Tensor-Product Adaptive Grids Based on Coordinate Transformations

PAUL A. ZEGELING
Mathematical Institute, Utrecht University
P.O. Box 80.010, 3508 TA, Utrecht, The Netherlands
email: zegeling@math.uu.nl

Abstract

In this paper we discuss a two-dimensional adaptive grid method that is based on a tensor-product approach. Adaptive grids are a commonly used tool for increasing the accuracy and reducing computational costs when solving both Partial Differential Equations (PDEs) and Ordinary Differential Equations (ODEs). A traditional and widely used form of adaptivity is the concept of equidistribution, which is well-defined and well-understood in one space dimension. The extension of the equidistribution principle to two or three space dimensions, however, is far from trivial and has been the subject of investigation of many researchers during the last decade. Besides the non-singularity of the transformation that defines the non-uniform adaptive grid, the smoothness of the grid (or transformation) plays an important role as well. We will analyze these properties and illustrate their importance with numerical experiments for a set of time-dependent PDE models with steep moving pulses, fronts, and boundary layers.

Keywords: method of lines, adaptive grid refinement, finite differences, moving grids, coordinate transformations

1 Introduction

Over the years a large number of adaptive grid methods have been proposed for time-dependent PDE models. Two *main* strategies of adaptive grid methods can be distinguished, namely, static-regridding methods and moving-grid or dynamic-regridding methods. In static-regridding methods (denoted by h-refinement) the location of nodes is fixed. A method of this type adapts the grid by adding nodes where they are necessary and removing them when they are no longer needed. The refinement or de-refinement is controlled by error estimates or error monitor values (which have no resemblance with the true numerical error). In dynamic-regridding methods (denoted by r-refinement) nodes are moving continuously in the space-time domain, like in classical Lagrangian methods, and the discretization of the PDE is coupled with the motion of the grid. A third approach, often combined with h-refinement in finite element methods or implemented in ODE-solvers, is characterized by the term p-refinement, which indicates the possibility of a variable order of approximation during the calculations.

In this paper we follow the second approach in a structured grid. One of the most popular techniques in one space dimension is the so-called 'equidistribution principle'. This method aims at 'equally' distributing a relevant quantity, called the 'weight function', in order to cluster grid points non-uniformly abd to reduce the numerical errors. In two space dimensions this technique is not very well-developed, see however ([10], [15], [1]) for some interesting extensions to higher space dimensions. Here, we let the grid be adapted in a tensor-like manner in the two spatial directions. The advantage is that nice properties of the method, such as non-singularity and smoothness, are preserved in each direction. The adaptive tensor-grid method is based on a semi-discretization of a system of two fourth-order PDEs for the grid variables and is being coupled to the physical PDE model re-written in a new co-ordinate system. We use the method-of-lines technique: first we discretize the PDEs in the space direction using a finite-difference approximation, so as to convert the PDE problem into a system of stiff, ordinary differential equations (ODEs) with time as independent variable. The discretization in time of this stiff ODE system then yields the required fully discretized scheme.

The layout of the paper is as follows. Section 2 is devoted to a description of the equidistribution principle in one space dimension. In section 3 we present the PDE model, the coordinate

transformation and the adaptive grid equations. The tensor-grid approach is enhanced with smoothing in both space and time direction and is defined as the solution of adaptive grid PDEs. Section 4 illustrates the importance of smoothness in terms of the truncation error and gives an numerical example for a 1d advection equation. Numerical experiments for the 2d tensor-grid are shown in section 5 for a series of test cases, among others, a 'whirlpool' model from meteorology, a parabolic PDE problem describing a rotating cone, and the so-called Gray-Scott model, a reaction-diffusion system from pattern formation. Furthermore, a 'counterexample' shows the main drawback of the proposed method. Finally, section 6 lists the conclusions and presents an outlook to future work.

2 The equidistribution principle

The general objective in structured r-refinement techniques is to find transformations (grids) that map steep solutions in the physical coordinates (waves, pulses, etc) into milder objects in the computational coordinates which can be treated 'easier' with numerical methods. The advantage is then that we may have good hope to prevent or reduce numerical oscillations near steep fronts or improve the local accuracy without increasing the number of spatial grid points too much. For special PDE models with well-known solution properties explicit transformations may be available to meet this goal. However, for general PDE systems with complicated time-dependent solution behaviour this is hardly possible to accomplish. To overcome this problem in one space dimension, a well-known principle that may be used to define the transformation *implicitly* is described by the so-called equidistribution principle. Equidistribution aims at 'equally' distributing a positive 'weight', 'grading' or 'monitor' function $\mathcal{W}$ on a spatial grid. Ideally, this $\mathcal{W}$ represents some measure of the numerical error in the discretization (but this is difficult for general situations, and, if already feasible, not computationally efficient). In other words, we would like to choose or compute a non-uniform adaptive grid $\{x_i : \ x_l = x_0 < x_1 < ... < x_{N-1} < x_N = x_r\}$ such that the contributions to the 'error' or some related quantity from each subinterval (x_k, x_{k+1}) are the same. This idea can be worked out in the formula (first, we suppress the time-dependence):

$$\Delta x_i \ \mathcal{W}_i = c, \quad i = 0, ..., N-1, \quad x_0 = x_l, \quad x_N = x_r; \quad \Delta x_i := x_{i+1} - x_i. \tag{1}$$

Equation (1) can also be recognized as a discrete version (using, for instance, the midpoint rule) of

$$\int_{x_i}^{x_{i+1}} \mathcal{W}(x) \ dx = c, \quad i = 0, ..., N-1; \quad x_0 = x_l, \quad x_N = x_r.$$

The constant c is then determined from

$$\int_{x_l}^{x_r} \mathcal{W} dx = \int_{x_0}^{x_1} \mathcal{W} dx + \int_{x_1}^{x_2} \mathcal{W} dx + ... + \int_{x_{N-1}}^{x_N} \mathcal{W} dx = c + c + ... + c \qquad (N \text{ times}),$$

giving $c = \frac{1}{N} \int_{x_l}^{x_r} \mathcal{W} dx$. The equidistribution principle becomes:

$$\int_{x_i}^{x_{i+1}} \mathcal{W} dx = \frac{1}{N} \int_{x_l}^{x_r} \mathcal{W} dx, \quad i = 0, ..., N-1,$$

which means that the monitor function $\mathcal{W}$ is equally distributed over all subintervals. The simplest way of describing equidistribution is to note that from (1) follows that grid cells Δx_i are small where $\mathcal{W}_i$ is large, and vice versa, since their product is kept constant. Moreover, equation (1) can be interpreted as a discrete (finite difference) approximation of the problem

$$\frac{dx}{d\xi} \ \mathcal{W}(x(\xi)) = c, \quad 0 < \xi < 1; \quad x(0) = x_l, \quad x(1) = x_r. \tag{2}$$

Since $\frac{dx}{d\xi} = 1/\frac{d\xi}{dx}$, this is equivalent with

$$\frac{d\xi}{dx} = c\ \mathcal{W}(\xi(x)), \quad x_l < x < x_r; \quad \xi(x_l) = 0, \quad \xi(x_r) = 1.$$

From this expression we can find an explicit formula for the (inverse) transformation $\xi(x)$. Note that

$$1 = \xi(x_r) - \xi(x_l) = \int_{x_l}^{x_r} \frac{d\xi}{dx} dx = c \int_{x_l}^{x_r} \mathcal{W} d\bar{x}$$

from which follows $c = \frac{1}{\int_{x_l}^{x_r} \mathcal{W} d\bar{x}}$ and thus

$$\frac{d\xi}{dx} = \frac{\mathcal{W}(x)}{\int_{x_l}^{x_r} \mathcal{W} d\bar{x}}.$$

