Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 219 (2006) 86–103

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Cost-effectiveness of fully implicit moving mesh adaptation: A practical investigation in 1D

Giovanni Lapenta *, Luis Chacón

Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87544, USA

Received 17 October 2005; received in revised form 13 March 2006; accepted 17 March 2006
Available online 9 August 2006

Abstract

The cost-effectiveness of moving mesh adaptation is studied in a number of 1D tests. We propose a method that is based on two established modern techniques. First, we use a moving mesh approach based on the classic equidistribution method. Second, we discretize the model equations for grid and physics using a conservative finite volume method and we solve the resulting equations with a preconditioned inexact Newton–Krylov method.

Using these state of the art methods, we consider the question of whether a real improvement in performance can be achieved using adaptive grids. We consider rigorous metrics of the accuracy and cost of a numerical solution on uniform and adaptive grids. For a number of classic but challenging problems we demonstrate that indeed adaptive grids can lead to a great improvement in cost-effectiveness.

PACS: 02.70.Bf; 02.60.Cb; 02.60.Lj

Keywords: Adaptive grids; Implicit methods; Grid generation; Newton–Krylov; Fully implicit methods; Moving mesh

1. Introduction

Adaptive grids are becoming an ever more common tool for high performance scientific computing. We focus here on the type of adaptation achieved by moving a constant number of points according to appropriate rules, an approach termed moving mesh adaptation (MMA).

A great literature body exists on the subject of moving mesh adaptation, and we refer the reader to the excellent textbooks on the subject [1,2]. The approach we consider here is based specifically on retaining a finite volume approach but allowing the grid to evolve in time according to a grid evolution equation obtained from minimization principles. The approach originates from the seminal papers by Brackbill and Saltzmann [3] and by Winslow [4].

In the present paper we consider the fundamental question in the application of MMA. Is it worth the effort? The literature is very rich and considerable results have been obtained in designing MMA approaches

* Corresponding author. Tel.: +1 505 667 5493; fax: +1 505 665 3107.
E-mail address: lapenta@lanl.gov (G. Lapenta).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.03.011

that provide grids that can indeed present the desired properties [5–10]. But the question of whether once the adaptive grids are used the simulations are actually more cost-effective remains largely unanswered.

Some efforts are notable in the literature in trying to ascertain the cost-effectiveness of grid adaptation. For example, a study of geophysical flow solvers based on MMA [11] has shown that the use of adaptive grids within an explicit time discretization approach is only marginally advantageous in 1D and becomes more cost-effective only in 2D where the gains obtained by adaptation in each direction are compounded. This is a crucial consideration, any gains obtained in 1D can theoretically be squared in 2D and cubed in 3D. Of course, in practice, complexities added by 2D and 3D reduce the theoretical gain, but the fact remains, as demonstrated in [11], that 1D adaptation presents the fundamental challenge in proving cost-effectiveness of grid adaptation. However, unlike Ref. [11], the present study relies on implicit time discretization and significant gains are demonstrated even in 1D.

We intend here to revisit the question of cost-effectiveness of adaptive grids in 1D problems bringing to bear the latest developments in modern numerical analysis. Three issues are key in affecting cost-effectiveness of grid adaptation.

First is the formulation of the moving grid equations. In 1D the problem is benign, as error minimization leads to error equidistribution and to a rigorous and simple minimization procedure [2]. However, in 2D and 3D the problem is more serious. In the present paper the focus is on 1D problems, but the approach followed here can be extended to higher dimensionality [12] using the classic approach by Brackbill–Saltzmann–Winslow [3,4]. A crucial feature of our approach is the formulation of the physics equations in a conservative form. The independent variables of the physics equations are changed from the physical to the logical space and the equations are rewritten in the logical space in a fully conservative form [2]. In a separate paper we consider the extension of the implicit MMA methods to 2D problems [12].

Second is the solution algorithm for the MMA method. Here we bring a new development. The moving mesh equations and the physics equations, derived by discretizing the problem under investigation on a moving grid, form a tightly coupled system of algebraic non-linear equations. Traditionally, the coupling is broken, the physics and grid equations being solved separately in a lagged time-splitting approach. Each time step is composed of two alternating steps: the physics equations are solved on the current grid, the grid equations are then solved using new information from the solution of the physics equations. However, in presence of sharp fronts or other moving features, breaking such coupling can lead to grid lagging with respect of the physics equations, with adaptation resulting behind rather than on the moving feature.

We avoid breaking the coupling and solve the full non-linear set of physics and grid equation using the preconditioned Newton–Krylov (NK) approach [13]. Fully non-linear Newton methods have been used before in different frameworks for MMA such as in the moving finite element approach [14], in finite difference approaches [7,5,8], particularly for radiation-hydrodynamics [6,9].

Third ingredient in a cost-effective grid adaptation is good error indicators. Among the three ingredients this appears, to our subjective reading of the literature, as the less mature of the three. Keeping our helm fixed on the goal of probing the cost-effectiveness of the MMA method, we decided not to rely on unproven experimental error indicators. We focus instead on proven albeit heuristic measures widely used in the literature, such as the arc-length or the curvature error indicators [15].

The rest of the paper is organized as follows. Section 2 describes the moving mesh formalism used. Section 3 describes a model advection–diffusion equations used in the investigation as well as its finite volume discretization. Section 4 describes our novel solution procedure based on modern preconditioned NK methods. Section 5 reports the results of our cost-effectiveness analysis for a suite of classic single and multiple scale problems. Conclusions are drawn in Section 6.

2. Moving mesh adaptation equations

Adaptation based on a fixed number of points and a fixed connectivity among the points can be obtained using the concept of mapping between a logical uniform space and the actual physical space. A large literature body is available on this approach, e.g. Ref. [2]. In multi-dimensions, the problem can only be tackled at a complex mathematical level. In 1D, instead, a simple derivation suffices. Below, we restrict the attention to 1D systems, but using methods that have been extended to 2D [3,16,17,12].

We derive adaptive grids by studying generating equations for the mapping $x(\xi)$ between a logical space ξ and the physical space x. On the logical space, the grid spacing is always uniform, and without loss of generality can be assumed unitary. The actual spacing on the physical grid is then defined by the mapping $x(\xi)$. Under this approach, the grid generation equations define the mapping.

Using the assumption of 1D geometry, we start by formulating the error functional as

$$\int_0^L \mathscr{M}(x)\mathrm{d}x = \int_0^1 \mathscr{M}[x(\xi)]\frac{\mathrm{d}\xi}{\mathrm{d}x}\mathrm{d}x \tag{1}$$

The monitor function $\mathscr{M}$ is a suitable measure or indication of the local numerical error in the solution. We assume a strictly positive definite monitor function $\mathscr{M} \in \mathbb{R}^+$.

The equation for the mapping follows with a straightforward textbook Euler–Lagrange minimization procedure [2,17] of the error functional

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\mathscr{M}\frac{\mathrm{d}x}{\mathrm{d}\xi}\right) = 0 \tag{2}$$

In 1D an obvious property follows by integrating the Euler–Lagrange equation

$$\mathscr{M}\frac{\mathrm{d}x}{\mathrm{d}\xi} = \text{constant} \tag{3}$$

It follows that in 1D minimization of the error functional is achieved on a grid where the volume integral of the monitor function is equidistributed:

$$\int_c \mathscr{M}\mathrm{d}x = \text{constant} \tag{4}$$

where the integral is conducted over any cell in the system.

The simplicity of the equidistribution property even in 1D is deceptive, as in general the monitor function $\mathscr{M}$ depends non-linearly on the unknown mapping $x(\xi)$ and on the solution of interest. A crucial property of the method is that any minimization procedure needs to acknowledge that $\mathscr{M}[x(\xi)]$ changes as the mapping changes. Indeed the error depends on the discretization scheme. As the mapping changes, the grid spacing changes and the error changes. Therefore, computing the minimization, i.e. solving the Euler–Lagrange equations, must include the fundamental fact that the monitor function $\mathscr{M}[x(\xi)]$ depends non-linearly on the solution. If the solution has discontinuities, the monitor function is likely to be discontinuous as well.

