

SPACE-TIME ADAPTIVE SOLUTION OF FIRST ORDER PDES

LARS FERM¹ and PER LÖTSTEDT² *

¹*Dept of Information Technology, Scientific Computing, Uppsala University, SE-75105 Uppsala, Sweden. email: ferm@it.uu.se*

²*Dept of Information Technology, Scientific Computing, Uppsala University, SE-75105 Uppsala, Sweden. email: perl@it.uu.se*

Abstract

An explicit time-stepping method is developed for adaptive solution of time-dependent partial differential equations with first order derivatives. The space is partitioned into blocks and the grid is refined and coarsened in these blocks. The equations are integrated in time by a Runge-Kutta-Fehlberg method. The local errors in space and time are estimated and the time and space steps are determined by these estimates. The error equation is integrated to obtain global errors of the solution. The method is shown to be stable if one-sided space discretizations are used. Examples such as the wave equation, Burgers' equation, and the Euler equations in one space dimension with discontinuous solutions illustrate the method.

Keywords: Runge-Kutta-Fehlberg method, shock problems, space adaptation, time adaptation

AMS subject classification: 65M20, 65M50

1 Introduction

A numerical method for solution of time dependent partial differential equations (PDEs) with space-time adaptivity is developed in this paper. The grid is refined and coarsened dynamically in patches and the equations are integrated with variable time steps. Adaptive methods are in general more efficient than fixed grid methods in particular in higher dimensions and for problems with steep gradients [7]. The adaptivity is often based on control of the discretization errors to decide

*Financial support has been obtained from the Swedish Research Council.

when a change of the grid or the time step is required. Moreover, no prior knowledge of the solution is in principle necessary when the initial grid is generated or when the initial time step is chosen.

There are essentially two methods of adapting the computational grid for PDEs: the moving grid method and adaptive mesh refinement (AMR) sometimes referred to as the r -method and the h -method. The grid points in a moving grid method are generated by equidistributing the space steps with a monitor function. In [32], [33], this function depends on the arc length or the curvature of the solution. An associated differential equation defined by the monitor function is solved for the position of the grid points in [1], [17]. The discretized PDE and the equation for the grid points are coupled and solved simultaneously with an implicit method for ordinary differential equations (ODEs) allowing variable time steps. Regularization of the grid is necessary after each time step. Two-dimensional (2D) problems are solved in [17]. An advantage of the method is that grid points are reallocated for better accuracy without increasing the memory requirements. The errors due to the time discretization are controlled in [1], [32], but there is usually no quantitative estimation of the spatial errors. The choice of monitor function has a significant influence on the accuracy of the results [1], [31]. Different equidistribution methods are compared in [32]. Solutions to one-dimensional (1D) conservation laws are computed with a semi-implicit method in [31]. Problems with accuracy and stability are investigated in [24] and spurious solutions on a fixed grid may not be removed by a moving grid method [4]. These methods may also have difficulties in higher dimensions with skewed cells. In such cells the accuracy of the discretization schemes is degraded and additional refinement is required.

In AMR methods, points or cells are added in the original grid when a finer resolution is necessary and removed when they are no longer needed. New cells are either introduced wherever a sensor is sufficiently large or in patches with many cells. The resulting grid in the first approach has irregular boundaries between the refined and coarsened areas and a special data structure organizes the cells [27]. In the second approach the data structure is simpler but a waste of cells cannot be avoided. The patches are aligned with the original grid in [2], [3], where solutions to time dependent flow problems are computed in 2D. The error is estimated by comparing the solutions obtained with twice the step size in space and time. If this is too large then the grid is refined. Substantial savings in computing time are reported in [20]. Other similar methods for different problems in physics are [16] and [18].

Our method is of AMR type with refinement in patches or blocks of the grid. The geometry of these blocks is predetermined, thus further simplifying the data structure compared to the method in [2]. The discretization (or truncation) errors are estimated in space and time. All cells in a block are refined or coarsened depending on the spatial error. The time steps are selected in the same way as in the numerical solution of ODEs [12], [13], [30]. Steady state solutions of flow

problems with shocks have been computed with finite volume methods in 2D and 3D [7], [8], [9]. Time-dependent hyperbolic equations are solved in 2D with an implicit time-stepping method in [25]. Such a method is suitable for PDEs with second derivatives and all blocks can be advanced in time by the same time step. The stability at the block boundaries is investigated in [10].

In this paper, we solve time-dependent problems with first order space derivatives in 1D. The equations are discretized with a second order finite volume method in space and an explicit second order Runge-Kutta-Fehlberg (RKF) method in time suitable for conservation laws. The time integration is proved to be stable for a convection equation and the accuracy is of second order also at the block interfaces. The local errors are estimated in space as in [8] by comparing the space discretization on two different grids and in time by the RKF method. Small step sizes around shocks may be the only way to reduce errors there and also in areas away from the shock. High order schemes suffer from a loss of order of accuracy for shock problems even in smooth parts of the solution [6]. The local errors are integrated by the error equation to obtain estimates of the global error. The techniques developed in this paper can be applied to 2D and 3D problems. The treatment of the block boundaries is the same as here for steady state problems in 2D in [8] and [9], and for time-dependent equations in 2D in [25].

The advantage of refining and coarsening the grid in patches is a simpler data structure compared to refinement of single cells. It is easier to maintain more than first order accuracy in cells with neighbors that are larger or smaller. Furthermore, for time-dependent problems the administration of the grid is reduced with patches. With single cells they will be reorganized in almost every time step while the grid remains constant for longer time with patches. How many extra cells that are used depends on the size of the patches. Compared to moving grid methods the AMR method is readily extended to higher dimensions, has few parameters, and no instability has been observed. There is an estimate of the error due to the spatial discretization and there is no need to solve an additional equation to determine the grid in the next time step. Disadvantages are the additional data structure and the treatment of the difference stencils at the block boundaries.

The contents of the remainder of the paper are as follows. In the next section, the discretization of the PDE is discussed and in Section 3 the error control is described. The numerical examples in Section 4 are a scalar convection equation, the wave equation, Burgers' inviscid equation, and the Euler equations of gas dynamics.

2 Discretization in space and time

The discretization of the PDE is here described for one space dimension and time. The time step is denoted by Δt and the space step by Δx . The PDE is written in conservation form here but this is not necessary for the time integration algorithm to be applicable.

2.1 Finite volume and Runge-Kutta discretization

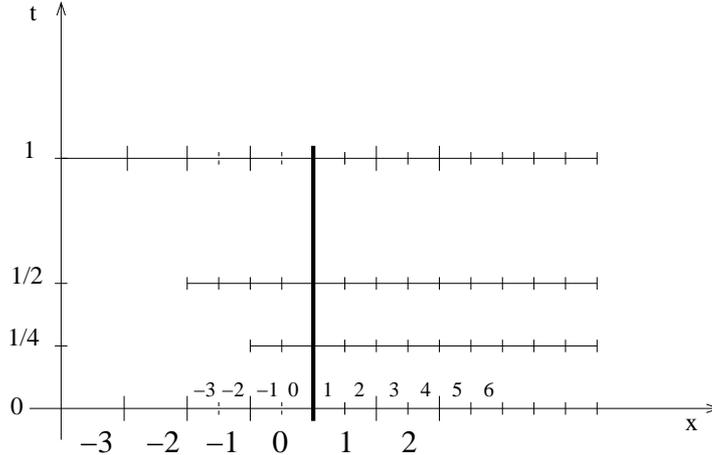


Figure 1: Space-time diagram of a domain with two blocks with different grid sizes. The indices of the cells in the fine grid are written with small digits and in the coarse grid with large digits. The time axis is divided into four levels corresponding to $t = 0, \Delta t/4, \Delta t/2, \Delta t$.

The computational domain in space is divided into a number of blocks. The size of the cells in a block varies smoothly but is allowed to jump at the block boundaries. At level 0 in Fig. 1 the spatial domain consists of two blocks: one with coarse cells to the left and one with fine cells to the right. The blocks are separated by the bold line in the figure and they overlap each other with two ghost cells. One ghost cell in the coarse grid is composed of two fine cells to the right of the block boundary and two fine ghost cells reside inside one coarse cell to the left. The new time level at 1 is reached by taking an intermediate step to level 1/2 in the fine block. The jump in the grid size is at most 2 in this paper but other quotients are possible employing similar techniques.

The PDE is approximated in space by a finite volume method and in time by an explicit Runge-Kutta method. Consider a PDE for u in conservation law form

$$u_t + f(u)_x = 0. \tag{1}$$

A subscript x or t denotes differentiation with respect to the variable. The conservation law is integrated over a cell ω_j of length Δx between $x_{j-1/2}$ and $x_{j+1/2}$ so that

$$\Delta x^{-1} \left(\int_{\omega_j} u \, dx \right)_t + \Delta x^{-1} (f(u(x_{j+1/2})) - f(u(x_{j-1/2}))) = 0. \quad (2)$$

The average u_j^{n+1} in ω_j at time $t = t^{n+1}$ is computed from u_i^n , $i = j - k, \dots, j + k$, with the Runge-Kutta scheme

$$u_j^* = u_j^n - \Delta t F_j(u^n), \quad (3a)$$

$$u_j^{n+1} = u_j^n - 0.5 \Delta t (F_j(u^n) + F_j(u^*)), \quad (3b)$$

where $F_j(u)$ is the discretization of the space derivative $f(u)_x$ in cell j . The method in (3) is second order accurate in time for smooth problems and has the TVD (total variation diminishing) property [11].

