letting $N \to \infty$ results in the continuous form

$$3\mu \left[\frac{u_{xx}}{\xi_x} \right]_A^B = \int_A^B \frac{\mu u_{xxx} + u_{xx}[\dot{x} + \mathcal{F}_u]}{\xi_x} dx. \tag{2.16}$$

Finally, differentiating results in the asymptotic node movement

$$\dot{x} = -\mathcal{F}_u + \mu \left(2\frac{u_{xxx}}{u_{xx}} - 3\frac{\xi_{xx}}{\xi_x}\right). \tag{2.17}$$

Note, that relation (2.17) is valid only in intervals without points with a zero asymptotic node density ($\xi_x = 0$) or with zero curvature ($u_{xx} = 0$). However, Baines [3] stated that grid points cannot pass a point with zero curvature with the consequence that grid points are confined to regions between two zero-curvature points (the so-called anti-cluster property of singular points). In fact, the grid points are even repulsed from the singular points.

If $\mu \neq 0$, then equation (2.17) can be integrated to obtain the asymptotic node distribution

$$\xi_x = K(t) \mid u_{xx} \mid^{2/3} \exp\left(\frac{1}{3\mu} \int (-\mathcal{F}_u - \dot{x}) dx\right). \tag{2.18}$$

For $\mu = 0$ and $F = F(u,t)$ equation (2.17) means that a grid point will propagate along the characteristic $\dot{x} = -\mathcal{F}_u$ and is *not* dependent on the grid distribution. This is the situation as described in Section 5.2.2. For $\mu \neq 0$ one can easily derive asymptotic node distributions for restricted choices of F. For instance, the node distribution of the so-called shifting pulse, which we used as an example in [20], once every point travels with the velocity of the pulse, is given by

$$\xi_x = K(t)\,|u_{xx}|^{2/3} \tag{2.19}$$

which can be derived from (2.18) provided that $\dot{x} = \dot{x}(t)$ and $F = F(x,t)$. So once every point travels with the velocity of the pulse, the nodes should be distributed by MFE according to (2.19) and the plots in [20] show indeed that MFE approximately equidistributes some power of the second derivative of the solution.

For convection-diffusion equations like (2.9), one can derive from (2.18), assuming that $\dot{x} = 0$, $u_t = 0$ and $F = F(u,t)$, a steady-state distribution

$$\xi_x = K\,|u_{xx}|^{2/3}\,|u_x|^{1/3}, \tag{2.20}$$

indicating that in this case a combination of first and second order derivatives is equidistributed.

5.3. Numerical examples

In our numerical experiments we have solved the implicit ODE system (2.6) with the (implicit) BDF integrator SPGEAR of the SPRINT package [4] in the usual way. This means, among others, that the resulting algebraic system is solved by a modified Newton process.

5.3.1. Example I ('Anisotropy')

Our first example is an anisotropic wave front (see Whitham [19, p.254]). In short, anisotropy means that a difference exists between the directions of the characteristic curves of the PDE (the movement of the 'fluid'-particles) and the movement of the wavefront. This phenomenon can not occur in one space dimension. In 2D, anisotropy may give rise to a distorted MFE grid eventually leading to a breakdown of the numerical time-stepping procedure.

Probably the best way to illustrate this effect is by giving a PDE-example. Consider, for this purpose,

$$\frac{\partial u}{\partial t} = -\beta_1 \frac{\partial u}{\partial x} - \beta_2 \frac{\partial u}{\partial y} + \varepsilon \Delta u, \tag{3.1}$$

on the domain $\Omega = (0,1)\times(0,1)$, with

$$\beta_1 = u,$$

$$\beta_2 = (\frac{3}{2} - u),$$

$$u\,|_{t=0} = u_{\text{exact}}\,|_{t=0},$$

$$u\,|_{\partial\Omega} = u_{\text{exact}}\,|_{\partial\Omega},$$

and

$$u_{\text{exact}} = \frac{3}{4} - \frac{1}{4}\,\frac{1}{1+\exp\left(\dfrac{-4x+4y-t}{32\varepsilon}\right)}.$$

The exact solution of this model problem (a scalar version of the system in [5,

p.89]) describes a wavefront with a steep transition area of thickness $O(\varepsilon)$, that moves, under an angle of 135^o with the positive x-axis, from the middle of Ω to the upper left corner. For $\varepsilon \downarrow 0$ the transition area becomes steeper, and for $\varepsilon=0$ a pure hyperbolic situation is created with a discontinuous moving shock.

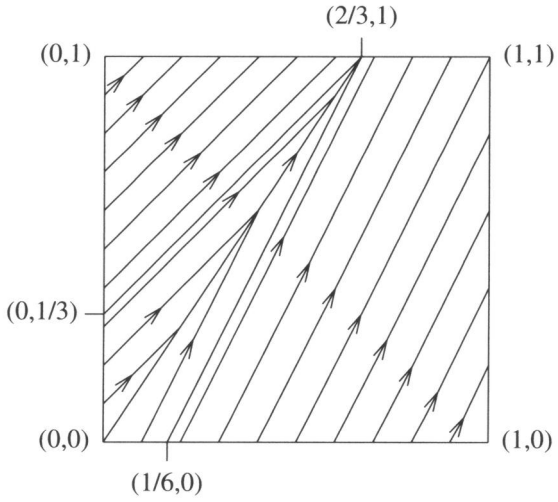


FIGURE 5.1. Node movement by characteristics from $t = 0$ to $t = 1$.

Formulation (2.8) reveals that the method of characteristics, and, to a great extent MFE as well, at first will send the grid points to the upper right corner of the domain. This can be seen very easily by writing out the equations (2.8) for this case ($\varepsilon = 0$):

$$\text{for } y > x + t \tag{3.2.a}$$

$$\dot{X}_i = U_i(t) \approx \frac{3}{4},$$

$$\dot{Y}_i = \frac{3}{2} - U_i(t) \approx \frac{3}{4},$$

and

$$\text{for } y < x + t \tag{3.2.b}$$

$$\dot{X}_i = U_i(t) \approx \frac{1}{2},$$

$$\dot{Y}_i = \frac{3}{2} - U_i(t) \approx 1.$$

The characteristic movement from $t = 0$ until $t = 1$ is pictured in Figure 5.1. This grid movement will lead to a coarse grid in the lower left corner of Ω, since all grid points are moved to the upper right corner. Further, at later points in time, a

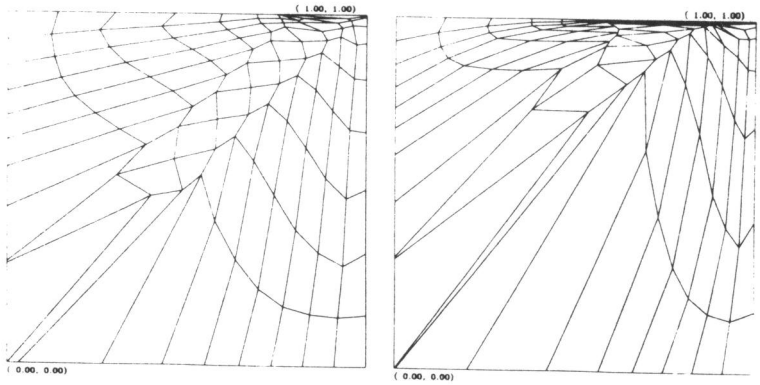


FIGURE 5.2. MFE grid for Example I at $t = 0.5$ and 1.0.
Dividing each quadrilateral by the diagonal from upper left to lower right gives the MFE triangles.

congestion of grid points near the upper side of the domain Ω will arise, due to the boundary effects. Since, in that area, the relative distance between the nodes will become very small, the penalty functions should keep the points from moving into each other and thus the ease with which the ODE system (2.6) can be solved (if at all), will become very dependent upon the correct choice of the penalty functions. It could easily result in a drastic drop of performance only caused by inadmissible triangle orientations during the Newton process. It must be noted, however, that for $\varepsilon \downarrow 0$ MFE will resemble the method of characteristics more and more, resulting in an almost exact solution in each grid point. In Figure 5.2 a boundary layer of points is shown, obtained by applying MFE to problem (3.1) with $\varepsilon = 5.10^{-3}$ and a uniform starting grid of 11×11 moving grid points. At $t \approx 1.02$ the computational process breaks down because of the unacceptable triangle orientations. This could be prevented by taking larger penalty values resulting in a less accurate solution.

It is obvious that for these situations a procedure to delete and create nodes could be added to MFE to prevent a congestion of grid points and to keep the finite element approximation of the solution accurate enough. Also for this special case, a solution to eliminate the anisotropy in the PDE could be found. One could think of applying a transformation to the PDE that describes a rotation of the variables over an angle $\phi = 135^o$. In the new variables the characteristic curves and the direction of the normal to the wave front would coincide (the anisotropy would then cease to exist). In general situations, however, it is, a priori, unclear how to choose ϕ, especially ϕ could even be time-dependent. So far, it has not been possible to reformulate MFE in a proper way to generate such a transformation automatically.

5.3.2. Example II ('Grid rotation')

Our second example, copied from [14], is concerned with the fact that in 2D an unwanted rotation of the grid can occur. To illustrate this, consider

$$\frac{\partial u}{\partial t} = -\beta_1 \frac{\partial u}{\partial x} - \beta_2 \frac{\partial u}{\partial y}, \tag{3.3}$$

on the domain $\Omega = (-0.5, 1.5)\times(-0.5, 1.5)$, with

$$\beta_1 = +\pi(y - \tfrac{1}{2}),$$

$$\beta_2 = -\pi(x - \tfrac{1}{2}),$$

$$u|_{t=0} = \exp(-80[(x-\tfrac{1}{2})^2 + (y-\tfrac{3}{4})^2]),$$

and

$$u|_{\partial\Omega} = 0.$$

Although the boundary condition is mathematically not consistent with the initial condition, it is expected that this will give no problems in numerical computations, since the difference is less than the machine precision.

The exact solution describes a pulse that moves around in circles with a constant speed. During this movement the shape of the pulse does not change. The characteristic curves are circles with centre $(\frac{1}{2}, \frac{1}{2})$, which can be derived immediately from (2.8) and (3.3):

$$\dot{U}_i = 0 \quad \text{and}$$

$$(X_i - \tfrac{1}{2})^2 + (Y_i - \tfrac{1}{2})^2 = r_i^2, \quad 0 < r_i < 1,$$

with $\dot{r}_i = 0, \ \forall i$.

In contrast with the previous example, the movement of the grid points might be called ideal. They follow the steep parts of the solution in an optimal way and MFE benefits by this property, resulting in a good approximation. However, since we fixed the corner points of the square, the grid will exhibit an unwanted spiral structure. This occurs when the pulse has moved down to the lower region of Ω. A consequence of this effect is the so-called line tangling, a 2D version of node crossing in 1D. The numerical procedure will break down whenever this occurs, again because of inadmissible triangle orientations during the Newton process. In this case, however, larger penalty values can only delay but not prevent the breakdown. We show this spiral effect in Figure 5.3, where we pictured the grids, produced by MFE, at various time values. The starting grid consists again of 11×11 points of which 5×5 are distributed uniformly around the cone in (0.25,0.75)×(0.5,1.0). At $t \approx 1.52$ the computation breaks down. Again the MFE approximation in each grid point is rather accurate and the performance of MFE in the time stepping process is satisfying until the spiral structure leads to line tangling.

Note that in this case intermediate grid rezoning or annihilation and creation of points, based only on the accuracy of the MFE approximation, would be no cure for

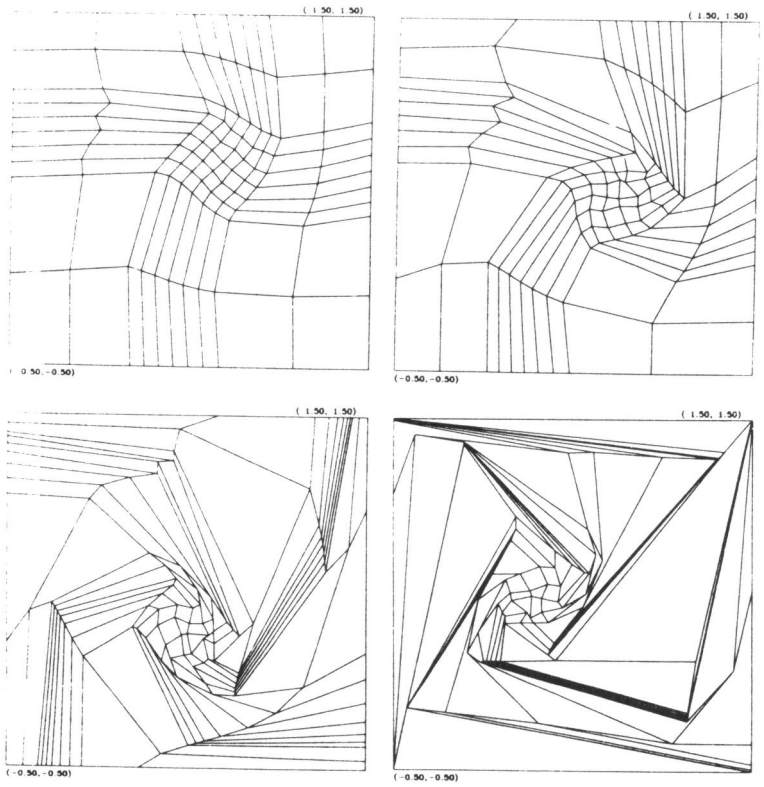


FIGURE 5.3. MFE grid for Example II at $t = 0.25$, 0.5, 1.0 and 1.5. Dividing each quadrilateral by the diagonal from upper left to lower right gives the MFE triangles.

the grid distribution problem. Of course, there are some other means to check this effect, again for this special case. First, one could allow the grid points on the boundary to move with the internal points (i.e., 'move around the corner'). For this problem, for instance, it is easy to replace Ω initially by a circular domain and allow free

movement of the boundary nodes. The grid then produces no longer spirals, but is congruent with the initial grid and rotated with the characteristic velocity, and the problem is solved without any trouble. Another trick to avoid that the numerical procedure breaks down, is described by Mueller and Carey [14]. They add an extra penalty term to the method, which brings on an anti-rotation to the grid movement. This regularization term, however, has only a limited working: with any choice of the constant, appearing in the penalty, there remains some point of time for which the line tangling takes place. Only, with larger penalty values the method would collapse at a later moment in the time-integration. But, larger penalty values also result in a worse resolution of the pulse, yielding larger errors during the computation.

5.3.3. Example III ('Parabolic pulse')

In the two previous examples we encountered difficulties in applying MFE due to its strong relation with the method of characteristics for hyperbolic equations. Next, we give an example of a PDE with an exact solution very similar to that of model (3.3), but now the PDE has a parabolic character. It has already been treated by several authors ([1, 17]), and is defined by

$$\frac{\partial u}{\partial t} = \Delta u + f(x,y,t), \tag{3.4}$$

on the domain $\Omega = (-0.5,1.5)\times(-0.5,1.5)$, with

$$u|_{t=0} = u_{\text{exact}}|_{t=0},$$

$$u|_{\partial\Omega} = u_{\text{exact}}|_{\partial\Omega},$$

The source $f(x,y,t)$ is chosen so that the exact solution is

$$u_{\text{exact}} = \exp(-80[(x-r(t))^2 + (y-s(t))^2]),$$

where

$$r(t) = (2+sin(\pi t))/4, \quad s(t) = (2+cos(\pi t))/4.$$

This solution is a rotating pulse and thus very similar to the solution of Example II. However, in contrast with the hyperbolic Example II, the grid points do not move according to a principle like (2.8). In particular, MFE, applied to (3.4), shows no spiral effect. The points are not stuck to their position on the pulse and the grid structure remains more or less the same during the time-stepping. This is illustrated in Figure 5.4, where we pictured the grid at several points in time. Although the error of the approximation is higher than in Example II (this can be repaired by increasing the number of points), the procedure does not break down because of grid tangling. On the contrary, once the grid has been forced around the cone, the time stepping process is satisfying, although the penalty choice is also of influence in this case.

Finally, noteworthy is that the concentration of triangles in regions with large second order derivatives indicates a similar equidistribution behaviour as stated in Section 5.2.3 for one dimension.

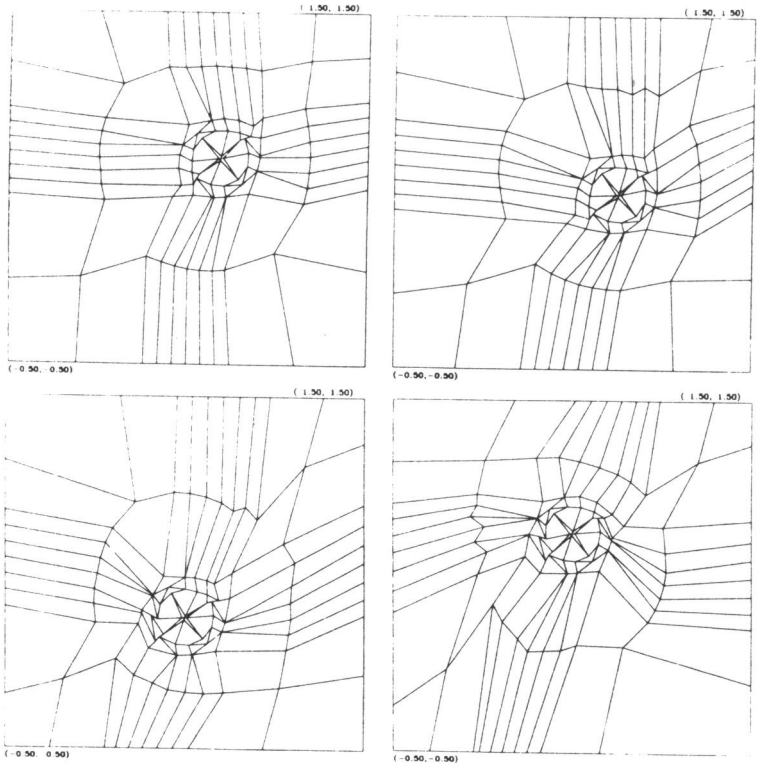


FIGURE 5.4. MFE grid for Example III at $t = 0.25$, 0.5, 1.0 and 2.0. Dividing each quadrilateral by the diagonal from upper left to lower right gives the MFE triangles.

5.4. CONCLUSIONS

For hyperbolic or strongly convection dominated convection-diffusion equations, the grid points are moved by MFE in a way similar to the method of characteristics. This results in a very good approximation of the solution but sometimes also in distorted grids, because the grid movement is independent of the grid distribution. Such grids then eventually cause the numerical time-stepping to fail. A procedure to delete and create points could in some cases be a remedy, but will on the other hand complicate the method considerably.

For scalar parabolic equations one can show that in 1D the MFE movement of a grid point does depend on the grid distribution. MFE approximates a transformation striving after equidistribution of derivatives of the solution. An example showed that possibly this remains valid also in 2D.

REFERENCES

1. S. ADJERID and J.E. FLAHERTY (1988). A Local Refinement Finite Element Method for Two Dimensional Parabolic Systems, *SIAM J. Sci. Stat. Comput.*, 9, 792-881.
2. M.J. BAINES (1989). *Moving Finite Elements and Approximate Legendre Transformations,* Numerical Analysis Report 5/89, University of Reading.
3. M.J. BAINES (1991). An Analysis of the Moving Finite Element Procedure, *SIAM J. Numer. Anal.*, 28, 1323-1349.
4. 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..
5. J.H.M. TEN THIJE BOONKKAMP (1988). *The Numerical Computation of Time-Dependent, Incompressible Fluid Flow,* PhD. Thesis, Universiteit van Amsterdam.
6. J.U. BRACKBILL and J.S. SALTZMAN (1982). Adaptive Zoning for Singular Problems in Two Dimensions, *J. Comput. Phys.*, 46, 342-368.
7. N. CARLSON and K. MILLER (1988). The Gradient Weighted Moving Finite Element Method in Two Dimensions, in *Finite Elements Theory and Application*, 152-164, ed. D.L. DWOYER, M.Y. HUSSAINI AND R.G. VOIGHT, Springer Verlag.
8. M.J. DJOMEHRI, S.K. DOSS, R.J. GELINAS, and K. MILLER (1985). Applications of the Moving Finite Element Method for Systems in 2D, *Preprint (submitted to J. Comput. Phys.).*
9. M.J. DJOMEHRI and K. MILLER (1981). *A Moving Finite Element Code for General Systems of PDE's in 2-D,* Report PAM-57, Center for Pure and Applied Mathematics, University of California, Berkeley.
10. H.A. DWYER (1983). A Discussion of Some Criteria for the Use of Adaptive Gridding, in *Adaptive Computational Methods for PDEs*, 111-122, ed. I. BABUŠKA, J. CHANDRA AND J.E. FLAHERTY, SIAM, Philadelphia.
11. K. MILLER (1981). Moving Finite Elements II, *SIAM J. Numer. Anal.*, 18, 1033-1057.

12. K. MILLER (1986). Recent Results on Finite Element Methods with Moving Nodes, in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, 325-338, ed. I. BABUŠKA, O.C. ZIENKIEWICZ, J. GAGO AND E.R. DE A. OLIVEIRA, John Wiley & Sons Ltd..
13. K. MILLER and R.N. MILLER (1981). Moving Finite Elements I, *SIAM J. Numer. Anal.*, 18, 1019-1032.
14. A.C. MUELLER and G.F. CAREY (1985). Continuously Deforming Finite Elements, *Int. J. Numer. Methods Eng.*, 21, 2099-2126.
15. L.R. PETZOLD (1987). Observations on an Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations, *Appl. Numer. Math.*, 3, 347-360.
16. R. THRASHER and K. SEPEHRNOORI (1986). On Equidistributing Principles in Moving Finite Element Methods, *J. Comp. Appl. Math.*, 16, 309-318.
17. R.A. TROMPERT and J.G. VERWER (1991). A Static-Regridding Method for Two-Dimensional Parabolic Partial Differential Equations, *Applied Numer. Maths.*, 8, 65-90.
18. J.G. VERWER, J.G. BLOM, R.M. FURZELAND, and P.A. ZEGELING (1989). A Moving-Grid Method for One-Dimensional PDEs based on the Method of Lines, in *Adaptive Methods for Partial Differential Equations*, 160-175, ed. J.E. FLAHERTY, P.J. PASLOW, M.S. SHEPHARD AND J.D. VASILAKIS, SIAM, Philadelphia.
19. G.B. WHITHAM (1974). *Linear and nonlinear waves*, Wiley-Interscience, New-York.
20. P.A. ZEGELING and J.G. BLOM (1990). *An Evaluation of the Gradient-Weighted Moving-Finite-Element Method in One Space Dimension,* Report NM-R9006 (Preprint), Centre for Mathematics and Computer Science (CWI), Amsterdam (to appear in J. Comput. Phys.).