In the formulation above, the grid generation equation is elliptic in nature and results in a grid that responds instantly to any change of $\mathscr{M}[x(\xi)]$. In time-dependent problems, it is generally preferred to deploy grids that change in time smoothly.

For these two reasons, complexity and ellipticity, the grid generation equation is usually modified by introducing a time evolution to it. In time-dependent problems, the mapping is governed by a time evolution equation as any other field. In stationary problems, a time-like parameter is often introduced just for the sake of simplifying the solution of the grid equations. In the present paper, we will focus on time-dependent problems and will not address the case of time-independent problems, referring the reader to the extensive literature on the subject [1].

A number of alternatives exist for the parabolization of the equation for the mapping, some of which (and their stability) are considered in [10]. Among the stable choices proposed in [10], we use

$$\frac{\partial}{\partial\xi}\left(\mathscr{M}\frac{\partial\dot{x}}{\partial\xi}\right) = -\frac{1}{\tau}\frac{\partial}{\partial\xi}\left(\mathscr{M}\frac{\partial x}{\partial\xi}\right) \tag{5}$$

Three desirable properties can be observed directly. First, a time scale τ is introduced to control the response of the grid to changes of $\mathscr{M}$. Second, in the limit of $\tau \to 0$ the original elliptic equation is obtained correctly. Third, discontinuities in $\mathscr{M}$ or in its time variation would not result in discontinuities in the time evolution of the mapping. The last property is the primary reason for choosing Eq. (5) over other alternatives suggested in [10]. We note that this choice of the discretization scheme results in an effective time smoothing of the discontinuities in the monitor function [18].

The grid generation equation can be discretized using elementary methods since the logical grid is uniform. We label the cell edges as i and the cell centers as $i+1/2$, and label n and $n+1$ the start and end of the time step, respectively. A spatial staggered grid is used: the derivatives appearing in Eq. (5) are computed using the grid nodes x_i, while the monitor function is computed in the cell centers $\mathscr{M}_{i+1/2}$. A time centered scheme is used.

We define

$$\begin{aligned} d^n_{i+1/2} &= x^n_{i+1} - x^n_i \\ \overline{\mathscr{M}}_{i+1/2} &= \tfrac{1}{2}(\mathscr{M}^n_{i+1/2} + \mathscr{M}^{n+1}_{i+1/2}) \end{aligned} \tag{6}$$

where we note that the monitor function is computed as the average between old and new time steps.

The grid evolution equation is discretized as

$$\begin{aligned} &(\overline{\mathscr{M}}_{i+1/2}d^{n+1}_{i+1/2} - \overline{\mathscr{M}}_{i-1/2}d^{n+1}_{i-1/2}) - (\overline{\mathscr{M}}_{i+1/2}d^{n}_{i+1/2} - \overline{\mathscr{M}}_{i-1/2}d^{n}_{i-1/2}) \\ &\quad = -\frac{\Delta t}{2\tau}\left((\overline{\mathscr{M}}_{i+1/2}d^{n+1}_{i+1/2} - \overline{\mathscr{M}}_{i-1/2}d^{n+1}_{i-1/2}) + (\overline{\mathscr{M}}_{i+1/2}d^{n}_{i+1/2} - \overline{\mathscr{M}}_{i-1/2}d^{n}_{i-1/2})\right) \end{aligned} \tag{7}$$

In the derivation above, it has been assumed that a suitable definition of the monitor function $\mathscr{M}$ is available. A robust algorithm for grid adaptation relies on a good definition of the error measure. Inappropriate choices of the monitor function can lead to results where the adaptive grid is inferior even to the uniform grid [17]. Our future work will investigate the use of different error estimators [19–21].

In the present work, the monitor function is based on simple but widely used heuristic monitor functions. In particular, unless specified otherwise, we will rely on the arc-length

$$\mathscr{M}^n_{i+1/2} = \sqrt{1 + \left(\frac{\psi^n_{i+1} - \psi^n_i}{x^n_{i+1} - x^n_i}\right)^2} \tag{8}$$

where a physical field ψ is used, the choice of ψ being based upon to the physics under investigation.

3. Model physics equation

The problems considered below belong to the general class of conservative advection–diffusion equations:

$$\frac{\partial}{\partial t}(\rho\phi) = \frac{\partial}{\partial x}\left(\epsilon\frac{\partial E(\phi)}{\partial x}\right) - \frac{\partial}{\partial x}(\rho\phi u) + S \tag{9}$$

where $E(\phi)$ is a general non-linear function of ϕ.

Once the transformation from physical to logical coordinates is made, the equation still in conservation form becomes [2]

$$\frac{\partial}{\partial t}(J\rho\phi) = \frac{\partial}{\partial \xi}\left(\frac{\epsilon}{J}\frac{\partial E(\phi)}{\partial \xi} - \rho\phi(u-w)\right) + JS \tag{10}$$

where the grid velocity is $w = \mathrm{d}x/\mathrm{d}t$. The Jacobian is $J = \partial x/\partial\xi$.

A crucial feature of our approach is the discretization of the physics equations in a Eulerian frame in the logical coordinates. The logical coordinates allow a conservative discretization of the equations, where the information about grid motion is all included in the Jacobian. The approach described below in 1D has been generalized to multiple dimensions [2,12].

Equations of the form above are discretized implicitly in time, using the Implicit Euler ($\theta = 1$) or the Crank–Nicolson scheme ($\theta = 1/2$). We define the independent variable on the vertices ξ_i and discretize the equation in each unitary cell with center $\xi_{i+1/2}$. The discretized equation becomes

$$\rho^{n+1}_i\phi^i_{n+i}J^{n+1}_i - \rho^n_i\phi^n_iJ^n_i = \Delta t(F^{n+\theta}_{i+1/2} - F^{n+\theta}_{i-1/2}) + J^{n+\theta}_iS^{n+\theta}_i\Delta t \tag{11}$$

where the superscript $n+1$ (n) stands for the end (start) of the time step and quantities computed at time level $n+\theta$ are averaged:

$$\psi^{n+\theta} = (1-\theta)\psi^n + \theta\psi^{n+1} \tag{12}$$

where ψ is any generic grid quantity.

The fluxes $F_{i+1/2}$ are obtained by discretizing the spatial operators. For the second-order diffusion operator we use the centered difference scheme

$$F_{i+1/2} = \frac{\epsilon_{i+1/2}}{J_{i+1/2}}(E(\phi_{i+1}) - E(\phi_i)) \tag{13}$$

where the time level is not written explicitly. The cell-centered diffusion coefficient is computed using the geometric average

$$\epsilon_{i+1/2} = 2\frac{\epsilon_{i+1}\epsilon_i}{\epsilon_{i+1}+\epsilon_i} \tag{14}$$

For the first-order advection operator we use the ZIP difference scheme

$$F_{i+1/2} = \tfrac{1}{2}(\rho_i\phi_i(u_{i+1} - w_{i+1}) + \rho_{i+1}\phi_{i+1}(u_i - w_i)) \tag{15}$$

The choice is dictated by the properties of non-linear stability and second-order accuracy of the ZIP algorithm [22].

In some examples, more advanced non-oscillatory discretizations of the first-order operator have been used in space as noted below.

4. Solution methodology

The physics equations and the grid motion equation form a single non-linear problem. Two approaches are common in the literature.

First, the physics equations are solved separately from the grid equations. In each time step the physics equations are solved on a given grid, and the grid equations are solved using the information available after solving the physics equations [3]. The approach breaks the link between the two and can lead to a lagging of the grid with respect to the evolution of the system. For example, in presence of a moving discontinuity the approach can lead to a grid that lags behind the discontinuity. This effect is particularly troubling for methods attempting to use large time steps, as implicit methods. A number of improvements [1] can be designed to circumvent such troubles but the fundamental limitation of the lagging approach remains.

Second, the coupled system can be solved as such, with advanced numerical methods [5]. Here lagging is removed altogether but at the cost of handling the complex non-linear coupling among the physics and grid equations.