Integrating once gives

$$\xi(x) = \int_{x_l}^{x} \frac{\mathcal{W}(\bar{x})}{\int_{x_l}^{x_r} \mathcal{W}(\bar{x}) d\bar{x}} d\bar{x} = \frac{\int_{x_l}^{x} \mathcal{W}(\bar{x}) d\bar{x}}{\int_{x_l}^{x_r} \mathcal{W}(\bar{x}) d\bar{x}}.$$

Taking the derivative w.r.t. ξ of (2) yields the two-point boundary value problem for $x(\xi)$:

$$\frac{d}{d\xi}[\frac{dx}{d\xi}\mathcal{W}(x(\xi)] = 0, \quad x(0) = x_l, \quad x(1) = x_r. \tag{3}$$

An important issue is the choice of the weight function $\mathcal{W}$. It may be obvious that there is no general rule for all cases. In literature many different functions have been used for all kinds of situations. For example, in ([2], [8], [11]) the popular arc-length monitor $\mathcal{W} = \sqrt{1 + u_x^2}$ is used. Pereyra et al [17] derived estimates for the local truncation error to define the weight function. Already in 1913, Sundman [20] (recently improved by Leimkuhler [14] realized the usefullness of a transformation with $\mathcal{W} = u^{p/q}$ for the three-body ODE system. The curvature monitor $\mathcal{W} = (1 + u_{xx}^{2p})^{\frac{1}{2p}}$ places the grid points in regions of large second-derivatives and is investigated in [6]. As an extension of the arc-length function for the gas dynamics equations in [19] a weight function is proposed that depends on the entropy $s := p\gamma/\rho$. Finally, Budd et al [5]) derive simple but extremely effective monitor functions, such as $\mathcal{W} = u^{p-1}$, $\mathcal{W} = |u|^2$, or $\mathcal{W} = x^\alpha |u|^\beta |u_x|^\gamma$ for PDE models in which scaling, blow up, or similarity properties should be preserved by the adaptive grid. This (incomplete) short overview of different weight functions in 1d indicates that a general rule for $\mathcal{W}$ can not be given.

In two space dimensions, however, the situation is even more complicated. Although several authors ([10], [15], [1]) have described new interesting ideas that all lead in some sense to equidistribution, when restricting the respective method to one space dimension, no proper extension of this principle is yet known for higher space dimensions. In the following section a simplified tensor-grid approach is proposed that yields equidistribution in each spatial direction with additional smoothness properties.

3 Tensor-product adaptive grids

Consider now the two-dimensional time-dependent PDE model

$$\partial_t u = \epsilon \Delta u - \beta(u, x, y, t) \cdot \nabla u + s(u, x, y, t), \tag{4}$$

where $(x, y) \in [x_l, x_r] \times [y_l, y_u]$, $t \in [0, T]$, $0 \leq \epsilon$ is the diffusion coefficient, β the velocity vector and s a nonlinear sourceterm.

It is common and useful in structured r-refinement methods to first apply a coordinate transformation to the physical PDE model (4). The adaptive grid can then be seen as a uniform

discretization of this mapping in the new variables. In this paper we make use of a transformation of variables ([24], [13]) in a dimensionally-split approach

$$\xi = \xi(x,t), \quad \eta = \eta(y,t), \quad \theta = t, \tag{5}$$

in which (x,y) and $(\xi,\eta) \in [0,1] \times [0,1]$ denote the physical and computational coordinates, respectively. Applying this transformation to equation (4) gives (a similar derivation can be made for a system of PDEs)

$$\mathcal{J}\ \partial_\theta u - \partial_\xi u\ \partial_\eta y\ \partial_\theta x - \partial_\eta u\ \partial_\xi x\ \partial_\theta y = \epsilon[\partial_\xi(\frac{\partial_\eta y\ \partial_\xi u}{\partial_\xi x}) + \partial_\eta(\frac{\partial_\xi x\ \partial_\eta u}{\partial_\eta y})] \tag{6}$$

$$-\beta_1 \partial_\eta y\ \partial_\xi u - \beta_2 \partial_\xi x\ \partial_\eta u + s(u, x(\xi,\theta), y(\eta,\theta), \theta), \tag{7}$$

where $\mathcal{J} := \partial_\xi x\ \partial_\eta y$ is the Jacobian of the transformation. Note that $\partial_x \xi = [\partial_\xi x]^{-1}$ and $\partial_y \eta = [\partial_\eta y]^{-1}$ measure the grid densities in each separate direction.

The adaptive grid in terms of the mapping is determined as a solution of two fourth-order PDEs in ξ and η with an additional time-dependent component. We set

$$\partial_\xi\left[(\mathcal{S}_1(\mathcal{J}_1) + \tau\ \partial_\theta \mathcal{J}_1)\mathcal{W}_1\right] = 0, \tag{8}$$

$$\partial_\eta\left[(\mathcal{S}_2(\mathcal{J}_2) + \tau\ \partial_\theta \mathcal{J}_2)\mathcal{W}_2\right] = 0, \quad (\tau \geq 0),$$

with suitable boundary conditions for x (similar conditions hold for y):

$$x(0,\eta) = x_l,\ x(1,\eta) = x_r,\ \partial_n x(0,\eta) = \partial_n x(1,\eta) = 0.$$

The operators $\mathcal{S}_1$ and $\mathcal{S}_2$ are direction-specific versions of the operator $\mathcal{S}$ defined as:

$$\mathcal{S} = \mathcal{I} - \sigma(\sigma+1)(\Delta\xi)^2 \partial^2_{\xi\xi} \qquad (\sigma \geq 0), \tag{9}$$

where $\mathcal{J}_1 := \partial_\xi x$ and $\mathcal{J}_2 := \partial_\eta y$ are the 'one-dimensional' Jacobians, respectively. As mentioned before, several choices for the weight functions in (8) can be made. Here, we simply take

$$\mathcal{W}_1 = \sqrt{1 + \alpha\ \max_\eta [\partial_\xi u]^2}, \tag{10}$$

$$\mathcal{W}_2 = \sqrt{1 + \alpha\ \max_\xi [\partial_\eta u]^2} \qquad (\alpha \geq 0).$$

The parameter α is an adaptivity parameter: $\alpha = 0$ yields $\mathcal{W}_1 = \mathcal{W}_2 = 1$ and thus a uniform grid distribution (this can easily be derived from (8) and (9)); for increasing values of α the derivatives $\partial_\xi u$ and $\partial_\eta u$ are stressed more and more with the effect of higher spatial grid adaptation. It can be shown that the transformation (5) as a solution of equations (8), (9), (10) satisfies the 'grid-consistency' condition

$$\mathcal{J} > 0, \quad \forall\ \theta \geq 0,\ \text{and}\ \ (\xi,\eta) \in [0,1] \times [0,1],$$

and also the 'local quasi-uniformity' property

$$\left|\frac{\partial^2_{\xi\xi} x}{\partial_\xi x}\right| \leq 1/\sqrt{\sigma(\sigma+1)\Delta\xi}, \tag{11}$$

$$\left|\frac{\partial^2_{\eta\eta} y}{\partial_\eta y}\right| \leq 1/\sqrt{\sigma(\sigma+1)\Delta\eta}.$$

To prove these theoretical properties of the grid, the results from [11] have shown to be very useful. The first property is equivalent to *non-singularity* of the mapping, which is, of course, a

minimum demand. The second property concerns the *smoothness* of the mapping (see below for more details). Note that for $\sigma = \tau = 0$ (i.e. without smoothing operators) the grid equations (8) reduce to

$$\partial_\xi \left[\mathcal{J}_1 \mathcal{W}_1\right] = 0, \tag{12}$$
$$\partial_\eta \left[\mathcal{J}_2 \mathcal{W}_2\right] = 0,$$

which can be easily solved, just as in the one-dimensional case, to obtain an explicit expression for the (inverse) coordinate transformation

$$\xi(x,t) = \int_{x_l}^{x} \mathcal{W}_1 \ \mathrm{d}\bar{x} / \int_{x_l}^{x_r} \mathcal{W}_1 \ \mathrm{d}\bar{x}, \tag{13}$$

$$\eta(y,t) = \int_{y_l}^{y} \mathcal{W}_2 \ \mathrm{d}\bar{y} / \int_{y_l}^{y_u} \mathcal{W}_2 \ \mathrm{d}\bar{y}.$$