The space derivative in (1) is discretized with

$$\begin{aligned} F_j(u) &= \Delta x^{-1} (h_{j+1/2} - h_{j-1/2}), \\ h_{j+1/2} &= h(u_{j-k+1}, \dots, u_{j+k}), \end{aligned} \quad (4)$$

where $f(u) = h(u, \dots, u)$ for consistency, see [23]. In our discretizations $k = 2$ and two ghost cells suffice on each side of a block interface. For the simple, hyperbolic, scalar model equation

$$u_t + au_x = 0, \quad (5)$$

we choose a second order accurate upwind formula

$$h_{j+1/2}(u_{j-1}, u_j, u_{j+1}, u_{j+2}) = \begin{cases} a(1.5u_j - 0.5u_{j-1}), & a > 0, \\ a(1.5u_{j+1} - 0.5u_{j+2}), & a < 0. \end{cases} \quad (6)$$

Burgers' equation is approximated by the first order Engquist-Osher scheme [5], [22], and the Euler equations of gas dynamics are discretized by Osher's method of second order [23], [26], in the numerical examples in Sect. 4. Both methods are in the form (4).

Assume that the step sizes in the coarse and fine blocks in Fig. 1, Δx and $\Delta x/2$, are constant. Let U_j denote a cell average in the coarse grid and u_j a value in the fine grid. Then the values in the ghost cells U_1 and U_2 (see Fig. 1) in the coarse block are exactly given by the values u in the fine block

$$U_1 = 0.5(u_1 + u_2), \quad U_2 = 0.5(u_3 + u_4). \quad (7)$$

The values u_0 and u_{-1} are determined by one-sided third order accurate interpolation

$$u_0 = (11U_0 - 4U_{-1} + U_{-2})/8, \quad u_{-1} = 2U_0 - u_0. \quad (8)$$

The interpolation coefficients are determined such that the Taylor expansions of the cell averages are matched including second order terms in Δx . Then $h_{1/2}$ is a third order accurate approximation at the block interface and $F_1(u)$ and $F_2(u)$ in the fine block are second order accurate there, see [25].

The scheme is not conservative at block boundaries with jumps in the step size, since the flux in the coarse grid there is different from the flux in the fine grid. When a shock passes the interface both blocks have the same step size, thus ensuring conservation at least locally.

The solution is advanced in time by (3). The simplest algorithm is to let Δt be the same in all blocks. The time step is constrained by some characteristic speed v in the problem, the CFL number depending on the time-stepping method and Δx in the following way

$$\Delta t \leq \text{CFL} \cdot \Delta x / v.$$

With the same global time step, Δt is probably limited by stability in the fine blocks. In the coarse blocks, Δt could be longer and the error would still satisfy an error bound. Computational work is saved if the solution is integrated in two steps with $\Delta t/2$ in the fine block, where the space step is $\Delta x/2$, and with Δt in the coarse block. The difficulty is how to calculate the missing values in the ghost cells.

Assuming that all values at level 0 are known, U^* in the coarse grid and u^* in the fine grid are computed with (3a). They are second order approximations of u^{n+1} and $u^{n+1/2}$, respectively. The values in the fine ghost cells $-3, -2, -1, 0$, at level $1/2$ are computed by linear interpolation in time from the corresponding interpolated values in the ghost cells at levels 0 and 1 using $U^0 = U^n$ and U^* . Then $u^{n+1/2}$ is computed in the fine grid including the ghost cells -1 and 0 with (3b). This is a globally second order accurate approximation of $u(x, t^{n+1/2})$. Take the step from level $1/2$ to 1 in the fine block by first computing u^* from (3a) with $u^{n+1/2}$ as the input solution. Both U^* and u^* are now known at level 1 including ghost cells and U^{n+1} and u^{n+1} can be computed by (3b). A complete time step Δt has been taken in both blocks with a second order time accurate solution at the new time t^{n+1} . This procedure is then generalized to computational domains with blocks with grid sizes $\Delta x/2^k$ for $k = 0, 1, 2, \dots$. The corresponding time steps are $\Delta t/2^k$. By restricting the jump in the time step to 2 at a block interface, the general case is not more complicated than what we have in Fig. 1.

2.2 Stability in time

The stability of the time integration is investigated for the model equation (5). Suppose that $a(x) > 0$ and that a Dirichlet boundary condition is given at $x = 0$ so that $u(0, t) = u_0$. The approximation of the space derivative is

$$F_j(u) = \Delta x^{-1} a \left(\frac{3}{2} u_j - 2u_{j-1} + \frac{1}{2} u_{j-2} \right). \quad (9)$$

Consider a grid partitioned into two blocks as in Fig. 1. Let u_l be the solution vector in the left coarse block and let u_r be the corresponding vector in the right fine block. Their indices increase for increasing x -values and after space discretization with (9), $u_l(t)$ and $u_r(t)$ satisfy

$$u_{lt} = \Delta x^{-1}(A_1 u_l + B_1 u_b), \quad u_{rt} = \Delta x^{-1}(A_2 u_r + B_2 u_l), \quad (10)$$

where A_1 and A_2 are lower triangular matrices with two subdiagonals and B_1 and B_2 have non-zero elements only in the upper left and right corners, respectively, and u_b depends on u_0 . The elements in B_2 are given by the interpolation (8). Introduce the quotient $\mu = \Delta t / \Delta x$ which is the same in both blocks. Then one step from n to $n + 1$ with (3) in the left block can be written

$$u_l^{n+1} = (I - \mu A_1 + 0.5\mu^2 A_1^2) u_l^n + \mu B_3 u_b, \quad (11)$$

with a sparse B_3 . The values in the ghost cells of the right block are determined only by the u_l -values to the left of the block interface. After some basic matrix algebra we have for u_r^{n+1}

$$u_r^{n+1} = (I - \mu A_2 + 0.5\mu^2 A_2^2) u_r^n + \mu B_4 u_l^n, \quad (12)$$

with a sparse B_4 . The conclusion from (11) and (12) is that the stability of the whole time-integration is guaranteed if the integration in each block separately is stable. The same conclusion can be drawn if the order of the blocks is interchanged and if $a < 0$. This is summarized in a proposition.

Proposition 1. *Assume that the grid is partitioned into two blocks with a jump in the step size at the interface. The equation (5) is integrated with (3) as described in the previous section and the space discretization is (6). If $\mu = \Delta t / \Delta x$ is such that the time-integration (3) is stable in each block separately, then the combined integration is also stable.*

Proof. The claim follows from the discussion above and (11) and (12). ■

Suppose that the eigenvalues of A_1 and A_2 are $\lambda_j(A_1)$ and $\lambda_j(A_2)$. The time step should be chosen sufficiently small so that $\mu\lambda_j(A_1)$ and $\mu\lambda_j(A_2)$ belong to the stability region of the the Runge-Kutta method (3). The generalization to several blocks with different time steps $\Delta t / 2^k, k \geq 0$, is straightforward.

3 Error control

The discretization errors and the control of them are discussed in this Section. The errors in the space and time discretizations are estimated and measured in certain norms. The grid size in the blocks and the time step are determined by these estimates.

3.1 The error equation

Let the integrated form (2) of the differential equation (1) be denoted by $G(u)$ and its discretization by $\Gamma(u)$. Then for any smooth u in a cell j we have

$$\tau_j(u) = G_j(u) - \Gamma_j(u).$$

If u is the analytical solution, then $G_j(u) = 0$ and

$$\Gamma_j(u) = -\tau_j(u).$$

The numerical solution \hat{u} and its smooth reconstruction from the cell averages satisfy $\Gamma_j(\hat{u}) = 0$ for all j . Hence, the error $\delta u = \hat{u} - u$ in \hat{u} fulfills the discrete error equation

$$\Gamma_j(\hat{u}) - \Gamma_j(u) = \Gamma_j(u + \delta u) - \Gamma_j(u) = \tau_j(u). \quad (13)$$

The continuous counterpart is

$$G_j(u + \delta u) - G_j(u) = \tau_j(\hat{u}) + \Gamma_j(\hat{u}) = \tau_j(\hat{u}). \quad (14)$$

If Γ and G are linear then $\Gamma_j(\delta u) = \tau_j(u)$ and $G_j(\delta u) = \tau_j(\hat{u})$. The discretization error consists of two parts τ_{Sj} and τ_{Tj} due to the space discretization and the time discretization so that

$$\tau_j = \tau_{Sj} + \tau_{Tj}. \quad (15)$$

For smooth solutions and the second order discretizations (3) and (6), $\tau_{Sj} = \mathcal{O}(\Delta x^2)$ and $\tau_{Tj} = \mathcal{O}(\Delta t^2)$. The assumption of smoothness is in general not valid for conservation laws, but by including viscosity it is possible to analyze adaptive schemes based on the solution error [19].