Here we adhere to the second approach and we solve the coupled set of non-linear equations using the preconditioned NK method.

A complete description of the NK method is beyond the scope of the present paper. Here we give a brief introduction to NK methods focusing on the specific details of the implementation we use in the examples below. We refer the reader to textbooks on the subject [13] for more extensive descriptions of the approach. Newton–Krylov methods solve a given non-linear system $\mathbf{G}(\mathbf{x}) = \mathbf{0}$ iteratively by solving successive linear systems of the form

$$\left.\frac{\partial \mathbf{G}}{\partial \mathbf{x}}\right|_k \delta\mathbf{x}_k = -\mathbf{G}(\mathbf{x}_k)$$

with $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta\mathbf{x}_k$.

Non-linear convergence is achieved when

$$\|\mathbf{G}(\mathbf{x}_k)\|_2 < \epsilon_\mathrm{a} + \epsilon_\mathrm{r}\|\mathbf{G}(\mathbf{x}_0)\|_2 = \epsilon_\mathrm{t} \tag{16}$$

where $\|\cdot\|_2$ is the ℓ_2-norm (euclidean norm) and $\mathbf{G}(\mathbf{x}_0)$ is the initial residual. In the present study we choose $\epsilon_\mathrm{a} = \sqrt{N} \times 10^{-15}$ (with N the total number of degrees of freedom) for the absolute tolerance to avoid converging below roundoff (however, this issue actually never arises here and $\epsilon_\mathrm{a} = 0$ could be used). The Newton relative convergence tolerance ϵ_r is set to 10^{-7} in this work.

Such linear systems are solved iteratively with Krylov methods (specifically GMRES [23] in present paper), which only require matrix–vector products to proceed. Because the linear system matrix is a Jacobian matrix, such matrix–vector products can be implemented Jacobian-free using the Gateaux derivative

$$\left.\frac{\partial \mathbf{G}}{\partial \mathbf{x}}\right|_k \mathbf{v} = \lim_{\epsilon\to 0} \frac{\mathbf{G}(\mathbf{x}_k + \epsilon\mathbf{v}) - \mathbf{G}(\mathbf{x}_k)}{\epsilon} \tag{17}$$

where in practice a small but finite ϵ is employed [13]. Thus, the evaluation of the Jacobian-vector product only requires the function evaluation $\mathbf{G}(\mathbf{x}_k + \epsilon\mathbf{v})$, and there is no need to form or store the Jacobian matrix.

An inexact Newton method [24] is used to adjust the convergence tolerance of the Krylov method at every Newton iteration according to the size of the current Newton residual, as follows:

$$\|J_k\delta\mathbf{x}_k + \mathbf{G}(\mathbf{x}_k)\|_2 < \zeta_k\|\mathbf{G}(\mathbf{x}_k)\|_2 \tag{18}$$

where ζ_k is the inexact Newton parameter and $J_k = \frac{\partial \mathbf{G}}{\partial \mathbf{x}}|_k$ is the Jacobian matrix. Thus, the convergence tolerance of the Krylov method is loose when the Newton state vector $\mathbf{x}_k$ is far from the non-linear solution, but tightens as $\mathbf{x}_k$ approaches the solution. Hence, the linear solver works the hardest when the Newton state vector is closest to the non-linear root. Superlinear convergence rates of the inexact Newton method are possible if the sequence of ζ_k is chosen properly [13]. Here, we employ the modified Eisenstat–Walker formula [25,13].

A further advantage of Krylov methods is that they can be preconditioned by considering the alternate (but equivalent) systems $J_k P_k^{-1} P_k \delta\mathbf{x}_k = -\mathbf{G}_k$ (right preconditioning) or $P_k^{-1} J_k \delta\mathbf{x}_k = -P_k^{-1}\mathbf{G}_k$ (left preconditioning). Such preconditioning step can be straightforwardly and efficiently implemented in the Krylov algorithm as two consecutive matrix–vector products.

A crucial feature of preconditioning is that, while it can substantially improve the convergence properties of the Krylov iteration if $P_k^{-1} \approx J_k^{-1}$, it does not alter the solution of the Jacobian system upon convergence (because the solution $\delta\mathbf{x}_k$ of the preconditioned system is the same as that of the original system). Therefore, one can explore suitable approximations in the preconditioner for efficiency purposes without compromising the accuracy of the final result.

In the present paper we rely to a simple right preconditioning approach. We define as our preconditioning matrix the block tridiagonal matrix formed by the main diagonal and the two adjacent diagonals of each block of the Jacobian. The blocks are given by the grid equation and each physics equation taken separately. The choice of a block tridiagonal preconditioning matrix simplifies its inversion.

The three diagonals of each block are computed numerically using the Gateaux derivative (17) with a displacement vector appropriately selected with non-zero unitary elements only corresponding to the desired elements of the Jacobian. Note that the computation of each diagonal of each block can be performed with just one function evaluation. Therefore only three matrix–vector multiplications are needed. We simplify the preconditioning technique further by approximating P_k with P_0 computed with the initial guess for the Newton iteration, thus avoiding the need to recompute the preconditioning matrix at every iteration.

5. Efficacy analysis in a suite of sample problems

Based on the approach described above, we intend to investigate whether the adaptive simulations can result in a more efficient approach. To this end we focus on two fundamental issues.

First, we settle on a number of classic problems widely used in the literature. We choose four in particular: a radiation diffusion problem, Burger's equations, the advection equation and Fisher's equation. These choices include both multiple spatial and time scale problems (radiation diffusion, Burger and Fisher) that develop steepened fronts, and single scale more benign problems (advection).

Second, for each problem we use the same diagnostics designed to probe in detail not only the accuracy of the adaptive moving mesh approach but also its cost-effectiveness. We use four types of diagnostics.

First is CPU time, measured in seconds on an Intel Pentium 4 CPU 2.00 GHz computer with 1 GB of RAM operating under the Windows XP operating system. All tests are written in Fortran 90 and are compiled using the Lahey Fortran LF95 compiler, version 5.7.

Second is the average number of Newton and Krylov iterations required by the preconditioned non-linear solver.

Third is accuracy defined as the L_2 norm of the difference between a reference solution and the computed solution. The reference solution is computed either from a fiducial solution on a very refined grid or from an analytical solution. In both cases the reference solution is available on a refined grid, different from the grid where the problem is being solved. To compute the L_2 norm, we interpolate linearly from the current grid where the solution is computed to the more refined grid where the reference solution is available. The norm is then computed on the reference grid comparing the interpolated solution with the reference solution. The error is computed at the end of the run.

Fourth is efficacy defined as the inverse of the product of the error defined above and of the CPU time. The efficacy is a direct measure of how accurate a solution is per unit CPU cost. The efficacy is larger for more accurate solutions obtained at a smaller cost.

The key goal of the present paper is to determine whether the adaptive grid solution can obtain a larger efficacy than the uniform grid. This means that the adaptive grid leads to a more accurate solution for the same cost of a solution on a uniform grid, or to the same accuracy for a reduced cost. Efficacy is a direct measure of cost-effectiveness and is the key parameter to decide in practice whether a uniform or an adaptive grid leads to a more economical solution method.

5.1. Radiation diffusion

A simple form of the radiation diffusion equation that assumes equilibrium between the matter temperature ϕ and radiation energy $E=\phi^4$ can be written in non-dimensional units as [26]

$$\frac{\partial}{\partial t}(\alpha\eta E+(1-\alpha)\phi)=\frac{\partial}{\partial x}\left(D(\phi)\eta\frac{\partial E}{\partial x}\right) \tag{19}$$

where we choose $\alpha=0$ and $\eta=1$ to consider a matter-dominated system. The diffusion coefficient is $D(\phi)=\phi^3/3$ and we apply a diffusion flux limiter to avoid infinite speed of radiation propagation [26]

$$D_L=\left(\frac{1}{D}+\frac{1}{E}\left|\frac{\partial E}{\partial x}\right|\right)^{-1} \tag{20}$$

The initial condition is $\phi=1-0.8x/L$ and the boundary conditions are $\phi(0)=1$ and $\phi(L)=0.2$ with $L=1$.