Moreover, equations (12) can be seen as the Euler-Lagrange equations of the quadratic 'grid-energy' functionals

$$\mathcal{E}_1(\xi) = \int_{x_l}^{x_r} \frac{1}{\mathcal{W}_1} (\partial_x \xi)^2 \mathrm{d}x, \quad \text{and} \quad \mathcal{E}_2(\eta) = \int_{y_l}^{y_u} \frac{1}{\mathcal{W}_2} (\partial_y \eta)^2 \mathrm{d}y. \tag{14}$$

Formulae (14) can be taken to represent the energy of a system of springs with spring constants $\mathcal{W}_{1,2}$ spanning each subinterval (grid points can then be seen as the mass points of the spring system). The grid point distribution resulting from 'equidistribution' thus represents the equilibrium state of the spring system, i.e., the state of minimum 'energy'. Note that, from (12), i.e. without any kind of smoothing, follows *directly* that, as $W_{1,2} > 1$: $\mathcal{J}_{1,2} > 0$, and therefore $\mathcal{J} = \mathcal{J}_1 \mathcal{J}_2 > 0$. Using the fact that $\Delta\xi$ and $\Delta\eta$ are constant, the continuous property $\mathcal{J} > 0$ both for equations (12) and for equations (8) can be translated in semi-discrete terms as

$$\Delta x_{i,j}(\theta) > 0, \quad \forall \theta \in [0,T], \quad \forall i,j, \tag{15}$$

$$\Delta y_{i,j}(\theta) > 0, \quad \forall \theta \in [0,T], \quad \forall i,j. \tag{16}$$

In other words, these relations state that the grid points can never cross one another. Property (11) can be read in semi-discrete terms as

$$\frac{1}{1+1/\sigma} \leq \frac{\Delta x_{i+1,j}(\theta)}{\Delta x_{i,j}(\theta)} \leq 1 + 1/\sigma, \quad \forall \theta \in [0,T], \quad \forall i,j, \tag{17}$$

and similar relations for the y-direction. Relation (17) means that the variation in successive grid cells in both directions can be controlled by the parameter σ at every point of time. The importance of this property will be discussed in the next section. Finally, the parameter τ in (8) has a smoothing effect in the time-direction to prevent the grid from adjusting too quickly to new values of the weightfunctions $\mathcal{W}_{1,2}$. It can be seen as a small delay factor for the grid movement.

4 A numerical experiment in one space dimension

In the previous section we have shown that the adaptive grid in terms of a coordinate transformation as defined by the solution of the PDEs (8) possesses nice properties, such as grid consistency and grid smoothness. In the following, we discuss the backgrounds and effects of the (non-)smoothness of the grid and give a numerical illustration in one space dimension to support

the theory. An important quantity in this respect is the so-called 'grid size ratio', sometimes also denoted by 'local stretching factor'. It is defined by

$$r := \frac{x_i - x_{i-1}}{x_{i+1} - x_i} = \frac{\Delta x_{i-1}}{\Delta x_i} := \frac{q}{p}.$$

For the numerical treatment of the physical PDE model we need to approximate spatial derivatives. For instance, the first-order derivatives can be approximated using central-finite-differences:

$$\partial_x u|_i = \frac{u_{i+1} - u_{i-1}}{p+q} + T.$$

The truncation error T can be expressed in several different, but mathematically equivalent, ways as follows:

$$\begin{aligned} T &= -\frac{\partial^2_{xx} u}{2}(1-r)\Delta x_i - \frac{\partial^3_{xxx} u}{6}(1-r+r^2)\Delta x_i^2 + ... = \\ &= \frac{\Delta\xi^2}{6}(3\ \partial^2_{\xi\xi} x\ \partial^2_{xx} u + [\partial_\xi x]^2\ \partial^3_{xxx} u) + \mathcal{O}(\Delta\xi^4) = \\ &= \frac{\Delta x_i^2}{6}(3\frac{\partial^2_{\xi\xi} x}{[\partial_\xi x]^2}\ \partial^2_{xx} u + \partial^3_{xxx} u) + \mathcal{H}.\mathcal{O}.\mathcal{T}. \end{aligned}$$

In the case of a uniform grid we have $r = 1$, and the relation for T reduces to the standard second-order expression

$$T = -\frac{\Delta\xi^2}{6}\ \partial^3_{xxx} u + \mathcal{O}(\Delta\xi^4).$$

On the other hand, for $r > 1$, i.e. a non-uniform grid, we can derive that the truncation error will only be of second-order

$$\Longleftrightarrow r = 1 + \mathcal{O}(\Delta x_i), \tag{18}$$

(which is called 'quasi-uniformity')

$$\Longleftrightarrow \frac{\partial^2_{\xi\xi} x}{[\partial_\xi x]^2} = \mathcal{O}(1).$$

A local version of this property is named 'local quasi-uniformity' (see also the previous section) and can be expressed as

$$\mathcal{O}(1) = \frac{1}{K} \leq r \leq K = \mathcal{O}(1). \tag{19}$$

It is easy to make the the link between (18) and (19). If we choose $\sigma = \mathcal{O}(1)$ in the smoothed adaptive transformation, then we automatically obtain with $K := 1 + 1/\sigma$: $K = \mathcal{O}(1)$. In this sense, the additional smoothing operator gives us the desired property for second-order approximation of the first spatial derivative (note that similar derivations can be made for $\partial^2_{xx} u|_i$ and other spatial derivatives appearing in the physical PDE).

As a numerical illustration we apply a 1d-version of the adaptive grid method defined by (8), where we have frozen the y-direction, to the following advection model:

$$\partial_t u + 4\cos(4\pi t)\partial_x u = 0.$$

An exact solution for this problem is given by $u^*(x,t) = \sin^{1000}(\pi(x - \frac{1}{\pi}\sin(4\pi t)))$ and describes an extremely sharp pulse that moves periodically in the time direction from left to right and backwards again through the domain. For the weight function we take the 1d-restriction of (10):

$$\mathcal{W} = \sqrt{1 + \alpha\ [\partial_\xi u]^2}.$$

N	$\alpha = 0$	$\alpha = 1,\ \sigma = 0$	$\alpha = 1,\ \sigma = 2$
50	0.721312	0.624699	0.387192
100	0.577044	0.432729	0.116723
200	0.509914	0.274196	0.033135
400	0.327693	0.142711	0.025296
800	0.109807	0.072737	0.017410
1600	0.027250	no solution	0.011549

Table 1: The maximum error $||e||_\infty$ at $t = 0.4$ for the 1d-advection model

Table 1 confirms the theoretical considerations: if we add the smoothing operator ($\sigma = 2$) to the adaptation procedure, the convergence of the method, when doubling the number of spatial grid points, is improved dramatically compared with the uniform grid case ($\alpha = 0$) and with the pure equidistribution case ($\sigma = 0$). In these runs at $t = 0.4$ the parameters τ and tol (the tolerance in the time-integrator DASSL [18]) were deliberately chosen very small to see the effect of the spatial part of the smoothing: $\tau = 10^{-6}$ and $tol = 10^{-8}$. It is observed that the non-smooth case experiences a severe degradation of performance, even resulting in a breakdown of the method for $N = 1600$. The solutions and the grid history for $N = 100$ are displayed in Figures 1 and 2. Note the much bigger error in the solution and the 'unsmooth' trajectories for $\sigma = 0$ compared with the plots for $\sigma = 2$. It is also clear that the uniform grid solution is far too inaccurate.

