Application of Adaptive Methods Based on Finite Difference Discretizations to Systems of PDAEs

P. M. P. Brito and A. A. T. G. Portugal *
Departamento de Engenharia Química
Largo Marquês de Pombal
3000 Coimbra, Portugal

Keywords: Adaptive methods, grid refinement, moving nodes, partial differential equations, finite difference approximations

AMS classification: 65M50

Abstract

In this paper two adaptive algorithms are presented for the solution of systems of evolutive one-dimensional Partial Differential/Algebraic Equations (PDAEs).

A spatial discretization based on finite difference approximations on arbitrarily spaced grids, transforms the original problem in a set of Ordinary Differential Equations (ODEs), solved via an implicit integrator package (DASSL). The temporal integration is coupled with a spatial adapting strategy. The identification of the spatial subdomains, where the introduction of grid adaptivity is needed, is done through the comparison of the solutions computed with two fixed grids of different sizes. The subproblems generated are solved by two adaptive strategies: the Grid Refinement Method (GRM), that refines the subgrids detected in the previous step, and the Moving Mesh Method (MMM), that includes an additional differential equation for the nodal mobility in each original subproblem.

In this paper, these algorithms were successfully applied to the solution of two problems: an isothermal tubular reactor model and a flame propagation system described by two PDEs referring to fuel mass density and temperature dynamics. The performance of each algorithm is compared to the results obtained by Duarte [1], based on the application of a formulation of the Moving Finite Elements Method, with cubic Hermite polynomials approximations. The MMM algorithm revealed its robustness in dealing with the chosen models. The GRM algorithm originated poorer results, mainly due to errors associated with the boundary conditions procedure.

1 Introduction

Several problems in Engineering can be properly simulated by the solution of evolutive Differential/Algebraic Systems where the influence of diffusional/convective phenomena is very important. For the sake of simplicity, in this paper, we only study one-dimensional problems, but the specific algorithms developed here, are easily extended to multidimensional domains. In the case of hyperbolic systems, the weight of the convective terms is dominant. This factor may lead to the development of steep moving waves or discontinuities on the solution profiles.

Most of the numerical strategies used for the solution of this kind of problems are directly

*Author to whom correspondence should be addressed: e-mail: eq1atp@eq.uc.pt, phone: 351-39-828392, fax: 351-39-841138

inspired by the widely known Method of Lines: first, the derivatives over one of the independent variables (generally the spatial one) are estimated by algebraic formulas (in this case, finite difference approximations). So, the continuous domain is discretized to a grid of points, where the solution is computed. By now, the original PDE problem is transformed in a complex, but easier to solve, system of ODEs, that is integrated over the remaining independent variable (in this work the temporal one) by a numerical integrator software (the DASSL implicit BDF formula package [5]).

The finite difference weights are estimated by a recursive scheme developed by Fornberg [2], for arbitrarily spaced grids, and by a strategy inspired by Schiesser [7], for the evaluation of weights associated with Neumann boundary conditions.

When the solution develops very large spatial gradients, in areas that move with time, the overall grid has to be very dense, to reproduce correctly the numerical results without introducing large numerical instability, which leads to unreasonable computational times. In these cases, such problems can be overcome by the introduction of mobility criteria for the positions of the nodes in the grid, on the regions of the spatial domain where the solution activity is higher (the spatial solution gradients are larger) and the advance of the temporal integration is more difficult. Therefore, the grid adapts itself to the specific characteristics of the solution in each region of the domain, and these type of algorithms are designated by Adaptive Methods.

In this paper, two adaptive algorithms are developed, that basically apply two important regridding techniques, widely studied by several authors:

- Grid Refinement and Relaxation [3] - Introduction of additional nodes and elimination of useless ones from an initial grid. Based on a estimation discretization error procedure for each time step, several grids are constructed with various sizes or refinement levels through all the spatial domain, over which the problem is solved. Nodes are added in the areas of major solution activity (Grid Refinement) and removed from regions where the spatial gradients are lower (Mesh Relaxation).

- Dynamical Node Displacement [4, 6] - Using selected properties of the solution, additional partial differential equations are deduced, that describe the movement of the nodal positions during the solution of the problem. These equations are computed together with the original differential system, that obviously has to be transposed to the related dynamical coordinate set of independent variables.