Clearly, the radiation equation is of the standard form (9) above, and is discretized as described in Section 3 above, using the implicit Euler method in time. In space, the first-order operator is handled with ZIP differencing, and the second-order operator with centered differencing. In the computation of the limited diffusion coefficients we use

$$D_{i+1/2}=\left(\frac{1}{D_{i+1/2}}+2\left|\frac{(E_{i+1}-E_i)}{(x_{i+1}-x_i)(E_{i+1}+E_i)}\right|\right)^{-1} \tag{21}$$

We change the number of cells but hold the time step at $\Delta t=10^{-3}$, continuing the simulation until the final time $t=0.8$.

The grid equations are solved using arc-length as monitor function applied to $\phi(x)$ and with $\tau=2\Delta t$.

The problem does not have an analytical solution and we regard the solution from the adaptive method described above with $N=500$ as our fiducial solution. The fiducial solution is shown in Fig. 1, where both the energy E and the diffusion coefficient D are shown.

As an example of the type of grid adaptation provided by the method described above, the evolution of each grid point is shown in Fig. 2.

The error obtained on uniform and adaptive grids with different number of cells is shown in Fig. 3. The error is measured in the whole domain in Fig. 3(a) and separately in the left half domain in Fig. 3(b). The adaptive grid not only consistently reduces drastically the global error, but turns the convergence rate of the global error from first order to second order, making grid refinement more effective than on uniform grids.

The reason for this remarkable feature can be found by analyzing in detail the source of error. Comparing the error in the whole domain (Fig. 3(a)) with the error in the left half only (Fig. 3(b)) two conclusions emerge. First, the greatest portion of the error comes indeed from the right half of the domain, where the sharp front in the solution lays: the error in the left part is about two orders of magnitude smaller. Second, the uniform

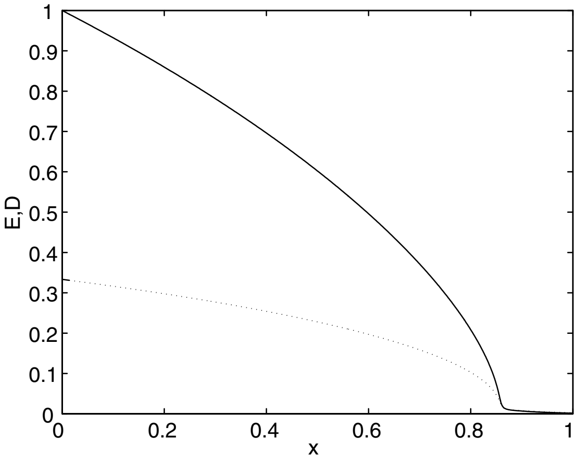

Fig. 1. Radiation diffusion: fiducial solution at the end ($t = 0.8$) of a reference simulation with $N = 500$ cells using the adaptive algorithm. The energy E is shown with a solid line and the diffusion coefficient D with a dotted line.

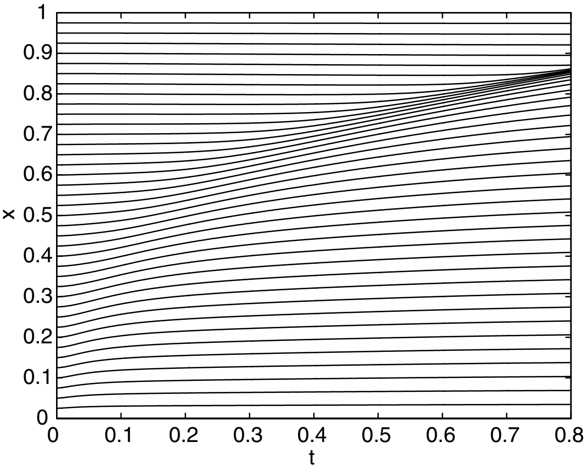

Fig. 2. Radiation diffusion: time dependence of the locations of the vertices x_i in the adaptive grid with $N = 40$.

solution does not resolve the sharp front even in the finest grid used. A known feature of convergence rates is that at unresolved features the converge rate drops to lower order [27]. In the present case, the uniform solution is second-order accurate in the smooth region but its convergence rate drops to first order near the front. Note that as the error in the smooth region is reduced, eventually the uniform solution looses its second-order convergence even in the smooth region, due to the parabolic nature of the equation and the consequent error propagation in the domain.

We note that in the results above we used an $N = 500$ adaptive grid as reference. The same conclusion is reached when a uniform grid solution is used as reference. However, the range where the uniform grid can be used as reference is limited, as the accuracy available to a uniform grid with the available computing tools never reaches the accuracy of the $N = 500$ adaptive grid. The convergence rate is indeed increased to second order by the adaptive grid.

The efficacy of the uniform and adaptive solution is shown in Fig. 4. Clearly the adaptive grid succeeds in providing a more cost-effective solution compared to the uniform grid. Thanks to the increased convergence rate, the improvement of efficacy increases as more refined grids are compared.

To test the performance of the preconditioned Newton–Krylov method, Table 1 shows the number of Newton and Krylov iterations for each grid considered. We note that in the uniform grid case the problem is

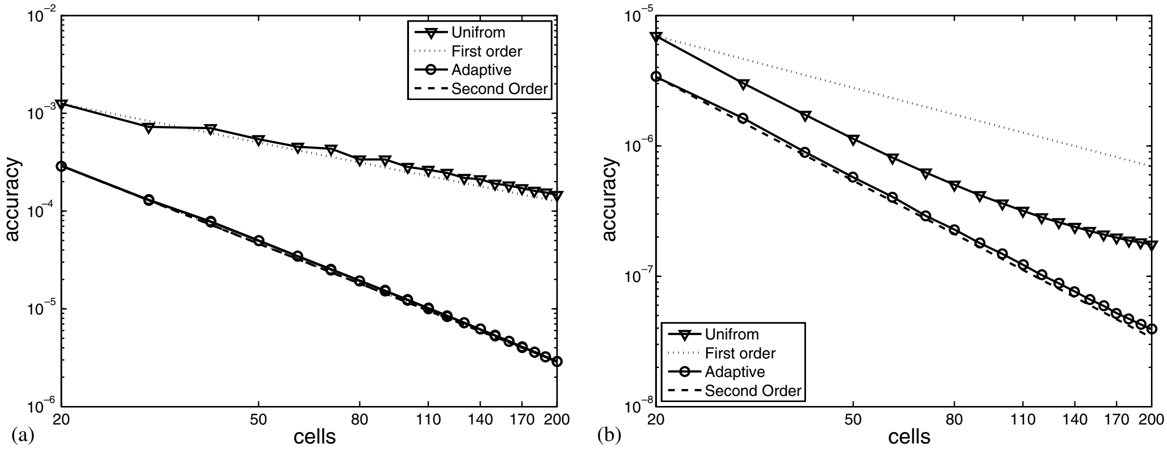

Fig. 3. Radiation diffusion: accuracy for different grid sizes, using uniform and adaptive grids. Reference curves for first-order and second-order convergence is shown to guide the eye. The L_2 error in the whole domain (a) and the left half (b) are shown separately for both methods.

Table 1
Number of Newton and Krylov (GMRES) iterations for uniform and non-uniform runs with different grid points N

N	Grid			
	Uniform		Non-uniform	
	Newton	Krylov	Newton	Krylov
20	3.00	1.00	3.26	4.00
30	3.00	1.00	3.20	3.97
40	3.00	1.00	3.08	3.91
50	3.00	1.00	3.00	3.89
60	3.00	1.00	3.00	3.91
70	3.00	1.00	3.00	3.92
80	3.00	1.00	3.00	3.92
90	2.94	1.00	3.00	3.92
100	2.79	1.00	3.00	3.92
200	2.46	1.00	3.00	3.91

tridiagonal and our block tridiagonal preconditioner solves the problem without needing any Krylov iteration. On the adaptive grid, instead, the coupling between the grid block and the physics block is neglected by the preconditioner, requiring the Krylov iteration. However, even in this case the number of Krylov iterations is modest.