5 Numerical experiments in two space dimensions

In this section we will demonstrate the usefulness of the adaptive tensor-grid method in a two-dimensional setting. The method is applied to a set of PDE models from different application areas. In the numerical tests, unless specified otherwise, the default choices in the experiments are the following: a uniform starting grid, a time-tolerance of 10^{-3}, and grid parameters $\alpha = \sigma = 1,\ \tau = 10^{-3}$. For this value of σ the grid cell ratio's always remain bounded between $1/2$ and $3/2$. We have discretized the spatial derivatives both in the physical and the adaptive grid PDEs with central finite differences. It is, of course, clear that more suitable approximations, w.r.t. accuracy and efficiency, can be made depending on each separate PDE model. The stiff time-integrator DASSL [18] takes care of the resulting ODE system. This code makes use of $h-p$-refinement in the time-direction, which means variable timestep and a variable order (less than 5) of approximation in the BDF-(Backward-Differentiation Formulas)integrator.

5.1 A whirlpool model

An interesting test model from meteorology is described by the hyperbolic PDE

$$\partial_t u = -\frac{v_t}{v_{t,max}}\frac{y}{r}\partial_x u + \frac{v_t}{v_{t,max}}\frac{x}{r}\partial_y u, \tag{20}$$

where

$$r = \sqrt{x^2 + y^2}, \quad v_t = \frac{\tanh(r)}{\cosh^2(r)}, \quad v_{t,max} = 0.385, \tag{21}$$

with initial and boundary conditions: $u|_{t=0} = -\tanh(\frac{y}{2})$, $\partial_n u|_{\partial\Omega} = 0$, on the domain $(x, y) \in [-4, 4] \times [-4, 4],\ t \in [0, 4]$. This model describes the formation of cold and warm fronts in a two-dimensional setting. Beginning with a narrow region of high gradients (a front), a fixed (in time) rotational velicity field will act to twist the front in a manner similar to that observed on daily-weather maps (positive solution values correspond to a warm front and negative values to a cold front). Details on this model can be found in [21]. A complicated structure with high spatial activity, similar to a whirlpool, develops in the center of the domain. Figure 3 shows the

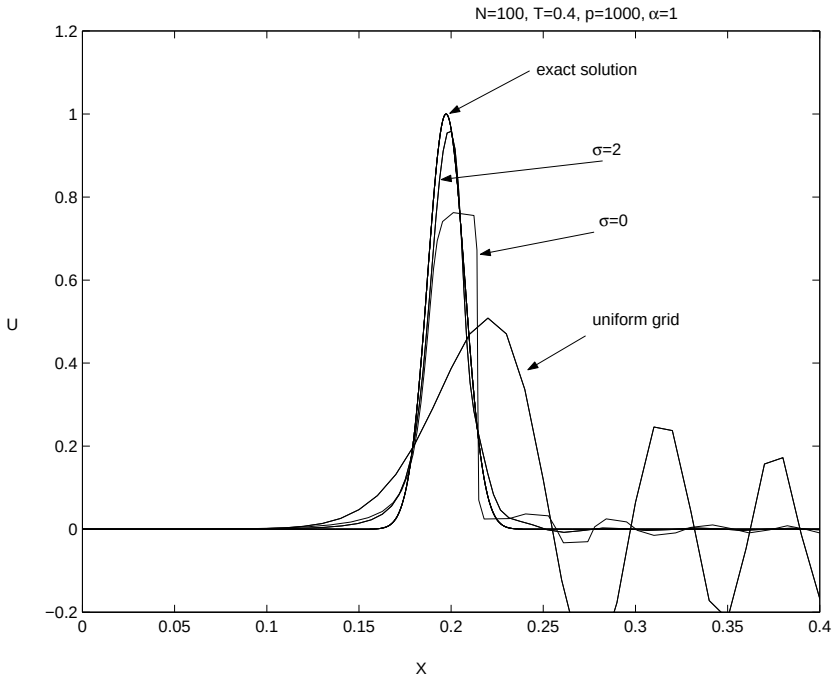

Figure 1: Numerical solutions for the 1d-advection model for $N = 100$ at $t = 4$ for different choices of α and σ (zoomed in several times around $x = 0.2$).

Grid size	$\alpha = 0$	$\alpha = 1,\ \sigma = 0$	$\alpha = 1,\ \sigma = 1$	$\alpha = 10,\ \sigma = 1$
19×19	0.99983	0.57177	0.62015	0.50516
29×29	0.74773	0.27647	0.25053	0.24930
39×39	0.52421	0.15113	0.15087	0.13512
49×49	0.29419	0.10606	0.09828	0.09244
59×59	0.19357	0.08476	0.08387	0.07491

Table 2: The maximum error $||e||_\infty$ at $t = 4.0$ for the whirlpool model

grids and numerical solutions on a 49×49 grid at $t = 0.0,\ 1.6,\ 2.8$ and 4.0. Also contourplots are given for comparison with the uniform grid and the adaptive grid case for $t = 4.0$ at which point of time the whirlpool has developed. The adaptive solution compares favorably to the uniform solution in which the inner-layer structure of the whirlpool is not resolved very well at all. Note that we haven taken here the re-scaled value $\alpha = 10$ instead of $\alpha = 1$, since the domain and the solution have larger scales. In Table 2 the maximum error is displayed at the final time for different values of α and σ. We see that for $\alpha = \sigma = 1$, which would be a 'standard' choice for a unit-square domain and solution values of $\mathcal{O}(1)$, the method performs not as good as for the re-scaled $\alpha = 10$ case. The difference between the smooth ($\sigma = 2$) and the non-smooth ($\sigma = 0$) case is not so profound for this model, because the steep parts of the solution remain concentrated in the centre of the domain for all time.

5.2 A combustion model

A reaction-diffusion system (see [12] for more details) of a so-called single one-step reaction of a mixture of two chemicals that stems from combustion theory often gives rise to moving and stationary layers in the solution. A simplified scalar version of this model is defined by

$$\partial_t u = \epsilon \Delta u + D(1 + \beta - u)\mathrm{e}^{-\delta/u}, \tag{22}$$

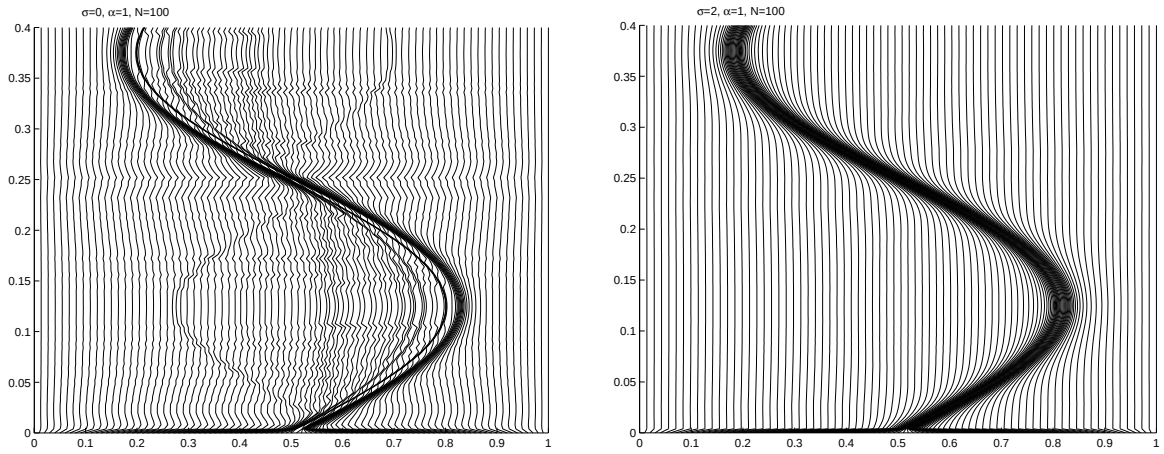

Figure 2: Numerical results for the 1d-advection model: irregular adaptive grid for $\sigma = 0$ (left) vs. smooth adaptive grid for $\sigma = 2$ (right).

where $D = \frac{Re^{\delta}}{\beta\delta}$ denotes the Damkohler number, R the reaction rate, δ the activation energy, and β the heat release, respectively. As an initial condition we take a sharp non-symmetric hump:

$$u|_{t=0} = 1 + \sin^{50}(\pi x)\sin^{10}(\pi y).$$

Note that smaller values of the diffusion coefficient ϵ yield steeper fronts in the model: we choose $\epsilon = 0.1$ which gives rise to very sharp transitions. At the boundary the Dirichlet condition $u|_{\partial\Omega} = 1$ is imposed. Due to the interaction between diffusion and reaction a moving steep layer is observed that moves quickly to the boundaries and then settles down in a steady-state. The other chemical parameters are typical for a standard test model: $R = 5$, $\beta = 1$ and $\delta = 20$. In Figure 4 the adaptive-grid solutions and the grids are depicted at different points of time on a 39×39 tensor-grid. It can be observed that the adaptive grid is nicely situated around the steep moving flame front, from the initial phase up to the boundary layer steady-state.