If the space operator is linear in u , then $F_j(u) = (A(t)u)_j$ with a matrix A . The error in the average of u approximately satisfies the differential equation form of the error equation

$$\begin{aligned} G_j(\delta u) &\approx \delta u_{jt} + F_j(u + \delta u) - F_j(u) = \delta u_{jt} + (A(t)\delta u)_j = \tau_j, \\ \delta u_j(0) &= 0, \quad j = 1, \dots, N, \end{aligned} \quad (16)$$

assuming that the initial conditions are exact. This is a system of ordinary differential equations. By Duhamel's principle [21], the solution of (16) can be written

$$\delta u(t) = \int_0^t S(t, s)\tau(s) ds, \quad (17)$$

with a solution operator S . We find that by changing the discretization error τ by a factor β , the error in the solution is also changed by the same factor.

3.2 Optimal error control

The grid size Δx and the time step Δt for the coarsest block are chosen such that a norm of τ satisfies an upper bound. Assume that there are M blocks with N cells covering the computational domain. The indices for the cells in block J are N_J to $N_{J+1} - 1$. Then the temporal errors between t^n and t^{n+1} are measured in the J :th block of length ℓ_J with $N_{J+1} - N_J$ cells with grid size Δx_j in the norm $\|\cdot\|_{r,J}$ defined by

$$\|\tau_T^{n+1}\|_{r,J}^r = \Delta t^{-1} \ell_J^{-1} \int_{t^n}^{t^{n+1}} \sum_{j=N_J}^{N_{J+1}-1} |\tau_{Tj}|^r \Delta x_j dt. \quad (18)$$

Assume that k_j time steps are taken between t^n and t^{n+1} and that the temporal order of accuracy is p with $\tau_{Tj} = c_t(x_j, t^n) \Delta t^p + \mathcal{O}(\Delta t^{p+1})$ for cell j . Then by (18)

$$\begin{aligned} \|\tau_T^{n+1}\|_{r,J}^r &= \Delta t^{-1} \ell_J^{-1} \sum_{j=N_J}^{N_{J+1}-1} \Delta x_j \sum_{i=1}^{k_j} k_j^{-1} \Delta t |c_t(x_j, t^n)|^r \Delta t^{pr} k_j^{-pr} + \mathcal{O}(\Delta t^{pr+1}) \\ &= \ell_J^{-1} \sum_{j=N_J}^{N_{J+1}-1} \Delta x_j |c_t(x_j, t^n)|^r \Delta t^{pr} k_j^{-pr} + \mathcal{O}(\Delta t^{pr+1}). \end{aligned} \quad (19)$$

For a block J with $\Delta x_j = \Delta x_J = \text{const.}$, $k_j = k_J = \text{const.}$, $j = N_J, \dots, N_{J+1} - 1$, and $\ell_J = \Delta x_J (N_{J+1} - N_J)$ we have

$$\|\tau_T^{n+1}\|_{r,J}^r = \ell_J^{-1} \Delta x_J \Delta t^{pr} k_J^{-pr} \sum_{j=N_J}^{N_{J+1}-1} |c_t(x_j, t^n)|^r + \mathcal{O}(\Delta t^{pr+1}). \quad (20)$$

Let ℓ be the length of the interval $\ell = \sum_{J=1}^M \ell_J$. Then for all M blocks the norm is

$$\begin{aligned} \|\tau_T^{n+1}\|_r^r &= \ell^{-1} \sum_{J=1}^M \ell_J \|\tau_T^{n+1}\|_{r,J}^r \\ &= \ell^{-1} \sum_{j=1}^N \Delta x_j |c_t(x_j, t^n)|^r \Delta t^{pr} k_j^{-pr} + \mathcal{O}(\Delta t^{pr+1}). \end{aligned} \quad (21)$$

The leading term $|c_t(x_j, t^n) (\Delta t/k_j)^p|$ is estimated in Sect. 3.4.

The spatial errors are measured in the same norm as above. Assume that the spatial order of accuracy is q so that $\tau_{Sj} = c_s(x_j, t^n) \Delta x_j^q + \mathcal{O}(\Delta x_j^{q+1})$ between t^n and t^{n+1} . Then in the same manner as above for all cells

$$\begin{aligned} \|\tau_S^{n+1}\|_r^r &= \Delta t^{-1} \ell^{-1} \int_{t^n}^{t^{n+1}} \sum_{j=1}^N |\tau_{Sj}|^r \Delta x_j dt \\ &= \ell^{-1} \sum_{j=1}^N \Delta x_j |c_s(x_j, t^n)|^r \Delta x_j^{qr} + \mathcal{O}(\Delta x_j^{qr+1}). \end{aligned} \quad (22)$$

For blocks with constant Δx_J the norm in (22) is

$$\begin{aligned} \|\tau_S^{n+1}\|_r^r &= \ell^{-1} \sum_{J=1}^M \Delta x_J^{qr} \sum_{j=N_J}^{N_{J+1}-1} \Delta x_J |c_s(x_j, t^n)|^r + \mathcal{O}(\Delta x_J^{qr+1}) \\ &= \ell^{-1} \sum_{J=1}^M \ell_J \|\tau_S^{n+1}\|_{r,J}^r. \end{aligned} \quad (23)$$

The leading term $|c_s(x_j, t^n)\Delta x_j^q|$ is estimated in Sect. 3.3.

The total error $\tau = \tau_T + \tau_S$ determines the error δu in the solution, see (13), (14), and (15). In special cases we have a fortuitous cancellation so that $\tau = 0$ but $\tau_T \neq 0$ and $\tau_S \neq 0$. Such a case is the first order discretization

$$\Delta t^{-1}(u_j^{n+1} - u_j^n) + a\Delta x^{-1}(u_j^n - u_{j-1}^n) = 0$$

of (5) with $\Delta x = a\Delta t$. The error τ grows when Δt decreases from the optimal choice where $\tau = 0$. The cancellation with a particular choice of Δx and Δt is impossible to achieve in general situations for systems of equations in several dimensions. This matter is discussed further in [14]. Therefore, we control the errors τ_T and τ_S separately by adjusting Δt and Δx so that

$$\|\tau\|_r \leq \|\tau_S\|_r + \|\tau_T\|_r \leq \epsilon.$$

Let Δx be the coarsest grid size and Δt the longest time step in all blocks. In a block with $\Delta x_J < \Delta x$ the number of time steps to reach Δt is $\Delta x/\Delta x_J$. Moreover, let v be the volume of a block in a Cartesian grid in a d -dimensional space, $\delta = d + 1$, and w_0 be the work per cell and time step. The computational work to advance such a problem in a time interval $[0, T]$ without any change of space or time steps is proportional to the number of time steps and the number of cells in the M blocks

$$W = w_0 \frac{T}{\Delta t} \sum_{J=1}^M \frac{\Delta x}{\Delta x_J} \frac{v}{\Delta x_J^d} = w_0 \Delta t^{-1} \Delta x v T \sum_{J=1}^M \frac{1}{\Delta x_J^\delta}. \quad (24)$$

The local error in each step is by (21) and (23)

$$\|\tau\|_r \leq \|\tau_T\|_r + \|\tau_S\|_r \leq C_t \Delta t^p + \sum_{J=1}^M C_J \Delta x_J^q, \quad (25)$$

where $\|\tau_T\|_r \leq C_t \Delta t^p$ and $\|\tau_S\|_{r,J} \leq C_J \Delta x_J^q$. The goal of the adaptation is to minimize the work in (24) subject to an upper bound ϵ on the local error in (25). The minimal work is characterized in the following proposition.

Proposition 2. *The work W in (24) is minimized with respect to Δt and Δx_J subject to the constraint*

$$C_t \Delta t^p + \sum_{J=1}^M C_J \Delta x_J^q \leq \epsilon$$

if

$$C_t \Delta t^p = \epsilon / (1 + (p\delta)/q), \quad \sum_{J=1}^M C_J \Delta x_J^q = \epsilon / (1 + q/(p\delta)).$$

Furthermore,

$$C_J \Delta x_J^q = \frac{C_J^{\delta/(q+\delta)}}{\sum_{J=1}^M C_J^{\delta/(q+\delta)}} \cdot \frac{\epsilon}{1 + q/(p\delta)}.$$

Proof. Since the parameters in W and the constraint are positive, the minimum is obtained with the constraint satisfied as an equality. Then the work can be written

$$W = \frac{w_0 C_t^{1/p} v T \Delta x}{(\epsilon - \sum C_J \Delta x_J^q)^{1/p}} \sum \frac{1}{\Delta x_J^\delta}.$$

At the optimum $\partial W / \partial \Delta x_J = 0$ for all J . The solution is

$$C_J \Delta x_J^{q+\delta} = \frac{p\delta}{q} \left(\sum \frac{1}{\Delta x_J^\delta} \right)^{-1} (\epsilon - \sum C_J \Delta x_J^q). \quad (26)$$