What is more important is that both Newton iterations and Krylov iterations are almost independent on the number of cells, resulting on an ideal scaling. We remark that without preconditioning, the number of Krylov iterations increases dramatically as the number of cells is increased, for both the uniform and the adaptive case. The preconditioner is key in controlling the cost of the simulations and in keeping the efficacy of the adaptive grid higher than for the uniform grid.

A property of the adaptive grid used here is to equidistribute the monitor function. As noted above this is a property of the original elliptic grid generation Eq. (2). The parabolic equation used here, Eq. (5), instead satisfies this property only asymptotically. The deviation from monitor function equidistribution is controlled by the choice of τ in the grid generation equation. By keeping τ close to the time step used in the solution of the physics equations, the equidistribution is still well satisfied.

The quality of equidistribution achieved here is shown in Fig. 5 as a function of space and time. The distribution of the monitor function times the cell size $\mathscr{M}_{i+1/2}(x_{i+1} - x_i)$ is shown.

A further test is conducted to investigate the premise that lagging can be a major culprit for a reduced efficacy of adaptive grid methods. To compare the fully implicit technique described above with a time-lagging technique, we consider a traditional scheme where the grid is solved using information from the old time level

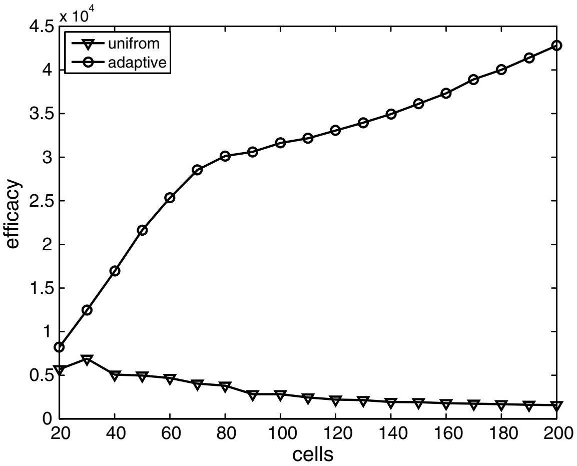

Fig. 4. Radiation diffusion: efficacy for different grid sizes, using uniform and adaptive grids.

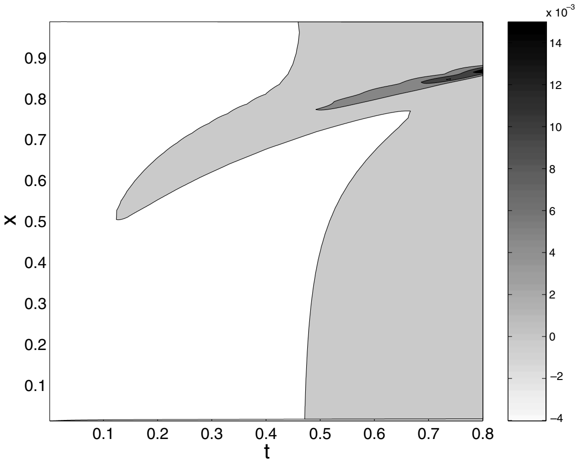

Fig. 5. Radiation diffusion: deviation from uniformity distribution of the monitor function. Grid $N = 40$.

only, removing the non-linear coupling of the grid equation with the physics equation. In practice, this change amounts to replacing in Eq. (7), $\overline{\mathscr{M}}_{i+1/2}$ with $\mathscr{M}^n_{i+1/2}$. To highlight the effect of lagging, we use now a much larger time step, $\Delta t = 0.1$ for both lagged and consistent approaches.

Fig. 6 shows the error in the solution for the case with a synchronous grid adaptation based on Eq. (7) and one with time lagging. Clearly, lagging the grid equation results in a reduced accuracy. Note that, when using a large time step, the error in the solution is coming from the temporal discretization. The temporal error could be reduced by using higher-order schemes in time, but that is beyond the scope of the present investigation. Instead, to remove the effect of the temporal error from the comparison, we define the error shown in Fig. 6 using the grid-converged (1000 cells) solution obtained with the same time step. Note that, in the lagging case, the temporal error introduced by the lagging itself is still present. The reduced accuracy of the time-lagging approach results also in a much reduced efficacy, shown in Fig. 7.

5.2. Burger's equation

Burger's equation for a quantity is formulated as

$$\frac{\partial \Upsilon}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{\Upsilon\Upsilon}{2}\right) + \frac{\partial}{\partial x}\left(\frac{1}{\alpha}\frac{\partial \Upsilon}{\partial x}\right) \tag{22}$$

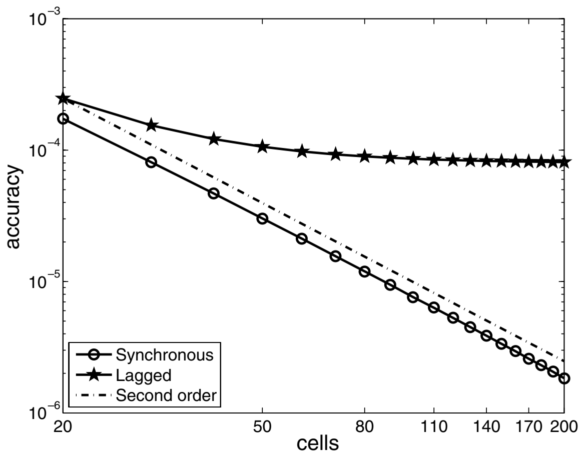

Fig. 6. Radiation diffusion: accuracy for different grid sizes, using uniform and adaptive grids. A reference curve for second-order convergence is shown to guide the eye.

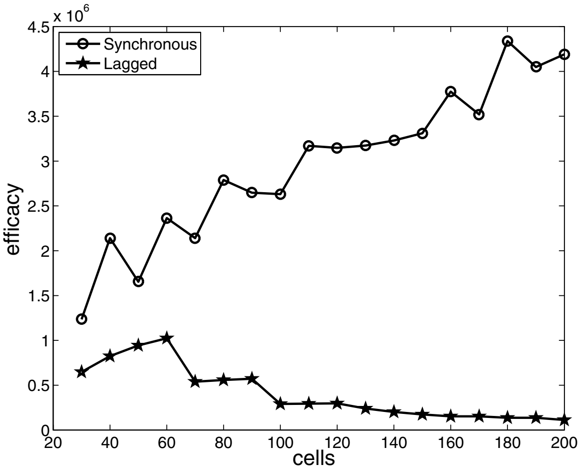

Fig. 7. Radiation diffusion: efficacy for different grid sizes, using uniform and adaptive grids.

The initial condition is $=(1+\tanh(-2\alpha(x-L/8)))/2$ and the boundary conditions are $(0)=1$ and $(L)=0$ with $L=10$ and $\alpha L=500$.

As for the radiation diffusion case, we use the implicit Euler scheme in time. The first-order spatial derivative is discretized with the ZIP method and the second-order derivative with the centered scheme.

We change the grid spacing but hold the time step at $\Delta t=0.03$, continuing the simulation until the final time $t=10$. As fiducial solution we use the results of an adaptive grid run with $N=500$. As the simulation progresses, the solution steepens and moves to the right as is typical of Burger's equation (result not shown). The grid equations are solved using arc-length as monitor function and with $\tau=\Delta t/2$.

The results of the present test are consistent with those of the previous test, resulting in a similar improvement in accuracy, and efficacy for the adaptive grid runs.

The accuracy of the uniform and adaptive solution is shown in Fig. 8. Both methods are second-order accurate when a sufficient number of cells is used. However, the adaptive grids attain the asymptotic scaling for fewer number of cells than uniform grids and the accuracy of the results is one order of magnitude better.