Chapter 6

Application of a Moving-Grid Method to a Class of 1D Brine Transport Problems in Porous Media

"Even op spoor 6 binnenkomen"

6.1. Introduction

The subject of this paper originates from the problem of disposal of hazardous waste, e.g., high-level radioactive waste, in salt formations. The most probable mechanism for release of these wastes to the biosphere is by transport via groundwater. Existing standard mathematical models for the study of groundwater flow and brine transport assume that the salt concentration is less than or equal to seawater concentration. This, however, is not true for flows in the vicinity of rock salt formations. In the vicinity of these formations, e.g., salt domes, the salt concentration may become very large and in fact to an extent that the groundwater flow is really influenced by the salt concentration. Recent theoretical and experimental hydrological studies indicate that for such high-concentration situations the involved basic equations of flow and transport need to be modified [9, 10]. This involves a significant effort in numerical modelling since the partial differential equations (PDEs) which show up cannot be solved by analytical means. The contents of the current paper has its origin in part of these numerical modelling studies.

We discuss the application of a numerical moving-grid method, originally developed for general time-dependent PDEs in one space dimension, to a specific class of nonlinear, brine transport problems borrowed from [7]. Our purpose is twofold. Firstly, while focusing on the application, we wish to show that this numerical method is a valuable tool for modelling nonlinear (brine) transport problems in one-space dimension, specifically so for problems having solutions with rapid transitions, such as a solute front transported in the soil or a sharp fresh-salt water interface. Secondly, while now focusing on the numerical analysis aspects, we wish to show that for the class of transport problems chosen, the grid-movement approach is successful and may provide a notable improvement compared to the more traditional approach of time-stepping on a fixed spatial grid.

The numerical method is based on the method-of-lines (MOL) approach for solving time-dependent PDEs (see, e.g., Ch. 10 in [5], and [11]). The method is of the finite-difference type and implicit, and thus applicable to wide classes of one-space dimensional PDE systems. In addition, the main feature of the method is that for evolving time it automatically refines the spatial grid in regions with large spatial transitions. Since it is a Lagrangian type method, in many cases of practical interest the grid movement also softens the solution behaviour in time, so that larger time steps can be taken than on a fixed spatial grid. The actual moving-grid algorithm underlies the principle of spatial equidistribution and is provided with appropriate grid regularization procedures to cater for smooth grid trajectories. The principal ideas for this regularization emanate from [6] and a further comprehensive discussion of the complete moving-grid algorithm can be found in [13] (see also [8] and the references therein).

An advantage of the moving-grid method is that it can be implemented in most of the MOL software packages based on sophisticated implicit stiff ODE/DAE solvers. We mention the BDF solvers developed by Gear, Byrne, Hindmarsh, Petzold and others (see, e.g., [4]). In the brine transport problem application we have used the FORTRAN package SPRINT [1]. SPRINT is a package developed for solving general algebraic, ordinary and partial differential equations. Its core is formed by implicit stiff ODE/DAE solvers (of BDF type). In [3] SPRINT has been provided with a software interface based on the moving-grid method here considered. This moving-grid interface (MGI), being an extension of the fixed-grid interface based on [12], is a most convenient tool for researchers who wish to concentrate on modelling their physics, since it automatically carries out the spatial discretization, thus relieving them from numerical choices to be made and saving programming time. The use of MGI merely requires that the mathematical problem be formulated in terms of FORTRAN statements. Consequently, both the spatial discretization and the temporal integration can then be left to the package and the user only has to set to some numerical control parameters, like a local tolerance parameter for the numerical integration in time, the number of points for the spatial discretization, and some parameters controlling the grid movement. In the experiments reported here we have used the MGI from [3].

Section 6.2 is devoted to the moving-grid method. An outline is given on important properties and principles of this method. The class of fluid-flow/salt-transport problems we focus on is discussed in Section 6.3. The physical properties involved here are advection-dispersion and in case of dominant advection solutions with rapid spatial and temporal transitions arise. In Section 6.4 we present results of numerical tests, emphasizing the occurrence of the rapid transitions and the use of the moving-grid method. Section 6.5 is devoted to concluding remarks.

6.2. THE MOVING-GRID ALGORITHM

6.2.1. The moving-grid algorithm

We will present the algorithm along the lines of the numerical method-of-lines approach for solving time-dependent PDEs. Consider an abstract Cauchy problem for a system of PDEs in one space dimension,

$$\frac{\partial u}{\partial t} = \mathcal{L}(u), \quad x_L < x < x_R, \quad t > 0, \tag{2.1}$$

where $u = u(x,t)$ and $\mathcal{L}$ is a spatial operator of at most order 2. We do not discuss boundary conditions here, since these are dealt with in the usual way. It is assumed that the solution u has (a sufficient number of) finite temporal and spatial derivatives and these are allowed to be very large. We thus focus on problems possessing solutions u with very large spatial and temporal variations, but do not consider problems with genuine discontinuous solutions.

The discretization of the PDE is carried out in two stages. In the first stage $\mathcal{L}(u)$ is discretized on a selected space mesh, which converts (2.1) into a Cauchy problem for an ODE system. The second stage then deals with the numerical integration in time of this semi-discrete system. Let us discuss the first stage, which here takes place in a moving reference frame. First we choose N time-dependent grid points $X_i(t)$, $1 \le i \le N$ defining the space grid

$$X: x_L = X_0 < ... < X_i(t) < X_{i+1}(t) < ... < X_{N+1} = x_R, \quad t \ge 0. \tag{2.2}$$

As yet the trajectories $X_i(t)$ are unknown, but they are supposed to be (sufficiently often) differentiable. Next, along each trajectory $x(t) = X_i(t)$ we introduce the total derivative

$$\dot{u} := \dot{x}\, u_x + u_t = \dot{X}_i u_x + \mathcal{L}(u), \quad 1 \le i \le N, \tag{2.3}$$

and spatially discretize the space operators $\partial/\partial x$ and L so as to obtain the Lagrangian semi-discrete system

$$\dot{U}_i = \dot{X}_i[(U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1})] + L_i, \quad t > 0, \; 1 \le i \le N. \tag{2.4}$$

Here, U_i and L_i represent the semi-discrete approximation to their exact counterparts u and $\mathcal{L}$ at the point $(x,t) = (X_i(t),t)$. The finite-difference replacement for $\mathcal{L}$ is, in principle, still free to choose. We discuss this in Section 6.2.5. Note that we use the standard, central finite-difference approximation for u_x. Also note that the boundary values U_0 and U_{N+1} are to be defined from the semi-discretization of the physical boundary conditions. The internal grid points X_i are still free to choose. The purpose is to let them move automatically such that the grid X becomes fine in regions of high spatial activity and coarse in regions where the spatial variation is low. One way to accomplish this is to apply equidistribution. For this purpose we introduce the point-concentration values [6]

$$n_i = (\Delta X_i)^{-1}, \quad \Delta X_i = X_{i+1} - X_i, \quad 0 \le i \le N, \tag{2.5}$$

and the equidistribution equation

$$n_{i-1}/M_{i-1} = n_i/M_i, \quad 1 \le i \le N, \tag{2.6}$$

where $M_i \ge \sqrt{\alpha} > 0$ represents a so-called monitor value that reflects the variation in space. The parameter $\alpha > 0$ serves to ensure that M_i remains positive. Trivially, n_i is proportional with M_i. Thus the equidistribution idea assumes that if some measure of the spatial error is available, here taken to be represented by M_i, then a good choice for the grid X would be one for which the error is equidistributed over X.

In applications the monitor M_i is usually taken to be a semi-discrete replacement of a solution functional $m(u)$ containing one or more spatial derivatives (note that the variables U_i and X_i are still time continuous). Lest we miss the obvious, the choice of monitor is important because it plays a decisive role in the actual local grid refinement. Following [2, 8, 13], in the present implicit MOL approach we advocate the first derivative monitor

$$M_i = (\alpha + \frac{1}{NPDE} \sum_{j=1}^{NPDE} \beta_j (\Delta U_i^j)^2 (\Delta X_i)^{-2})^{1/2}, \quad \Delta U_i^j = U_{i+1}^j - U_i^j, \tag{2.7}$$

where $NPDE$ denotes the number of PDEs in (2.1) and U_i^j is the j-th component of the vector variable U_i. Note that, at a given point of time, (2.7) is a semi-discrete replacement of $m(u) = (\alpha + \|u_x\|^2)^{1/2}$, where $\|.\|$ is the involved weighted Euclidean norm. With $\alpha = 1$ we have the well-known arc-length monitor which places grid points along uniform arc-length intervals. We use α as a parameter which can eventually be used for tuning purposes. In fact, the main purpose of this tuning parameter is to keep the monitor values positive, saying that a small value of α suffices. Clearly, α should not be taken too 'large' compared to the maximum of $\|u_x\|^2$, since this would result in a uniform grid, approximately. The weighting parameters β_i in (2.7) serve to make it possible to let certain components dominate the grid movement. This may be desirable in case of a badly scaled problem, for example. The actual choice of the monitor parameters α, $\beta_1, \ldots, \beta_{NPDE}$ will influence the outcome of a numerical simulation and, therefore, their optimal choice is problem dependent. On the other hand, our experience is that with the monitor (2.7) the method is quite robust and a bad choice merely effects the resulting accuracy. This means that given a well described problem class, like the brine transport problems, a close-to-optimal choice is normally easy to determine.

6.2.2. Grid smoothing

The Cauchy ODE problem resulting from the first MOL stage thus reads

$$\dot{U}_i = \dot{X}_i[(U_{i+1} - U_{i-1})/(X_{i+1} - X_{i-1})] + L_i, \quad t>0,\ 1 \le i \le N, \tag{2.8a}$$

$$n_{i-1}/M_{i-1} = n_i/M_i, \quad t > 0,\ 1 \le i \le N. \tag{2.8b}$$

After prescribing initial data for U_i and X_i, $1 \le i \le N$, and the boundary values U_0 and U_{N+1} from a semi-discretization of the physical boundary conditions, system (2.8) can be numerically integrated in time so as to obtain the final fully discrete solution on the moving grid X. However, since (2.8b) prescribes X in an implicit way in terms of the unknowns U_i, there is little control over the grid movement. For example, it may happen that the grid distance ΔX_i varies extremely rapidly over X or

that for evolving time the trajectories $X_i(t)$ tend to oscillate. A too large variation in ΔX_i may be detrimental to spatial accuracy and temporal grid oscillations do hinder the numerical time-stepping since the grid trajectories are computed automatically by numerical integration. Following [6,8,13], we therefore employ two so-called grid-smoothing procedures, one for generating a spatially smooth grid and the other for avoiding temporal grid oscillations. This involves a modification of the grid-equation system (2.8b).

The modified grid-equation system is given by

$$(\tilde{n}_{i-1} + \tau \frac{d}{dt}\tilde{n}_{i-1})/M_{i-1} = (\tilde{n}_i + \tau \frac{d}{dt}\tilde{n}_i)/M_i, \quad t>0,\ 1 \le i \le N, \tag{2.9}$$

where $\tilde{n}_i = n_i - \kappa(\kappa+1)(n_{i+1} - 2n_i + n_{i-1})$ with $n_{-1} = n_0$, $n_{N+1} = n_N$. We note in passing that in the actual implementation n_i is replaced by $(\Delta X_i)^{-1}$ and $\dot{n}_i$ by $-\dot{\Delta X}_i/(\Delta X_i)^2$. The modification thus results in a 5-point coupled, time-dependent grid-equation system. A consequence of the grid-smoothing is that, in addition to the monitor parameters α, $\beta_1, \ldots, \beta_{NPDE}$, two new grid parameters have been introduced, namely κ and τ.

The parameter $\kappa > 0$ is connected with the spatial grid-smoothing. Any grid X solving (2.9) satisfies

$$\frac{\kappa}{\kappa+1} \le \frac{n_{i-1}}{n_i} \le \frac{\kappa+1}{\kappa}, \tag{2.10}$$

showing that we have control over the variation in ΔX_i. Through κ we can control grid clustering and grid expansion. Loosely speaking, the monitor function still determines the shape of X and κ the level of clustering. Note that the extreme value $\kappa = \infty$ yields a uniform grid. Of importance is to emphasize that for a given number of points N, and any given distribution of monitor function values M_i , κ determines the minimal and maximal interval lengths (see, e.g., [13]). In actual application the minimum should of course be related to the expected small scale features in the sought solution. In our application we choose $\kappa = 2$. With this value of κ we not only obtain a rather modestly graded space grid, but also keep a sufficient number of points within the actual transitions of $\partial u/\partial x$.

The parameter $\tau \ge 0$ is connected with the temporal grid-smoothing and serves to act as a delay factor for the grid movement. More precisely, the introduction of the temporal derivative of the grid X forces the grid to adjust over a time interval of length τ from old to new monitor values, which provides a tool for suppressing grid oscillations and hence to obtain a smoother progression of $X(t)$. However, choosing τ too large will result in a grid X that lags too far behind any moving steep spatial transition. In fact, it can be shown that for $\tau \to \infty$ a nonmoving grid results. In situations where temporal grid-smoothing is really advisable, one should therefore choose τ not too large. For practical purposes a good choice is one which is close to the minimal temporal stepsize taken in the numerical integration, so that the influence of past monitor values is felt only over one or a few time steps.

6.2.3. Integration in time

We now have semi-discretized (2.1) on a moving grid. The semi-discrete formulation consists of the combined equations (2.8a) and (2.9), where $n_i = (\Delta X_i)^{-1}$ and $\dot{n}_i = -\Delta\dot{X}_i/(\Delta X_i)^2$ are used to convert the dependence on the point concentration values into a 'natural' dependence on the grid points X_i. Recall that the boundary values U_0 and U_{N+1} are to be defined from the spatial discretization of physical boundary conditions. The equations can be written in the linearly implicit ODE system form

$$\mathcal{A}(Y)\dot{Y} = G(Y), \quad t > 0, \; Y(0) \text{ given}, \tag{2.11}$$

where Y assumes the natural ordering of unknowns U_i^j, X_i, i.e., $Y = (\cdots, U_i^1, \cdots, U_i^{NPDE}, X_i, \ldots)^T$. Form (2.11) is a standard format for various well-known stiff ODE/DAE solvers. Note that without temporal grid-smoothing (2.9) is of purely algebraic form, so that (2.11) then becomes a DAE system. The numerical results in this paper have been obtained with the LSODI-based BDF solver of the SPRINT package. A similar solver is DASSL [4] which we have also applied elsewhere [13]. It is of interest to note that in our moving-grid application these solvers are employed in essentially the same way as in the conventional nonmoving MOL approach.

6.2.4. A moving-grid interface

As the integration in time is done automatically by the stiff integrator, it makes sense to also automize the spatial discretization of the PDE operator with its boundary conditions. This is particularly attractive for researchers who wish to concentrate on modelling their physics, since it saves programming time and relieves them from numerical choices to be made. Such a FORTRAN interface for use with the moving-grid method has been developed in [3]. We have also used this interface, called MGI, in the tests reported in Section 6.4.

MGI is an extension of the fixed-grid interface from [12] which is available in the SPRINT package and covers the following PDE system:

$$\sum_{k=1}^{NPDE} C_{jk}(x,t,u,u_x)\frac{\partial u^k}{\partial t} = x^{-m}\frac{\partial}{\partial x}(x^m R_j(x,t,u,u_x)) - Q_j(x,t,u,u_x),$$

$$x_L < x < x_R, \; t>0. \tag{2.12}$$

Index j runs from 1 to *NPDE*, u_k is the k-th component of the vector-valued function u, and R_j, Q_j can be thought of as flux and source or sink terms, respectively. The parameter m serves to cover polar co-ordinates ($m = 1$ or 2). In our present application we work in Cartesian co-ordinates and thus $m = 0$. The coefficient functions C_{jk}, R_j, Q_j are supposed to be at least continuous. The boundary conditions should fit in the MGI master form

$$\chi_j(x,t)\, R_j(x,t,u,u_x) = \gamma_j(x,t,u,u_t,u_x), \quad \text{at } x=x_L, \; x_R, \tag{2.13}$$

and the standard initial condition $u(x, 0) = u_0$ is supposed. The underlying spatial discretization of MGI is briefly discussed in Section 6.2.5. For a more detailed description of the use of MGI we refer to [3].

6.2.5. The spatial discretization in MGI

In order to reduce accuracy problems that arise for coefficients like x^{-m} in (2.12) when x is near zero and $m>0$, a spatial discretization method is used which is second order in space. The nonlinear Galerkin-based method is extensively described in Skeel and Berzins [12]. In the following we give a summary of this discretization method when applied to the PDE class (2.12) transformed to its Lagrangian form. We omit, however, the error analysis which can be found in Skeel and Berzins.

First we apply the Lagrangian transformation.

Let w be defined by $w := u_x\dot{x}$ and S_j by

$$S_j = S_j(x,t,u,u_x,\dot{u},w) := \sum_{k=1}^{NPDE} C_{j,k}(\dot{u}^k - w^k) + Q_j.$$

Then system (2.12) becomes

$$S_j = x^{-m}(x^m R_j)_x \quad \text{for } j=1,...,NPDE, \tag{2.14}$$

with $C_{j,k}$, Q_j, and R_j defined as before. On the spatial grid (2.2) we will formulate the Galerkin method for (2.14). Introduce the approximation U^k of u^k

$$U^k(x) = \sum_{i=0}^{N+1} U_i^k \phi_i^{(m)}(x).$$

Let $\psi_i^{(m)}$ denote the test functions. The trial functions $\phi_i^{(m)}$ and the test functions $\psi_i^{(m)}$ are given in the Appendix. Introduce the weight function x^m and integrate (2.14) on $[x_L, x_R]$ partially, so as to obtain

$$\int_{x_L}^{x_R} x^m \psi_i^{(m)} S_j dx = x_R^m \psi_i^{(m)}(x_R) R_j \,|_{x=x_R} - x_L^m \psi_i^{(m)}(x_L) R_j \,|_{x=x_L} - \tag{2.15}$$

$$\int_{x_L}^{x_R} x^m \frac{d\psi_i^{(m)}}{dx} R_j dx$$

for $j=1,...,NPDE$ and $i=0,...,N+1$.

Using the fact that $\psi_i^{(m)}(x)=0$ for $x \le X_{i-1}$ and $x \ge X_{i+1}$ we get for $i=1,...,N$ and $j=1,...,NPDE$

$$\int_{X_{i-1}}^{X_{i+1}} x^m \psi_i^{(m)} S_j dx = -\int_{X_{i-1}}^{X_{i+1}} x^m \frac{d\psi_i^{(m)}}{dx} R_j dx \tag{2.16}$$

The integration over an interval $[X_{i-1}, X_i]$ is performed by numerical quadrature using 1 quadrature point $\xi_{i-1/2}$. After applying the numerical integration on $[X_{i-1}, X_i]$ and $[X_i, X_{i+1}]$ and lumping (that is evaluation of $\dot{u}^k$ takes place in X_i rather than in ξ) (2.16) yields

$$S_j^{i-1/2} \int_{X_{i-1}}^{X_i} x^m \psi_i^{(m)} dx + S_j^{i+1/2} \int_{X_i}^{X_{i+1}} x^m \psi_i^{(m)} dx =$$

$$- \xi_{i-1/2}^{\mu} R_j^{i-1/2} \int_{X_{i-1}}^{X_i} x^{m-\mu} \frac{d\psi_i^{(m)}}{dx} dx - \xi_{i+1/2}^{\mu} R_j^{i+1/2} \int_{X_i}^{X_{i+1}} x^{m-\mu} \frac{d\psi_i^{(m)}}{dx} dx + E, \quad (2.17)$$

where

$$\begin{aligned} S_j^{i-1/2} &= S_j(\xi_{i-1/2}, t, U(\xi_{i-1/2}), U_x(\xi_{i-1/2}), \dot{U}_i, W_i), \\ S_j^{i+1/2} &= S_j(\xi_{i+1/2}, t, U(\xi_{i+1/2}), U_x(\xi_{i+1/2}), \dot{U}_i, W_i), \\ R_j^{i+1/2} &= R_j(\xi_{i+1/2}, t, U(\xi_{i+1/2}), U_x(\xi_{i+1/2})), \end{aligned} \quad \text{with } W_i = \frac{U_{i+1} - U_{i-1}}{X_{i+1} - X_{i-1}} \dot{X}_i,$$

and E stands for the total error due to interpolation, quadrature and lumping. For the definition of μ we make a distinction between two special cases
i) the regular case ($m = 0$ or $x_L > 0$): $\mu = m$
ii) the singular case ($m > 0$ and $x_L = 0$): $\mu = -1$.
The choice of ξ depends on μ. On an interval $[X_i, X_{i+1}]$ we choose $\xi_{i+1/2} = \gamma_{-\mu, i+1/2}$, where $\gamma_{p,i+1/2}$ denotes the Gauß-point for the weight function x^p, i.e.,

$$\int_{X_i}^{X_{i+1}} (x - \gamma_{p,i+1/2})\, x^p dx = 0. \quad (2.18)$$

The numerical integration in (2.17) is then second order accurate. If we neglect the error E in (2.17), we arrive at a semi-discrete approximation of (2.14), for $i = 1,...,N$

$$\begin{aligned} &S_j^{i-1/2} \frac{X_i^{m+1} - \zeta_{i-1/2}^{m+1}}{m+1} + S_j^{i+1/2} \frac{\zeta_{i+1/2}^{m+1} - X_i^{m+1}}{m+1} = \\ &\zeta_{i+1/2}^{m-\mu}\, \xi_{i+1/2}^{\mu}\, R_j^{i+1/2} - \zeta_{i-1/2}^{m-\mu}\, \xi_{i-1/2}^{\mu}\, R_j^{i-1/2}, \end{aligned} \quad (2.19)$$

with

$$\zeta_{i+1/2}^{m+1} = - \int_{X_i}^{X_{i+1}} x^{m+1} \frac{d\psi_i^{(m)}}{dx} dx, \quad \text{and} \quad \zeta_{i+1/2}^{0} = 1.$$

For a list of the test functions $\psi_i^{(m)}$, the trial functions $\phi_i^{(m)}$, the quadrature points $\xi_{i+1/2}$ and the integrals $\zeta_{i+1/2}^{m+1}$ see appendix A. In [12] a justification is given for all choices of the parameters and functions. There it is shown that the spatial discretization method is second order accurate, both in the regular and the singular case.