The right hand side is independent of J and is denoted by α . Therefore,

$$\Delta x_J = (\alpha / C_J)^{1/(q+\delta)}. \quad (27)$$

Insert (27) into the expression for the spatial error

$$\sum_{J=1}^M C_J \Delta x_J^q = \alpha^{q/(q+\delta)} \sum_{J=1}^M C_J^{\delta/(q+\delta)}.$$

It follows from the right hand side of (26) that

$$\epsilon = \left(1 + \frac{q}{p\delta}\right) \alpha^{q/(q+\delta)} \sum C_J^{\delta/(q+\delta)}.$$

Hence,

$$\begin{aligned} C_t \Delta t^p &= \frac{q}{p\delta} \alpha^{q/(q+\delta)} \sum C_J^{\delta/(q+\delta)} = \epsilon / (1 + p\delta/q), \\ \sum_{J=1}^M C_J \Delta x_J^q &= \epsilon / (1 + q/p\delta). \end{aligned}$$

The expression for $C_J \Delta x_J^q$ is obtained from (27). ■

The optimal distribution of the errors between the time and the space discretization is to let

$$\|\tau_T\|_r \leq \epsilon_T = \epsilon / (1 + (p\delta)/q), \quad \|\tau_S\|_r \leq \epsilon_S = \epsilon / (1 + q/(p\delta)).$$

The optimal error bounds for the spatial error in each block are

$$\|\tau_S\|_{r,J} \leq \kappa_J \epsilon / (1 + q/(p\delta)), \quad \kappa_J = C_J^{\delta/(q+\delta)} / \sum_{J=1}^M C_J^{\delta/(q+\delta)}.$$

In the numerical examples in Sect. 4, $p = q = 2$, $d = 1$, and the optimal bounds are $\epsilon_T = \epsilon/3$ and $\epsilon_S = 2\epsilon/3$. The spatial error in each block is required to be less than a constant tolerance regardless of the errors in the other blocks and the optimal distribution in the proposition is not utilized. The norm in Sect. 4 is the L_1 -norm $\|\cdot\|_1$.

3.3 Space step error estimate

The local errors in the space discretization are estimated by comparing F_j in a coarse cell with the sum of F_j in the corresponding fine cells. The difference is the leading term in the discretization error τ_S . It is estimated at a time level where the solution has been advanced by Δt and computed in all blocks. Let u_j and u_{j+1} be the solutions in the fine grid cells j and $j+1$. Create a coarse cell j' by removing the cell wall between j and $j+1$ and let $U_{j'} = 0.5(u_j + u_{j+1})$. Then it follows from [8] that

$$\tau_{Sj} = \frac{1}{3}(F_{j'}(U) - 0.5(F_j(u) + F_{j+1}(u))) + \mathcal{O}(\Delta x^3) \quad (28)$$

in a fine grid cell. Excessive refinement based on τ_S at e.g. shocks is avoided by introducing a smallest grid size. The grid size in a block is based on the estimate τ_{Sj} and the fact that it is proportional to Δx^2 .

The grid size in all cells in a block J is changed by a factor 2 depending on whether $\|\tau_S\|_J$ is greater or less than a tolerance ϵ_S . Let $\lceil x \rceil$ denote the smallest integer number i such that $x \leq i$. Then the algorithm for block J is

```

if  $\|\tau_S\|_J > \epsilon_S$ 
  make  $r$  refinements with  $r = \lceil \log_2(\theta \|\tau_S\|_J / \epsilon_S) / p \rceil$ 
elseif  $2^p \|\tau_S\|_J < \theta \epsilon_S$ 
  make  $c$  coarsenings with  $c = \lceil -\log_2(\|\tau_S\|_J / \epsilon_S) / p \rceil - 1$ 
endif

```

A safety factor $\theta = 0.8$ is introduced to avoid unnecessary coarsening of the grid and to ensure that the grid is sufficiently fine.

The error estimate has an irregular behavior at the block boundaries due to the interpolation in the ghost cells. Therefore, the two cells closest to the boundary are excluded from the estimate.

The estimate τ_S works well for smooth problems, see [8],[9],[25], but it is not a fool-proof shock detector as the next example shows. Consider the solution in Fig. 2 of (1) with $f''(u) > 0$ and its space discretization (4). The solution on the upper fine grid is restricted to the lower coarse grid by averaging. Let the solution in the cells be denoted as follows

$$\begin{aligned} u_1 &= u_2 = u_L, & u_3 &= u_M, & u_4 &= u_5 = u_6 = u_R, \\ u_{12} &= u_L, & u_{34} &= 0.5(u_M + u_R) = u_N, & u_{56} &= u_R. \end{aligned}$$

The values are ordered as $u_L > u_M > u_N > u_R$. The shock speed $s = (f(u_L) - f(u_R)) / (u_L - u_R)$ is positive and therefore $f(u_L) > f(u_R)$. The estimate (28) for a two-point scheme in cell 34 is

$$\begin{aligned} \tau_{S34} &= \frac{1}{6\Delta x}(h(u_N, u_R) - h(u_L, u_N) \\ &\quad - (h(u_R, u_R) - h(u_M, u_R) + h(u_M, u_R) - h(u_L, u_M))) \\ &= \frac{1}{6\Delta x}(h(u_N, u_R) - h(u_L, u_N) - f(u_R) + h(u_L, u_M)). \end{aligned} \quad (29)$$

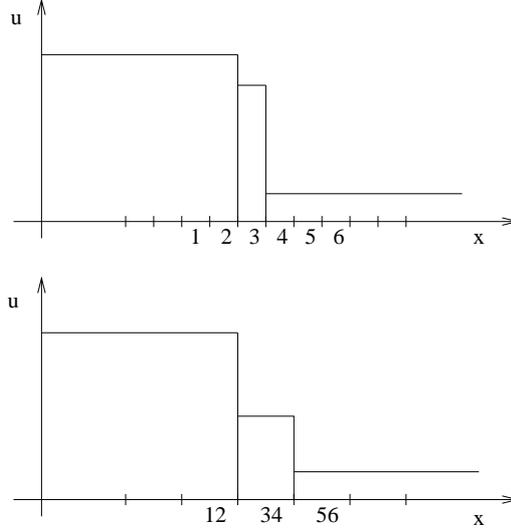


Figure 2: A shock in a scalar conservation law is moving to the right. The solution on the fine grid (upper) and the solution on the coarse grid (lower).

The numerical flux h for the Godunov scheme [23] is for our problem

$$h(u_l, u_r) = \max_{u_r \leq u \leq u_l} f(u).$$

Then there are two cases: $f'(u_R), f'(u_L) > 0$ and $f'(u_R) < 0, f'(u_L) > 0$. In the first case f is increasing monotonically. In the second case we have $f(u_L) > f(u_M)$ and $f(u_L) > f(u_R) > f(u_N)$. From (29) we derive

$$\tau_{S34} = \begin{cases} \frac{1}{6\Delta x}(f(u_N) - f(u_R)), & f'(u_R) > 0, f'(u_L) > 0, \\ 0, & f'(u_R) < 0, f'(u_L) > 0. \end{cases} \quad (30)$$

The error estimate (28) does not detect the shock in the second case and the grid will not be refined there. The same failure will occur if the coarse grid in Fig. 2 is shifted one step to the left or right. The problems are the same with the Engquist-Osher discretization.

As a remedy the discretization error estimate (28) is complemented by a sensor for detection of shocks so that the grid is refined there even if τ_S is small as in (30). A suitable condition is

$$|D_+ D_- u_j| = |\Delta x_j^{-2}(u_{j+1} - 2u_j + u_{j-1})| > 1/\chi. \quad (31)$$

If (31) is satisfied in one cell in a block, then the finest grid is used in that block irrespective of τ_S . In the examples in Sect. 4, χ is chosen to be $2 \cdot 10^{-4}$. To avoid interpolation in shocks at block interfaces and to preserve the conservation of the scheme when a shock crosses the interface, the finest grid size is used also in a neighboring block if the shock is close to the interface.