The efficacy of the uniform and adaptive solution is shown in Fig. 9. The efficacy is similar for uniform and adaptive grid at small cell numbers, but it becomes much larger for the adaptive grids with respect to the uni-

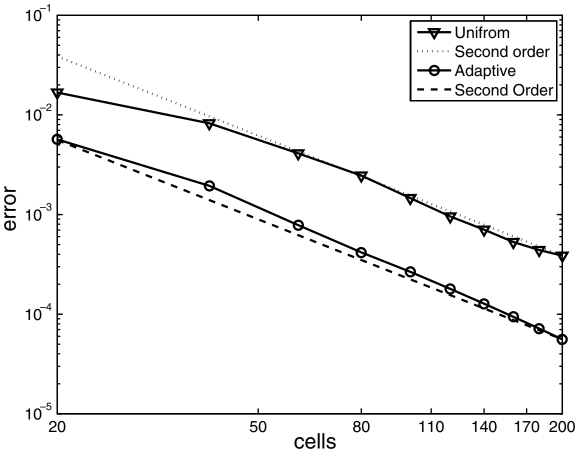

Fig. 8. Burger: accuracy for different grid sizes, using uniform and adaptive grids.

form grid as the number of cells increases. The reduced efficacy for small numbers of cells is linked to the fact that the accuracy at small number of cells has not yet attained the second-order asymptotic limit.

Table 2 reports the number of Newton and Krylov iterations for the cases considered. Again the Jacobian of the uniform grid case is actually tridiagonal, resulting in the preconditioner providing the exact answer without needing any Krylov iterations. In the adaptive grid runs, this is no longer the case, and a few Krylov iterations are needed. The number of Krylov iterations is again only very weakly dependent on the number of cells, showing the nearly ideal scaling properties of the preconditioner deployed here. The number of Newton iterations, is also nearly perfectly independent on the grid size.

To investigate the effects of the parabolization of the moving mesh equations and the choice of τ we investigate also the departure from exact equidistribution of the monitor function. Table 2 reports in the last column the variation of the monitor function multiplied by the cell volume divided by its average:

$$\frac{\delta\mathscr{M}}{\overline{\mathscr{M}}} = N\frac{\max_i\mathscr{M}_{i+1/2}(x_{i+1}-x_i) - \min_i\mathscr{M}_{i+1/2}(x_{i+1}-x_i)}{\sum_i\mathscr{M}_{i+1/2}(x_{i+1}-x_i)} \tag{23}$$

5.3. Advection equation

We consider the advection equation

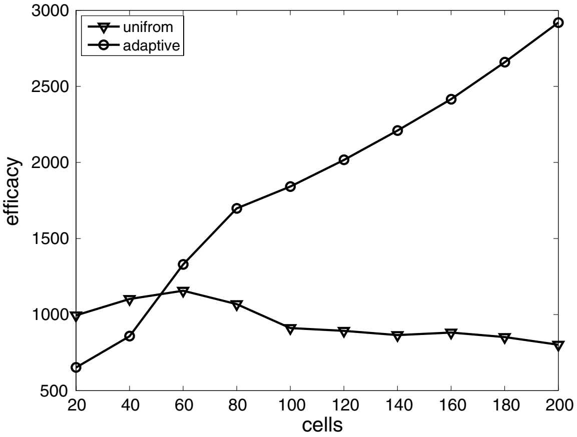

Fig. 9. Burger: efficacy for different grid sizes, using uniform and adaptive grids.

Table 2
Number of Newton and Krylov (GMRES) iterations for uniform and non-uniform runs with different grid points N

N	Grid				
	Uniform		Non-uniform		
	Newton	Krylov	Newton	Krylov	$\langle \delta\mathcal{M}/\overline{\mathcal{M}} \rangle$
20	3.00	1.00	3.87	4.95	7.88×10^{-5}
40	3.17	1.00	3.93	6.01	1.23×10^{-4}
60	3.39	1.00	3.93	6.52	2.11×10^{-4}
80	3.54	1.00	3.95	6.59	2.31×10^{-4}
100	3.67	1.00	3.95	6.69	2.05×10^{-4}
120	3.68	1.00	3.95	6.87	2.30×10^{-4}
140	3.68	1.00	3.94	7.13	2.46×10^{-4}
160	3.70	1.00	3.94	7.37	2.45×10^{-4}
180	3.70	1.00	3.93	7.41	2.27×10^{-4}
200	3.70	1.00	3.93	7.44	2.51×10^{-4}

For the non-uniform runs, the deviation form uniformity in the distribution of the monitor function is listed.

$$\frac{\partial \phi}{\partial t} = -\frac{\partial}{\partial x}(u\phi) \tag{24}$$

with constant (in time and space) velocity field $u = 1/2$ in a domain of size $L = 10$. Note that, unlike the previous two problems, simple advection has only one time scale introduced by the advection speed and only one spatial scale introduced by the choice of the initial condition. For this reason, the present case is actually a more challenging test for the adaptive algorithm considered in the present paper. By allowing less room for improvement here, we aim at testing if in a single time scale case the adaptive grid can still perform at least on par with the uniform grid.

For the present problem, the implicit method is still needed to handle the non-linearity and the coupling of the grid equation with the physics equation. Furthermore, in principle the additional grid equation introduces a new time scale, that of grid motion, which can be addressed by the implicit method without introducing additional stability constraints.

Following a previous study of adaptive grid performance [11], the initial condition is chosen as a Gaussian beam $\phi = \mathrm{e}^{-(x-x_0)^2/\sigma^2}$ with $x_0 = L/2$ and $\sigma^2 = 1/2$. The initial field is advected for a total time $t = 3$.

Unlike previous cases, we use a second-order method, with Crank–Nicolson time discretization and a more advanced non-oscillatory advection scheme for the first-order derivative. We use the quadratic upwind scheme QUICK [28], bounded by the SMART [29] algorithm to avoid an oscillatory behavior introduced by the high-order scheme. We keep the time step constant at $\Delta t = 0.01$ corresponding to a varying CFL number from CFL = 0.04 at $N = 40$ to CFL = 0.2 at $N = 200$. Different numbers of cells are considered, and the grid equation is solved using the function itself as monitor function $\mathcal{M} = 1 + \phi$ with $\tau = 5\Delta t$. We remark that this choice was motivated by the current problem where the function being advected was small in large parts of the system: we chose a monitor function that would tend to put the grid points where the function is non-zero. We note also that this choice satisfied the required property of being positive definite. In results not reported here we also tested the use of the arc-length as monitor function, but the results were inferior. We remark that the level of efficacy achieved by the grid adaptation is strongly linked with the monitor function used.

As the beam moves, its shape should be conserved and the final shape should be simply shifted, providing the reference analytical solution. Fig. 10 reports the error defined as the L_2 of the difference between computed solution and exact solution at the same time and point. The uniform and adaptive grid achieve similar convergence rates, with the adaptive grid achieving a marginally larger convergence rate. If the convergence rate is similar, the accuracy for a given number of cells is superior in the adaptive grid case.

The reduction of error achieved by the grid is not offset by the higher cost, and the adaptive grid results in a superior efficacy in the present example (see Fig. 11). This finding is particularly relevant as the present physics problem has no multiple scales and only the advection velocity determines the time step. The introduction of grid motion complicates the problem by introducing the additional time scale of the grid motion. The implicit time discretization is needed to avoid the problem and step over such unphysical scales introduced by the grid

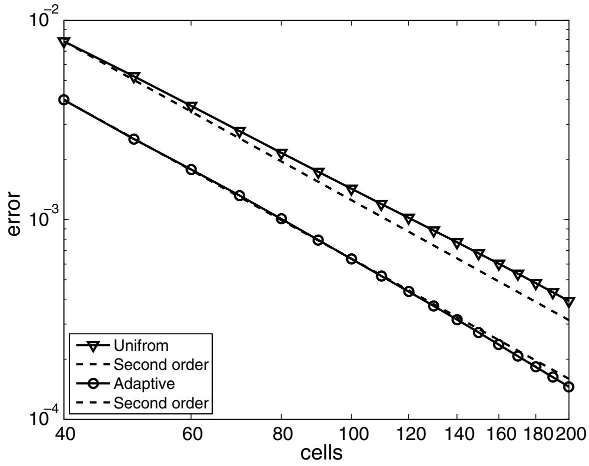

Fig. 10. Advection: accuracy for different grid sizes, using uniform and adaptive grids.