The right boundary equation in (2.13) $\chi_j(x_R, t)\, R_j|_{x=x_R} = \gamma_j|_{x=x_R}$ is combined with the semi-discrete approximation of (2.15) with $i=N+1$

$$S_j^{(N+1)-1/2} \frac{x_R^{m+1} - \zeta_{N+1/2}^{m+1}}{m+1} + \zeta_{N+1/2}^{m-\mu}\, \xi_{N+1/2}^{\mu}\, R_j^{N+1/2} = x_R^m\, R_j|_{x=x_R} \quad (2.20)$$

to eliminate $R_j|_{x=x_R}$.

In the regular case the same procedure is applied to the left boundary equation in (2.13) $\chi_j(x_L, t)\, R_j|_{x=x_L} = \gamma_j|_{x=x_L}$ and

$$S_j^{0+1/2} \frac{\zeta_{1/2}^{m+1} - x_L^{m+1}}{m+1} - \xi_{1/2}^{m} R_j^{1/2} = - x_L^m\, R_j|_{x=x_L}, \quad (2.21)$$

from which we can eliminate $R_j|_{x=x_L}$. If x_L and m are both zero we take x_L^m=1. In

the singular case, however, we use just the semi-discrete approximation of (2.15) with $i=0$ which gives

$$S_j^{0+\frac{1}{2}}/(m+1) - \xi_{\frac{1}{2}}^{-1} R_j^{\frac{1}{2}} = 0. \tag{2.22}$$

This means, that for $X_1 \to x_L$ the boundary equation tends to $R_j|_{x=x_L}=0$, which is a natural constraint for polar problems.

Equations (2.19) and (2.9) are combined to the now fully semi-discretized system (2.11).

6.3. The 1D fluid-flow/ salt-transport problem

Disposal of radioactive wastes in rock salt formations is being considered as a serious possibility by a number of countries. An integral part of the safety assessment of waste disposal is the study of mathematical models for nuclide transport to the geosphere via groundwater flow. Existing standard models for groundwater flow and salt transport assume that the salt concentration is less than or equal to seawater concentration. In such low-concentration situations the models in use have been sufficiently validated and in many cases of interest the fluid flow and the salt concentration equation can be treated uncoupled. However, for flows in the vicinity of rock salt formations the salt concentration may become high and influence the fluid density to an extent that it effects the fluid flow. On the other hand, salt is transported by the fluid and thus fluid flow and salt transport are mutually coupled. The existing standard models and their uncoupled treatment are then no longer adequate for safety assessment which makes it interesting to study this intricate situation. Recent theoretical and experimental hydrological studies [9,10] indicate that for such high-concentration situations the involved basic equations of flow and transport need to be modified, which requires a significant effort in numerical modelling. Here the moving-grid method enters the scene, because in the high-concentration situations also large concentration gradients prevail, making the use of fixed-grid methods inefficient.

In the modelling of transport of M solutes by groundwater flow generally $M + 1$ sets of equations appear, viz., one set for each solute and a set for the flowing fluid [9]. The set for the fluid (brine) constitutes the fundamental balance of mass property of the fluid supplemented with a Darcy-law expressing conservation of momentum. Similarly, for each solute the associated set constitutes the balance of mass property supplemented with conservation of momentum through a Fickian-type law. If temperature changes are important, then an energy equation should be added. Also, if deformation effects of the porous medium and porosity changes are important, then an additional set of equations for the solid phase of the porous medium has to be provided. In the present study we do not consider temperature or deformation effects and assume only one solute, the salt. We thus consider an isothermal, single-phase, two-component saturated flow model in the idealized case of one space dimension. It is further assumed that no external body forces except gravity exist and that the two brine components, water and salt, do not react or adsorb. This specific model, which we have borrowed from the RIVM report [7], has been selected for demonstration purposes.

The model comprises the following set of equations. For the fluid and salt we have, respectively,

$$\frac{\partial}{\partial t}(n\rho) + \frac{\partial}{\partial x}(\rho q) = 0, \qquad q = -\frac{k}{\mu}(\frac{\partial p}{\partial x} + \rho g), \tag{3.1}$$

$$\frac{\partial}{\partial t}(n\rho\omega) + \frac{\partial}{\partial x}(\rho\omega q + \rho J) = 0, \qquad J = -\lambda |q| \frac{\partial \omega}{\partial x}. \tag{3.2}$$

The fluid density ρ is supposed to obey the equation of state

$$\rho = \rho_0 \, exp\,(\beta(p - p_0) + \gamma\omega), \tag{3.3}$$

with constant reference density ρ_0, constant reference pressure p_0, constant compressibility coefficient β, and constant salt coefficient γ. Other constants are porosity n, permeability k, viscosity μ, gravity g and dispersion length λ. The various variables are the (Darcy) velocity of the fluid q, the hydrodynamic pressure p and the salt-mass fraction ω. We thus consider the medium to be homogeneous with respect to porosity, permeability and viscosity. However, inhomogeneities, and also sources and sinks, can easily be taken into account.

The set of equations can be formulated as a system of two PDEs with pressure p and salt concentration ω as independent variables. To this end we compute, from (3.3),

$$\frac{\partial \rho}{\partial t} = \rho\beta \frac{\partial p}{\partial t} + \rho\gamma \frac{\partial \omega}{\partial t} \tag{3.4}$$

and subsitute into (3.1) to obtain the fluid-mass balance equation

$$n\rho\beta \frac{\partial p}{\partial t} + n\rho\gamma \frac{\partial \omega}{\partial t} = -\frac{\partial}{\partial x}(\rho q). \tag{3.5}$$

A further substitution yields the salt transport equation

$$n\rho \frac{\partial \omega}{\partial t} = -\rho q \frac{\partial \omega}{\partial x} - \frac{\partial}{\partial x}(\rho J). \tag{3.6}$$

We have used this form as input for the numerical solution method. A few comments are in order. First, substitution of the expression for J into (3.6) yields the advection-dispersion equation

$$n\rho \frac{\partial \omega}{\partial t} = -\rho q \frac{\partial \omega}{\partial x} + \frac{\partial}{\partial x}(\rho\lambda |q| \frac{\partial \omega}{\partial x}), \tag{3.7}$$

showing that in the present model the physical salt-transport phenomena are advection and dispersion. Molecular diffusion is absent here. It is easily built in, however, since this merely amounts to adding a small constant to $\lambda |q|$. Assuming 'frozen' coefficients, we see that the Peclet number is

$$Pe = |\frac{L\rho q}{\rho\lambda |q|}| = \frac{L}{\lambda}, \tag{3.8}$$

where L denotes the physical length of the medium. Hence, for $\lambda \ll L$ advection dominates and this is just the physical situation that gives rise to steep concentration gradients. Another point worth to mention is that the compressibility coefficient β is

very small compared to the salt coefficient γ. In fact, it is often zero, in which case the balance equation (3.5) reduces to

$$n\rho\gamma\frac{\partial\omega}{\partial t} = -\frac{\partial}{\partial x}(\rho q) \tag{3.9}$$

and $\partial p/\partial t$ is absent. We then have two equations for $\partial\omega/\partial t$ of which (3.5) can be rewritten to a PDE containing only spatial derivatives. Hence this rules out the possibility of explicit time-stepping. Note that if we would also put $\gamma = 0$, that then the density ρ is constant and the mass balance equation reduces to the simple pressure equation $p_{xx} = 0$. Of course, a zero salt coefficient γ is not realistic in our application.

	Non-scaled	Scaled
Time	$0 < \mathbf{t} < \mathbf{T}$ [*sec*]	$t = \mathbf{t}/\mathbf{t_0}$, $\mathbf{t_0} = \boldsymbol{\mu}\mathbf{L}^2/(\mathbf{k_0}\mathbf{p_0}) = 10^4$
Space	$0 < \mathbf{x} < \mathbf{L}$ [*m*]	$x = \mathbf{x}/\mathbf{L}$
End time	$\mathbf{T}$ [*sec*]	$T = \mathbf{T}/\mathbf{t_0}$
Domain length	$\mathbf{L} = 1$ [*m*]	$L = 1$
Pressure	$\mathbf{p}$ [$kg/m/sec^2$]	$p = \mathbf{p}/\mathbf{p_0}$
Salt concentration	$\boldsymbol{\omega}$	$\omega = \boldsymbol{\omega}/\boldsymbol{\omega_0}$
Density	$\boldsymbol{\rho}$ [kg/m^3]	$\rho = \boldsymbol{\rho}/\boldsymbol{\rho_0}$
Permeability	$\mathbf{k} = \mathbf{k_0} = 10^{-12}$ [m^2]	$k = \mathbf{k}/\mathbf{k_0} = 1$
Viscosity	$\boldsymbol{\mu} = \boldsymbol{\mu_0} = 10^{-3}$ [$kg/m/sec$]	$\mu = \boldsymbol{\mu}/\boldsymbol{\mu_0} = 1$
Reference pressure	$\mathbf{p_0} = 10^5$ [$kg/m/sec^2$]	$p_0 = 1$
Salt inflow concentration	$\boldsymbol{\omega_0} = 0.26$	$\omega_0 = 1$
Reference density	$\boldsymbol{\rho_0} = 10^3$ [kg/m^3]	$\rho_0 = 1$
Gravity force	$\mathbf{g} = 9.81$ [m/sec^2]	$g = (\boldsymbol{\rho_0}\mathbf{L}\mathbf{g})\mathbf{p_0} = 0.0981$
Porosity	$\mathbf{n} = 0.2$	$n = 0.2$
Salt coefficient	$\boldsymbol{\gamma} = 0.69$	$\gamma = \boldsymbol{\gamma}\boldsymbol{\omega_0} = 0.1794$
Dispersion length	$\boldsymbol{\lambda}$ [*m*]	$\lambda = \boldsymbol{\lambda}/\mathbf{L}$
Compressibility coefficient	$\boldsymbol{\beta} = 10^{-10}$ [$msec^2/kg$]	$\beta = \boldsymbol{\beta}\mathbf{p_0} = 10^{-5}$

TABLE 6.1. Model data. Bold face notation is used for the non-scaled quantities.

To complete the model description, we must give the initial and boundary conditions. Defining the space-time domain as $[0,L]\times[0,T]$, the initial and boundary conditions we have imposed for $\omega(x,t)$ and $p(x,t)$ are, respectively,

$$\omega(x,0) = 0, \quad 0 \le x \le L, \tag{3.10a}$$

$$\omega(0,t) = \omega_0 > 0, \quad \frac{\partial\omega}{\partial x}(1,t) = 0, \quad 0 < t \le T, \tag{3.10b}$$

and

$$p(x,0) = p_0[(1-x/L)p_{left} + (x/L)p_{right}], \quad 0 \le x \le L, \tag{3.11a}$$

$$p(0,t) = p_0 p_{left}, \quad p(L,t) = p_0 p_{right}, \quad 0 < t \le T, \tag{3.11b}$$

where ω_0 is the left-end salt concentration and p_{left} and p_{right} are pressure coefficients.

We have selected these conditions with the aim of generating a travelling salt front. Note that at $t = 0$ there is no salt in the medium and that the inflow value $\omega_0 > 0$. Hence, assuming appropriate model data, this should give rise to a travelling front. The steepness and speed of the front will of course be determined by the complete set of physical data. A characteristic set is given in Table 6.1 which comprises all data needed to run the problem, except for the end time T, the dispersion length λ, and the pressure coefficients p_{left} and p_{right}. Finally, numerically we have treated the problem in scaled, dimensionless form. We refer to Table 6.1 for the scaling relations with the dimensionless values of all quantities involved. From these relations one can check that all equations are left invariant (note that this also holds for (3.3) due to the fact that after scaling $p_0 = 1$). Hence, in the remainder we have worked with the same set of equations as discussed above. The pressure coefficients p_{left} and p_{right} are left unchanged and will be specified with the numerical examples.

6.4. Numerical examples

We will present results of three numerical examples. To simplify the demonstration, these results have been obtained with a fixed set of numerical control parameters:

$$\text{TOL} = 10^{-5} \quad \text{(temporal integration)} \tag{4.1a}$$

$$\kappa = 2,\ \tau = 10^{-3},\ \alpha = 10^{-2},\ \beta_1 = 0,\ \beta_2 = 1 \quad \text{(grid movement)} \tag{4.1b}$$

$$X_i(0) = \frac{i}{N+1}, \quad 0 \le i \le N+1 \quad \text{(uniform initial grid)} \tag{4.1c}$$

SPRINT was called in standard mode, thus providing automatically an initial stepsize and Jacobian evaluation. The Euclidean norm was used for local error control while (4.1a) was imposed for all components of the vector Y (cf. (2.11); $NPDE=2$, $u^1 = p$, $u^2 = \omega$). Note that TOL $= 10^{-5}$ is quite small. However, to accurately simulate the rapid birth of the salt front, which arises from the inconsistency between the initial and left-end salt concentration, a small tolerance value is natural. We also emphasize that we always started on a uniform grid, just for convenience of use. This means that immediately after start the method should rapidly cluster most of the grid points near the left boundary.

The grid parameters take on more or less standard values, except for β_1. The choice $\beta_1 = 0$ means that the pressure gradient $\partial p/\partial x$ is not taken into account in the monitor (2.7). We decided to omit $\partial p/\partial x$ in the monitor since in our examples $\partial p/\partial x$ varies very slowly and thus acts more or less in the same manner as the constant regularization parameter α. In such cases a too large value for the near constant pressure gradient yields a unnecessarily large regularization effect. This, in turn, would imply that variations in the concentration gradient $\partial\omega/\partial x$ become of

lesser importance in the spatial equidistribution than desired.

6.4.1. Example I

The first example is defined by the data of Table 6.1, together with $\lambda = 1.E{-}3$, $T = 5$ and $p_{left} = 1.7$, $p_{right} = 1.0$. With this choice of pressure initial function the arising salt front travels to the right boundary and finally renders a steady state for p and ω with p equal to the linear initial pressure and $\omega = \omega_0 = 1$. The steady state starts to settle at about $t = 2$, far before the end time $T = 5$ has been reached. Consequently, due to the uniform salt concentration, at about $t = 2$ the grid should again become uniform. Hence this example provides an interesting test for the moving-grid method. The pressure p undergoes only a marginal change for $t > 0$ and below we will therefore only plot ω.

Figure 6.1 depicts the grid and salt concentrations at some values of t for $N = 25$ and 50. We see that the grid accurately reflects the anticipated solution behaviour. At very early times the grid points rapidly cluster near $x = 0$, then the cluster travels with the front and when the steady state is reached, a uniform grid appears. While $N = 25$ results in a little overshoot at the top and in a little smearing at the foot, $N = 50$ gives already very accurate salt concentration profiles. The profiles for $N = 100$ (not shown here) do equal those for $N = 50$ up to plotting accuracy.

Table 6.2 shows integration history for $N = 25$, 50 and 100 and serves to provide insight in the costs of the implicit numerical integration method. The given data have the following meaning: STEPS = number of integration steps; JACS = number of Jacobian updates; RESIDS = total number of evaluations of the ODE system, including those needed for the Jacobian updates; NITER = total number of Newton iterations; CPU = central processing time on an ALLIANT/FX4 computer, using one processor. Note that our decision to start on a uniform initial grid has its price. For example, for $N = 100$ more than half the number of steps is used to reach $t = 0.1$. In fact, at $t = 10^{-4}$, 10^{-3}, 10^{-2}, we have, respectively, STEPS = 39, 152, 271. A great deal of these steps is needed simply to adjust the initial grid to the very steep concentration profile at the very early times (see the right upper plot in Figure 6.1). Therefore, somehow adjusting the initial grid to the expected solution profile at the first forward time level will reduce STEPS significantly. We also wish to remark that the method efficiently detects the steady state, since for $t > 2$ the temporal step-sizes are rapidly increased and very few steps are required to complete the integration. Finally, we have also tabulated $\omega_{\max} - \omega_0$, which is the maximal overshoot at the given points of time. We see that already for $N = 25$ the overshoot is very little.

A further inspection of the salt concentration plots shows that, as expected, the first derivative monitor (2.7) places quite a number of points just within the front where $\partial\omega/\partial x$ is largest. Fortunately, the spatial grid-smoothing, resulting in relation (2.10), has the nice side-effect of keeping a substantial number of points at the foot and top of the front, where $\partial\omega/\partial x$ becomes smaller and finally zero. This only works, of course, if κ is taken not too small. Note that there should be enough points at the foot and top so as to avoid wiggles, since the spatial discretization is based on a common central finite-volume scheme. For a comparison of results obtained with a second-derivative monitor based on $m(u) = (\alpha + \|u_{xx}\|^2)^{1/4}$ and with a fixed grid, the

reader is referred to [14]. There it is concluded that the moving-grid approach with the first-derivative monitor is to be preferred with respect to accuracy and efficiency.

6.4.2. Example II

The second example is also defined by the data of Table 6.1, but now $p_{left} = 1.11$, $p_{right} = 1.0$, $T = 500$ and $\lambda = 1.E{-}4$. The smaller pressure gradient in the initial function has two effects. First, it yields a smaller fluid velocity resulting in a larger time scale, which explains the larger value for T. The second and more interesting effect is that the travelling salt front now comes to a stand still before it has reached the right boundary. This happens at about $t = 150$, at which point of time the front lies near $x = .6$. The reason is that the fluid velocity q tends to zero, uniformly in x, which settles the system into a steady state and this takes place long before the salt front has reached the right boundary. We note that this phenomenon is rather special in the sense that it heavily depends on the initial pressure gradient. The stand still of the salt front is lost with a relatively slight change in this gradient. Also note that this stand still requires a zero molecular diffusion which in reality is not true, of course. However, the simulation of this rather subtle situation provides a nice numerical test as it requires an accurate balancing of gravity force ρg and pressure gradient force $\partial p/\partial x$ in the Darcy velocity expression in (3.1). Finally, we have made the dispersion length ten times smaller than in the previous example, giving a Peclet number of $1.E{+}4$ and a much steeper front (recall that the spatial discretization of the MGI is based on a common central finite-volume scheme). Figure 6.2 shows the computed grid and salt concentration profiles at some values of time for $N = 25$ and 50. Like in the previous example, we see that the grid movement accurately reflects the anticipated solution behaviour. For early times it is completely similar, while for later times the cluster around the steep salt front remains in position. We also see that $N = 25$ now results in more overshoot, due to the fact that the dispersion length is ten times smaller than in the previous example. However, $N = 50$ again gives a very accurate solution and the profiles for $N = 100$ (not shown here) do equal those for $N = 50$ up to plotting accuracy.

Table 6.3 contains part of the integration history for $N = 25$, 50 and 100, providing the same information as before. With this table we wish to call attention for an inherent model difficulty stemming from the absolute value function in the dispersion-flux expression $\rho J = -\rho\lambda |q| \omega_x$. In the table this difficulty manifests itself in the large number of time steps and Jacobian updates used over the 'near steady-state interval' [200, 500] for $N = 100$ (recall that the steady state starts to settle at about $t = 150$). While the code easily detects the numerical steady state solution with 25 and 50 points, which can be concluded from the few number of steps needed to integrate from $t = 200$ to $t = 500$, this is clearly not the case with 100 points. In fact, with 100 points this 'near steady-state part' of the integration interval requires 1038 - 423 = 615 integration steps and 707 - 105 = 602 Jacobian updates, which is rather extreme. What has happened here is that the iterative Newton algorithm repeatedly fails to converge, so that the strategy of the SPRINT code keeps the temporal stepsize down and keeps asking for new Jacobians.