5.3 The Gray-Scott model

From pattern formation the following reaction-diffusion system ([8], [16]) exhibits complicated solution behaviour:

$$\partial_t u = 8 \times 10^{-5}\Delta u - uv^2 + 0.02(1-u), \tag{23}$$

$$\partial_t v = 4 \times 10^{-5}\Delta v + uv^2 - 0.086v. \tag{24}$$

In this model self-replicating spots have been observed. These are regions in which the (chemical) concentrations of some of the species exhibit large amplitude perturbations from a surrounding homogeneous state. Depending on system parameters, these regions can enlarge and split so that the spots replicate in a complex, and as yet incompletely understood, manner. Starting with a pulse, spot, or a small block in the middle of the domain as an initial condition, we see at some point of time that splitting may occur due to a complicated interaction between diffusion and reaction terms, resulting in four spots, later on in eight spots, etcetera. The initial conditions are

$$u(x,y,0) = \begin{cases} 0.5 & \text{if } 0.45 \le x \le 0.55 \quad \text{and} \quad 0.45 \le y \le 0.55, \\ 1 & \text{elsewhere,} \end{cases}$$

$$v(x,y,0) = \begin{cases} 0.25 & \text{if } 0.45 \le x \le 0.55 \quad \text{and} \quad 0.45 \le y \le 0.55, \\ 0 & \text{elsewhere,} \end{cases}$$

on the spatial domain $[0,1] \times [0,1]$. Note that it would be almost impossible for a uniform grid of moderate size to deal with this tiny initial block. The adaptive-grid solutions and grids in Figure 5 demonstrate clearly the capability of the tensor-grid to follow the splitting of the spots. For this experiment, we have used a 39×39 grid with endpoint of time $t = 500$.

5.4 A 'counterexample'

As we have seen, the proposed tensor-product grid method can be applied successfully to a set of different PDE models in two space dimensions. However, it is also obvious that this method may fail dramatically for a large class of other important models. A relatively simple model that can be used as a 'counterexample' (see [23] for details) is given by

$$\partial_t u = \epsilon \Delta u - u\ \partial_x u - (\frac{3}{2} - u)\ \partial_y u, \qquad 0 < \epsilon \ll 1. \tag{25}$$

The exact solution for this problem reads

$$u^*(x,y,t) = \frac{3}{4} - \frac{1}{4}\ \frac{1}{1 + \mathrm{e}^{\frac{-4x+4y-t+2}{32\epsilon}}}.$$

This is a scalar version of the 2D system of Burgers' equations. The solution describes a wave front with a steep transition area of thickness $\mathcal{O}(\epsilon)$ that moves under an angle of 135^0 with the positive x-axis. With the adaptive tensor-product grid, the grid points find an 'optimal' position at $t = 2$ as shown in Figure 6: a uniform grid distribution. Any other choice for the weight function or method parameters yields a preference direction which definitely will not improve the grid distribution, since the solution is a skew wave. It is clear that a skew wave (and many other types of more complicated layers than this one) can *not* be resolved with a tensor-product grid. In this figure we show fully-2d adaptive grids for this model as well, which are based on the more general transformation $x(\xi,\eta,\theta)$ and $y(\xi,\eta,\theta)$. A description of that method will appear in the subsequent paper [25]. This example shows clearly that 2d grid adaptation based on the 'full' tranfsormation is needed to cope with general solution structures, such as waves and moving layers.

5.5 A rotating cone

To show the effects of the adaptivity parameter α and the weight function $\mathcal{W}$ it is of interest to examine the linear parabolic equation described by

$$\partial_t u = \Delta u + f(x,y,t), \quad (x,y) \in [-1,2] \times [-1,2]. \tag{26}$$

The source term f is chosen so that the exact solution is

$$u^*(x,y,t) = \mathrm{e}^{-80[(x-r(t)))^2 + (y-s(t))^2]},$$

where

$$r(t) = \frac{1}{4}(2 + \sin(\pi t)), \qquad s(t) = \frac{1}{4}(2 + \cos(\pi t)).$$

This solution is a rotating cone with initial condition $\mathrm{e}^{-80((x-0.5)^2+(y-0.75)^2)}$ that moves around in circles with a constant speed. During the movement, the shape of the cone does not change. Another option for the weight functions, especially for this example, could be of the form

$$\tilde{\mathcal{W}}_1 = \tilde{\mathcal{W}}_2 := \sqrt{1 + \tilde{\alpha}\ u^2}$$

to stress the solution values at the peak itself instead of the gradients. In Table 3 and Figure 7, numerical results are displayed for different choices of the weight function and the adaptivity

Grid size	$\alpha = 0$	$\alpha = 1,\ \sigma = 0.2$	$\alpha = 1,\ \sigma = 1$	$\tilde{\alpha} = 1,\ \sigma = 0$	$\tilde{\alpha} = 10,\ \sigma = 0$
19×19	0.43590	0.71109	0.11618	0.43998	0.18423
29×29	0.25521	0.48283	0.16219	0.26116	0.03766
39×39	0.14363	0.21347	0.10206	0.15026	0.02522
49×49	0.08629	0.20423	0.08304	0.09600	0.01725
59×59	0.05636	0.09953	0.06081	0.06624	0.01147

Table 3: The maximum error $||e||_\infty$ at $t = 2.0$ for the rotating cone model

parameters α and $\tilde{\alpha}$. A few observations can be made from these simulations. First, we see again, that decreasing the smoothness, i.e., taking the rather small value $\sigma = 0.2$, negatively influences the maximum error for the case of an arc-length weight function. Second, the alternative choice $\tilde{\mathcal{W}}_{1,2}$ where the solution value is emphasized, and not the gradient of the solution, gives much better numerical results, although the adaptive grids themselves, perhaps surprisingly, look not too different. Note that, if the value of $\tilde{\alpha}$ is increased, the error in the numerical solution is reduced significantly. The solutions for $\alpha = 1$ exhibit a strange decrease in amplitude both at the top of the pulse and at the foot of the pulse, although for $\sigma = 1$ this behaviour is less pronounced than for the unsmooth run with $\sigma = 0.2$. For the alternative weight function with $\tilde{\alpha} = 10$, this effect is almost annihilated. This experiment indicates that an optimal choice for the weight function and an optimal value of the adaptivity parameter can not be given beforehand. This issue needs further attention and will be investigated in a subsequent report.