2 Adaptive Numerical Algorithms

In this paper, two adaptive mesh algorithms are described for one-dimensional evolutive systems of Algebraic-Differential Equations that can be resumed by the following general model:

$$F(\underline{u}_t, \underline{u}, \underline{u}_z, \underline{u}_{zz}) = 0 \tag{1}$$

$$G(\underline{u}) = 0 \tag{2}$$

subjected to the boundary conditions:

$$\underline{u}(z^L, t) = \underline{u}^L(t) \tag{3}$$

$$\underline{u}(z^R, t) = \underline{u}^R(t) \tag{4}$$

and the initial condition:

$$\underline{u}(z, 0) = \underline{u}^0(z); \qquad z \in [z^L, z^R] \tag{5}$$

Both algorithms are structured in two main stages:

- Stage I - Estimation of the spatial discretization error and identification of the subdomains where adaptation is needed.
- Stage II - Solution of the subproblems generated in the former stage, by the introduction of an adaptive grid technique.

2.1 Stage I - Discretization Error Estimation

This stage is similar in both algorithms and it is based on the comparison of the solution obtained by solving the original problem on two different grids: a fine and a coarse grid (Grids of level 2 and 1, respectively). Initially, the fine grid is constructed by the bissection of each interval of the coarse one. The nodes in the level 1 grid, that do not satisfy the error criterium, are grouped together with the level 2 nodes placed between them, to form the subdomains over which the adaptive subproblems are generated and then solved.

2.2 Stage II - Adaptive Integration of the Subproblems

The second stage refers to the actual adaptive procedure that is different for each algorithm studied in this work.

2.2.1 Grid Refinement Method (GRM)

This adaptive procedure is based on the work presented by Guiné [3], that generates subproblems of increasing refinement level, by repeating the procedure described in Stage I, until every node in every grid verifies the tolerance condition associated with the error estimated by the expression:

$$EU^i_{j,k+1} = Wh^i_{j,k+1} - W2h^i_{j,k+1}; \qquad j = 1, \cdots, NP_{n-1}, \qquad i = 1, \cdots, NPDE \tag{6}$$

In this case, $EU^i_{j,k+1}$ represents the approximation to the spatial error, in a node j of a grid of refinement level n; $Wh^i_{j,k+1}$ and $W2h^i_{j,k+1}$ are the approximations to the component i of the solution, obtained through integration between the times t_k and t_{k+1}, on the finer (level n) and the coarser (level n-1) grids, respectively; NP_{n-1} is the number of nodes in the grid of level n-1; and $NPDE$ is the number of partial differential equations of the problem.

The subdomains of level n+1 are obtained by joining all nodes n-1 that satisfy the tolerance condition:

$$| EU^i_{j,k+1} | > TOL_i; \qquad i = 1, \cdots, NPDE \tag{7}$$

In each refinement procedure, the profiles of the solution are computed by interpolation of the profiles of level 2, at all the intermediary positions.

2.2.2 Moving Mesh Method (MMM)

In this method, the subproblems are generated in Stage I and solved by a two step procedure:

1. Conversion of the problem to a moving set of coordinates by the relation:

$$\dot{u} = u_\tau + u_z \cdot \dot{z} \tag{8}$$

2. Inclusion of a moving grid differential equation in the transformed problem, that originates the dynamical problem that we want to solve.

The Moving Mesh Equations used here were deduced by Petzold [6], by generalization of a scheme developed by Hyman [4]. In this case, the velocities $\dot{z}$ are chosen to minimize the time rate of change of u and z in the new coordinates. The nodal movement is smoothed by the addition of a penalty function to the minimization, which attempts to give neighbouring nodes nearly equal velocities. Thus $\dot{z}$ for the node j satisfies:

$$\min_{\dot{z}_j} \left[\|\dot{u}_j\|_2^2 + \alpha \cdot \|\dot{z}_j\|_2^2 + \lambda \cdot \left(\left\| \frac{\dot{z}_j - \dot{z}_{j-1}}{z_j - z_{j-1}} \right\|_2^2 + \left\| \frac{\dot{z}_{j+1} - \dot{z}_j}{z_{j+1} - z_j} \right\|_2^2 \right) \right] \tag{9}$$