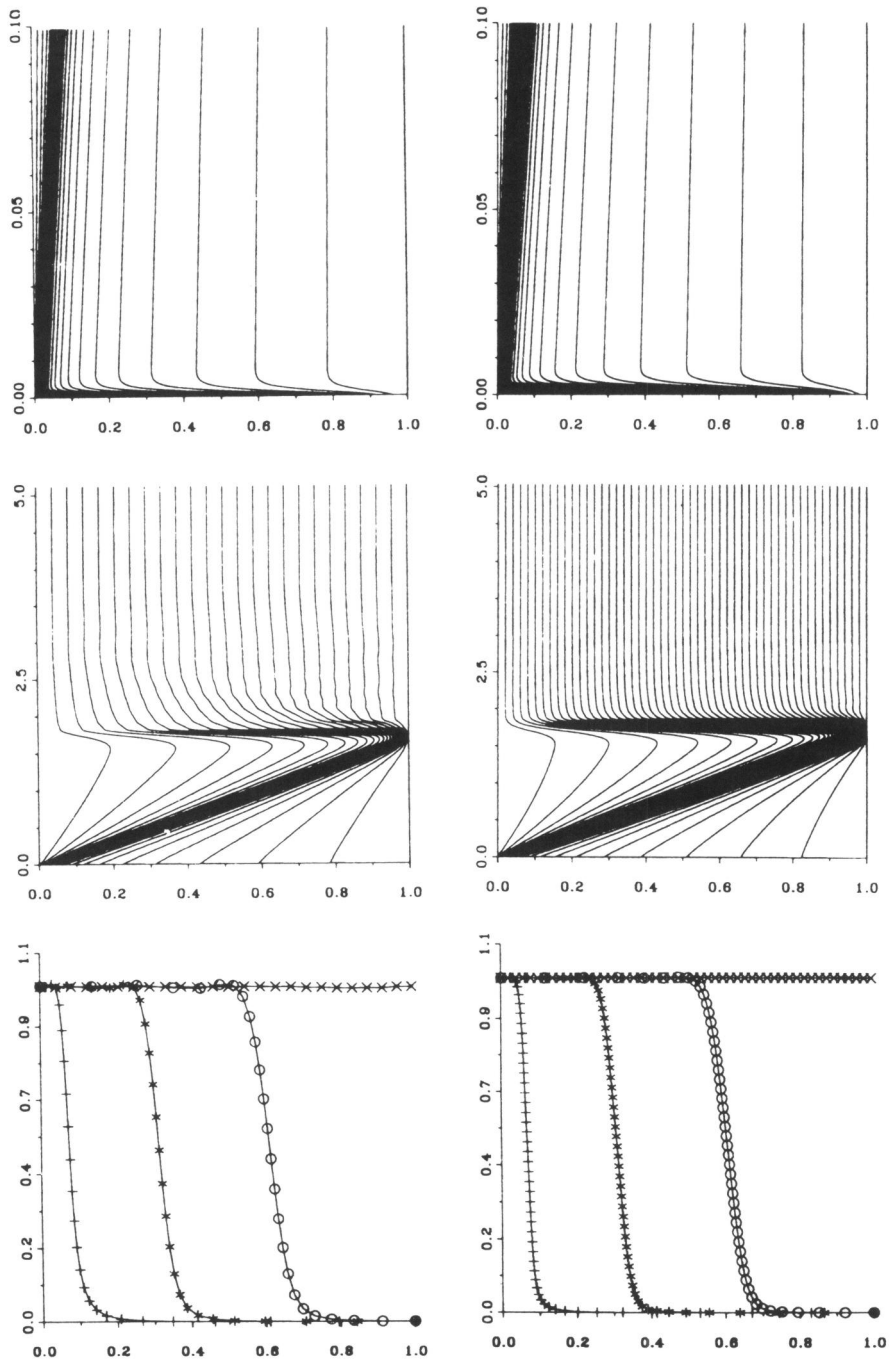


FIGURE 6.1. Example I: Gridlines and salt concentration profiles at $t = 0.1$, 0.5, 1.0, 5.0. The left part of the figure corresponds with $N = 25$ and the right part with $N = 50$. Note the difference in scaling in each of the two gridline plots.

		STEPS	JACS	RESIDS	NITER	OVERSHOOT	CPUtime (sec.)
N=25	t=0.1	149	39	931	407	7.0E-3	--
	t=0.5	198	47	1175	545	7.0E-3	--
	t=1.0	220	52	1302	607	6.0E-3	--
	t=5.0	565	144	3532	1610	--	140
N=50	t=0.1	202	53	1282	574	8.0E-4	--
	t=0.5	225	58	1413	638	1.0E-3	--
	t=1.0	234	61	1481	667	1.0E-3	--
	t=5.0	450	118	2904	1335	--	236
N=100	t=0.1	301	83	1988	882	3.0E-4	--
	t=0.5	317	87	2085	927	3.0E-4	--
	t=1.0	326	89	2140	956	6.0E-4	--
	t=5.0	529	142	3533	1648	--	566

TABLE 6.2. Example I: Integration histories.

The Newton convergence failure is caused by $|q|$ if $q \approx 0$. The following observations explain this. Due to the absolute value function, entries of the Jacobian matrix contain sign(q). Consequently, if $q \approx 0$, then during the Newton iteration approximate values for q readily change sign. Since the size of entries is large, as they contain terms $(\Delta X_i)^{-2}$, and ΔX_i can be very small, it happens that during the iteration process entries frequently change their value from large positive to large negative, or vice versa. No doubt this severely hinders the convergence of the iterative Newton process and, as we have observed, often will lead to convergence failures and requests for a Jacobian update. This explains why the march to steady state in the case of 100 points is so troublesome. However, we stipulate that also with 25 and 50 points the march to steady state eventually becomes troublesome. It all depends on the size of the computed velocities q and is a matter of accuracy. With lesser points the computed velocities arrive in the troublesome regime for larger values of time when the system has become sufficiently stationary or, in other words, when the numerical velocities have become sufficiently small. Ironically, with 100 points the accuracy is sufficiently good to have the troublesome Newton convergence behaviour already for $200 < t < 500$.

We emphasize that the troublesome march to steady state originates from the Jacobian matrix needed in the iterative solution process and not from the integration formula itself. In fact, we have also run the problem with $|q|$ replaced by $\sqrt{q^2 + \varepsilon}$ with $\varepsilon = 10^{-5}$, which completely remedies the situation and a normal march to steady state is observed with very large stepsizes towards the end value T, even up

to $T = 10^{12}$. When the modelling does not allow this slight modification in the dispersion-flux expression, an alternative remedy is to change the expression for $|q|$ only in the entries of the Jacobian, so as to avoid the sign changes. This involves a little change of the Jacobian matrix and thus should not interfere significantly with the convergence behaviour of the iterative Newton method.

6.4.3. Example III

This example is derived from Example I by changing the salt concentration value $\omega(0,t) = 1$ to the step function

$$\omega(0,t) = \begin{cases} 1, & 0 < t \leq 0.75, \\ 0, & 0.75 < t \leq 5.0. \end{cases} \tag{4.2}$$

Thus for $0 < t \leq 0.75$ the two solutions are equal and at $t = 0.75$ the step function generates a second front at $x = 0$ resulting in a block-form concentration profile. The block then travels to the right boundary and eventually the system runs into steady state with uniform zero salt concentration. For the moving-grid method this solution is more difficult to compute, since now two travelling fronts are present which appear and disappear at different values of t. Hence, instead of two times, four times the solution shape is drastically changed and the automatic grid movement and step-size control should be able to cope with these drastic changes. For example, without neglecting the already existing first front, at $t = 0.75$ the method must rapidly cluster grid points at the left boundary and decrease the time step to timely see the onset of the second front. Therefore, for the same accuracy, roughly twice the number of grid points and time-stepping effort will be needed as for Example I.

We have used $N = 25$, 50, 100. Apparently, 25 points is not enough, but with 50 points the solution is already fairly accurate. A comparison for 50 and 100 points reveals only minor differences at the top of the computed salt block profile and we may conclude that the results are very satisfactory. The gridline plot in Figure 6.3 for $N = 100$ nicely reveals the onset of the second front where very small time steps have been taken, similar as at $t = 0$ (see Figure 6.1). The arrival of the two fronts at the right boundary can also be clearly recovered from the plot, like the change to the uniform steady state grid. Note that also here small time steps are needed to accurately simulate the rapid solution change. The integration costs tabulated in Table 6.4 indeed show that the time-stepping effort is about twice as large as for Example I. As anticipated, comparison of Tables 6.2 and 6.4 reveals that the costs are mainly determined by the drastic changes in the solution shape. Once the front exists, the time stepping is done very efficiently, as can be deduced from the number of Jacobian updates listed in Table 6.2 at $t = 0.1$ and 1.0.

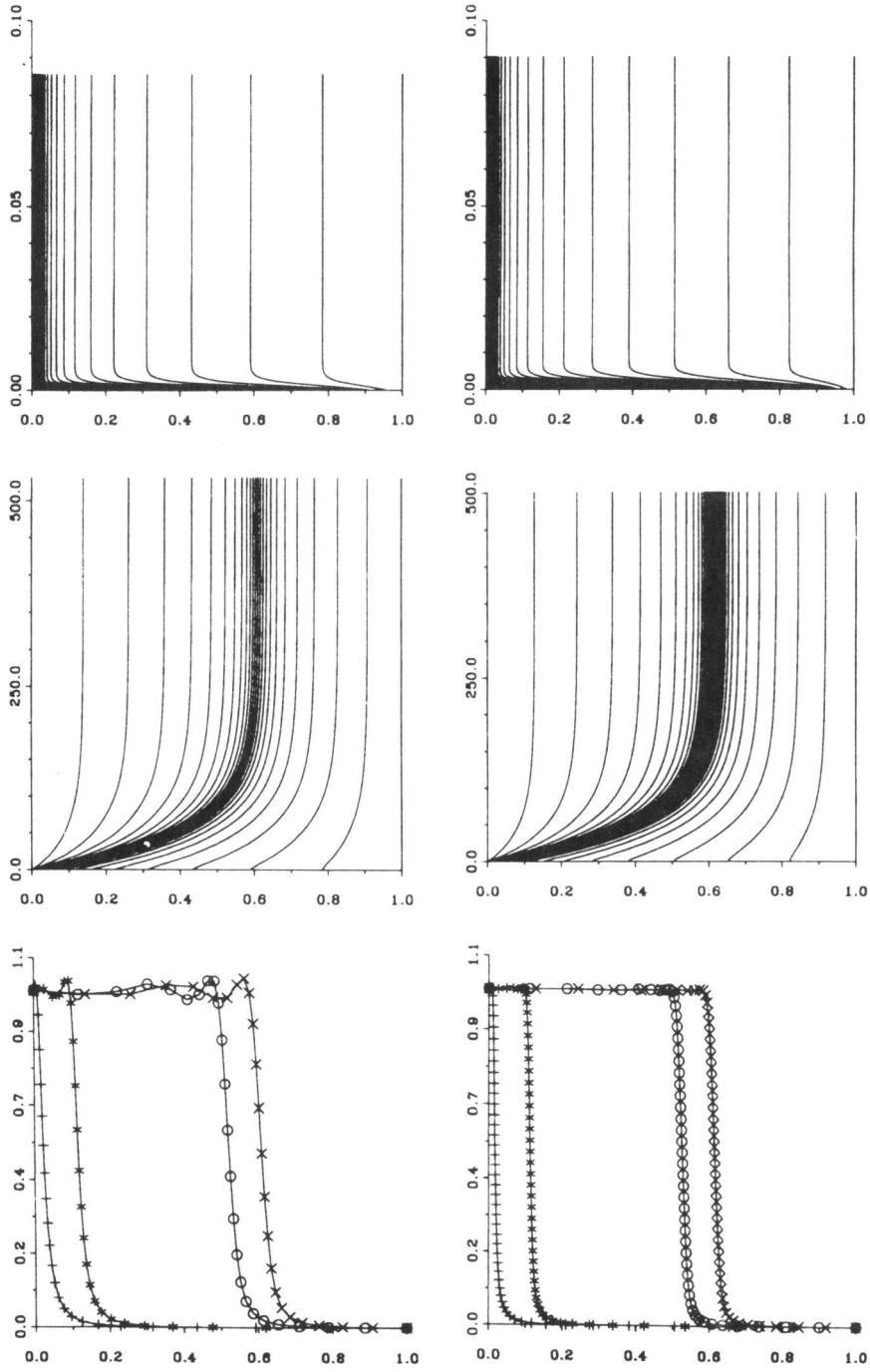


FIGURE 6.2. Example II: Gridlines and salt concentration profiles at t=1, 10, 200, 500. The left part of the figure corresponds with $N = 25$ and the right part with $N = 50$. Note the difference in scaling in each of the two gridline plots.

		STEPS	JACS	RESIDS	NITER	OVERSHOOT	CPUtime (sec.)
N=25	t=1	160	40	958	421	2.0E-2	--
	t=10	239	64	1509	654	3.0E-2	--
	t=100	335	92	2155	930	3.0E-2	--
	t=200	354	95	2244	980	4.0E-2	--
	t=500	373	101	2371	1072	4.0E-2	95
N=50	t=1	249	59	1473	685	4.0E-3	--
	t=10	300	72	1787	826	2.0E-3	--
	t=100	338	79	1983	931	2.0E-3	--
	t=200	355	81	2048	968	2.0E-3	--
	t=500	379	88	2204	1033	2.0E-3	185
N=100	t=1	312	85	2058	926	4.0E-4	--
	t=10	346	93	2253	1015	4.0E-4	--
	t=100	410	103	2538	1168	7.0E-4	--
	t=200	423	105	2592	1196	8.0E-4	--
	t=500	1038	707	12700	3344	9.0E-4	1753

TABLE 6.3. Example II: Integration histories.

		STEPS	JACS	RESIDS	NITER	CPUtime (sec.)
N=100	t=1.0	554	159	3734	1624	--
	t=2.0	819	262	6008	2511	--
	t=5.0	1009	308	7259	3154	1146

TABLE 6.4. Example III: Integration histories.

6.5. CONCLUDING REMARKS

We have applied a moving-grid finite-volume method to a particular class of one-space dimensional fluid-flow/salt-transport problems with rapid spatial and temporal transitions in the salt concentration. The success of this method rests on two sorts of automatic grid-adaptation. The first adaptation is connected with the space grid and serves to cope with the rapid spatial transitions. These are dealt with by

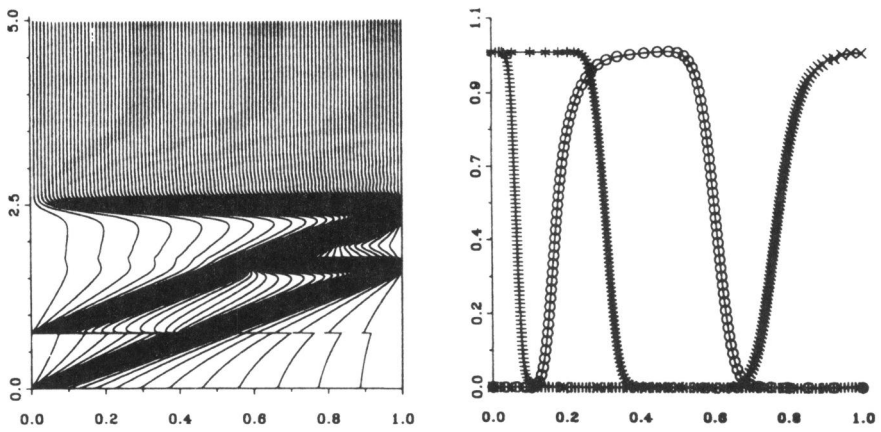


FIGURE 6.3. Example III: Gridlines and salt concentration profiles at $t = 0.1, 0.5, 1.0, 2.0, 5.0$ for $N = 100$.

integrating on grids that spatially equidistribute a relevant measure of the error. The equidistribution is realized in a dynamic Lagrangian approach where the grid is adapted continuously in time. This feature is important since it makes it possible to accurately and efficiently follow steep travelling fronts. The second adaptation serves to cope with rapid temporal transitions and is just the use of variable stepsizes in the numerical integration. Variable stepsizes are a prerequisite when drastic solution changes have to be dealt with, like the onset of a steep front. The numerical integration has been performed with the LSODI based stiff ODE solver of the SPRINT package [1].

Our findings reported in Section 6.4 have convincingly shown that the method is very well suited to solve 1D brine transport models involving high concentration gradients. Because we have worked with an a priori chosen set of numerical control parameters, it is most likely that tuning of these parameters will further enhance the efficiency and accuracy for the specific model at hand. Since the method has been originally developed for general, one-space dimensional PDE systems [6,13], the method is also an excellent candidate for solving fluid-flow/solute-transport problems from other fields of application. In this connection it is worth to emphasize the user-friendly computational environment of the SPRINT package and the moving-grid interface MGI [3], which together provide a numerical software tool that requires a minimum of programming effort.

REFERENCES

1. M. BERZINS, P.M. DEW, and R.M. FURZELAND (1989). Developing Software for

Time-Dependent Problems Using the Method of Lines, *Appl. Numer. Math.*, 5, 375-398.

2. J.G. BLOM and J.G. VERWER (1989). *On the Use of the Arclength and Curvature Monitor in a Moving-Grid Method which is Based on the Method of Lines,* Report NM-N8902, Centre for Mathematics and Computer Science (CWI), Amsterdam.
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 (CWI), Amsterdam (submitted to ACM Trans. Math. Software).
4. K.E. BRENAN, S.L. CAMPBELL, and L.R. PETZOLD (1989). *Numerical Solution of Initial-Value Problems in Differential Algebraic Equations*, North-Holland.
5. K. DEKKER and J.G. VERWER (1984). *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*, North-Holland.
6. E.A. DORFI and L. O'DRURY (1987). Simple Adaptive Grids for 1-D Initial Value Problems, *J. Comput. Phys.*, 69, 175-195.
7. J.C.H. VAN EIJKEREN, P.A. ZEGELING, and S.M. HASSANIZADEH (1991). *Practical Use of SPRINT and a Moving-Grid Interface for a Class of 1D Nonlinear Transport Problems,* Report nr. 959101001, RIVM, Bilthoven, The Netherlands.
8. R.M. FURZELAND, J.G. VERWER, and P.A. ZEGELING (1990). A Numerical Study of Three Moving Grid Methods for One-Dimensional Partial Differential Equations which are based on the Method of Lines, *J. Comput. Phys.*, 89, 349-388.
9. S.M. HASSANIZADEH (1990). Experimental Study of Coupled Flow and Mass Transport: a Model Validation Exercise, in *Calibration and Reliability in Groundwater Modeling*, ed. K. KOVAR, IAHS Publication No. 195, Wallingford, Oxfordshire, U.K..
10. S.M. HASSANIZADEH and A. LEIJNSE (1988). On the Modeling of Brine Transport in Porous Media, *Water Resources Research*, 24, 321-330.
11. J.M. SANZ-SERNA and J.G. VERWER (1989). Stability and Convergence at the PDE / Stiff ODE Interface, *Appl. Numer. Math.*, 5, 117-132.
12. R.D. SKEEL and M. BERZINS (1990). A Method for the Spatial Discretization of Parabolic Equations in One Space V ariable, *SIAM J. Sci. Stat. Comput.*, 11, 1-32.
13. J.G. VERWER, J.G. BLOM, R.M. FURZELAND, and P.A. ZEGELING (1989). A Moving-Grid Method for One-Dimensional PDEs based on the Method of Lines, in *Adaptive Methods for Partial Differential Equations*, 160-175, ed. J.E. FLAHERTY, P.J. PASLOW, M.S. SHEPHARD AND J.D. VASILAKIS, SIAM, Philadelphia.
14. P.A. ZEGELING, J.G. VERWER, and J.C.H. V. EIJKEREN (1991). *Application of a Moving Grid Method to a Class of Brine Transport Problems in Porous Media,* Report NM-R9112, Centre for Mathematics and Computer Science (CWI), Amsterdam (to appear in Int. J. for Numer. Meth. in Fluids).

APPENDIX

$$\psi_0^{(m)}(x) = \left\{ \begin{array}{ll} 0 & \text{if singular} \\ \displaystyle\int_x^{X_1} y^{-m}dy \Big/ \int_{x_L}^{X_1} y^{-m}dy & \text{if regular} \end{array} \right\} \quad x_L \le x \le X_1 \quad ,$$

$$0 \qquad \text{elsewhere}$$

$$\psi_i^{(m)}(x) = \left\{ \begin{array}{ll} 1 - \psi_{i-1}^{(m)}(x) & X_{i-1} \le x \le X_i \\ \displaystyle\int_x^{X_{i+1}} y^{-m}dy \Big/ \int_{X_i}^{X_{i+1}} y^{-m}dy & X_i \le x \le X_{i+1} \quad i = 1, \ldots, N, \\ 0 & \text{elsewhere} \end{array} \right.$$

$$\psi_{N+1}^{(m)}(x) = \left\{ \begin{array}{ll} 1 - \psi_N^{(m)}(x) & X_N \le x \le x_R \\ 0 & \text{elsewhere} \end{array} \right. .$$

Note that

$$\int_a^b y^{-m}dy = \left\{ \begin{array}{ll} b - a & m = 0 \\ \ln(b/a) & m = 1 \\ -(b^{-1} - a^{-1}) & m = 2 \end{array} \right. .$$

Integrals $\zeta_{i+\frac{1}{2}}^{m+1}$:

$$\zeta_{i+\frac{1}{2}}^{m+1} = \left\{ \begin{array}{ll} 0 & \text{if singular and } i = 0 \\ \displaystyle\int_{X_i}^{X_{i+1}} y dy \Big/ \int_{X_i}^{X_{i+1}} y^{-m}dy = \frac{X_{i+1}^2 - X_i^2}{2} \Big/ \int_{X_i}^{X_{i+1}} y^{-m}dy & \text{otherwise} \end{array} \right. .$$

Trial functions $\phi_i^{(m)}$:

In the regular case

$$\phi_i^{(m)}(x) = \psi_i^{(m)}(x), \quad i = 0, \ldots, N+1$$

and in the singular case

$$\phi_0^{(m)}(x) = \begin{cases} \dfrac{X_1^2 - x^2}{X_1^2 - x_L^2} & x_L \le x \le X_1 \\ 0 & \text{elsewhere} \end{cases},$$

$$\phi_i^{(m)}(x) = \begin{cases} 1 - \phi_{i-1}^{(m)}(x) & X_{i-1} \le x \le X_i \\ \dfrac{X_{i+1}^2 - x^2}{X_{i+1}^2 - X_i^2} & X_i \le x \le X_{i+1} \\ 0 & \text{elsewhere} \end{cases} \quad i = 1, \ldots, N,$$

$$\phi_{N+1}^{(m)}(x) = \begin{cases} 1 - \phi_N^{(m)}(x) & X_N \le x \le x_R \\ 0 & \text{elsewhere} \end{cases}.$$

Quadrature points $\xi_{i+1/2}$:
In the regular case

$$\xi_{i+1/2} = \begin{cases} \dfrac{X_{i+1} + X_i}{2} & m = 0 \\ \dfrac{X_{i+1} - X_i}{\ln(X_{i+1}/X_i)} & m = 1 \\ \dfrac{\ln(X_{i+1}/X_i)}{-(X_{i+1}^{-1} - X_i^{-1})} & m = 2 \end{cases}$$

and in the singular case

$$\xi_{i+1/2} = \frac{2}{3} \frac{X_{i+1}^3 - X_i^3}{X_{i+1}^2 - X_i^2}.$$

Chapter 7

Moving-Finite-Element Solution of Time-Dependent Partial Differential Equations in Two Space Dimensions

"This is duck soup for MFE"

7.1. Introduction

The aim of this paper is to show the capability of the 2D moving-finite-element method (MFE) to solve different kinds of time-dependent partial differential equations (PDEs) having solutions with steep moving fronts, rotating pulses, or other features involving fine scale structures. MFE belongs to the class of moving-grid methods, which is a subclass of the class of (time-dependent) adaptive-grid methods. Adaptive-grid methods are numerical methods for PDEs which strive to resolve the sharp transitions in the PDE solution to acceptable degrees of accuracy thereby avoiding the use of an excessive amount of spatial grid points. Fixed-grid methods are in such situations computationally inefficient, since, to afford an accurate approximation, they should contain an unacceptably large number of nodes. Adaptive methods use non-fixed, non-uniform or locally uniform grids and automatically concentrate the grid in regions of high spatial activity during the time-integration process.