6 Conclusions and comments

In this study we have used an adaptive tensor-grid approach for the numerical solution of time-dependent PDE models with steep fronts, rotating or splitting pulses from pattern formation, and boundary layers. The adaptive moving grid used in the experiments is based on the equidistribution principle in two directions enhanced with smoothing operators in the space- as well as in the time-direction. Both theory and experiments in a one-dimensional situation indicated the importance of these additional features of the method. We have shown the effectiveness of the adaptive tensor-grid in 2d to deal with regions of high spatial activity in the PDE solution. The main advantage of the proposed method is the fact that non-singularity and smoothness of the underlying transformation can always be guaranteed, which is one of the known bottlenecks in the theory behind existing structured r-refinement methods. The method is very useful for solutions with 'rectangular' or 'pulse-like' steep components moving through the spatial domain. A 'counterexample' with a 'non-rectangular' wave showed the main handicap of the tensor-grid approach, and indicates the need for full 2d grid-adaptation for these situations (see for example [25]). Another point of discussion is the choice of the weight function. Results were shown for which the traditional choice such as the arc-length weight function is inferior to much simpler functions in which the solution itself is emphasized instead of the gradients. Further theoretical research is needed to get a deeper understanding of this important issue. Moreover, it is necessary to increase the robustness of the adaptivity parameter in the weight functions. Interesting new developments in this respect can be found in [2], where the parameter is defined as a time-varying constant depending on the volume under the 2d solution surface. Furthermore, in [9] it is claimed that under certain conditions of the non-uniform tensor-grid so-called 'supra-convergence' should occur. This higher-order convergence behaviour was not observed in our experiments, however. It could, therefore, be of interest to investigate whether the grid defined by our transformation does or does not satisfy the, rather untransparant, conditions for 'supra-convergence'. The computational efficiency, especially for more complicated models in 2D and 3D, can be improved by switching from a direct solver to iterative solvers ([4], [22]) used

for the linear systems in the implicit time-integrator. Finally, we note that an efficient combination of the adaptive tensor-grid with a local-uniform grid (h-)refinement, such as [3], could be of importance for general applications. The effectiveness of $h-r$-refinement techniques was demonstrated in [7] in the case of a stationary model. For time-evolutionary models, however, this is still an open question and should be investigated in more detail.

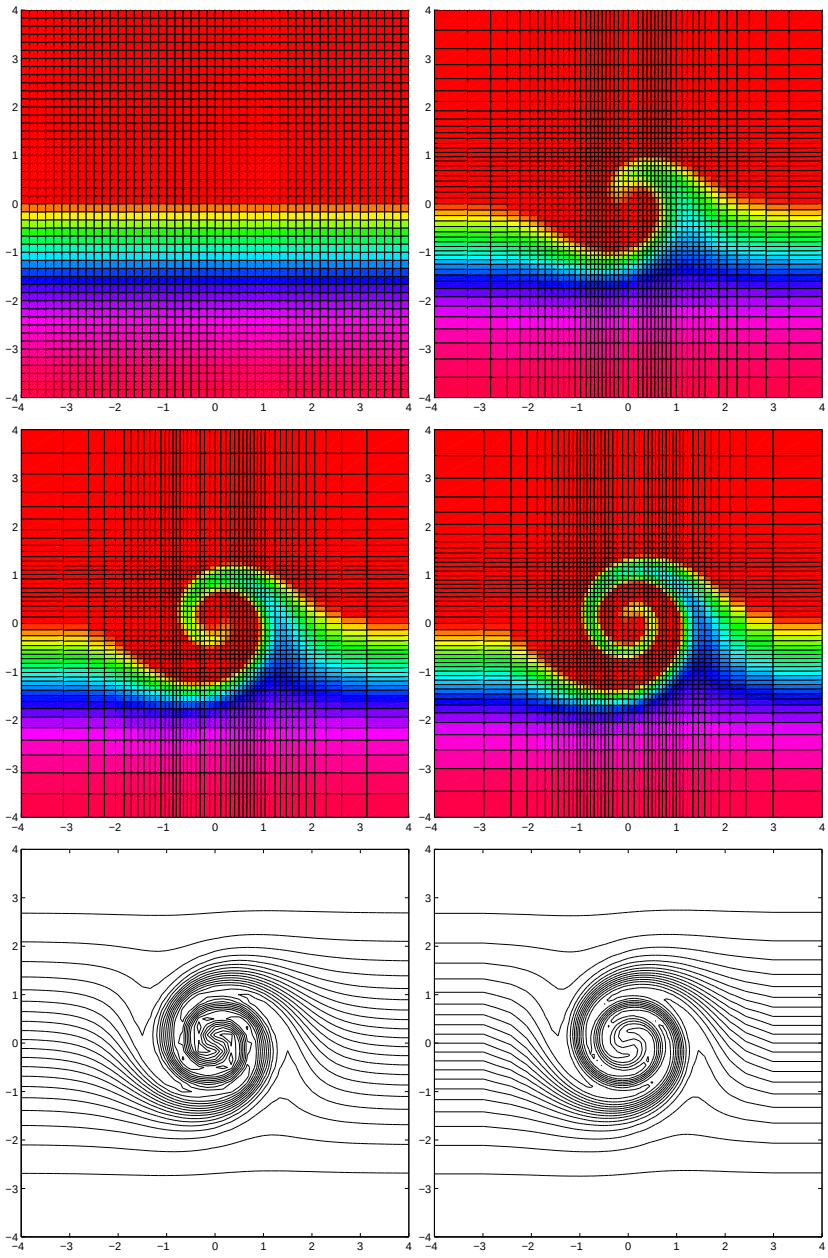

Figure 3: Tensor-grid solutions for the whirlpool model at $t = 0.0, 1.6, 2.8, 4.0$ and contourplots at $t = 4$: uniform grid (left) vs. adaptive grid (right).

References

[1] M.J. Baines, "Least Squares and Approximate Equidistribution in Multidimensions", *Num. Meth. for PDEs*, **15**, 605-615 (1999).

[2] G. Beckett, J.A. Mackenzie, A. Ramage and D.M. Sloan, "On the Numerical Solution of One-Dimensional PDEs Using Adaptive Methods Based on Equidistribution", *J. of Comp. Phys.*, **167**, 372-392 (2001).

[3] J. G. Blom, R. A. Trompert and J. G. Verwer, "Algorithm 758: VLUGR2, A Vectorizable Adaptive Grid Solver for PDEs in 2D", *ACM Trans. Math. Software*, **22**, 302-328 (1996).

[4] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, "Using Krylov Methods in the Solution of Large-Scale Differential-Algebraic Systems", *SIAM J. Sci. Comp.*, **15**, 1467-1488 (1994).

[5] C.J. Budd and M.D. Piggott, "Geometric integration and its applications", *Preprint Dept. of Math. Sc.*, Univ. of Bath (2000).

[6] J.G. Blom and J.G. Verwer, "On the Use of the Arclength and Curvature Monitor in a Moving-Grid Method which is Based on the Method of Lines", *Report NM-N8902*, CWI, Amsterdam (1989)

[7] P.C. Capon and P.K. Jimack, "On the Adaptive Finite Element Solution of Partial Differential Equations Using $h-r$-Refinement", *School of Computer Studies Research Report*, **96.13**, Univ. of Leeds (1996).

[8] A. Doelman, T.J. Kaper and P.A. Zegeling, "Pattern Formation in the One-Dimensional Gray-Scott Model", *Nonlinearity*, **10**, 523-563 (1997).

[9] J.A. Ferreira, "On Moving Finite Difference Methods for Partial Differential Equations on Two-Dimensional Domains", *Preprint 97-10*, Dep. de Math., Univ. de Coimbra (1997).

[10] W. Huang, "Variational Mesh Adaptation: Isotropy and Equidistribution", *Math. Res. Rep. 01-04-01*, Dep. of Maths., Univ of Kansas (2001).

[11] W. Huang and R.D. Russell, "Analysis of Moving Mesh Partial Differential Equations with Spatial Smoothing", *SIAM J. Num. Anal.*, **34**, 1106-1126 (1997).

[12] A.K. Kapila, "Asymptotic Treatment of Chemically Reacting Systems", *Pitman, Boston* (1983).

[13] P. Knupp and S. Steinberg, "Fundamentals of Grid Generation", *CRC-Press, Inc.* (1993).

[14] A. Kvaerno and B. Leimkuhler, "A Time-Reversible, Regularized, Switching Integrator for the N-Body Problem", *SIAM J. Sci. Comput.*, **22**, 1016-1035 (2000).