The quadratic equation in $\dot{z}$ can be minimized in each mesh point. Therefore, for $\lambda > 0$, (9) leads to

$$\alpha \cdot \dot{z}_j + \dot{u}_j \bullet u_{zj} + \lambda \cdot \left(\frac{\dot{z}_j - \dot{z}_{j-1}}{(z_j - z_{j-1})^2} - \frac{\dot{z}_{j+1} - \dot{z}_j}{(z_{j+1} - z_j)^2} \right) = 0 \tag{10}$$

Here, α is a positive scaling parameter, usually set to 1. The effect of the penalty term is similar to an extra diffusional factor that smoothes out differences in the mesh velocities and tries to keep points from crossing, but it does not entirely eliminates these crossings.

Additionally, it is introduced an adjustement of the time step to prevent node crossings and a final redefinition strategy of the level 2 base grid that locally refines the intervals where the spatial step exceeds a predetermined value: $\Delta z > \Delta z_{MAX}$, by equidistributing two additional nodes, or moves away nodes that get too close from one another: $\Delta z < \Delta z_{MIN}$.

2.3 Boundary Conditions Treatment in the Generated Subproblems

2.3.1 Introduction of Artificial Dirichlet Conditions (GRM)

The GRM algorithm is coupled with a strategy for the treatment of boundary conditions in the refinement subproblems that simply defines fixed Dirichlet conditions on each internal bound. Every spatial derivative is computed exactly as in the general problem and the positions of each bound, for the refinement level n+1 (for $n = 2, \cdots, N_{MAX} - 1$, where N_{MAX} is the maximum refinement level allowed in relation to the local spatial steps defined in the level 1 grid) are coincident with the positions of the first nodes of level n-1 that verify the specified tolerance. The constant value of the boundary conditions is given by the solution obtained in the integration over the level n-1 grid. This kind of procedure is very simple and prevents discontinuities on the overall profiles but tends to introduce significative errors in the solution.

2.3.2 Variable Boundary Conditions (MMM)

For the MMM algorithm, it is adopted a more complex procedure that allows a semi-free evolution of the solution on the internal boundaries. This temporal evolution is only constricted by the spatial derivatives estimation operation on the boundary (and internally adjacent) nodes, that uses the time evolution of the solution on the externally adjacent nodes to the subdomains, computed on the static integration step of the original problem. The selection of these nodes depends on the finite difference formula chosen and the relative position of the subdomain in the overall domain. The temporal profiles are approximated by linear interpolation. When possible, the estimation formula for the evaluation of the spatial derivatives is the same for every node in the subdomains (internal or boundary nodes). This procedure does not garantee the continuity of the overall profile because the solution over the bounds is allowed to change significantly (as it's computed by static or dynamic integration). So, this procedure is actually an iterative strategy, where convergence is only reached when a specific tolerance is verified on both boundaries of each dynamic subproblem.

The algorithms briefly described in this section are resumed in Figure 1.

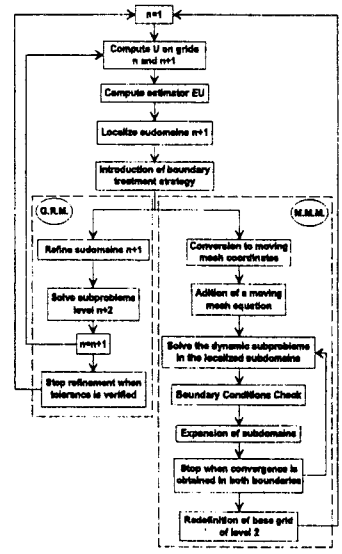

Figure 1: *Schematic resume of the adaptive numerical algorithms.*

3 Numerical Results

The two adaptive algorithms described in the previous section were applied to the solution of two problems: a escalar parabolic/hyperbolic P.D.E. and a system of two P.D.E.'s. The quality of the results, defined by the profiles precision and the computational effort demanded, is established by comparison with the results obtained with a formulation of the Moving Finite Elements Method (MFEM) developed by Duarte [1], based on Hermite polynomials approximations. The software designed for the application of every algorithm analised in this work, was executed in the same computer, a *Workstation* **SUN** *Sparcstation* of architecture **RISC** with 16 Mb of **RAM** memory.