In contrast with the one-dimensional case (see, e.g., [20, 24, 35]), application of moving-grid methods in two space dimensions is less trivial. For instance, there are many possiblities to treat the one-dimensional boundary and to discretize the spatial domain each having their own difficulties for specific PDEs. Therefore, 2D moving-grid methods have hardly been applied to real-life problems. The MFE method [10, 14, 20, 24], considering its general approach, allows in principle a large class of PDE problems to be dealt with. However, because of the intrinsic coupling between the discretization of the PDE and the grid selection, the application of MFE, as for any other method, is not without difficulties. The main difficulty we are referring to is the threat of grid distortion. Grid distortion can occur in many different ways due to the quite complex solution behaviour of 2D-evolution problems [38]. This paper describes some aspects of the MFE method when applied to

various kinds of PDEs with different underlying background. More precisely, three main PDE characteristics are recognized, i.e., convection, diffusion, and reaction. For each of this notions MFE acts differently with respect to efficiency (time-integration process), grid movement, etc..

MFE is based on the well-known numerical method-of-lines (MOL) approach for solving time-dependent PDEs. The MFE-method used here restricts its finite-dimensional approximation to a piecewise linear function on a hexagonally connected triangularization of the 2D spatial domain. The grid movement is generated by a least-squares minimization of the so-obtained PDE residual with respect not only to the time-derivatives of the solution amplitudes, as in the standard (fixed-grid) Galerkin case, but also to the now unknown grid velocities. This procedure yields, according to the MOL idea, a large system of stiff ODEs, which may be integrated with a sophisticated implicit stiff ODE/DAE solver, for example the SPGEAR module in the SPRINT package [8, 9].

In literature, several tests of the MFE method in 2D are described. These can be found, e.g., in [3, 6, 10, 14, 18, 29, 38]. Theoretically, however, little is known about the moving-finite-element method in 2D. An exception in this respect is the work by Baines and Wathen [4, 7, 36, 37], and Miller [23]. A very important theoretical property is the relation of MFE in both 1D and 2D, for hyperbolic PDEs, with the method of characteristics [4]. Secondly, for convection/diffusion equations with a small diffusion coefficient it can be shown that MFE resembles a perturbed method of characteristics [4, 38]. Additionally, in the case of parabolic equations there is a strong link of MFE with equidistribution principles [38]. All these properties have their influence on the performance of the method when applied to different types of PDEs, as we will see in this paper.

The paper is divided into four sections. In Section 7.2.1 we describe the MFE method for a general system of PDEs in two space dimensions. The treatment of second-order operators is discussed in Section 7.2.2. Section 7.3.1 shows an application of MFE to convection-reaction equations. A special feature of this section is the use of a non-rectangular domain for the so-called 'Molenkamp-test' (see also [26, 30]), which is an important testproblem in meteorology. In Section 7.3.2 MFE is applied to a reaction-diffusion equation from combustion theory [19, 27]. An interesting aspect in this section is the effect of the regularization parameters (penalties) on the grid movement, quality of the solution, and the numerical time-stepping procedure, respectively. Section 7.3.3 deals with convection-diffusion equations, and shows the effect of a small diffusion coefficient in the PDE on the semi-discrete MFE system. Also in this section, MFE is applied to a system of nonlinear brine transport problems in a porous medium, of importance in the field of hydrology [34, 39]. Finally, Section 7.4 is devoted to some concluding remarks.

7.2. Description of MFE in two space dimensions

7.2.1. The method

In this section a description is given of the moving-finite-element method in two space dimensions. For more details the reader is referred to [10, 11, 20, 24]. The method is presented along the lines of the numerical method-of-lines (MOL) approach.

Consider an abstract Cauchy problem for a system of PDEs in two space dimensions,

$$\frac{\partial u^{(j)}}{\partial t} = \mathcal{L}_j, \qquad j=1, \cdots, p \qquad (x,y)\in\Omega, \ t>0, \tag{2.1}$$

where $\mathcal{L}_j$ is a spatial differential operator containing at most second-order derivatives. The MFE-approximation to $u^{(j)}$ is chosen to be piecewise linear on a hexagonally connected triangularization of Ω

$$u^{(j)} \approx U^{(j)} = \sum_{l=1}^{M} U_l^{(j)}(t)\, \alpha_l(x,y,\{X_l(t),Y_l(t)\}), \qquad j=1, \cdots, p, \tag{2.2}$$

where M denotes the total number of gridpoints, and α_l are the standard piecewise linear hat functions. Differentiating (2.2) with respect to t by the chain rule, and using the time-dependence of the gridpoints $(X_l(t), Y_l(t))$ we obtain

$$\dot{U}_l^{(j)} = \sum_{l=1}^{M} \dot{U}_l^{(j)} \alpha_l + \dot{X}_l \beta_l^{(j)} + \dot{Y}_l \gamma_l^{(j)}, \qquad j=1, \cdots, p. \tag{2.3}$$

The basis functions $\beta_l^{(j)}$ (see Figure 7.1) have the same support as α_l, i.e., the hexagon of triangles surrounding the j–th node. They are discontinuous at the center and on the inner edges of the hexagon; they are zero on the hexagonal boundary and take on (for each PDE-component) the six different values of $-\partial U^{(l)}/\partial x$ at the central vertices of the six triangles. Note that due to the piecewise linear approximation (2.2), $-\partial U^{(l)}/\partial x$ has a constant value on each triangle. A similar description holds for $\gamma_l^{(j)}$, but now related to $-\partial U^{(l)}/\partial y$. The equations determining the semi-discrete unknowns $U_l^{(j)}$, X_l and Y_l are now obtained in the standard Galerkin way by minimizing the PDE residual R with respect to $\dot{U}_i^{(j)}$, $\dot{X}_i$ and $\dot{Y}_i$, where

$$R := \sum_{j=1}^{p} w_j \, \| \dot{U}_l^{(j)} - \mathcal{L}_j(U) \|^2_{L_2(\Omega)} + \sum_k P_k^2. \tag{2.4}$$

Here w_j are non-negative weight factors and P_k is the grid-regularization term (penalty) (see [11])

$$P_k := \varepsilon_k \dot{\Delta}_k - S_k, \tag{2.5a}$$

with

$$\varepsilon_k^2 := \frac{\varepsilon_1^2}{\Delta_k}, \tag{2.5b}$$

and

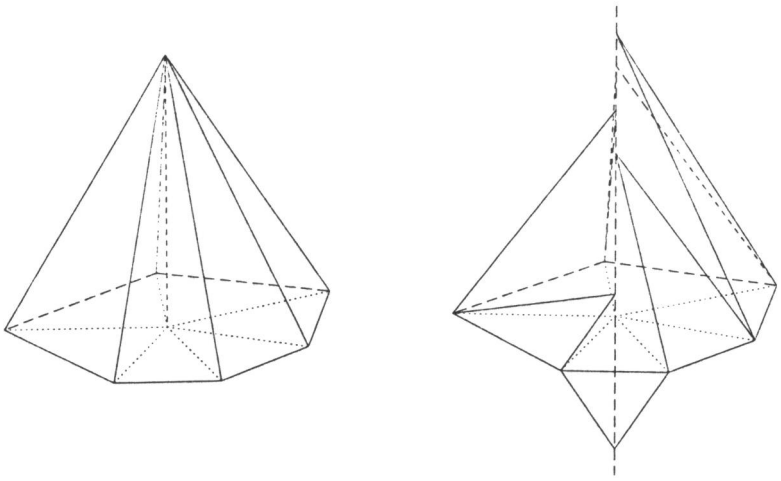


FIGURE 7.1. The basis functions α_l (left) and $\beta_l^{(j)}$ (right).

$$\varepsilon_k S_k := (\frac{\varepsilon_2}{\Delta_k})^2 , \tag{2.5c}$$

where ε_1^2 and ε_2^2 are small user-specified constants. The effects of adding this penalty term are explained below. Note that the second sum in (2.4) is taken over the three perpendiculars Δ_k of each triangle. The minimization of (2.4) is performed by setting

$$\frac{\partial R}{\partial \dot{U}_i^{(j)}} = 0, \qquad \text{for } i=1, \cdots, M;\ j=1, \cdots, p,$$

$$\frac{\partial R}{\partial \dot{X}_i} = \frac{\partial R}{\partial \dot{Y}_i} = 0, \qquad \text{for } i=1, \cdots, M$$

and results in a large system of $(p+2)\times M$ ordinary differential equations in the unknowns $U_i^{(j)}$, X_i and Y_i:

$$\sum_{l=1}^{M} <\alpha_i, \alpha_l> \dot{U}_l^{(j)} + <\alpha_i, \beta_l^{(j)}> \dot{X}_l + <\alpha_i, \gamma_l^{(j)}> \dot{Y}_l = <\alpha_i, \mathcal{L}_j(U)>, \tag{2.6a}$$

$$\text{for } i=1, \cdots, M;\ j=1, \cdots, p,$$

$$\sum_{j=1}^{p} w_j \sum_{l=1}^{M} <\beta_i^{(j)}, \alpha_l> \dot{U}_l^{(j)} + <\beta_i^{(j)}, \beta_l^{(j)}> \dot{X}_l + <\beta_i^{(j)}, \gamma_l^{(j)}> \dot{Y}_l + \sum_k P_k \frac{\partial P_k}{\partial \dot{X}_i} = \tag{2.6b}$$

$$\sum_{j=1}^{p} w_j <\beta_i^{(j)}, \mathcal{L}_j(U)>, \qquad \text{for } i=1, \cdots, M,$$

$$\sum_{j=1}^{p} w_j \sum_{l=1}^{M} <\gamma_i^{(j)}, \alpha_l> \dot{U}_l^{(j)} + <\gamma_i^{(j)}, \beta_l^{(j)}> \dot{X}_l + <\gamma_i^{(j)}, \gamma_l^{(j)}> \dot{Y}_l + \sum_k P_k \frac{\partial P_k}{\partial \dot{Y}_i} = \tag{2.6c}$$

$$\sum_{j=1}^{p} w_j <\gamma_i^{(j)}, \mathcal{L}_j(U)>, \qquad \text{for } i=1, \cdots, M,$$

where $<.,.>$ denotes the usual L_2-innerproduct. It is clear that (2.6a) without the $\dot{X}$- and $\dot{Y}$-innerproducts is just the standard Galerkin method applied to (2.1) using piecewise linear basis and trial functions on a nonuniform triangular grid. The time-dependency of the grid is reflected in the $\dot{X}$- and $\dot{Y}$-innerproducts in (2.6a) and the complete equations (2.6b) and (2.6c).

Working out the innerproducts and defining the vector

$$\eta := (\cdots, U_i^{(1)}, \ldots, U_i^{(p)}, X_i, Y_i, \cdots)^T,$$

we arrive at the semi-discrete MFE system

$$\mathcal{A}(\eta, \varepsilon_1^2)\, \dot{\eta} = G(\eta, \varepsilon_2^2), \qquad t>0, \ \eta(0) \text{ given}, \tag{2.7}$$

where $\mathcal{A}$ is a symmetric matrix, the so-called mass-matrix, containing quantities from the left-hand sides of (2.6), whereas the only problem-specific terms are contained in the vector G. It must be noted that, for $\varepsilon_1=\varepsilon_2=0$, i.e., without regularization terms, there exist fundamental difficulties with solving system (2.7).

The first difficulty is a possible singularity in the mass-matrix $\mathcal{A}$. The matrix $\mathcal{A}$ is singular in the so-called case of 'parallelism'. Parallelism occurs in the absence of curvature in the solution of the PDE. In this case, the basisfunctions $\beta_i^{(j)}$ and α_i (and/or $\gamma_i^{(j)}$ and α_i) become linearly dependent, which means that the parametrization of the time-derivative $U_t^{(j)}$ in (2.3) degenerates. In other words, the minimization procedure then has no unique solution, resulting in a zero mass-determinant: $\det \mathcal{A}(\eta,0)=0$. Therefore, to avoid the problem of solving a DAE system of index 1 or higher, the ε_l^2-term (2.5b), also called intratriangular viscosity, was introduced in the residual (2.4). It can be shown, that, for $\varepsilon_1 \neq 0$, $\mathcal{A}$ is positive-definite and thus regular. The second degeneracy of $\mathcal{A}$ arises whenever the triangles get too thin or lose their positive orientation, i.e., '2D mesh-tangling'. The matrix $\mathcal{A}$ will then become very ill-conditioned and numerically singular, because in such cases the triangle area tends to zero, giving widely varying eigenvalues. Note, that this behaviour is time-dependent. Both singularities of the mass-matrix are also discussed theoretically in [37]. Since it appears in the left-hand side of (2.7), the parameter ε_1 can also be seen to serve as a tool to control the grid-point motion.

Another difficulty in (2.7) could arise from a possible singularity of the nonlinear steady-state system: $G(\eta,0)=0$. In the already described case of parallelism, which appears in applications with, e.g., a constant steady-state, the system $G=0$ could have non-unique solutions. The parameter ε_2 in (2.5c) ('intratriangular spring force') serves to prevent this degeneracy.

As for any other method, the regularization is somewhat heuristic and necessarily problem-dependent. For example, if ε_1 is chosen too large, the grid movement is restricted; $\varepsilon_1 \to \infty$ gives a non-moving grid, with the result that there may not be sufficient refinement in regions of large spatial activity. On the other hand, if ε_1 is

too small, the mass-matrix $\mathcal{A}$ may become numerically singular. The parameter ε_2 could be chosen equal to zero in most applications. For PDEs with 'flat' steady-state solutions a small non-zero value of ε_2 suffices to keep the semi-discrete ODE system regular. As for ε_1, too large values for ε_2 could restrict the grid movement: $\varepsilon_2 \to \infty$ gives a uniform (non-moving) grid.

The weight-factors w_j in (2.4) serve to make it possible to let certain PDE components dominate the grid movement equations (2.6b) and (2.6c). This may be desirable for a badly scaled problem, or if one PDE component is strongly varying and a second component has only a mildly varying solution, for example.

7.2.2. Second-order operators

The MFE method used here has serious difficulties, when solving PDEs with second-derivative terms. Due to the piecewise linear approximation (2.2), second order derivatives fail to be defined in the usual sense. For example, Δu is zero on the interior of each triangle, but has a measure of constant strength $-(\frac{\partial u^+}{\partial n}+\frac{\partial u^-}{\partial n})$ on each inner edge. Here the + and - denote the two triangles over which the jump is made. Furthermore, the basis function β_i has a discontinuity along each inner edge. This can be derived from the relation $\beta_i = -U_x \alpha_i$. These two properties combined indicate that the innerproduct $<\beta_i, \Delta u>$ even fails to be defined in the sense of distributions, which makes it impossible to evaluate $\int_\Omega \beta_i \Delta u \, d\Omega$. Note that these considerations hold for the right-hand side γ-innerproducts in (2.6c) as well.

There are several 'tricks' to get around this fundamental difficulty:

1. Miller [20, 24] uses the idea of 'mollification' to regularize the undefined innerproducts. Mollification can be interpreted as using a limiting equation obtained by applying the minimizing condition to a PDE residual underlying a smoothed (mollified) piecewise linear function and then letting the limiting equation approach its now well-defined limit (using a small perturbation parameter, the so-called δ-mollifier).
2. Johnson [17] and Mueller [28] apply Green's theorem in a clever way to work out the troublesome innerproducts. Their treatment, however, can only be used for special PDE operators.
3. Sweby [31] uses the idea of 'recovery'. This involves fitting a piecewise polynomial to the first derivative and then differentiating this better defined quantity. The process is simple in one dimension, but in two dimensions the expressions may become very complex.
4. Higher order test functions. This means, that instead of applying a minimization of the PDE residual, i.e., a projection on a space of piecewise linears, we do a projection on a higher order function space. Then, no problems are encountered when evaluating the right-hand side innerproducts. This Petrov-Galerkin approach works well in 1D (see [16]), and seems to improve the nodal movement and position of the grid points in steep fronts. In 2D, however, the evaluation of the innerproducts is not so simple and straightforward as in one space dimension.

It is interesting to note, that for the relatively simple case of a Laplacian operator ideas 1. and 2. yield identical semi-discrete equations. This is expressed by the

following lemma:

LEMMA 7.1. The right-hand side innerproducts in (2.6b) for $\mathcal{L}(u) = \Delta u$ using mollification are identical to the ones using idea 2. In other words:

$$<\beta_i, \Delta u>_{1.} = <\beta_i, \Delta u>_{2.}. \tag{2.8}$$

PROOF By definition we have:

$$<\beta_i, \Delta u>_{2.} = \int_\Omega \beta_i u_{xx}\, d\Omega + \int_\Omega \beta_i u_{yy}\, d\Omega = -\int_{6\tau} u_x \alpha_i u_{xx}\, d\Omega - \int_{6\tau} u_x \alpha_i u_{yy}\, d\Omega,$$

where 6τ are the six triangles surrounding the grid point (X_i, Y_i). Applying the chain rule, the divergence theorem, and the small support of α_i, respectively, this expression can be written as

$$\sum_{6\tau} \left(\frac{1}{2}\alpha_{i,x}(\tau)(u_x^2(\tau) - u_y^2(\tau))\, A(\tau) + \alpha_{i,y}(\tau) u_x(\tau) u_y(\tau)\, A(\tau)\right), \tag{2.9}$$

where $A(\tau)$ is the area of triangle τ. Next, using relations between $\alpha_{i,x}$, $\alpha_{i,y}$, $A(\tau)$, the lengths L of an edge, and the unit normal vector n on an edge, (2.9) can be rewritten as

$$\sum_{6\tau} \sum_{2\ i.e.} \left[\left(n_1 \left(\frac{\partial u}{\partial n}\right)^2 - 2 n_2 \frac{\partial u}{\partial n} \frac{\partial u}{\partial \tau}\right) \frac{L}{4}\right] \qquad (i.e. = \text{inner edge}),$$

where n_1 and n_2 denote the x- and y-component of the normal vector n, $\partial u/\partial n$ the normal and $\partial u/\partial \tau$ the tangential derivative on each edge, and the second sum is taken over the two inner edges of τ. Expressing $\partial u/\partial n$ and $\partial u/\partial \tau$ in terms of u_x, u_y and n, finally results in

$$\sum_{6\ edges} \left(\frac{\partial u^+}{\partial n} + \frac{\partial u^-}{\partial n}\right) \frac{u_x^+ + u_x^-}{2}\, \frac{L}{2}.$$

This is equal to $<\beta_i, \Delta u>_{1.}$, since the mollified form consists of $-(\partial u^+/\partial n + \partial u^-/\partial n)$ (the constant measure of Δu on each inner edge) and $-(u_x^+ + u_x^-)/2$, which can be interpreted as a mean value for $-u_x$ along an edge (see [11]). □

To apply MFE to real-life problems, such as the brine transport problem in Section 7.3.3, involves the treatment of a general second order flux term in combination with the piecewise discontinuous basis functions β_i and γ_i. In this case, option 2. can not be used, and options 1. and 3. yield extremely complicated nonlinear semi-discrete terms in the right-hand side vector G in equation (2.7). We have used another idea, which is in our opinion the simplest justifiable option. The idea makes use of the fact, that, owing to the piecewise linear approximation, the first order spatial derivatives are constant on each triangle. For instance, the innerproduct of β_i with a flux term $\nabla \cdot \phi(u, \nabla u)$ is treated as follows:

$$\begin{aligned} <\beta_i, \nabla \cdot \phi(u, \nabla u)> &:= \int_\Omega \beta_i\, \nabla \cdot \phi(u, \nabla u)\, d\Omega = \int_{6\tau} -U_x\, \alpha_i\, \nabla \cdot \phi(u, \nabla u)\, d\Omega \\ &\approx \sum_{6\tau} -U_x(\tau) \int_\tau \alpha_i\, \nabla \cdot \phi(u, \nabla u)\, d\tau, \end{aligned} \tag{2.10}$$

where the last integral may be approximated without difficulties in the usual way by some quadrature rule, e.g., the mid-point rule. It must be noted, that using (2.10) is a bit tricky, because the jumps of β_i-values over triangle edges are, more or less, averaged in the approximating integral. In some sense, when replacing ϕ by ∇u, giving back the Laplacian, (2.10) is an averaged approximation to formula (2.8). Moreover, 'freezing' one space dimension in (2.10), say the y co-ordinate, approximates the mollified interpretation of $<\beta_i, u_{xx}>$, since applying (2.10) to $<\beta_i, u_{xx}>$ yields:

$$<\beta_i, u_{xx}> = \int_{X_{i-1}}^{X_{i+1}} -u_x \alpha_i u_{xx}\, dx \approx \sum_{2el} -U_{x,el} \int_{el} \alpha_i u_{xx}\, dx$$

$$\approx -\bar{U}_{x,2el} \sum_{2el} \int_{el} \alpha_i u_{xx}\, dx \quad (el = \text{element}),$$

where $\bar{U}_{x,2el}$ is the average value of U_x over the two elements. Defining m_i to be U_x on the element (X_{i-1}, X_i) and working out the integral by partial integration, we obtain

$$-\frac{m_i + m_{i+1}}{2}(m_{i+1} - m_i) = -\frac{1}{2}(m_{i+1}^2 - m_i^2).$$

The last expression is equivalent to $<\beta_i, u_{xx}>_1$. (see [24]).