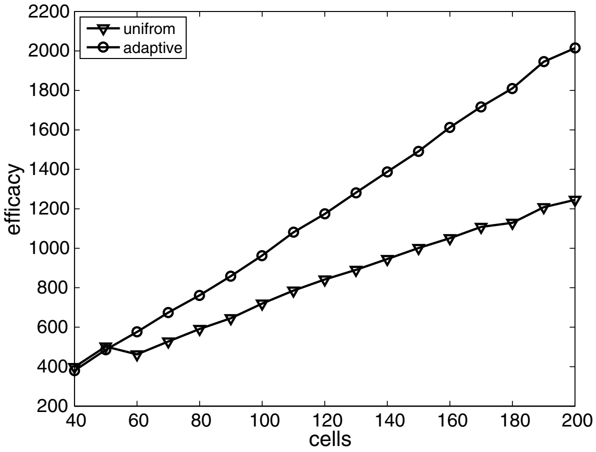

Fig. 11. Advection: efficacy for different grid sizes, using uniform and adaptive grids.

motion. The preconditioned NK solver presented above allows the efficient solution of the coupled problem, resulting in a considerable improvement of the overall efficacy of the adaptive grid compared with the uniform grid.

The number of Newton and Krylov iterations in the present example is shown in Table 3. The use of a higher-order advection algorithm makes the Jacobian non-tridiagonal even for the uniform grid case, requiring some Krylov iterations. However, in both uniform and adaptive grid runs, the number of Krylov and Newton iteration is virtually independent of the number of cells.

We note that the present case affords a special solution to the grid motion equation. One could test the present approach in the case the grid is moving with the local speed, removing the advection term altogether. In results not reported, the adaptive co-moving grid reproduces the exact solution as expected.

5.4. Fisher equation

We consider the Fisher equation

$$\frac{\partial \phi}{\partial t} = \beta \frac{\partial^2 \phi}{\partial x^2} + \alpha \phi (1 - \phi) \tag{25}$$

Table 3
Advection: number of Newton and Krylov (GMRES) iterations for uniform and non-uniform runs with different grid points N

N	Grid				
	Uniform		Non-uniform		
	Newton	Krylov	Newton	Krylov	$\langle \delta\mathscr{M}/\overline{\mathscr{M}} \rangle$
50	5.00	1.80	6.09	2.49	6.79×10^{-2}
60	4.97	1.79	6.04	2.48	5.80×10^{-2}
70	4.98	1.79	6.03	2.48	5.06×10^{-2}
80	5.00	1.80	6.04	2.49	4.50×10^{-2}
100	5.00	1.80	6.00	2.49	3.62×10^{-2}
200	5.00	1.80	6.00	2.49	3.57×10^{-2}

For the non-uniform runs, the deviation form uniformity in the distribution of the monitor function is listed.

with constant $\alpha = 100$ and $\beta = 10^{-3}$ in a domain of size $L = 1$. This problem admits an exact solution of the form

$$\phi = \left(1 + \mathrm{e}^{\sqrt{\frac{\alpha}{6\beta}}x - \frac{5}{6}\alpha t}\right)^{-2} \tag{26}$$

that presents a front with thickness $\delta = \sqrt{6\beta/\alpha}$ and moving with speed $c = \sqrt{25\alpha\beta/6}$ (see Fig. 12). The initial state and the boundary conditions are also imposed using the exact solution.

This test proved challenging in the past attempts to use adaptive grids [5]. It was suggested that for a wide class of adaptive methods based on the approach by Dorfi and Druri [7] the discretized coupled grid–physics model could be numerically unstable and provide less accurate results than a uniform grid [5]. However, the results below prove that no such instability is present in our approach.

Here we remove any possibility for temporal and spatial instabilities by considering a fully implicit BDF2 time discretization that preserve positive solutions and allows a second-order time dicretization [30]. The BDF2 method is a two-step method needing two previous time levels. We use Crank–Nicolson at the first time step to provide the first time level after the initial condition. The spatial discretization is also second order and based on centered differences for the diffusion term and the ZIP algorithm for the advection term introduced by grid motion. The use of the ZIP algorithm assures also the non-linear stability, besides the linear stability of

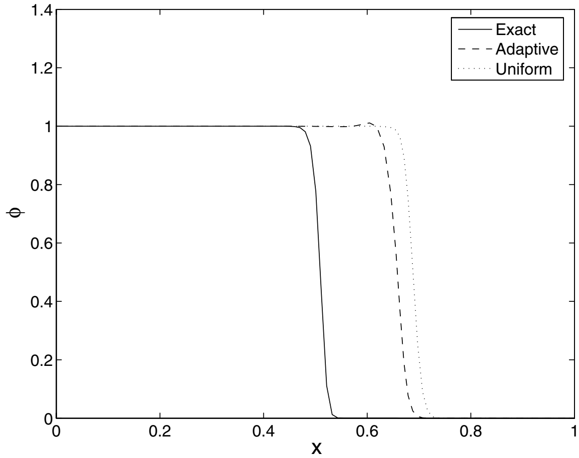

Fig. 12. Fisher: exact solution (solid line) compared with a two computed solutions on 100 cells, one with a uniform grid (dotted) and one with adaptive grid using the monitor function in Eq. (27).

the numerical scheme [22]. This proves a crucial improvement over previous efforts [5] based on centered schemes that do not ensure non-linear stability.

We use a time step of $\Delta t = 0.005$ and compute the evolution until a final time $t = 0.8$. The exact solution and two solutions with uniform and adaptive grids are shown in Fig. 12. Clearly, unlike the approach presented in Ref. [5] the present approach does not show any presence of numerical instability and the adaptive grid results in better accuracy than the uniform grid. Note that the principal error comes from the front propagating too fast and the area under each curve increases in time as the front moves. A natural definition of the error in the present case is the difference of the front speed from the exact value. Equivalently (but more conveniently), the error can be computed as the error in the growth of the area under the solution curve. Below we adopt this definition and use as reference solution for each method the grid-converged solution with the same time step ($\Delta t = 0.005$) but with 1000 cells (we compared with a reference solution with 500 cells and obtained similar results). This choice removes the effect of the temporal discretization error.

To investigate the performance, different numbers of cells are considered, and the grid equation is solved using two different monitor functions, the arc-length and a renormalized curvature defined as

$$\mathscr{M} = 1 + \frac{\left|\frac{\partial^2 \phi}{\partial x^2}\right|}{\max_x \left|\frac{\partial^2 \phi}{\partial x^2}\right|} \tag{27}$$

with $\tau = 10\Delta t$.

Fig. 13 shows the error for various grid sizes and for three types of runs: with a uniform grid, with an adaptive grid using the arc-length monitor function and with the renormalized curvature monitor function.

The arc-length MMA fails to improve the solution when compared with a uniform grid with the same number of points but remains numerically stable and converges to the reference solution. Instead, the MMA solution based on the curvature monitor function improves the solution not only quantitatively by producing a smaller error for a given number of points.

In contrast with Ref. [5], we interpret the inferior performance of the arc-length monitor function simply as an inadequacy of the monitor function itself. Indeed, Fig. 13 shows that the curvature monitor function defined in Eq. (27) is superior in accuracy to the uniform grid. We find no numerical instability, a consequence of our choice of second-order discretization schemes and in particular of the choice of the ZIP discretization scheme, stable not only linearly but also non-linearly. Clearly, if a mathematically motivated monitor function representing a true error estimator (and not a heuristic guess of the error) was used, more improvements could be achieved.

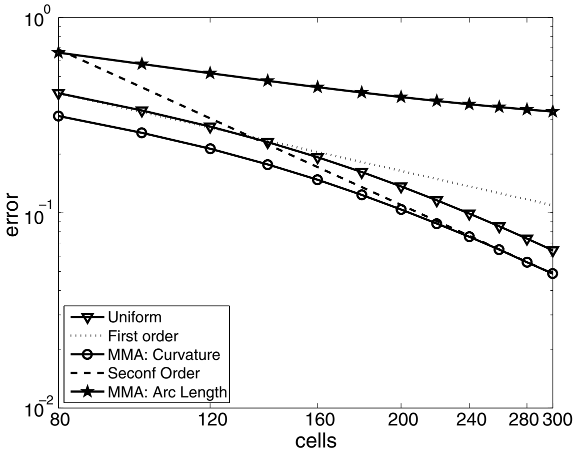

Fig. 13. Fisher: accuracy for different grid sizes, using uniform and two types of adaptive grids, one based on the arc-length monitor function and one using the monitor function in Eq. (27). The error is defined as the relative difference of the front speed with respect to the reference solution.