3.1 Example 1: Plug-Difusional Reactor

This model [1] simulates the startup of an isothermal tubular reactor, subjected to a step disturbance in the reagent concentration of the feed stream. The reagent is consumed in a homogeneous first order reaction A $\rightarrow$ B and it is assumed that the influence of axial diffusion can not be neglected. Thus, the problem is the following microscopic mass balance:

$$\frac{\delta u}{\delta t} = \frac{1}{P\epsilon} \cdot \frac{\delta^2 u}{\delta z^2} - \frac{\delta u}{\delta z} - Da \cdot u \qquad (11)$$

with the boundary conditions: $\frac{\delta u(0,t)}{\delta z} = P\epsilon \cdot (u-1)$ and $\frac{\delta u(1,t)}{\delta z} = 0$,
and the initial condition: $u(z,0) = 0$.

Here, u is the concentration of component A and Da and $P\epsilon$ are the Damkhöler and Peclet adimensional numbers, respectively.

3.1.1 Grid Refinement Method

The Peclet's number ($P\epsilon$) quantifies the relation between the convective and difusional mechanisms that describe the movement of the reactive fluid inside the reactor. If the value of $P\epsilon$ is small, difusion is dominant, so the disturbance massic wave introduced at the initial time tends to spread through the spatial domain (the length of the reactor) with a comparative lower velocity and the total time needed for the system to reach a steady-state is longer.

For large values of $P\epsilon$ the flow is mainly convective, thus diffusion has limited influence on mass displacement and the fluid movement assumes mainly plug-flow behaviour. The wave propagates with hardly any distortion on its shape and the front gradients tend to be infinite due to the initial step disturbance injected in the system, in spite of being slightly smoothed by the little diffusion effect that still remains. In this case, $P\epsilon = 10^4$, thus the behaviour of the massic wave is mainly convective and the moving fronts generated are abrupt.

The conditions of the run are the following: centered finite difference formula with five points, for the spatial discretization; linear interpolations, for the evaluation of the solution on the higher level grids; method's tolerance fixed at 1×10^{-3}; maximum refinement level of $N_{MAX} = 9$; and the initial base grid of first level defined as an equidistributed grid with 41 nodes.

By the analysis of the results (vd. Figures 2 and 3) it can be noticed that the thickness of the fronts is slightly higher than expected, due to numerical dissipation. Overall the results are acceptable but demand an initial base grid relatively dense and still very slight oscillations are noticible on the upper section of the fronts that desappear with the time integration. The use of linear interpolation is more suitable to reproduce the large and brusque variations on the profiles spatial gradients, originated by the existence of the abrupt front.

The refinement activity of the base grids is very intense (vd. Figure 3) and the refinement profiles follow the propagation movement of the abrupt massic front.

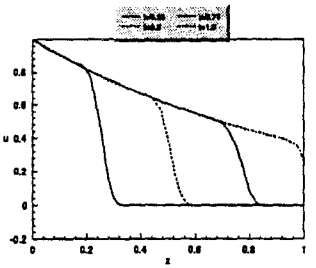

Figure 2: *Concentration profiles.*

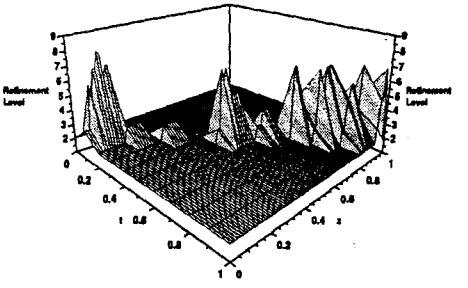

Figure 3: *Refinement distribution.*

3.1.2 Moving Mesh Method

The present example was also solved by the Moving Mesh Algorithm with the following conditions and parameters: five points centered spatial discretization; linear interpolations for the grid redefinitions; tolerance 1×10^{-4}; initial non-uniform first level grid with 16 nodes mainly concentrated on the feeding area ($z \approx 0$); $\Delta z_{MIN} = 1 \times 10^{-5}$ and $\Delta z_{MAX} = 7 \times 10^{-3}$; and viscosity factor $\lambda = 0.75$.