For the brine-transport application, we evaluate the flux innerproducts according to the idea above. In the other numerical experiments in this paper the second idea is used for the Laplacian-innerproducts.

7.3. An evaluation of MFE in 2D

System (2.6) is implemented according to the ideas in [11]. This holds also for the treatment of the boundary terms. To obtain the fully discretized solution, ODE system (2.7) must be integrated numerically. It is known, that this system will usually be extremely stiff. For integration in time, therefore, a suitable stiff ODE solver must be used. In our numerical experiments we have solved the implicit ODE system (2.7) with the (implicit) BDF integrator SPGEAR of the SPRINT package [8, 9] in the usual way. This means amongst others that the resulting algebraic system is solved by a modified Newton process. In the description of the experiments, which all were done on an SGI Indigo workstation, the following notation is used:

STEPS = number of successful time-integration steps

JACS = number of Jacobian evaluations

BS = total number of Newton iterations

TOL = time-tolerance (absolute and relative) for the SPGEAR integrator

7.3.1. Application to convection-reaction equations

For this type of PDEs a standard form for the right-hand side operator $\mathcal{L}$ in (2.1) is

$$\mathcal{L}(u) = -\gamma \cdot \nabla u + g(u,x,y,t), \tag{3.1}$$

where γ defines the convection term, and may depend on u, x, y, and t. This means, that (2.1) for this choice of $\mathcal{L}$ is of the hyperbolic type. MFE applications to this class of PDEs can be found in [3, 6, 28, 29, 38]. It is known, see, e.g., [4], that, for this choice of PDE operator, there is a strong link between the semi-discrete MFE system (without regularization terms) and the characteristic equations of the PDE. More specifically, it can be shown that for PDE operators $\mathcal{L}$ of the form (3.1) with γ linear in u, x, y, while setting aside boundary effects, the ODE system (2.7) is equivalent to a discretized version of

$$(\dot{x}, \dot{y})^T = \gamma^T, \tag{3.2a}$$

$$\dot{u} = g. \tag{3.2b}$$

In many cases, this characteristic behaviour is very beneficial. However, there are some situations in 2D for which this behaviour could give problems. Two of these problems are described in [38]. The first problem has to do with a possible difference between the directions of the characteristic curves of the PDE (the movement of the 'fluid'-particles) and the movement of the solution front. The second problem can be summarized by the term 'grid rotation', and may occur in PDE problems where the characteristic curves are circles, spirals, etc.. Furthermore, equations (3.2) show, that for γ=0, i.e., when]Y are dealing with an ODE instead of a PDE, there will be no grid movement at all. This is a desirable property, since pure ODEs can not produce propagating wave solutions.

The following numerical example from this PDE class is the so-called 'Molenkamp-test', which is a standard test problem in meteorology (see also [25, 26, 30]). In fact, this example is identical to Example II in [38] enhanced with a linear reaction term. It was this example (without reaction term) for which MFE produced a strongly distorted grid when applied on a square with fixed corner points. It will be shown that, if we let the boundary points 'move around the corner', MFE produces a very accurate solution on a well-adapted grid. The effect on MFE of adding the reaction term will be examined as well.

EXAMPLE I ('Molenkamp-test'):

The operator $\mathcal{L}$ reads for this test-problem:

$$\mathcal{L}(u) = -\pi\,(y-\frac{1}{2})\frac{\partial u}{\partial x} + \pi\,(x-\frac{1}{2})\frac{\partial u}{\partial y} - c\,u, \tag{3.3}$$

with initial and boundary conditions

$$u\,|_{t=0} = u_0(x,y) = \exp(-80[(x-\frac{1}{2})^2 + (y-\frac{3}{4})^2]),$$

# rotations	STEPS	JACS	BS	$U_{\max}$
1	100	68	374	1.00000
3	205	162	842	1.00000
5	289	234	1205	0.99989
7	390	323	1664	0.99989

TABLE 7.1. Example I: Integration history for c=0.

$u\,|_{\partial\Omega}$=0.

The domain Ω is chosen to be a circle with center (1/2,1/2) and radius $\sqrt{2}$, in contrast with the experiments in [38], where Ω was represented by a square. It must be noted that the choice of a circular domain is certainly not restrictive. On $\partial\Omega$ equation (2.6a) is replaced by $u = 0$. Equations (2.6b) and (2.6c) are not altered, so that the grid points on the boundary are now free to move.

The exact solution describes a pulse that moves around in circles with a constant speed. During the movement the shape of the pulse changes, depending on the value of c in the reaction term. For c=0, the shape of the pulse is unchanged, whereas, for c>0, the peak of the pulse will decrease. The characteristic curves are circles with centre (1/2,1/2)

$$(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2 = r^2,\ \ 0 < r < \sqrt{2}.$$

On these curves the solution varies according to $\dot{u} = -c\,u$, resulting in the exact solution: $u(x,y,t) = u_0\, e^{-c\,t}$.

From literature it is known, that many numerical methods have severe problems with solving this test example (see [25, 30]). Two important drawbacks of standard numerical techniques to solve (3.3) are, that either they damp out the solutions dramatically, because of numerical diffusion, and/or they exhibit strong oscillations in the solution during the integration process.

In Table 7.1 and Figure 7.2 the results for MFE applied to model (3.2) on a circular domain with c=0 are given. In the below described runs the standard choices TOL=1.E−3, ε_1^2=1.E−4 and ε_2^2=1.E−9 are made. In this example the effects of the penalty terms are not essential; they are only needed to keep the semi-discrete system (2.7) non-degenerate. The starting grid consists of only 11×11 points of which 5×5 are distributed uniformly around the cone in (0.25,0.75)×(0.5,1.0).

A notable point is that the integration costs remain almost constant during each rotation of the pulse. This can be explained by the property of MFE to follow the characteristic curves of the PDE, thereby yielding almost linearly (in x and y) varying grid speeds during the calculations. Note, that this is the optimal way to follow the rotating pulse. Also, the maximum value of the pulse shows no tendency to decrease as for other methods. In fact, the error in the peak of the pulse is less than

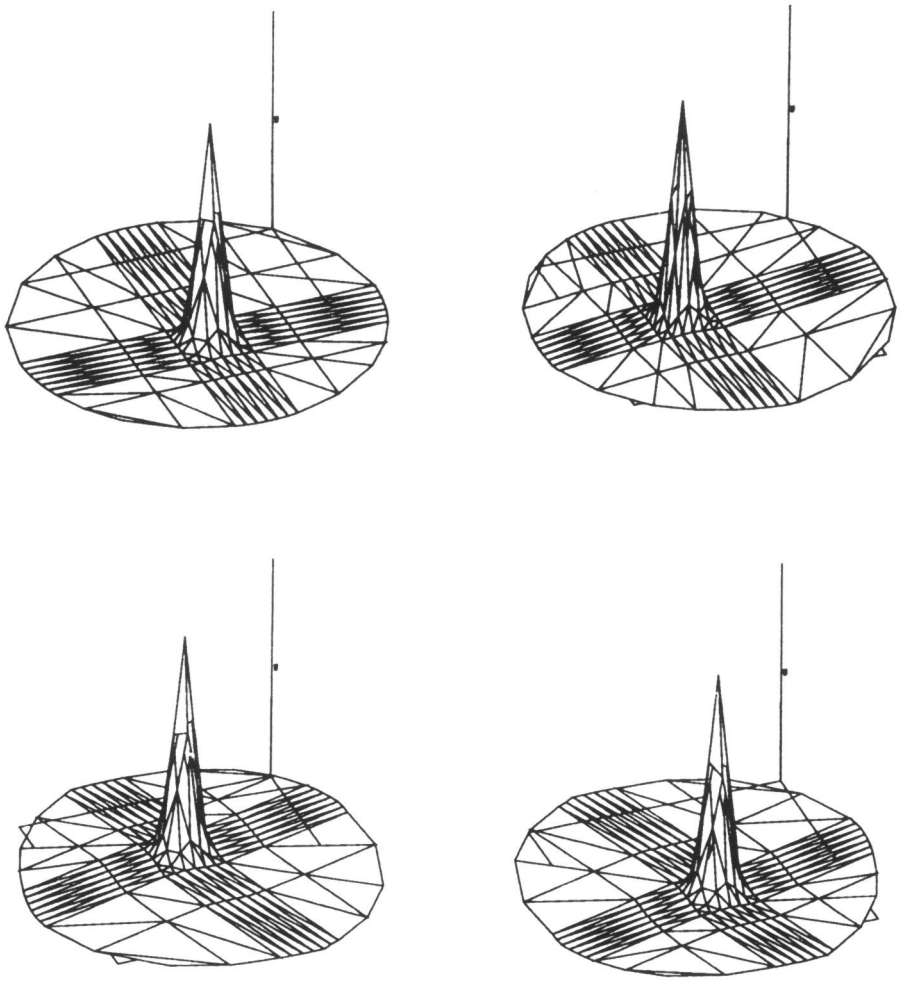


FIGURE 7.2. MFE solution for Example I ($c=0$) at $t=0.0$, 0.5, 1.0 and 2.0.

$1.E-4$ even after several rotation periods. Both effects can be explained by referring to equation (3.2) with $g=0$. In contrast with [38], the grid structure now remains undistorted and well-adapted to the shape of the pulse.

If we take a non-zero reaction term, for instance $c=1$ or $c=10$ in (3.2), the results do not change dramatically. As for the previous case, the error induced by MFE for

the maximum amplitude is less than $1.E-4$ for both $c=1$ and $c=10$. The time-integration costs are, say for $c=1$: $STEPS=95$, $JACS=68$, $BS=361$ for the time-period $0\leq t\leq 2$. Again, this could be explained by the fact that the semi-discrete MFE system (2.7) is strongly related with equations (3.2), but now with $g\neq 0$.

7.3.2. Application to reaction-diffusion equations

For this type of PDEs a standard form for the right-hand side operator $\mathcal{L}$ in (2.1) is

$$\mathcal{L}(u) = \nabla\cdot(D(u)\nabla u) + f(u,x,y,t), \tag{3.4}$$

with a diffusion coefficient D. MFE applications to this class of PDEs can be found in [2, 10, 17, 18, 38].

In the following numerical example, MFE is applied to a scalar reaction-diffusion equation. The effect of the penalty parameters ε_1 and ε_2 on the time-stepping process, the movement of the grid, and the quality of the solution, are examined, respectively.

EXAMPLE II ('Flame propagation'):

This example of the reaction-diffusion type is a model of a so-called single one-step reaction of a mixture of two chemicals [19] and stems from combustion theory. The right-hand side operator $\mathcal{L}$ in (2.1) reads in this case

$$\mathcal{L}(u) = d\,\Delta u + D(1+a-u)e^{-\delta/u}, \tag{3.5}$$

on the domain $\Omega = (0,1)\times(0,1)$, subjected to the initial and boundary conditions

$$u|_{t=0}=1,$$

$$\frac{\partial u}{\partial n}=0, \text{ at } x=0,\ y=0,$$

$$u=1, \text{ at } x=1,\ y=1.$$

The dependent variable u here represents the temperature of the mixture. The parameter a is the heat release, $D=Re^{\delta}/a\delta$ the Damkohler number, δ the activation energy, and R is the reaction rate. For small times the temperature gradually increases in a circular area around the origin. Then, provided the reaction rate is large enough, at a finite time 'ignition' occurs causing the temperature to suddenly jump from near unity to $1+a$, while simultaneously a reaction front is formed which circularly propagates towards the outer Dirichlet boundary. When the front reaches the boundary the problem runs into steady-state. Following [1] we select the parameter values $a=1$, $\delta=20$, $R=5$, but choose a different value for the diffusion parameter d. While in [1] the diffusion coefficient $d=1$, we have here put $d=0.1$ as in [33]. A smaller diffusion parameter has the effect that the temperature front becomes steeper, particularly so upon approaching steady-state. With this choice of parameters the 'ignition' takes place at about $t=.24$ and the solution is in 'steady-state' at about $t=.35$. It is known (see, e.g., [13] for the one-dimensional case), that BDF codes need a rather small time tolerance TOL of, say $1.E-5$, because of the different time-scales in the model. This is especially so to detect the start of ignition

accurately. Small errors at this time point result in significantly larger global errors later on. This 'local instability' of the model can be explained by inspection of the reaction term: for $1 \leq u \leq 1.8$ its derivative $\partial f / \partial u$ is positive, for example $1.E+3$ for $u \approx 1.6$, resulting locally in growing solutions and for $u \geq 1.8$ negative, for example $-5.5E+3$ for $u \approx 2.0$, resulting in locally decreasing solutions, respectively.

Since the initial solution is constant, we let MFE start on a uniform grid consisting of 21×21 moving grid points. The experiments shown below are separated in two different subcases. First, a standard value for ε_2^2 is chosen, viz., $1.E-9$, while varying the first penalty parameter ε_1^2 from $1.E-6$ to 1. Second, with a fixed value for ε_1^2, viz., $1.E-4$, the effects of varying ε_2^2 within a range from 0 to 1 are studied.

Figure 7.3 shows the grid and solution generated by MFE at some interesting points of time for the standard choice of parameters $\varepsilon_1^2 = 1.E-4$ and $\varepsilon_2^2 = 1.E-9$. Although there is no exact solution available, the numerical solution may be compared with a 'reference' solution obtained by [33], where an adaptive grid with local refinement is used. Both solutions resemble very well, and both adaptive grids, although underlying totally different adaptation principles, indeed generate refinements in the same regions. Moreover, just as in Example III of [38], we see for this PDE (especially in steady-state, i.e., at $t=0.35$) a concentration of triangles in regions with large second derivatives. This corroborates the conjecture that for parabolic equations in 2D there is a close relation between MFE and equidistribution principles. This is, unfortunately, only an experimental evidence, in contrast with the one-dimensional situation for which there is some theory available in this respect.

Tables 7.2 and 7.3 show integration data for increasing values of ε_1^2 and ε_2^2, respectively. From Table 7.2 we note that the efficiency of MFE depends highly on the choice of ε_1. For very small values of this parameter the solution is still accurate, but is computed on a grid which moves not very smoothly in time. We also see, that for larger values of ε_1, the adaptivity of the method is influenced, while for too large choices the grid does not move at all. In this respect, the parameter ε_1, originally introduced in the minimization procedure to ensure the regularity of the mass-matrix, can also be seen as a smoothing parameter for the grid movement. The integration performance of MFE is in one aspect disappointing. If we calculate the ratio *JACS/STEPS* as a function of ε_1, we see from the table that for $\varepsilon_1 = 1.E-4$ this ratio is 0.34, i.e., when using MFE optimally, whereas for $\varepsilon_1 = 1$ (a fixed grid) *JACS/STEPS* is only 0.12. We may conclude, that the efficiency of MFE, at least for this testproblem, although resulting in accurate solutions, is strongly influenced by the choice of the intratriangular viscosity constant. The effect of the parameter ε_2, introduced to keep the steady-state MFE system regular, is less important (see Table 7.3). In fact, ε_2 could have been chosen equal to zero for this example, since the steady-state solution still possesses a steep profile. Also, the time-integration process is not much influenced by this parameter, in contrast with the penalty constant ε_1. This can be seen in the second column of Table 7.3, where the number of time steps is almost constant for the smaller values of this parameter. Note, that a fixed (uniform) grid, with an inaccurate solution, is obtained, if we let ε_2 tend to infinity. In the case of a flat steady-state, as in the next numerical example, this so-

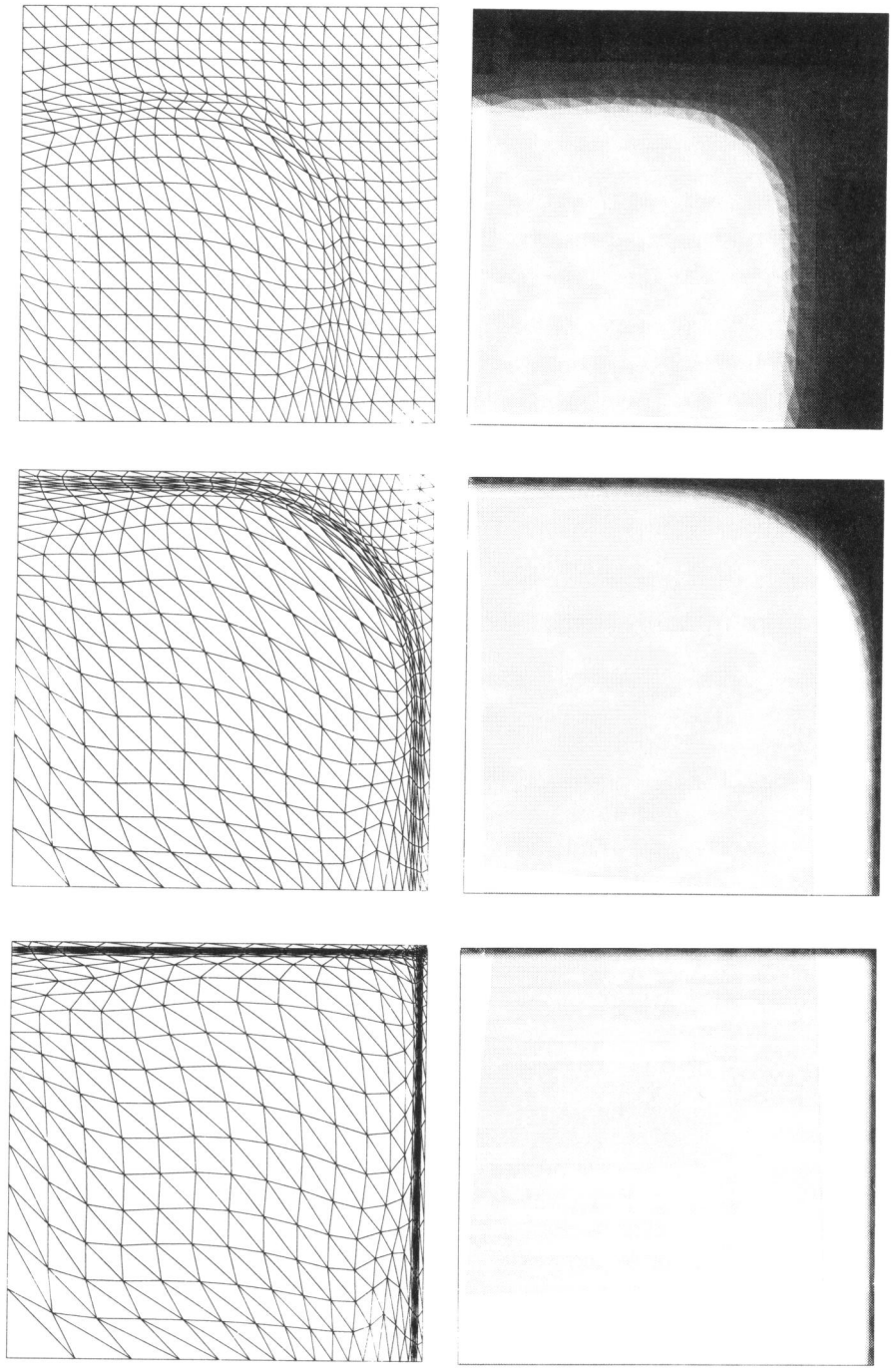


FIGURE 7.3. Grids and solutions of Example II at $t = 0.25237,\ 0.27653,\ 0.35000$.

called spring constant plays a more significant role.

ε_1^2	*STEPS*	*JACS*	*BS*	*SOLUTION*(*)	*GRID*(**)
1.E-6	1984	579	5928	O.K.	NON-SMOOTH
1.E-4	618	209	2188	O.K.	ADAPTIVE
1.E-2	502	93	1370	O.K.	TOO SLOW
1.E-0	423	52	1066	BAD	FIXED

TABLE 7.2. Example II: Variation of ε_1^2.
(*): compared with the solution in [32]
(**): this is, of course, a subjective notion

ε_2^2	*STEPS*	*JACS*	*BS*	*SOLUTION*	*GRID*
0	609	188	2105	O.K.	ADAPTIVE
1.E-9	618	209	2188	O.K.	ADAPTIVE
1.E-6	635	186	2212	O.K.	LESS NON-UNIFORM
1.E-3	783	255	2839	INACCURATE	ALMOST UNIFORM
1.E-0	469	75	1250	BAD	UNIFORM

TABLE 7.3. Example II: Variation of ε_2^2.