[15] F. Liu, S. Ji, and G. Liao, "An adaptive grid method and its application to steady Euler flow calculations", *SIAM J. Sci. Comput.*, **20**, 811-825 (1998).

[16] J.A. Pearson, "Complex Patterns in a Simple System", *Science*, **261**, 189-192 (1993).

[17] V. Pereyra and E.G. Sewell, "Mesh selection for Discrete Solution of Boundary Problems in Ordinary Differential Equations", *Num. Math.*, **23**, 261-268 (1975).

[18] L.R. Petzold, "A Description of DASSL: A Differential/Algebraic System Solver", in: *IMACS Transactions on Scientific Computation*, Eds.: R.S. Stepleman et al., 65-68 (1983).

[19] J.M. Stockie, J.A. Mackenzie and R.D. Russell, "A Moving Mesh Method for One-Dimensional Hyperbolic Conservation Laws", *SIAM J. Sc. Comp.*, **22**, 1791-1813 (2001).

[20] K.F. Sundman, "Mémoire sur le Problème des Trois Corps", *Acta Math.*, **36**, 179, (1913).

[21] P. Tamamadis and D.N. Assanis, "Evaluation of Various High-Order-Accuracy Schemes with and without Flux Limiters", *Int. J. for Num. Meth. in Fluids*, **16**, 931-948 (1993).

[22] H.A. van der Vorst, "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems", *SIAM J. Sci. Stat. Comput.*, **13**, 631-644 (1992).

[23] P.A. Zegeling and J.G. Blom, "A Note on the Grid Movement Induced by MFE", *Int. J. for Num. Meths. in Eng.*, **35**, 623-636 (1992).

[24] P.A. Zegeling and R. Keppens, "Adaptive MOL for Magneto-Hydrodynamic PDE Models", in: *Adaptive Method of Lines*, Eds.: A. Vande Wouwer, Ph. Saucez, W.E. Schiesser, Chapman & Hall/CRC Press, 117-137 (2001).

[25] P.A. Zegeling and H.P. Kok, "Adaptive Moving Mesh Computations for Reaction-Diffusion Systems" *submitted to J. of Comp. Appl. Maths.* (2002).

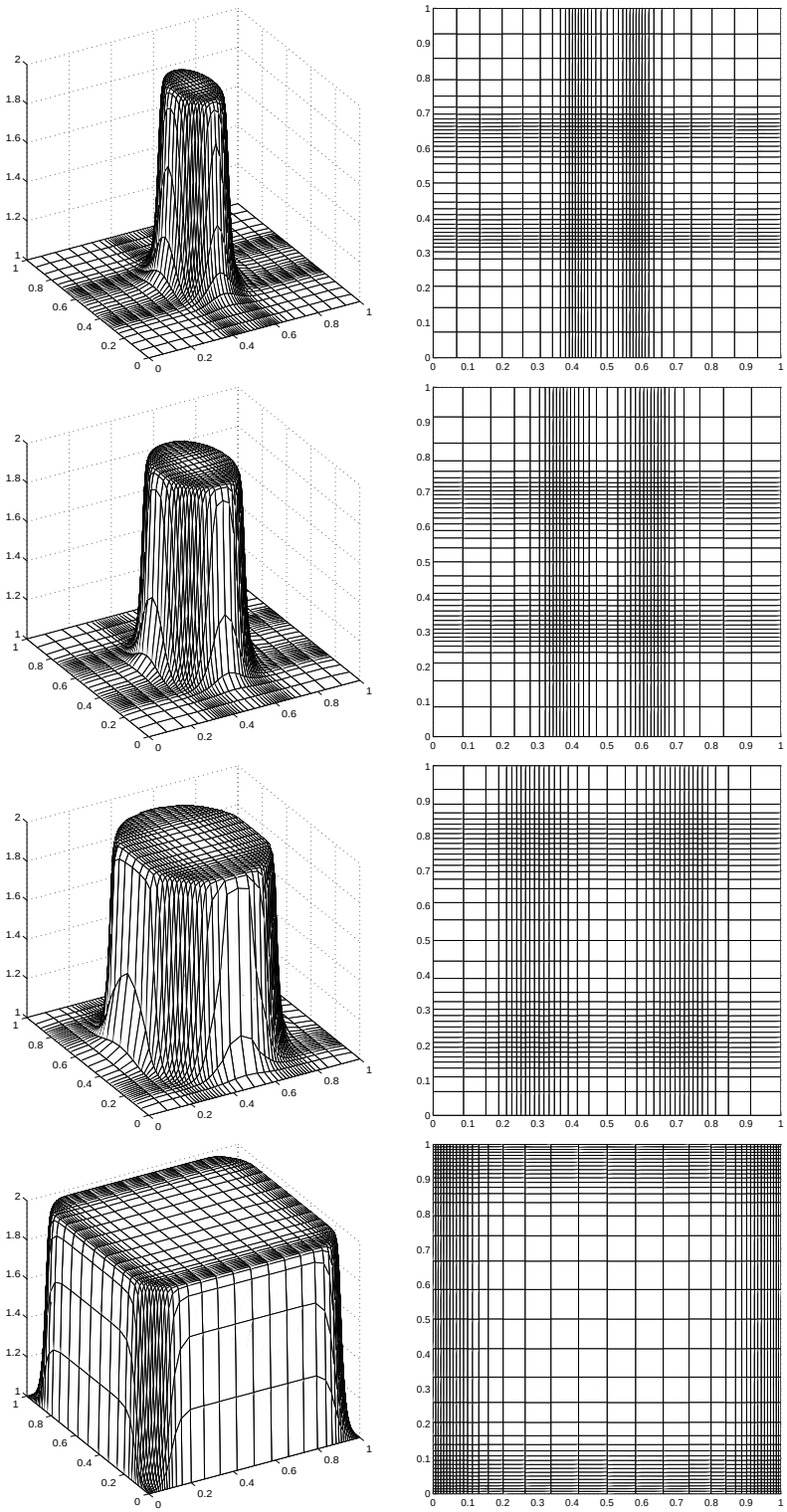

Figure 4: Tensor-grid solutions for the combustion model at $t = 0.01, 0.02, 0.04, 0.08$.