From the results obtained and presented in Figures 4 and 5, we notice some minor instability at the upper section of the fronts for earlier times. These disturbances are smoothed with time and eventually disappear for later steps. The method is able to develop very abrupt fronts that are similar to the ones presented by Duarte [1] using a MFEM formulation. The evolution of the dynamic second level grid in time (without considering any additional nodes introduced) is showed in Figure 5 and it is visible that the nodes follow the massic wave propagation.

The sudden variations of the spatial gradients are correctly reproduced by the linear interpolations. On the other end, it is verified that the definition of a small Δz_{MAX} is essential in order to maintain the stability of the upper section of the profiles near the front. Thus, the computational effort increases dramatically through the integration, due to the need of building very dense grids. So, the integration of the fixed grids in Stage I, becomes the limiting factor of the procedure, leading to relatively high computational times for the run execution.

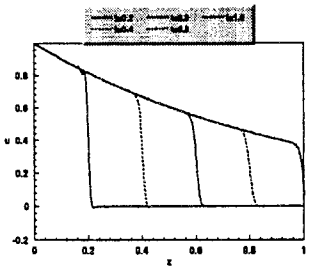

Figure 4: *Concentration profiles.*

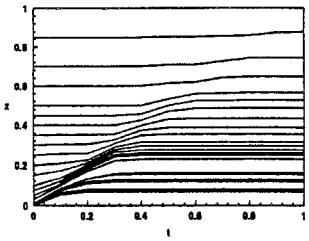

Figure 5: *Temporal evolution of the grid.*

3.1.3 Comparison of the Numerical Performance Obtained by Each Algorithm

In Table 1, we compare the numerical perfomance of the presented case (A) with a diffusional example (Case B - $Pe = 10^2$) that does not represent any major difficulty in its solution, due to the smoothness of the profiles. The GRM provides identical results than the MFEM, using a very small amount of computational effort. Thus, for the this example solved in the Case B conditions (mainly difusional problem), GRM reveals to be the most effective algorithm.

For Case A, GRM and MMM proved to be faster than the MFEM. However, it is visible the influence of numerical dissipation in the GRM results. On the other end, MMM gives very accurate results reproducing the shape and movement of the abrupt massic fronts and proving to be the most effective algorithm for the solution of this example.

Table 1: *Computational performances for example 1.*

Case	*Method*	*Tcpu(s)*
A	G.R.M.	7368.6
($Pe = 10^4$)	M.M.M.	7000.7
	M.E.F.M. [1]	10436.2
B	G.R.M.	707.4
($Pe = 10^2$)	M.E.F.M. [1]	5927.5

3.2 Example 2: Flame Propagation

The second problem [1, 6] is a model of flame propagation that consists of two coupled equations for mass density and temperature. Initially, a ramp disturbance in the temperature is introduced in the system, on the right boundary of the spatial domain. Therefore, the problem is described by the following partial differential system:

$$\frac{\delta u}{\delta t} = \frac{\delta^2 u}{\delta z^2} - 3.52 \times 10^6 \cdot u \cdot \epsilon^{-\frac{4}{v}} \quad (12)$$

$$\frac{\delta v}{\delta t} = \frac{\delta^2 v}{\delta z^2} + 3.52 \times 10^6 \cdot u \cdot \epsilon^{-\frac{4}{v}} \quad (13)$$

subject to the boundary conditions: $\frac{\delta u(0,t)}{\delta z} = 0$, $\frac{\delta v(0,t)}{\delta z} = 0$, $\frac{\delta u(1,t)}{\delta z} = 0$ and $v(1,t) = 0.2 + \frac{t}{0.0002}$ for $t \leq 0.0002$ or $v(1,t) = 1.2$ for $t > 0.0002$;
and the initial conditions: $u(z,0) = 1$ and $v(z,0) = 0.2$.

Here, u is the fuel mass density and v represents the flame temperature.