7.3.3. Application to convection-diffusion equations

For this type of PDEs a standard form for the righthand-side operator $\mathcal{L}$ in (2.1) is

$$\mathcal{L}(u) = \varepsilon \Delta u - \beta \cdot \nabla u, \tag{3.6}$$

where, in general, ε is a small coefficient, and β a linear or nonlinear function of u. Depending on the proportion $\varepsilon/\|\beta\|$, the PDE defined by (3.6) can be classified numerically as hyperbolic or parabolic. The numerical difficulties for this type of PDEs, in fact, arise because of this 'double' property: for $\varepsilon \downarrow 0$ the PDE will be of the hyperbolic type, whereas otherwise the PDE will possess parabolic properties. MFE test results for this class of PDEs can be found in [10, 14, 21, 22, 28, 29, 38]. Since the displacement of solution fronts for this type of equations is mainly determined by the convection term $\beta \cdot \nabla u$, the complete movement may be defined by a perturbed characteristic ODE system. In formulas:

$$\dot{\mathbf{r}} = \beta + \varepsilon \mathbf{f}, \quad \dot{u} = \varepsilon h, \quad 0 < \varepsilon \ll 1, \tag{3.7}$$

where $\mathbf{r}$ represents the position vector $(x,y)^T$, and $\mathbf{f}$, h are some functions containing first and second order spatial derivatives of the solution u. Similar to the pure convective case (see Section 7.3.1), the semi-discrete MFE equations are now related to equations (3.7), of course without penalty terms ($\varepsilon_1=\varepsilon_2=0$). In one space dimension, we can formulate the function $\mathbf{f}$ explicitly, both in the continuous and semi-discrete case, see, e.g., [12, 38]. In [4] it is shown that, in two dimensions, the MFE grid speeds are, in a certain sense, approximations to

$$\dot{\mathbf{r}} = \beta + \varepsilon\,((\Delta u)_x/u_{xx},(\Delta u)_y/u_{yy})^T.$$

It is clear, that the perturbation term $\mathbf{f}$, though multiplied by the small diffusion coefficient ε, may be of importance in subregions of the spatial domain Ω, where the solution of the PDE model possesses large first and second order derivatives, such as in boundary layers and steep transitions. Therefore, a proper treatment of the Laplacian-innerproducts in (2.7) is indispensable.

The following two numerical PDE examples will show the performance of MFE for convection-diffusion equations. First, a simple linear PDE is discussed with the accent on the effects of the semi-discrete diffusion term, especially its effect in a boundary layer and for steady-state situations. The second example is a strongly nonlinear system of two PDEs, describing brine transport in a porous medium. Here, the weightfactors w_j, which were defined in (2.4), play an important role. An additional aspect in this example is the use of general second order flux terms.

EXAMPLE III ('A linear model'):

For this example we have chosen the convection term β to be constant: $\beta = (\beta_1,\beta_2)^T = (1,1/2)^T$, and two different values for ε: $1.E{-}2$ and $1.E{-}3$. The domain Ω is the unit square and the boundary and initial conditions satisfy

$t = 0$: $\quad u = 1$ for $0 \le x \le 1/11$ and $0 \le y \le 1/11$,

$\quad u = 0$ elsewhere,

$t > 0$: $\quad u = 1$ for $x = 0$ and $0 \le y \le 1/11$, and for $y = 0$ and $0 \le x \le 1/11$,

$$\frac{\partial u}{\partial n}\Big|_{\partial\Omega} = 0, \text{ elsewhere.}$$

With these choices, the solution is a front that, starting as a small-sized block near (0,0), moves approximately with speed $\|\beta\|$ and direction β. After having reached the boundary, the solution u tends to a constant steady-state value of 1 for $t\to\infty$, as we have Neumann conditions on the remaining part of the boundary. Note, that this part of the computation is difficult, since the solution at the boundary is lifted from 0 to 1, while still obeying the Neumann conditions.

In Table 7.4 the time-integration history is presented for $\varepsilon=1.E{-}2$. Figure 7.4 shows the grids and solutions at t=0.3, which is halfway the propagation phase of the front. Using the standard values $\varepsilon_1^2=1.E{-}4$, $\varepsilon_2^2=1.E{-}9$ and $TOL=1.E{-}3$, MFE is applied to this test problem on a starting grid of 25×25 grid points, of which 13×13 are distributed uniformly over the block, and the remaining ones uniformly over the rest of the domain.

There are some interesting remarks to be made regarding the figures in the table. At first, we see a start-up phase, in which MFE tries to cope with the initial discontinuity u and the initial nonuniform grid. After that, the propagation phase of the solution dominates the performance of the moving-grid method. This can be seen from the almost constant time integration between t=0.25 and t=0.75. In this phase, MFE approximates a perturbed characteristic movement (see (3.7)), with a small perturbation term $\varepsilon \mathbf{f}$. Around t=1.0 the front hits the boundary, at which point the perturbation $\mathbf{f}$ is no longer negligible. As indicated above, this part of the computation combined with the approach to steady-state takes many time steps. It must be noted, that, when taking a smaller diffusion coefficient, these effects appear even stronger. For ε=1.E−3, the propagation phase until t=0.75 takes only *STEPS* = 41, *JACS* = 24 and *BS* = 142, letting the characteristic movement of the grid points show better. On the other hand, the steady-state phase becomes proportionally more expensive. Until t=100.0, the diffusion takes over the integration process: *STEPS* = 333, *JACS* = 212 and *BS* = 1031.

t	*STEPS*	*JACS*	*BS*	*PHASE*
0.25	32	20	106	START-UP
0.50	45	25	149	PROPAGATION
0.75	57	34	200	PROPAGATION
1.00	64	37	222	BOUNDARY EFFECTS
100.00	119	64	381	STEADY-STATE

TABLE 7.4. Example III: Integration history for $\varepsilon=10^{-2}$.

EXAMPLE IV ('Brine transport in a porous medium'):

This problem stems from hydrology and models the transport of salt in a porous medium [15]. In the present application we consider a particular model for isothermal, single-phase (only fluids), two component (water and salt) saturated flow, which is constituted by a system of two PDEs basic to groundwater flow: a continuity equation for the brine mass and a transport equation for the salt mass concentration. These equations are supplemented with two basic laws, viz., Darcy's law and Fick's law. After a few simplifications (see [34, 39]), the following system of two nonlinear PDEs determines the model

$$n\rho\frac{\partial\omega}{\partial t} = -\nabla\cdot(\rho\mathbf{J}) - \rho\mathbf{q}\cdot\nabla\omega \ , \tag{3.8a}$$

$$\beta n\rho\frac{\partial P}{\partial t} + \gamma n\rho\frac{\partial\omega}{\partial t} = -\nabla\cdot(\rho\mathbf{q}) \ , \tag{3.8b}$$

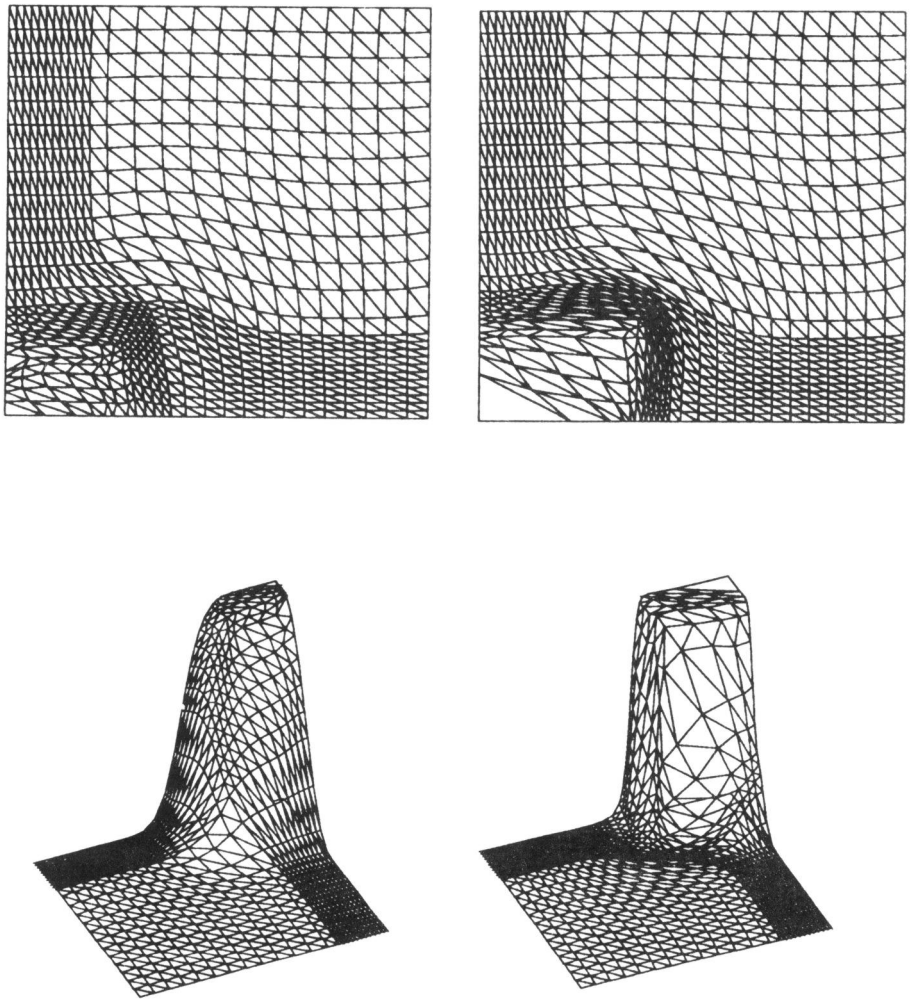


FIGURE 7.4. Grids and solutions of Example III at t=0.3 for $\varepsilon = 10^{-2}$ (left), 10^{-3}(right).

where

$$\mathbf{q} = -\frac{K}{\mu}(\nabla P - \rho \mathbf{g}) \quad \text{(Darcy's law)}, \tag{3.9a}$$

$$\mathbf{J} = -\mathcal{D}\nabla\omega \quad \text{(Fick's law)}. \tag{3.9b}$$

The 2×2 dispersion tensor is defined by

$$\mathcal{D} = (nD_{mol} + \alpha_T |\mathbf{q}|)I + \frac{\alpha_L - \alpha_T}{|\mathbf{q}|}\mathbf{q}\mathbf{q}^T, \quad |\mathbf{q}| = (\mathbf{q}^T\mathbf{q})^{1/2}, \tag{3.9c}$$

To complete the physical model, an equation of state is added, given by

$$\rho = \rho_0 e^{\beta(P-P_0)+\gamma\omega} . \tag{3.9d}$$

In equations (3.8) and (3.9), the following notation is used: the quantities ω, P, ρ, n, μ, K, $\mathbf{q}$, $\mathbf{J}$ and $\mathbf{g}$ represent, respectively, salt mass fraction, pressure, density, porosity, viscosity, permeability, Darcy velocity, dispersive flux and the gravity vector. Further, D_{mol} stands for the molecular diffusion, α_T for the transversal dispersion coefficient, α_L for the longitudinal dispersion coefficient, β for the compressibility, γ for a salt coefficient, and ρ_0, P_0 for reference density and pressure values.

Some characteristic properties of PDE system (3.8) and (3.9) may be derived. First, it can be seen easily, that the matrix $\mathcal{D}$ is positive definite for $|\mathbf{q}| \neq 0$. In addition, however, it is noted that for $|\mathbf{q}| \downarrow 0$ the PDE system may, in some circumstances, be not well-posed [32]. This ill-posedness could occur in special cases, such as in stagnation points or in vortices. In the present application the model does not suffer from these singular situations. Second, both dispersion coefficients α_T and α_L highly determine the character of equation (3.8a). Equation (3.8a) is of the convection-diffusion (advection-dispersion) type and, as usual, numerically difficult to solve if it is advection dominated. With the below selected initial and boundary conditions and the actual choice for α_T and α_L, steep travelling salt fronts are generated. Further, a special feature of the model is that the compressibility coefficient β is very small or even zero. If $\beta=0$, then the 2×2-matrix multiplying the time-derivative vector $(\omega_t, P_t)^T$ is singular and (3.8b) is effectively replaced by an equation without temporal derivatives. As we will use a stiff (implicit) ODE/DAE solver, viz., the BDF module of the SPRINT package, there is no need to distinguish between $\beta=0$ and $\beta\neq 0$. Finally, it is known, that the pressure equation (3.8b) can, in certain circumstances, be approximated in a so-called Boussinesq sense, replacing it by the standard continuity equation $\nabla\cdot\mathbf{q} = 0$. With the present conditions this results in a very smoothly varying pressure distribution $P(x,y,t)$. For more information with regard to the PDE model, the interested reader is referred to [34, 39].

The application of MFE to this model needs an extra explanation. There are three additional aspects worth mentioning:

1) The first aspect is the appearance of a matrix, not equal to the identity matrix, in front of the time-derivative. However, for this, only a minor modification is needed to apply the minimization procedure, which was explained in Section 7.2.1. No special difficulties are encountered when working out the innerproducts for the new semi-discrete formulation.

2) The second aspect deals with the treatment of general second-order operators. This has already been treated in Section 7.2.2.

3) The third new aspect in this MFE-application is the use of the weight factors w_j (see equations (2.6)). The w_j were introduced to make it possible to let the grid movement equations (2.6b) and (2.6c) be dominated by certain PDE components. In the brine transport equation this possibility is very welcome. Since the pressure

gradients $\partial P/\partial x$ and $\partial P/\partial y$ are expected to vary only slowly, the weight factor w_2 is taken zero. This means, that pressure effects are not taken into account in the moving-grid equations (2.6b) and (2.6c), and that the salt concentration changes take the responsibility for the grid movement (w_1=1).

In the present MFE-application constant values are taken for the permeability (K=1.E−10), viscosity (μ=1.E−3), porosity (n=0.4) and the molecular diffusion (D_{mol}=0). Further, the model is subjected to initial and boundary conditions on the unit square (0,1)×(0,1):

$$\omega(x,y,0) = 0, \qquad P(x,y,0) = p_0 - p_1 y,$$

$$y = 0 \text{ and } 2/11 < x < 8/11 \quad : \quad \partial\omega/\partial n = 0, \quad P = p_0,$$

$$y = 0 \text{ and } x \text{ elsewhere} \quad : \quad \omega = \omega_0, \quad P = p_0,$$

$$x = 0,\ 1 \text{ and } 0 < y < 1 \quad : \quad \partial\omega/\partial n = \partial P/\partial n = 0,$$

$$y = 1 \text{ and } 0 < x < 1 \quad : \quad \partial\omega/\partial n = 0, \quad P = p_0 - p_1,$$

where ω_0, p_0 and p_1 are constants: ω_0=0.25, p_0=1.7E+5, p_1=0.7E+5. The remaining problem data are: β=4.5E−10, γ=0.6923, ρ_0=9.98E+2, P_0=1.E+5 and $\mathbf{g}=(0,-9.81)^T$. The dispersion coefficients are chosen: α_T=2.E−3 and α_L=1.E−2.

Under these conditions, the model describes an injection of salt water of a high concentration through two gates at the bottom of the domain. Due to the small values of α_T and α_L, and the boundary conditions imposed on the pressure component P, the solution $\omega(x,y,t)$ of the PDE model is a travelling fresh/salt water front, moving from the lower boundary to the upper boundary. After having reached that boundary, the dispersion and the Neumann boundary condition take over the process, resulting in a smoothing out of the two fronts. For t sufficiently large the fronts disappear completely which means that the whole medium is filled with the high-salt concentration fluid.

MFE is applied to this model with a starting grid of 31×21 gridpoints, of which 11 gridpoints in the y-direction are distributed uniformly between 0 and 0.1 and the remaining ones uniformly over the rest of the domain. We have scaled the equations in such a way, that ω and P are of the same order of magnitude. Therefore, we can select one time-tolerance value for all DAE components, viz., TOL=1.E−3. The penalty parameter values are ε_1^2=1.E−4 and ε_2^2=1.E−8. Figure 7.5 shows the MFE solutions for ω at some characteristic points of time. Since the solutions for p do not change dramatically, we leave them out of the discussion. We clearly see the movement of the grid points following the two salt fronts. The lower three plots picture the steady-state process, for which finally MFE renders a uniform grid. Table 7.5 gives the time-integration history belonging to this run. It is interesting to remark that the figures in Table 7.5 are almost identical to the ones in Table 7.4. The four phases described in the previous example can also be recognized in this application: 1. the start-up phase, which costs a considerable number of *STEPS* for MFE to cope with the initial solution and grid distribution, 2. the propagation phase of the fronts, with a nearly constant time behaviour, for which the characteristic movement of the grid points is responsible, 3. the phase, in which the salt front hits the boundary and

t	*STEPS*	*JACS*	*BS*	*PHASE*
0.25	40	19	118	START-UP
0.50	48	23	144	PROPAGATION
1.00	59	25	174	PROPAGATION
10.00	96	44	287	BOUNDARY EFFECTS
1.E+6	146	71	423	STEADY-STATE

TABLE 7.5. Example IV: Integration history.

4. the steady-state phase. The last two phases account for almost 50% of the integration effort.

For practical simulations of the model there is a need for more general boundary conditions (flux-conditions) and smaller values for the dispersion coefficients. Smaller α_T and α_L have the effect to give yet steeper fronts in ω. However, to apply MFE on such cases, we need to consider a more careful treatment of the general second order flux terms (2.10), since they have a strong influence on the grid movement, especially in the steep parts of the moving front solution, and near steady-state.

7.4. CONCLUSIONS

As a member of the class of moving-grid methods, the moving-finite-element method (MFE) is able to accurately approximate solutions of PDE models in two space dimensions possessing steep local transitions. In this paper MFE has been applied to PDEs with a different underlying background, viz., containing convection, reaction and/or diffusion terms. In all examined cases, steep moving front solutions are satisfactorily followed by the semi-discrete moving-grid points. It must be stressed that compared with standard fixed-grid methods, a notable advantage of MFE is, that it can be used with a relatively small number of spatial grid points, when applied to PDEs with sharp transitions.

The effect of the penalty parameters on the semi-discrete ODE system depends on the PDE to be solved. The first parameter, the intratriangular viscosity constant, ε_1, has an important influence on the efficiency of the time-stepping procedure. For small values the grid movement may become irregular, whereas for slightly larger values of ε_1 it can be seen as a grid-smoothing parameter. Too large values of ε_1 yield an (unwanted) non-moving grid. The second parameter, the intratriangular spring constant, ε_2, needs only to be used for PDEs with a flat steady-state solution, and, therefore, affects the time-integration process for large points of time. Further, the steady-state 'flame front' solution strengthens the conjecture, that, for parabolic equations, MFE is closely related to equidistribution principles.

For convection-diffusion equations MFE resembles a perturbed method of

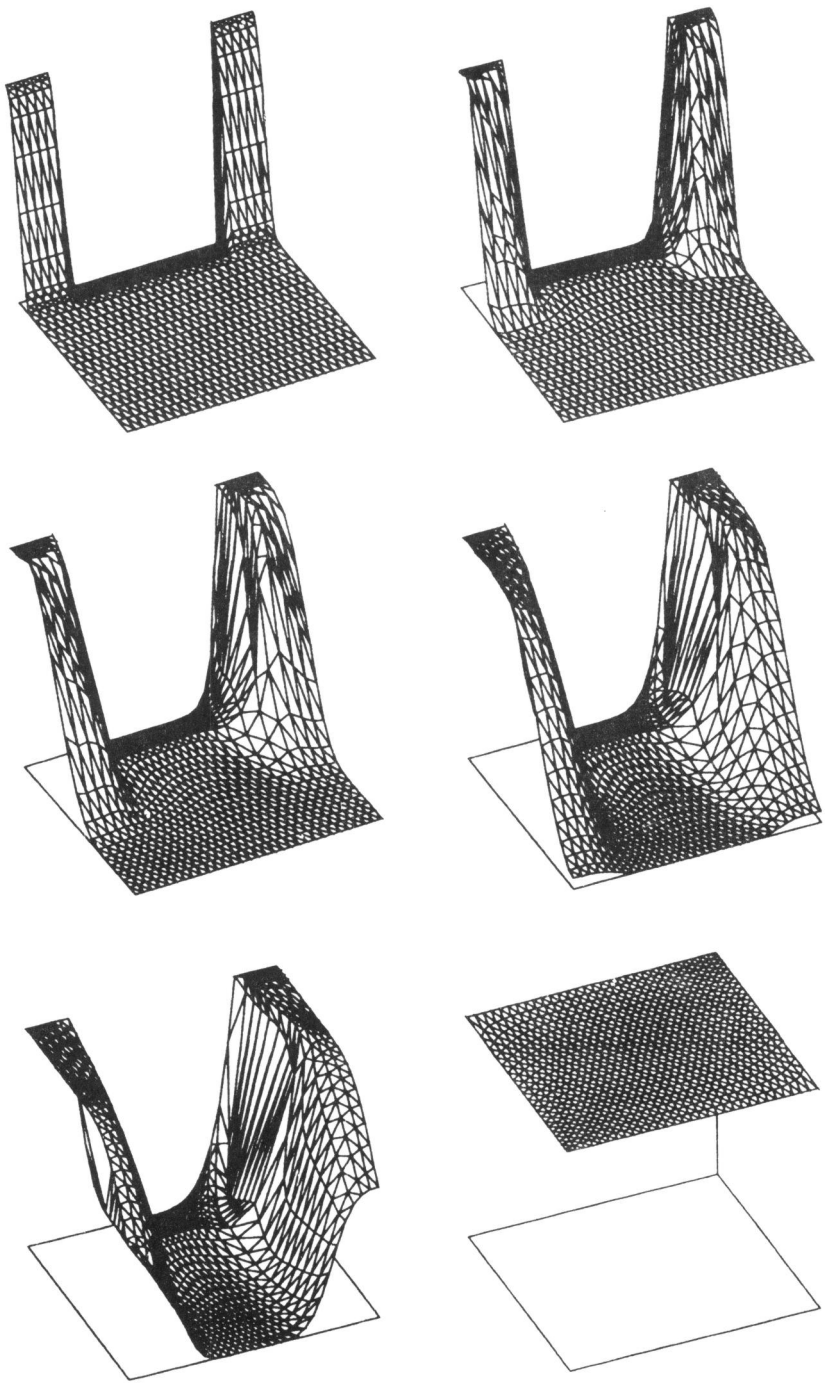


FIGURE 7.5. MFE solution of Example IV at
$t = 0,\ 0.1,\ 0.25,\ 0.5,\ 0.75,\ 1.E{+}6.$

characteristics. The PDE diffusion coefficient plays an important role in this respect. Since MFE uses piecewise linear basis functions, second derivative terms are not well-defined. Several regularizations are possible to treat the troublesome innerproducts. The diffusion term and, thus the choice of regularization, influences the grid movement around steep transitions, near boundary layers and also in near steady-state situations.