3.4 Time step error estimate

The local error due to the time steps is estimated by comparing the second order method (3) with a third order method in a Runge-Kutta-Fehlberg pair [13]. The third order scheme is

$$u_j^* = u_j^n - 0.25\Delta t(F_j(u^n) + F_j(u^*)), \quad (32a)$$

$$u_j^{n+1} = u_j^n - \Delta t\left(\frac{2}{3}F_j(u^*) + \frac{1}{6}(F_j(u^n) + F_j(u^*))\right). \quad (32b)$$

The variable u^* is computed in (3a) and the sum $F_j(u^n) + F_j(u^*)$ is reused from (3b). At a block boundary, U^* in the coarse block in Fig. 1 is computed at level 1/2 and u^* in the fine block at level 1/4 including two ghost cells. Then U^{n+1} and $u^{n+1/2}$ are given by (32b). In every block at t^n we have a second and a third order approximation u_{2j} and u_{3j} , $j = N_J, \dots, N_{J+1} - 1$, at $t^n + \Delta t/k_j$. The local error τ_T is estimated by the difference between u_{2j}^{n+1} from (3) and u_{3j}^{n+1} from (32)

$$\tau_{Tj}^{n+1} = (\Delta t/k_j)^{-1}(u_{2j}^{n+1} - u_{3j}^{n+1}) = c_t(x_j, t^n)(\Delta t/k_j)^2 + \mathcal{O}(\Delta t^3). \quad (33)$$

The time steps are chosen so that $\|\tau_T\|_r$ in (21) is less than a given tolerance ϵ_T at every time level t^n where the solution is known in all blocks. Following [12] and [30] the new time step is determined by a PI-controller. The time step to reach t^n was Δt and the new step to t^{n+1} is

$$\Delta t_{new} = \left(\frac{\theta\epsilon_T}{\|\tau_T^{n+1}\|_r}\right)^{0.3/p} \cdot \left(\frac{\|\tau_T^n\|_r}{\|\tau_T^{n+1}\|_r}\right)^{0.4/p} \Delta t, \quad (34)$$

where $p = 2$ the order of the method and the safety factor θ is 0.8. If $\Delta t_{new} > \Delta t$ then the step from $n - 1$ to n is accepted and the next step is computed with Δt_{new} . If $\Delta t_{new} < 0.9\Delta t$ then u^{n+1} is rejected and recomputed with Δt_{new} since the error is too large in the last step. Traditionally, only the first factor in (34) raised to $1/p$ determines the new time step, but it is shown convincingly for ordinary differential equations in [12] and [30] that the expression in (34) leads to smoother step sequences.

It turned out that in our test problems in Sect. 4, the time steps are bounded by the stability requirement in many cases. Smooth initial solutions usually lead to longer time steps well above the theoretical stability limit. This implies growing, and in space oscillating, estimated time errors. After a number of time cycles they are large enough to induce a reduction of the time step.

The time step errors at a shock are overestimated by (33), see [15]. Small time steps are not necessary at shocks to obtain the correct shock speed. This is accomplished by the conservative formulation of the space discretization. Therefore a filter multiplying the estimate (33) is introduced. The filter Φ depends on

$|D_+D_-u_j|$ as in (31). In cell j we let

$$\Phi_j = 1/(\max(1, \sigma \max_j |D_+D_-u_j|)). \quad (35)$$

With a suitable parameter σ , Φ_j is small at a shock and is 1 in all other points. The time step selection in (34) is then based on $\Phi_j\tau_{Tj}^{n+1}$ instead of τ_{Tj}^{n+1} .

4 Numerical results

The adaptive method is applied to four equations in this Section. The first two equations have constant coefficients and the last two equations are Burgers' inviscid equation and the Euler equations of gas dynamics. The errors are measured in the L_1 -norm ($r = 1$ in (21) and (23)).

4.1 A scalar model equation

The equation (5) with $a = 1$

$$u_t + u_x = 0, \quad x \in [0, 1], \quad t > 0,$$

propagates the initial distribution $u_0(x) = u(x, 0)$ to the right. With periodic boundary conditions, the solution is periodic. The initial condition is a Gauss pulse in Fig. 3.a. The block partitioning of the computational domain in the x -direction is indicated by vertical dashed lines. The number of cells in the blocks is displayed above each block. The global error δu in the solution is computed with the error equation (14) and the estimated local error τ . The result at $T = 0.08$ after 179 time cycles or 1432 fine time steps is compared to the true global error in Fig. 3.b. The error tolerances are $\epsilon_S = 1/40$ and $\epsilon_T = .0001$ leading to small time steps well below the stability limit. Small time steps are necessary for an accurate integration of (14) and a good agreement between the estimated and true global errors.

The solution is computed after one period ($T = 1$) in Fig. 4. The computational work and the true global error are plotted in Fig. 4.a in a \log_2 scale versus the local error tolerance $\epsilon = \epsilon_T = \epsilon_S$. As the values of ϵ decrease by a factor 4, the measured global error decreases by the same factor as expected from (17). The computational work is measured by the total number of times the solution is advanced one time step in a cell by the basic Runge-Kutta solver (3).

When ϵ is changed by a factor β then we would expect Δt and Δx_j in a block to be modified by $\sqrt{\beta}$. The computational work in (24) will then change by a factor $\beta^{-3/2}$. The observation in Fig. 4.a is that the work behaves as β^{-1} . An explanation is that every block is not refined when β decreases and fewer cells are added than anticipated above.

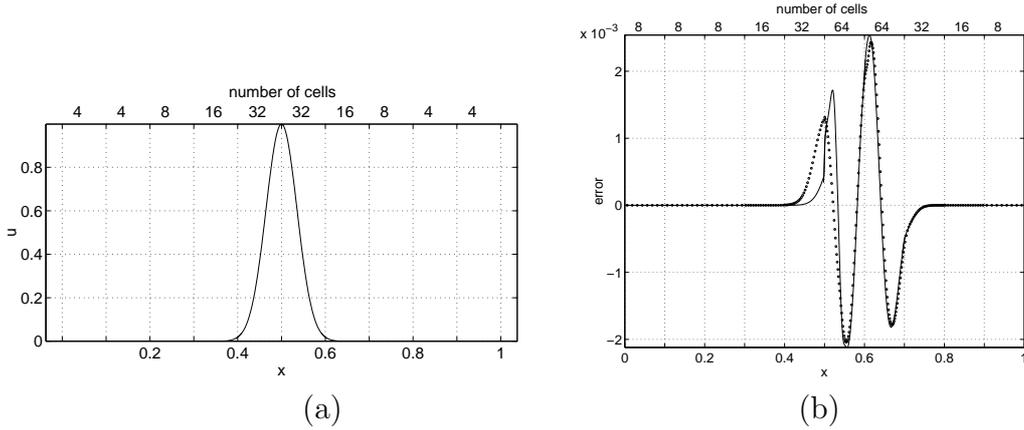


Figure 3: The scalar model equation. (a) The pulse at the initial position. (b) The estimated (dotted) and observed global errors (solid) at $T = 0.08$.

Fig. 4.b shows the corresponding results at $T = 0.25$ without a logarithmic scale versus ϵ_T , while $\epsilon_S = 1/16$ is fixed. The work is minimized when $\epsilon_T \approx \epsilon_S/2$. This is in excellent agreement with the prediction of Proposition 2 in Sect. 3. The almost doubled work for larger ϵ_T is caused by the recalculation of almost every time cycle. The time step is here chosen close to the stability limit with too optimistic a prediction of the next time step.

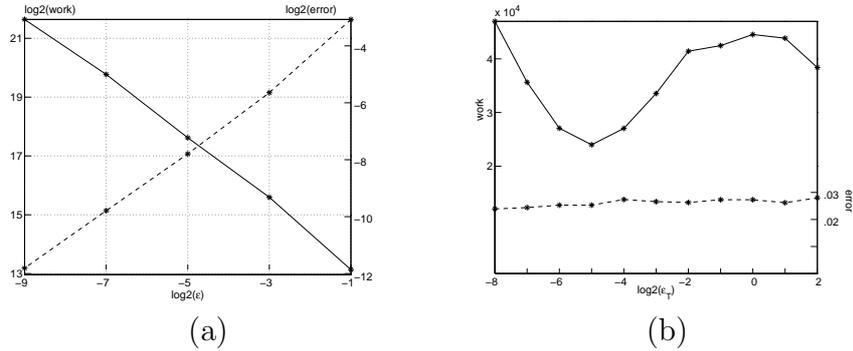


Figure 4: The computational work and global error for the scalar model equation for different $\epsilon = \epsilon_T = \epsilon_S$ (a), and for different ϵ_T when $\epsilon_S = 1/16$ (b).

The effect of the PI-regulation (34) of the time step is investigated in Fig. 5. The equation is solved up to $T = 0.25$ with adaptivity in space and time. There are between 1 and 3 grid levels in the experiments and the error tolerance in space and time is $\epsilon_T = \epsilon_S = 1/16$. A comparison is made between the PI-control (34) and the standard P-control without the memory factor in (34).

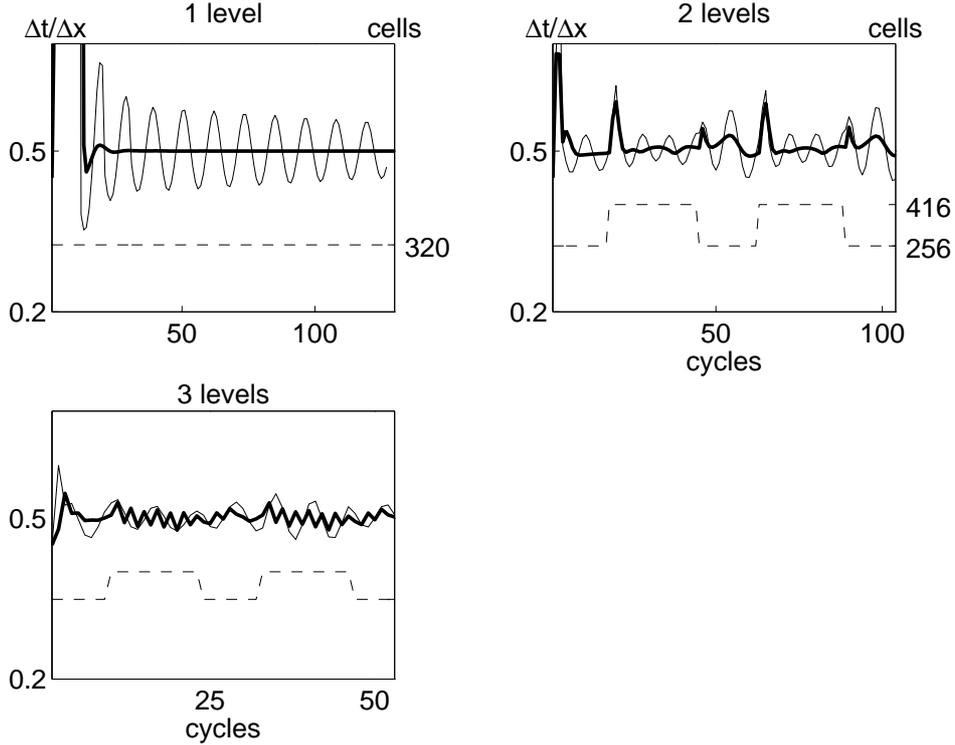


Figure 5: The adapted time step and the number of cells (dashed) for the scalar model equation. For Δt , the standard control (thin) is compared to PI-control (thick).