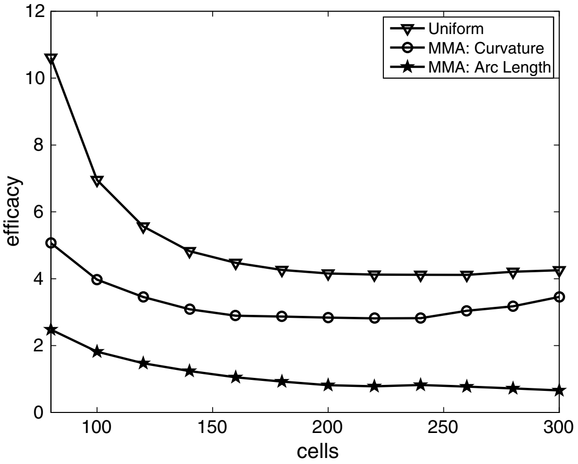

Fig. 14. Fisher: efficacy for different grid sizes, using uniform and two types of adaptive grids, one based on the arc-length monitor function and one using the monitor function in Eq. (27).

The need for a true error estimator in the present problem is further motivated by analyzing the efficacy, in Fig. 14. Unlike all previous cases, the improvement in accuracy is insufficient to offset the increased cost of deploying the adaptive grid. More accuracy gain is needed to make the adaptive grid competitive for the present test. Future work will be directed towards using rigorous error estimators to achieve this goal.

The important conclusion of the test on the Fisher problem is not the inability of adaptive grids to achieve higher efficacy without a rigorous error estimator, but that even for this problem the adaptive grid can achieve better accuracy than a uniform grid with the same number of points. No instability such as that reported by Ref. [5] is present in our implementation.

6. Conclusions

The conclusions of our paper are twofold.

First, we have demonstrated a general MMA approach based on the finite volume discretization of the model equations, including a grid motion equation, and their solution with a preconditioned NK method.

Second, we have shown in a number of typical test problems that a rigorous definition of efficacy demonstrates that even in 1D the proposed MMA is indeed cost-effective. In most of the cases reported, the MMA has a superior efficacy than uniform grids. We have shown also a notable exception, the Fisher equation, where better accuracy but not better efficacy could be achieved.

We note that in performing the tests it has emerged that preconditioning the NK is key to success. Without preconditioning the adaptive grid cases require very many Krylov iterations. Any improvement in accuracy is swamped by the exploding cost of NK without preconditioning.

The good news is that even the trivial block-tridiagonal preconditioned used here attains nearly ideal performances, erasing almost completely the higher cost introduced by adapting the grid. In particular, the preconditioner results in a scalable solver since the number of Krylov iterations and Newton iterations is kept nearly constant as the number of cells is increased. The results shown here apply to 1D problems. However, we have demonstrated a similar ability to precondition effectively the grid equations in 2D using a multigrid preconditioner [12]. A similar study in 2D is left to future work.

The greatest remaining challenge is error estimation. As noted in the test case of the Fisher equation, an adaptive strategy based on heuristic monitor functions can result in limited improvements over uniform grids. In that case, the MMA did not result in cost-effective performance for the lack of a good guidance on where adaptation should focus.

Despite the need for further work in this direction, the fundamental finding remains that in presence of good monitor functions, the gain in accuracy provided by fully implicit MMA more than offsets the higher costs resulting in a much increased efficacy.

Acknowledgments

This research is conducted for the United States Department of Energy, under Contract W-7405-ENG-36. The work is supported by the Laboratory Directed Research and Development (LDRD) program at the Los Alamos National Laboratory.

References

[1] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, Boca Raton, FL, 1994.

[2] V.D. Liseikin, Grid Generation Methods, Springer, Berlin, New York, 1999.

[3] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in 2 dimensions, J. Comput. Phys. 46 (3) (1982) 342–368.

[4] A. Winslow, Adaptive mesh zoning by the equipotential method, Technical Report- UCID-19062, Lawrence Livemore Laboratory, 1981.

[5] S. Li, L. Petzold, Y. Ren, Stability of moving mesh systems of partial differential equations, SIAM J. Sci. Comput. 20 (1998) 719–738.

[6] M. Gehmeyr, D. Mihalas, Adaptive grid radiation hydrodynamics with titan, Physica D 72 (1994) 320.

[7] E.A. Dorfi, L.O. Drury, Simple adaptive grids for 1-d initial value problems, J. Comput. Phys. 69 (1) (1987) 175–195.

[8] S. Li, L. Petzold, Moving mesh methods with upwinding schemes for time-dependent PDEs, J. Comput. Phys. 131 (2) (1997) 368–377.

[9] K.H.A. Winkler, M.L. Norman, M.J. Newman, Adaptive mesh techniques for fronts in star formation, Physica D 12 (1–3) (1984) 408–425.

[10] W.Z. Huang, Y.H. Ren, R.D. Russell, Moving mesh partial differential equations (MMPDES) based on the equidistribution principle, SIAM J. Num. Anal. 31 (3) (1994) 709–730.

[11] J. Iselin, J. Prusa, W. Gutowski, Dynamic grid adaptation using the MPDATA scheme, Mon. Weather Rev. 130 (2002) 1026–1039.

[12] L. Chacón, G. Lapenta, A fully implicit, nonlinear adaptive grid strategy, J. Comput. Phys. 212 (2006) 703–717.

[13] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.

[14] M.J. Baines, Moving Finite Elements, Oxford University Press, Oxford, 1994.

[15] W. Cao, W. Huang, R. Russell, Comparison of two-dimensional *r*-adaptive finite element methods using various error indicators, Math. Comput. Simul. 56 (2001) 127–143.

[16] J.U. Brackbill, An adaptive grid with directional control, J. Comput. Phys. 108 (1993) 38–50.

[17] G. Lapenta, Variational grid adaptation based on the minimization of local truncation error: time independent problems, J. Comput. Phys. 193 (2004) 159.

[18] K. Lipnikov, M.J. Shashkov, The error-minimization-based strategy for moving mesh methods, Comm. Comput. Phys., in press.

[19] M. Ainsworth, J.T. Oden, A posteriori error estimation in finite element analysis, Comput. Methods Appl. Mech. Eng. 142 (1–2) (1997) 1–88.

[20] G. Lapenta, A recipe to detect the origin of error in discretization schemes, Int. J. Num. Meth. Eng. 59 (2004) 2065.

[21] L. Ferm, P. Lötstedt, Adaptive error control for steady state solutions of inviscid flow, SIAM J. Sci. Comput. 23 (2002) 1777.

[22] C.W. Hirt, Heuristic stability theory for finite-difference equations, J. Comput. Phys. 2 (1968) 339–355.

[23] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.

[24] R. Dembo, S. Eisenstat, R. Steihaug, Inexact Newton methods, J. Numer. Anal. 19 (1982) 400.

[25] S. Eisenstat, H. Walker, Globally convergent inexact Newton methods, SIAM J. Optim. 4 (1994) 393.

[26] D. Mihalas, B. Weibel-Mihalas, Foundations of Radiation Hydrodynamics, Dover, New York, 1999.

[27] R. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, UK, 2002.

[28] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Comput. Meth. Appl. Mech. Eng. 19 (1979) 59–98.

[29] P. Gaskell, A. Lau, Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm, Int. J. Numer. Meth. Eng. 8 (1988) 617–641.

[30] R.B. Lowrie, A comparison of implicit time integration methods for nonlinear relaxation and diffusion, J. Comput. Phys. 196 (2) (2004) 566–590.