MFE is applied to a brine transport problem in a porous medium. The weight factors, introduced to emphasize or de-emphasize certain PDE components, are utilized to let the grid-movement of the method merely be driven by the first component, the salt mass fraction, which consists of steep moving fronts, and not by the second component, the pressure, which varies only little during the whole time-period. A special treatment, of the general flux term is carried out, using the average of the first derivatives of the solution, to cope with undefined innerproducts.

Future developments to improve the performance of MFE, could contain: 1. the use of higher order basis/testfunctions, 2. the incorporation of an initial grid procedure (see [5]), which could also be used for regridding after grid distortion, and 3. the implementation of general flux (Robin) boundary conditions. Finally, there is a need for more theoretical results with regard to moving-finite-elements in two space dimensions.

References

1. S. Adjerid and J.E. Flaherty (1988). A Local Refinement Finite Element Method for Two Dimensional Parabolic Systems, *SIAM J. Sci. Stat. Comput.*, 9, 792-881.
2. R. Alexander, P. Manselli, and K. Miller (1979). Moving Finite Elements for the Stefan Problem in Two Dimensions, *Academia Nazionale Dei Lincei, VIII*, LXVII, 57-61.
3. R.D. Alstead (1987). *Moving Finite Element Modelling of the 2D Shallow Water Equations,* Numerical Analysis Report no. 5/87, University of Reading.
4. M.J. Baines (1991). An Analysis of the Moving Finite Element Procedure, *SIAM J. Numer. Anal.*, 28, 1323-1349.
5. M.J. Baines (1991). Algorithms for Best Piecewise Discontinuous Linear and Constant L2 Fits to Continuous Funcions with Adjustable Nodes in One and Two Dimensions, *Preprint (submitted to SIAM J. Sci. Stat. Comput.)*, University of Reading.
6. M.J. Baines and A.J. Wathen (1986). Moving Finite Element Modelling of Compressible Flow, *Applied Numer. Maths.*, 2, 495-514.
7. M.J. Baines and A.J. Wathen (1988). Moving Finite Element Methods for Evolutionary Problems I: Theory, *J. Comput. Phys.*, 79, 245-269.
8. 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..
9. M. Berzins and R.M. Furzeland (1986). *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.

10. N. CARLSON and K. MILLER (1988). The Gradient Weighted Moving Finite Element Method in Two Dimensions, in *Finite Elements Theory and Application*, 152-164, ed. D.L. DWOYER, M.Y. HUSSAINI AND R.G. VOIGHT, Springer Verlag.
11. M.J. DJOMEHRI and K. MILLER (1981). *A Moving Finite Element Code for General Systems of PDE's in 2-D,* Report PAM-57, Center for Pure and Applied Mathematics, University of California, Berkeley.
12. J.K. DUKOWICZ (1984). A Simplified Adaptive Mesh Technique Derived from the Moving Finite Element Method, *J. Comput. Phys.*, 56, 324-342.
13. R.M. FURZELAND, J.G. VERWER, and P.A. ZEGELING (1990). A Numerical Study of Three Moving Grid Methods for One-Dimensional Partial Differential Equations which are based on the Method of Lines, *J. Comput. Phys.*, 89, 349-388.
14. R.J. GELINAS, S.K. DOSS, J.P. VAJK, J. DJOMEHRI, and K. MILLER (1983). Moving Finite Elements in 2D: Fluid Dynamics Examples, in *Scientific Computing*, 29-36, ed. R. STEPLEMAN ET AL..
15. S.M. HASSANIZADEH, A. LEIJNSE, W.J. DE VRIES, and R.A.M. STAPPER (1990). *Experimental Study of Brine Transport in Porous Media, Intraval Test Case 13,* Reportnr. 725206003, National Institute of Public Health and Environmental Protection, Bilthoven, The Netherlands.
16. B.M. HERBST, S.W. SCHOOMBIE, and A.R. MITCHELL (1982). A Moving Petrov-Galerkin Method for Transport Equations, *Int. J. Numer. Methods Eng.*, 18, 1321-1336.
17. I.W. JOHNSON (1985). *Moving Finite Elements for Nonlinear Diffusion Problems in One and Two Dimensions,* Numerical Analysis Report no. 12/85, University of Reading.
18. I.W. JOHNSON, A.J. WATHEN, and M.J. BAINES (1988). Moving Finite Element Methods for Evolutionary Problems II: Applications, *J. Comput. Phys.*, 79, 270-297.
19. A.K. KAPILA (1983). *Asymptotic Treatment of Chemically Reacting Systems*, Pitman Advanced Publ. Company.
20. K. MILLER (1981). Moving Finite Elements II, *SIAM J. Numer. Anal.*, 18, 1033-1057.
21. K. MILLER (1983). Alternate Modes to Control the Nodes in the Moving Finite Element Method, in *Adaptive Computational Methods for PDEs*, 165-182, ed. I. BABUŠKA, J. CHANDRA AND J.E. FLAHERTY, SIAM, Philadelphia.
22. K. MILLER (1986). Recent Results on Finite Element Methods with Moving Nodes, in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, 325-338, ed. I. BABUŠKA, O.C. ZIENKIEWICZ, J. GAGO AND E.R. DE A. OLIVEIRA, John Wiley & Sons Ltd..
23. K. MILLER (1988). *On the Mass Matrix Spectrum Bounds of Wathen and the Local Moving Finite Elements of Baines,* Centre for Pure and Applied Mathematics report PAM-430, UCB.

24. K. MILLER and R.N. MILLER (1981). Moving Finite Elements I, *SIAM J. Numer. Anal.*, 18, 1019-1032.
25. W.J. MINKOWICZ, E.M. SPARROW, G.E. SCHNEIDER, and R.H. PLETCHER (1988). *Handbook of Numerical Heat Transfer*, John Wiley & Sons, Inc..
26. C.R. MOLENKAMP (1968). Accuracy of Finite-Difference Methods Applied to the Advection Equation, *Journal of Appl. Meteorology*, 7, 160-167.
27. P.K. MOORE and J.E. FLAHERTY (1992). Adaptive Local Overlapping Grid Methods for Parabolic Systems in Two Space Dimensions, *J. Comput. Phys.*, 98, 54-63.
28. A.C. MUELLER (1983). *Continuously Deforming Finite Element Methods for Transport Problems,* Dissertation, The Univ. of Texas at Austin.
29. A.C. MUELLER and G.F. CAREY (1985). Continuously Deforming Finite Elements, *Int. J. Numer. Methods Eng.*, 21, 2099-2126.
30. TH. L. V. STIJN, J.C.H. V. EIJKEREN, and N. PRAAGMAN (1987). *A Comparison of Numerical Methods for Air-Quality Models,* RIVM-Report 958702007, National Institute of Public Health and Environmental Protection.
31. P.K. SWEBY (1987). *Some Observations on the Moving Finite Element Method and its Implementation,* Numerical Analysis Report no. 13/87, University of Reading.
32. R.A. TROMPERT (1991). *Private Communication.*
33. R.A. TROMPERT and J.G. VERWER (1991). A Static-Regridding Method for Two-Dimensional Parabolic Partial Differential Equations, *Applied Numer. Maths.*, 8, 65-90.
34. R.A. TROMPERT, J.G. VERWER, and J.G. BLOM (1992). *Application of an Adaptive-Grid Method to Brine Transport in Porous Media,* Report NM-R9201, Centre for Mathematics and Computer Science (CWI), Amsterdam (submitted to Int. J. for Numer. Meth. in Fluids).
35. J.G. VERWER, J.G. BLOM, R.M. FURZELAND, and P.A. ZEGELING (1989). A Moving-Grid Method for One-Dimensional PDEs based on the Method of Lines, in *Adaptive Methods for Partial Differential Equations*, 160-175, ed. J.E. FLAHERTY, P.J. PASLOW, M.S. SHEPHARD AND J.D. VASILAKIS, SIAM, Philadelphia.
36. A.J. WATHEN (1986). Mesh-Independent Spectra in the Moving Finite Element Equations, *SIAM J. Numer. Anal.*, 23, 797-814.
37. A.J. WATHEN and M.J. BAINES (1985). On the Structure of the Moving Finite Element Equations, *IMA J. of Numer. Anal.*, 5, 161-182.
38. P.A. ZEGELING and J.G. BLOM (1990). *A Note on the Grid Movement Induced by MFE,* Report NM-R9019, Centre for Mathematics and Computer Science (CWI), Amsterdam (to appear in Int. J. for Numer. Meth. in Eng.).
39. P.A. ZEGELING, J.G. VERWER, and J.C.H. V. EIJKEREN (1991). *Application of a Moving Grid Method to a Class of Brine Transport Problems in Porous Media,* Report NM-R9112, Centre for Mathematics and Computer Science (CWI), Amsterdam (to appear in Int. J. for Numer. Meth. in Fluids).

FAB

Samenvatting

Standaard numerieke methoden om tijdsafhankelijke partiële differentiaalvergelijkingen (PDV) op te lossen gebruiken een uniform rooster dat vast wordt gehouden over het gehele tijdsinterval. Echter, als de oplossing in bepaalde delen van het domein een sterke variatie vertoont, dan kan het gebruik van een standaard vaste-rooster methode zeer inefficiënt zijn. Immers, om een nauwkeurige benadering van de oplossing te verkrijgen, moet de methode een grote hoeveelheid roosterpunten bevatten. Het rooster moet in zulke gevallen lokaal verfijnd worden. Als er bovendien steile fronten aanwezig zijn, die zich in de tijd verplaatsen dan is het nodig dat de methode het rooster ook in de tijdvariabele laat aanpassen aan het steile profiel. In deze situaties, die in de praktijk, bijv. in de chemie, meteorologie of hydrologie, frequent optreden, worden adaptieve roostermethoden gebruikt.

Men kan ruwweg twee klassen van tijdsafhankelijke adaptieve roostermethoden onderscheiden: I bewegende-rooster ('dynamic-regridding') methoden en II zogenaamde 'static-regridding' methoden. De laatstgenoemde klasse bevat methoden die het rooster slechts aanpassen op discrete tijdsniveaus. Klasse I methoden hebben de specifieke eigenschap om het plaatsrooster continu in het ruimte-tijd domein te laten bewegen, waarbij de discretisatie van de PDV en de roosterselectie procedure intrinsiek aan elkaar gekoppeld zijn. Beide klassen hebben natuurlijk voor-en nadelen. Dit hangt, o.a., af van het op te lossen PDV-model, de te gebruiken computer en de vorm van het ruimtedomein in het model.

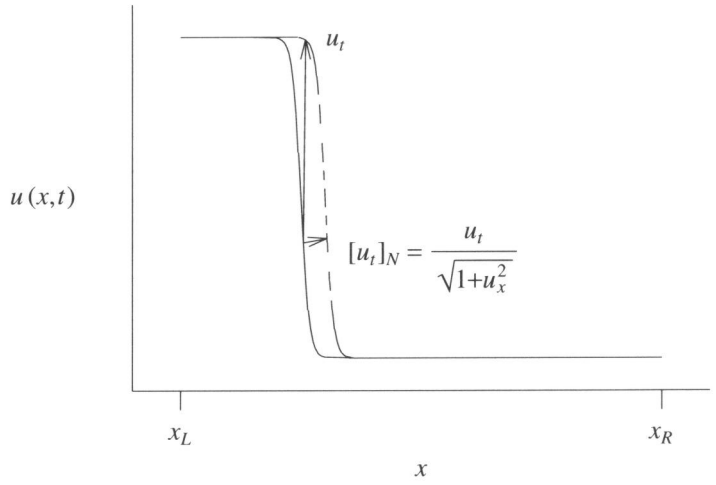
Het grootste voordeel van klasse II methoden is hun conceptuele eenvoud en robuustheid. Dit laatste komt vooral tot uiting wanneer tegelijkertijd meerdere fronten gevolgd moeten worden. Een nadeel echter is, dat men niet onder interpolatie uit komt, wanneer numerieke grootheden van oude naar nieuwe roosters moeten worden vertaald. Verder produceren ze, in tegenstelling met bewegende-rooster methoden, geen 'smoothing' in de tijdrichting. Voor deze methoden zal de tijdstapnauwkeurigheid dus in het algemeen kleinere tijdstappen eisen dan voor bewegende-rooster methoden.

Klasse I methoden gebruiken een vast aantal roosterpunten, waarbij geen interpolatie nodig is, en laten ze continu in de tijd bewegen met de steile fronten. In het geval van meerdere steile fronten die actief zijn in verschillende gebieden van het ruimtedomein, kan dit problemen opleveren bij de numerieke berekeningen, als het rooster het ene front aan het volgen is, terwijl ergens anders een nieuw front ontstaat. Een ander probleem is van topologische aard, n.l., de zogenaamde 'rooster-ontaarding'. Dit treedt bijvoorbeeld op wanneer roosterlijnen elkaar gaan snijden (exact) of bijna gaan snijden (numeriek). Om dit te voorkomen zijn vaak extra regularisatie parameters nodig, hetgeen de robuustheid niet ten goede komt. Anderzijds, ofschoon meer berekeningen per roosterpunt nodig zijn, kunnen bewegende-rooster methoden zeer efficiënt zijn, omdat in het algemeen het tijdsintegratie-proces 'gladder' verloopt en omdat veel minder roosterpunten gebruikt worden.

Dit proefschrift behandelt bewegende-rooster ('moving-grid') methoden voor tijdsafhankelijke partiële differentiaalvergelijkingen in 1 en 2 ruimtedimensies.

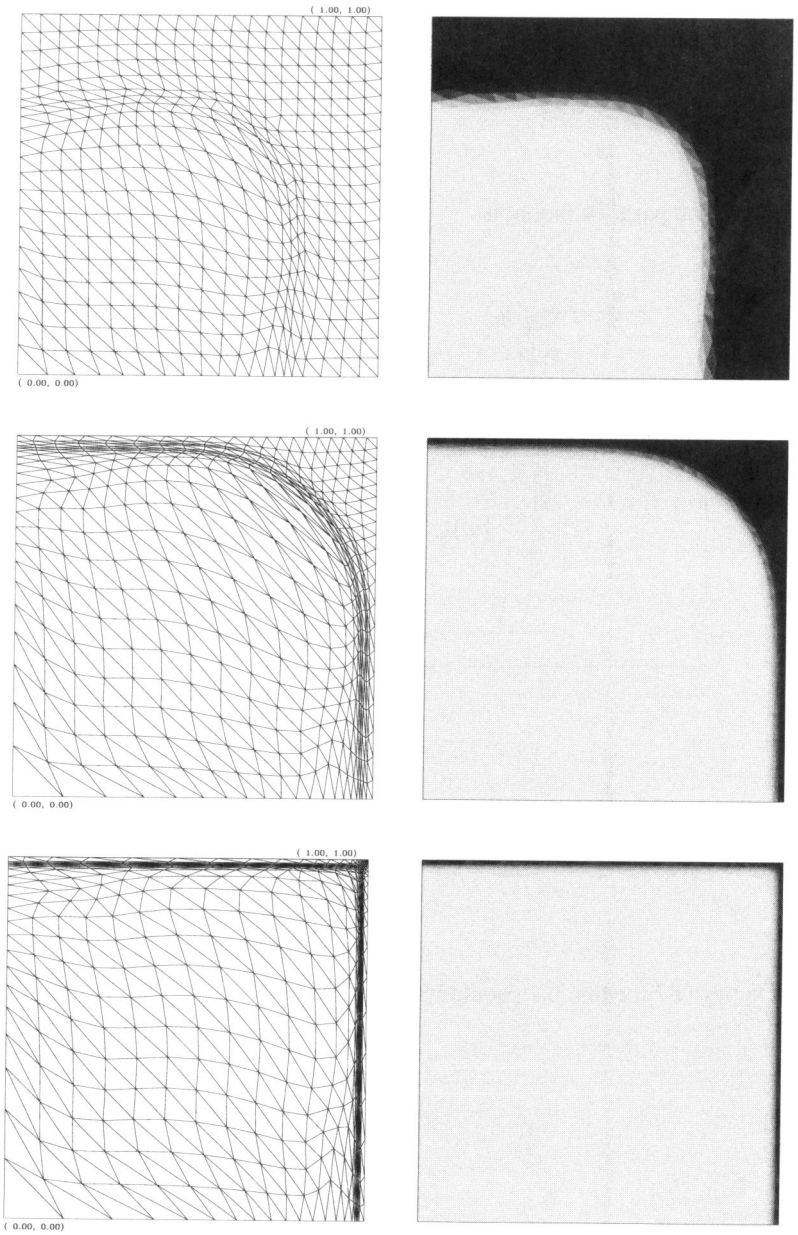
Errata

1. The figure on page 74 should be



2. On page 87 the first 0.1 should be –0.1.

3. Figure 7.3 on page 154 should be



Stellingen behorende bij het proefschrift

Moving-Grid Methods for Time-Dependent Partial Differential Equations

van P.A. Zegeling

1. De 2D adaptieve roostermethode van Brackbill & Saltzman (zie ref. [11] in hoofdstuk 1 van dit proefschrift) wordt beschreven door een stelsel elliptische differentiaalvergelijkingen. Indien we deze methode beperken tot 1 ruimtedimensie, dan definieert de equidistributierelatie $x_\xi w = constant$ (zie bijv. hoofdstuk 2) een integraalkromme voor dit stelsel.

2. Het formeel minimaliseren van het PDV-residu (zie hoofdstuk 4), dat bij de MFE-methode de roosterbeweging genereert, levert altijd een (locaal) minimum op.

3. Zij $\mathcal{A}(\varepsilon_1^2,\eta)\dot{\eta} = G(\varepsilon_2^2,\varepsilon,\eta)$ voor $\varepsilon_1^2 \neq 0$, $\varepsilon_2^2 \neq 0$, $\varepsilon > 0$, het semi-discrete stelsel, dat wordt verkregen door het toepassen van de MFE methode op de vergelijking van Burgers $\frac{\partial u}{\partial t} = \varepsilon \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x}$ met homogene Dirichlet randvoorwaarden. Dan geldt:

i) $\Delta X_i = h := \frac{1}{N+1}$, $U_i = 0$ $(\forall i)$ is een asymptotisch stabiele stationaire oplossing van het semi-discrete stelsel. Na linearisering van dit stelsel worden de $2N$ eigenwaarden gegeven door:

$$\lambda_{1,k} = -\frac{12\varepsilon}{h^2}\frac{\sin^2(k\pi h/2)}{2-\cos(k\pi h)}, \qquad k=1, \cdots, N,$$

$$\lambda_{2,k} = -\frac{2\varepsilon_2^2}{\varepsilon_1^2 h^2}, \qquad k=1, \cdots, N;$$

ii)

$$\dot{X}_i = O(\frac{1}{\varepsilon_1^2}) \quad (\forall i) \quad \text{voor } \varepsilon_1^2 \to \infty,\ \varepsilon_2^2 = O(1) \quad \text{(een vast rooster);}$$

iii)

$$\Delta X_i = h + O(\frac{1}{\varepsilon_2^2}) \quad (\forall i) \quad \text{voor } \varepsilon_1^2 = O(1),\ \varepsilon_2^2 \to \infty \quad \text{(een uniform rooster).}$$

4. Voor de Burgers' vergelijking zonder diffusie (ε=0) is de gradient-weighted MFE methode (zie hoofdstuk 4) identiek met de MFE methode. De gradient-weighting heeft dus slechts invloed op het 'diffusiegedeelte' ($\varepsilon \neq 0$) van de roosterbeweging. Een asymptotische roostersnelheid $\dot{x}$ wordt in dit geval gegeven

door:

$$\dot{x} = u + \varepsilon\left\{ 2\frac{u_{xxx}}{u_{xx}} - 3\frac{\xi_{xx}}{\xi_x} + 3\frac{w(u_x)_x}{w(u_x)}\right\}.$$

5. Laat gegeven zijn de 'gesmoothde' equidistributierelatie $\tilde{n}_i/M_i = constant$ met $\tilde{n}_i := n_i - \kappa(\kappa+1)(n_{i+1}-2n_i+n_{i-1})$, $n_i := 1/\Delta X_i$ en $\kappa > 0$ (zie hoofdstuk 3). Er geldt voor iedere positieve monitorfunctie M_i:

$$\frac{\kappa}{\kappa+1} \le \frac{\Delta X_{i+1}}{\Delta X_i} \le \frac{\kappa+1}{\kappa}, \ \forall i.$$

In het bijzonder volgt hieruit dat 'teveel smoothing' ($\kappa \to \infty$) een uniform rooster geeft.

6. Er bestaan minstens twee niet-triviale stationaire oplossingen $(T(z), P(z))$ voor $z \in [0,1]$ die voldoen aan het 'coronale-lus' model:

$$\varepsilon\frac{P}{T}\frac{d}{dz}(T^{3/2}P\frac{dT}{dz}) + \alpha - P^2\chi(T) = 0,$$

$$\frac{dP}{dz} = -Mg \ , \ T(0)=T_0, \ \frac{dT(1)}{dz}=0, \ \int_0^1 \frac{P}{T}dz = \mathcal{L}_0.$$

(F. Verhulst & P.A. Zegeling, Math. Meth. Appl. Sc., V13, 431-439, 1990)

7. Houtje-touwtje-methoden hebben het voordeel, dat ze gemakkelijk aan elkaar te knopen zijn. Het nadeel is echter, dat ze vaak rammelen.

8. Het gebruik van ' '-tjes en " "-tjes verhult het onvermogen exact te formuleren en 'bevordert' de "duidelijkheid".

9. Gezien de vele ups en downs bij het uitvoeren van wetenschappelijke experimenten zou men beginnende OIO's en AIO's kunnen adviseren 'niet te vroeg te juichen, maar ook niet te snel in de put te gaan zitten'. Daarom zou een doos met dobbelstenen dan ook niet alleen tot de "standaarduitrusting" van een statisticus, maar van elke onderzoeker moeten behoren.

10. Overeenkomstig de suggestie van Platonov verdient het aanbeveling de eerste in plaats van de vijfde set van een volleybalwedstrijd te spelen volgens het zogenaamde rallypoint systeem.

11.