For one and two grid levels, the time steps are smoother with PI-control as expected from [12], [30]. For three levels the time steps are somewhat more oscillatory. The reason may be the increased number of internal time steps in each time cycle and interpolation errors in space at the block boundaries inducing perturbations in the time step regulation. In all cases, the algorithm selects a time step close to the maximum CFL-number 0.5.

4.2 The wave equation

The wave equation is written in first order form

$$\mathbf{U}_t + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \mathbf{U}_x = 0, \quad x \in [0, 1], \quad (36)$$

where $\mathbf{U} = (u_1, u_2)^T$. The component u_2 is initially zero, while the initial state of u_1 is displayed in the uppermost plot in Fig. 6.a. The boundary conditions are periodic at $x = 0$ and $x = 1$. For $t > 0$ there are two pulses traveling in opposite directions for each component illustrated in the other two plots in Fig. 6.a at $T = 0.25$. The two pulses of u_2 cancel each other twice in every time unit. Due

to the restriction on the jump in cell size at block interfaces there are only three block levels at $T = 0.25$, but five levels initially and later when the two pulses meet after one period.

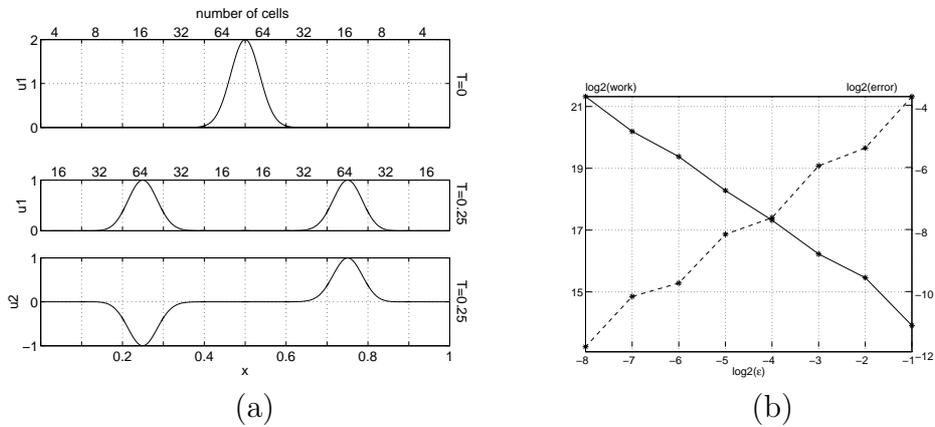


Figure 6: The wave equation. (a) From above: The initial state of u_1 , the traveling u_1 -pulses at $T = 0.25$, the traveling u_2 -pulses at $T = 0.25$. (b) The computational work and the observed error versus the prescribed local error tolerance.

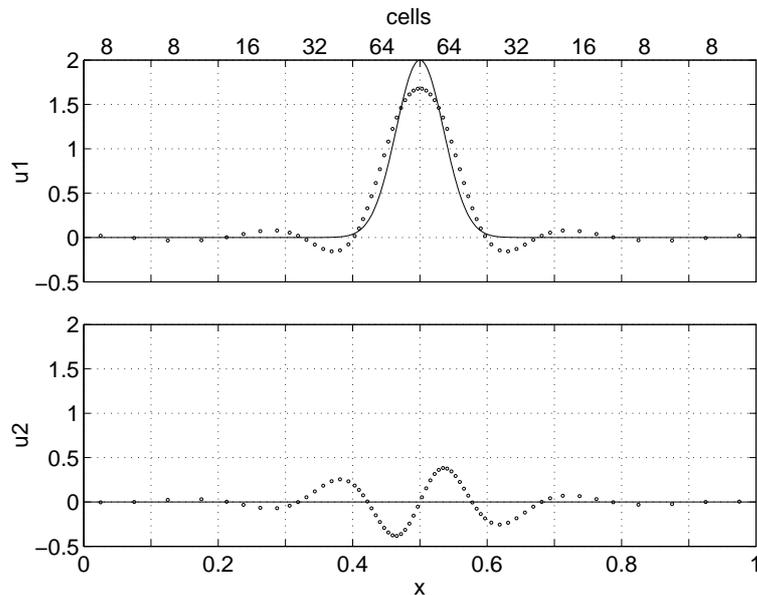


Figure 7: The computed solution (dotted) of the wave equation is compared to the exact solution (solid) at $T = 10$.

The measured global error and the total computational work at time $T = 0.5$ are plotted in Fig. 6.b versus \log_2 of the local error tolerance ϵ . The conclusion

is that the error is proportional to ϵ and the work is proportional to ϵ^{-1} in the same manner as in Fig. 4.

The solution after 10 periods ($T = 10$) with $\epsilon_T = \epsilon_S = 1/16$ is displayed in Fig. 7. An error has been accumulated in the integration. This is particularly visible in u_2 with the exact solution $u_2(x, 10) = 0$. The time history of the number of cells, the time step compared to the space step at the coarsest level, and the error growth are recorded in Fig. 8. The time series of the number of cells has a period of 0.5 as expected from the analytical solution. The time step oscillates in the neighborhood of the CFL-limit. The global error growth is linear in time.

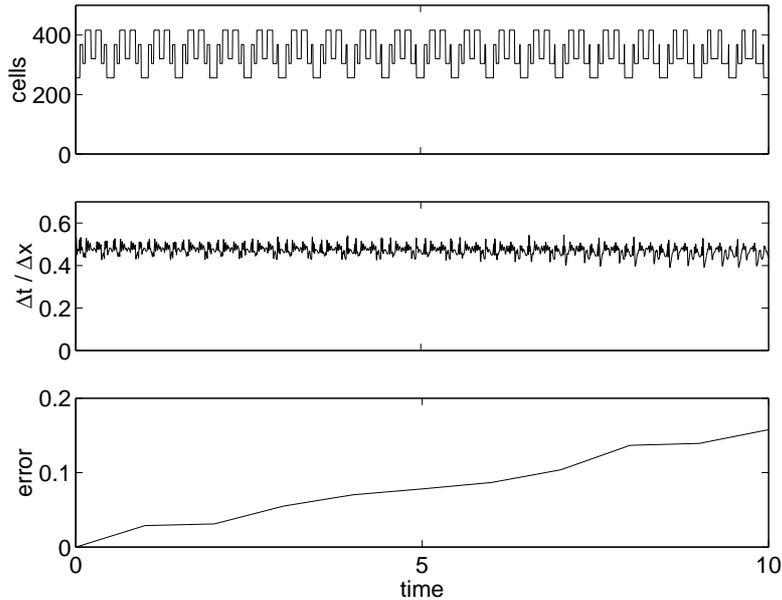


Figure 8: The number of cells (upper), the ratio $\Delta t/\Delta x$ (middle) and the L_1 -error in the adaptive solution of (36) for $t \in [0, 10]$.

4.3 Burgers' equation

Consider Burgers' inviscid equation

$$u_t + (u^2/2)_x = 0 \tag{37}$$

with initial state

$$u(x, 0) = \begin{cases} 1.1 & \text{for } x < 0.15, \\ -0.1 & \text{for } x > 0.15. \end{cases} \tag{38}$$

The solution for $t > 0$ is a shock traveling to the right with shock speed 0.5. The equation is discretized in space by the Engquist-Osher scheme [5]. The interval is partitioned into 10 blocks with at most 32 cells in one block. The solution at

$T = 1$ is found in Fig. 9.a. The block with the shock has 32 cells while the grids in other blocks in the neighborhood are determined by the jump conditions.