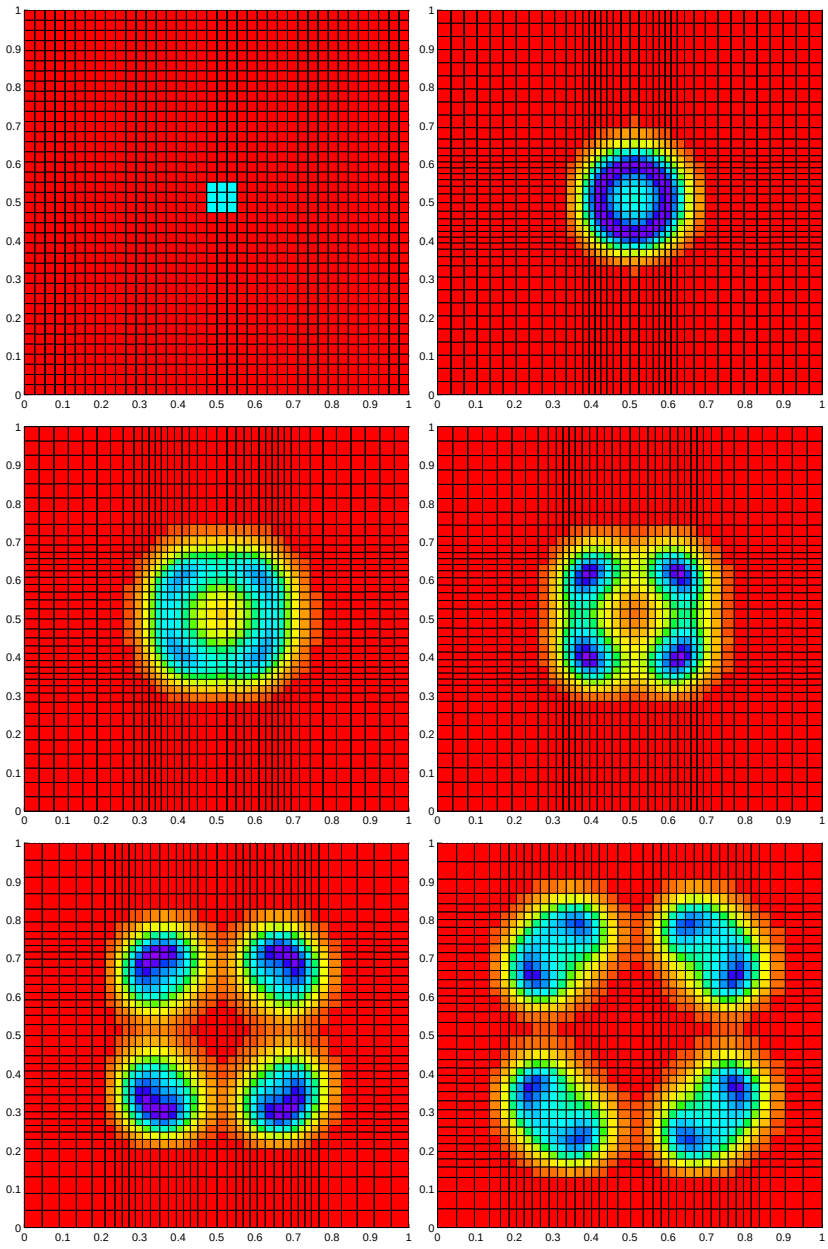

Figure 5: Tensor-grid solutions for the Gray-Scott model: from initial tiny block at $t = 0$ (upper-left), via four spots, to eight spots at $t = 500$.

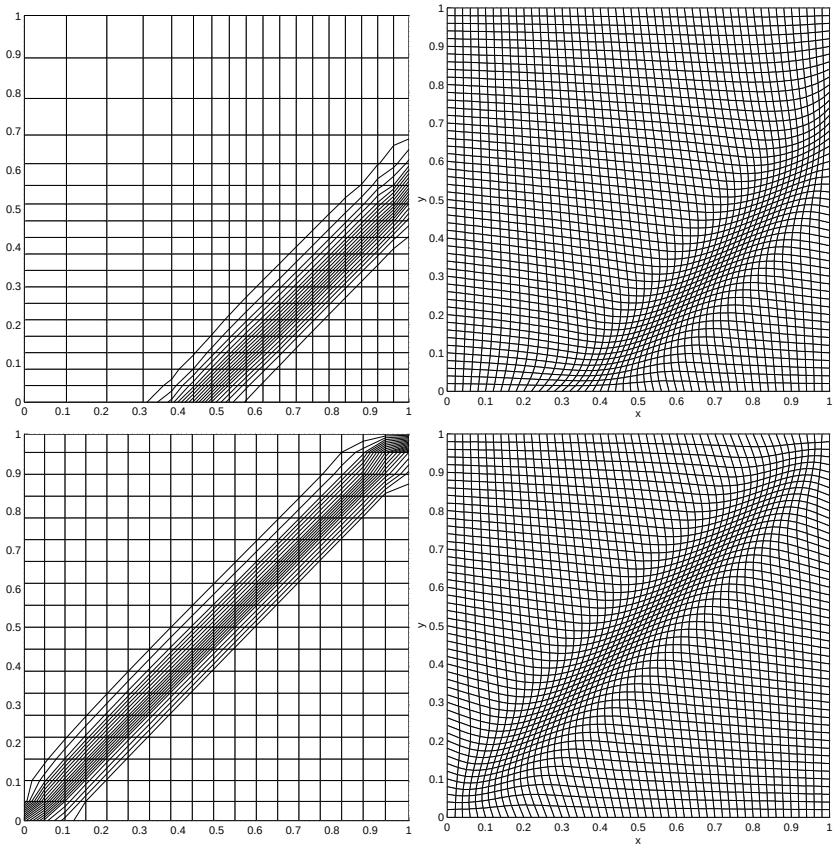

Figure 6: Tensor-grids with exact contour lines of the solution (left) compared with fully 2d adaptive grids (right) for the scalar 2d Burger's equation at $t = 0.5$ and $t = 2$.

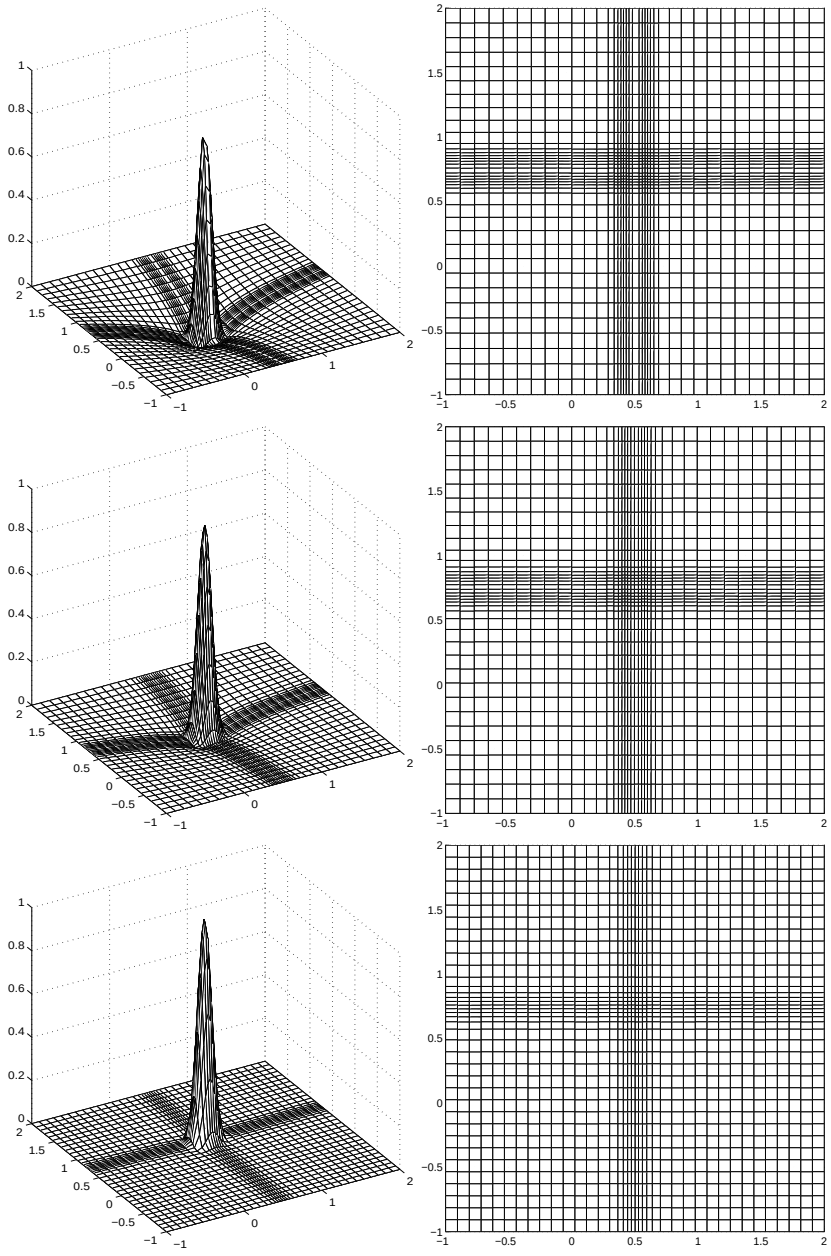

Figure 7: Tensor-grid solutions for the rotating cone model: solutions (left) and grids (right) after one rotation for, respectively, $(\alpha,\ \sigma) = (1, 0.2)$, $(\alpha,\ \sigma) = (1, 1)$, and $(\tilde{\alpha},\ \sigma) = (10, 0)$.