3.2.1 Grid Refinement Method

The problem presented above was solved by the GRM algorithm under the following conditions: five points centered finite difference formula for the spatial discretization on both variables (u and v); tolerance - 1×10^{-2} for both variables; linear interpolations; and an uniform first level base grid with 21 nodes.

The results obtained are presented in Figures 6–8. We notice that the profiles show two anomalous features: the propagation speed of the waves depends directly on the basic temporal step (Δt) and the thickness of the moving fronts is practically nill. The errors generate two opposing behaviours:

1. A visible delay in the wave movement, due to the excessive low density of spatial base grid ($NP_1 = 21$).

2. An artificial advance of the waves promoted by the anticipation of their effect over each refinement subproblem.

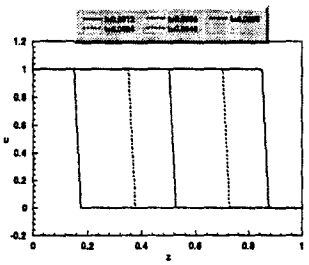

Figure 6: *Mass density profiles (u).*

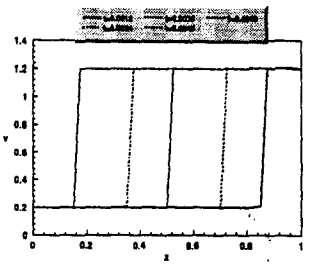

Figure 7: *Temperature profiles (v).*

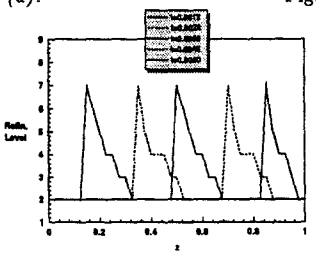

Figure 8: *Refinement distribution.*

The definition of fixed Dirichlet boundary conditions in the integration of the refinement subproblems over grids of level greater or equal than 3, induces the introduction of the moving fronts disturbance effect on the right boundary of these subdomains at the initial instant of each time step. Thus, by increasing the refinement level, the algorithm promotes a left displacement on the refinament profiles in relation to the original third level subdomain selected in Stage I (vd. Figure 8). Therefore, the procedure develops fronts of infinite gradient and pushes the waves to the left in each refinement operation, which accelerates their propagation speed.

In this case, for $\Delta t = 0.0012$ the effect 1 is dominant, but if we decrease the Δt, the error 2 becomes the most important factor. In fact, for $\Delta t = 0.0006$, the number of time steps (and

the number of refinement operations) is high enough to push the waves until they already reach the left boundary at $T_{final} = 0.0060$, which is manifestly incorrect.

3.2.2 Moving Mesh Method

The results obtained by the application of the MMM under the following conditions: five points centered discretizations for both variables; tolerance 5×10^{-3} for both variables; initial base grid nonuniform with 20 nodes, mainly concentrated near the left boundary; linear type interpolations; $\Delta z_{MIN} = 1 \times 10^{-4}$ and $\Delta z_{MAX} = 1 \times 10^{-2}$; and $\lambda = 0.6$, are depicted in Figures 9–12.

The algorithm reproduces correctly the wave shape and its movement (vd. Figures 9 and 10) and the propagation velocity is accurate. Again, the linear interpolations reveal to be the best strategy in dealing with this kind of abrupt moving fronts. There is a slight overshoot on the v profiles (vd. Figure 10) at earlier times, also visible in the results presented by Petzold [6] for this example. Besides that, the solution profiles are very similar to the ones presented by Duarte [1].

The dynamic grids (Figure 11 and 12) follow efficiently the wave propagation movement. Both grids, associated with each variable u and v are very similar but not entirely coincident, which significantly increases the total computational effort of the procedure, because it has to be repeated for each variable and grid.

Again, it is verified a considerable number of nodal additions in the initial base grids, during the advance of the temporal integration. As in the former problem, the static grids integration becomes the limiting step of the algorithm evolution.