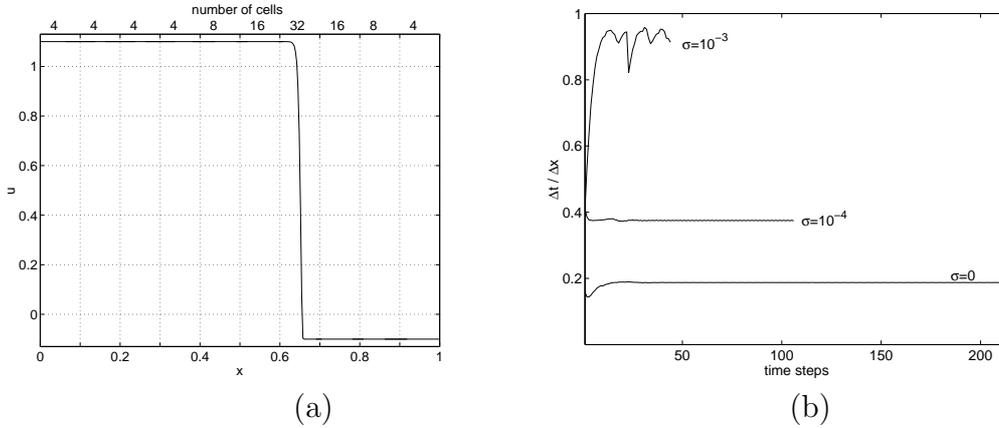


Figure 9: Burgers' equation. (a) Solution with blocked region. (b) Test of filters with different σ to avoid reduction of the time step at shocks.

Fig. 9.b shows the time steps obtained in calculations with different values of σ in the filter in (35). With $\sigma = 0$ the filter is turned off and $\Delta t / \Delta x$ is almost a constant but relatively small. With $\sigma = 10^{-4}$ the time step is approximately doubled and still almost constant. For $\sigma = 10^{-3}$ we are close to the theoretical stability limit $\Delta t / \Delta x = 1$ but the time steps are more oscillatory.

cells	L_1 error	L_∞ error
32	.0044	.53
64	.0022	.53
128	.0011	.54

Table 1: The accuracy of the solutions of Burgers' equation in L_1 and L_∞ norms.

The L_1 and the maximum errors are determined by subtracting the exact solution from the computed solution at $T = 1$. The results for different number of cells in the finest block are collected in Table 1. The maximum error is not reduced by refining Δx due to the shock but the L_1 -error is of $\mathcal{O}(\Delta x)$. This is better than the theoretical prediction in a general case [28] where the L_1 -error is of $\mathcal{O}(\sqrt{\Delta x})$.

The choice of tolerance in space ϵ_S does not have a large effect here, since the number of cells in the finest block at the shock is fixed, and jump conditions determine the grid size of the adjacent blocks. Experimental results for $\sigma = 10^{-4}$ show that the work has a minimum of $2.7 \cdot 10^4$ at $\epsilon_T = 1/8$. With a constant time step and the same small step size in all blocks the minimal work is $1.8 \cdot 10^5$.

4.4 The Euler equations

The Euler equations in one space dimension for a compressible fluid are

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{pmatrix}_x = 0.$$

The variables are the density ρ , the velocity u , the pressure p and the total energy

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2,$$

where γ , the ratio of specific heat, is 1.4 in air, see [23]. Let

$$\mathbf{U} = (\rho, \rho u, E)^T.$$

Then the initial state is

$$\mathbf{U}(x, 0) = \begin{cases} \begin{pmatrix} 0.445 \\ 0.311 \\ 8.928 \end{pmatrix} & \text{for } x < 0, \\ \begin{pmatrix} 0.5 \\ 0 \\ 1.4275 \end{pmatrix} & \text{for } x \geq 0 \end{cases} \quad (39)$$

simulating a shock tube with a membrane separating the two states for $t < 0$ as in [29]. A rarefaction wave is moving to the left when $t > 0$, and a contact discontinuity and a shock are propagating to the right, see Fig. 10. The equation is discretized in space by Osher's method of second order [26] and two different tolerances $\epsilon_T = \epsilon_S = 1/4$ and $1/32$. The solution with tolerance $1/32$ is interpolated to the coarser grid of the solution with tolerance $1/4$ for a more transparent comparison. The resolution of the solution with the lower ϵ is improved where the rarefaction wave ends in the constant state and at the contact discontinuity. The time series of the number of cells and $\Delta t/\Delta x$ in Fig. 11 have a smooth behavior with $\epsilon_T = \epsilon_S = 1/32$ in spite of the discontinuities in the solution.

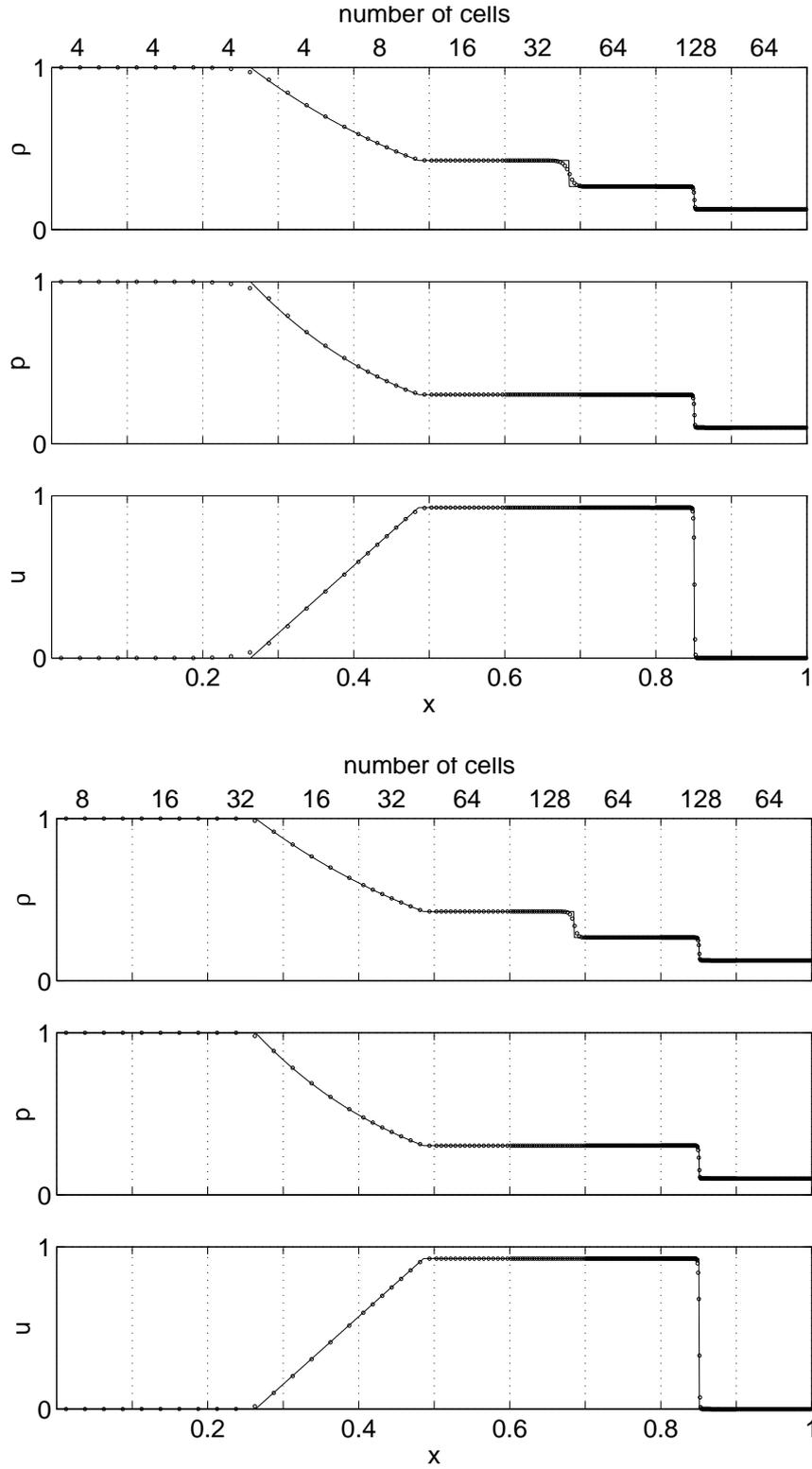


Figure 10: The analytical solution (solid) and the numerical solution (dots) of the Euler equation at $T = 0.4$ with initial conditions (39). The error tolerance is $\epsilon = 1/4$ (top) and $\epsilon = 1/32$ (bottom).