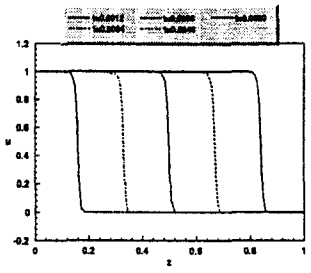

Figure 9: *Mass density profiles (u).*

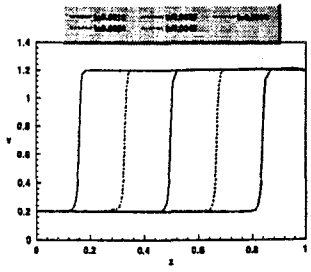

Figure 10: *Temperature profiles (v).*

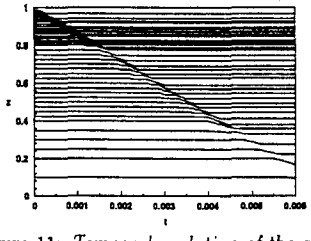

Figure 11: *Temporal evolution of the grid (u).*

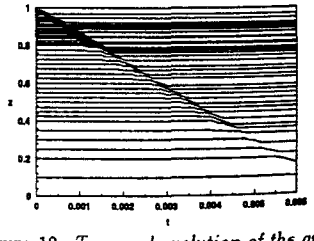

Figure 12: *Temporal evolution of the grid (v).*

3.2.3 Comparison of the Numerical Performance Obtained by Each Algorithm

In Table 2 we resume the computational times obtained for each algorithm. The GRM algorithm is very fast but the precision of the results obtained is not satisfactory. On the other hand the MMM results are very similar to the obtained by Duarte [1] with his formulation of the MFEM. However, the computational time for MMM is somewhat higher than the MFEM one. Therefore, we can conclude that, overall, the MEFM algorithm is more effective for this example

but it is important to remind that this is a significantly more complex procedure, from a formal perspective.

Table 2: *Computational performances for example 2.*

Method	*Tcpu(s)*
G.R.M.	526.7
M.M.M.	12788.6
M.E.F.M. [1]	8226.9

4 Conclusions

From the solution of the examples presentes in this work, we can conclude that the GRM algorithm reveals some difficulties in describing high gradient profiles and it may develop numerical dissipation and some instability, mainly due to the simplicity and imprecision of the Dirichlet boundary treatment strategy for the refinement subproblems. On the other hand, GRM is a very robust and efficient method for models that involve relatively smooth profiles.

The MMM algorithm is very suitable to reproduce moving abrupt fronts or waves. The results obtained are very exact with hardly any numerical instability. The boundary condition procedure coupled with the MMM algorithm, based on linear interpolations on time for the nodes near the subdomains' boundaries proved to be effective and exact.

As it was expected linear interpolations are the most suitable to deal with abrupt fronts characterized by large variations in space, on the solution profiles.

Acknowledgment: This work was supported by a grant from JNICT (Junta Nacional de Investigação Científica e Tecnológica).

References

[1] B. P. M. Duarte, "Método dos Elementos Finitos Móveis Aplicado à Resolução de Modelos de Frente de Reacção", *PhD Thesis*, Universidade de Coimbra, F. C. T., Coimbra, 1994.

[2] B. Fornberg, "Generation of Finite Difference Formulas on Arbitrarily Spaced Grids", *Math. of Comput.* **51** (1988) 699–706.

[3] R. P. F. Guiné, "Resolução de Sistemas de Equações Diferenciais de Derivadas Parciais — Algoritmos de Refinamento Espacial", *MSc Thesis*, Universidade de Coimbra, F. C. T., Coimbra, 1996.

[4] J. M. Hyman, "Adaptive Moving Mesh Methods for Partial Differential Equations", Los Alamos National Laboratory Report LA-UR-82-3690, Los Alamos, New Mexico 1982 – not published.

[5] L. R. Petzold, "A Description of DASSL: A Differential/Algebraic System Solver", Sandia Tech. Rep. 82-8637, 1982.

[6] L. R. Petzold, "Observations on an Adaptive Moving Grid Method for One-Dimensional Systems of Partial Differential Equations", *Appl. Numer. Math.* **3** (1987) 347–360.

[7] W. E. Schiesser, *The Numerical Method of Lines: Integration of Partial Differential Equations*, Academic Press Inc., San Diego, 1991.