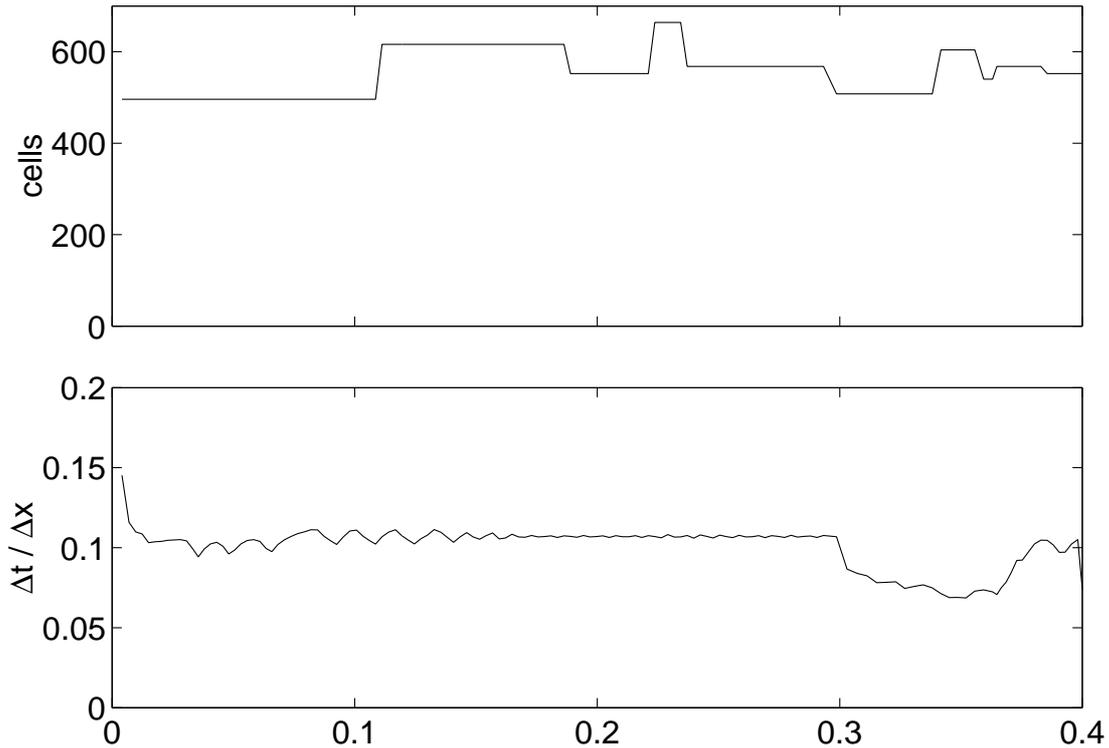


Figure 11: Time evolution for $t \in [0, 0.4]$ of the number of cells (upper) and the ratio $\Delta t/\Delta x$ (lower) for the solution of the Euler equations with $\epsilon = 1/32$ (bottom).

In these calculations, the error control in time primarily ensures the stability by suppressing small spatial oscillations. When they are detected then the time step is reduced. The reduction is often relatively large and may lead to a recalculation of the last time cycle. Too large a tolerance often incurs repeated recalculations.

cells	ϵ_S	L_1 error	L_∞ error
128	1/4	.0137	.43
128	1/8	.00799	.47
128	1/16	.00496	.47
128	1/32	.00395	.47

Table 2: Errors in the Euler calculations in the L_1 and L_∞ norm.

The global errors measured in the L_1 norm and the maximum norm are collected in Table 2 for different tolerances on the spatial error. The number of cells in the finest block is fixed at 128. As in Table 1, the L_1 -error is reduced when ϵ_S is lowered but the finest step size is fixed here. This explains why halving ϵ_S

does not improve the error by two. The L_∞ -error is not affected by ϵ_S but only by the magnitude of the discontinuities.

References

- [1] G. BECKETT, J. A. MACKENZIE, A. RAMAGE, D. M. SLOAN, On the numerical solution of one-dimensional PDEs using adaptive methods based on equidistribution, *J. Comput. Phys.*, 167 (2001), p. 372–392.
- [2] M. BERGER, P. COLELLA, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.*, 82 (1989), p. 64–84.
- [3] M. BERGER, R. LEVEQUE, Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems, *SIAM J. Numer. Anal.*, 35 (1998), p. 2298–2316.
- [4] C. J. BUDD, G. P. KOOMULLIL, A. M. STUART, On the solution of convection-diffusion boundary value problems using equidistributed grids, *SIAM J. Sci. Comput.*, 20 (1998), p. 591–618.
- [5] B. ENGQUIST, S. OSHER, Stable and entropy satisfying approximations for transonic flow calculations, *Math. Comp.*, 34 (1980), p. 45–75.
- [6] B. ENGQUIST, B. SJÖGREEN, The convergence of finite difference schemes in the presence of shocks, *SIAM J. Numer. Anal.*, 35 (1998), p. 2464–2485.
- [7] L. FERM, P. LÖTSTEDT, Efficiency in the adaptive solution of inviscid compressible flow problems, in Proceedings of WCNA 2000, *Nonlinear Analysis*, 47 (2001), p. 3467–3478.
- [8] L. FERM, P. LÖTSTEDT, Adaptive error control for steady state solutions of inviscid flow, *SIAM J. Sci. Comput.*, 23 (2002), p. 1777–1798.
- [9] L. FERM, P. LÖTSTEDT, Anisotropic grid adaptation for Navier-Stokes' equations, *J. Comput. Phys.*, 190 (2003), p. 22–41.
- [10] L. FERM, P. LÖTSTEDT, Accurate and stable grid interfaces for finite volume methods, Report 2002-012, Dept. of Information Technology, Uppsala University, Uppsala, Sweden, 2002, available at <http://www.it.uu.se/research/reports/2002-012/>, to appear in *Appl. Numer. Math.*
- [11] S. GOTTLIEB, C.-W. SHU, Total variation diminishing Runge-Kutta schemes, *Math. Comp.*, 67 (1998), p. 73–85.

- [12] K. GUSTAFSSON, Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods, *ACM Trans. Math. Software*, 17 (1991), p. 533–554.
- [13] E. HAIRER, S. P. NØRSETT, G. WANNER, *Solving Ordinary Differential Equations*, 2nd ed., Springer-Verlag, Berlin, 1993.
- [14] K. HÖRNELL, *Runge-Kutta Time Step Selection for Flow Problems*, PhD thesis, Uppsala Dissertations 16, Faculty of Science and Technology, Uppsala University, Uppsala, Sweden, 1999.
- [15] K. HÖRNELL, P. LÖTSTEDT, Time step selection for shock problems, *Commun. Numer. Meth. Engng*, 17 (2001), p. 477–484.
- [16] R. D. HORNING, J. A. TRANGENSTEIN, Adaptive mesh refinement and multilevel iteration for flow in porous media, *J. Comput. Phys.*, 136 (1997), p. 522–545.
- [17] W. HUANG, R. D. RUSSELL, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, *SIAM J. Sci. Comput.*, 20 (1999), p. 998–1015.
- [18] J. P. JESSEE, W. A. FIVELAND, L. H. HOWELL, P. COLELLA, R. B. PEMBER, An adaptive mesh refinement algorithm for the radiative transport equation, *J. Comput. Phys.*, 139 (1998), p. 380–398.
- [19] C. JOHNSON, A. SZEPESSY, Adaptive finite element methods for conservation laws based on a posteriori error estimates, *Comm. Pure Appl. Math.*, 48 (1995), p. 199–234.
- [20] R. KEPPENS, M. NOOL, G. TÓTH, J. P. GOEDBLOED, Adaptive mesh refinement for conservative systems: multi-dimensional efficiency evaluation, *Comput. Phys. Comm.*, 153 (2003), p. 317–339.
- [21] H.-O. KREISS, J. LORENZ, *Initial Boundary Value Problems and the Navier-Stokes Equations*, Academic Press, Boston, 1989.
- [22] B. VAN LEER, On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher, and Roe, *SIAM J. Sci. Comput.*, 5 (1984), p. 1–20.
- [23] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 2002.
- [24] S. LI, L. PETZOLD, Y. REN, Stability of moving mesh systems of partial differential equations, *SIAM J. Sci. Comput.*, 20 (1998), p. 719–738.

- [25] P. LÖTSTEDT, S. SÖDERBERG, A. RAMAGE, L. HEMMINGSSON-FRÄNDÉN, Implicit solution of hyperbolic equations with space-time adaptivity, *BIT*, 42 (2002), p. 134–158.
- [26] S. OSHER, F. SOLOMON, Upwind difference schemes for hyperbolic systems of conservation laws, *Math. Comp.*, 38 (1982), p. 339–374.
- [27] K. G. POWELL, A tree-based adaptive scheme for solution of the equations of gas dynamics and magnetohydrodynamics, *Appl. Numer. Math.*, 14 (1994), p. 327–352.
- [28] F. SABAC, The optimal convergence rate of monotone finite difference methods for hyperbolic conservation laws, *SIAM J. Numer. Anal.*, 34 (1997), p. 2306–2318.
- [29] G. SOD, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *J. Comput. Phys.*, 27 (1978), p. 32–78.
- [30] G. SÖDERLIND, Automatic control and adaptive time-stepping, *Numer. Alg.*, 31 (2002), p. 281–310.
- [31] J. M. STOCKIE, J. A. MACKENZIE, R. D. RUSSELL, A moving mesh method for one-dimensional hyperbolic conservations laws, *SIAM J. Sci. Comput.*, 22 (2001), p. 1791–1813.
- [32] A. VANDE WOUWER, P. SAUCEZ, W. E. SCHIESSER, Some user-oriented comparisons for partial differential equations in one space dimension, *Appl. Numer. Math.*, 26 (1998), p. 49–62.
- [33] A. VANDE WOUWER, P. SAUCEZ, W. E. SCHIESSER, eds., *Adaptive Method of Lines*, Chapman & Hall, Boca Raton, 2001.