Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 168 (2004) 519–528

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Adaptive moving mesh computations for reaction–diffusion systems

P.A. Zegeling*, H.P. Kok[1]

Mathematical Institute, Utrecht University, P.O. Box 80.010, Utrecht 3508 TA, The Netherlands

Received 30 September 2002; received in revised form 6 June 2003

Abstract

In this paper we describe an adaptive moving mesh technique and its application to reaction–diffusion models from chemistry. The method is based on a coordinate transformation between physical and computational coordinates. The transformation can be viewed as a solution of adaptive mesh partial differential equations (PDEs) which is derived from the minimization of a mesh-energy integral. For an efficient implementation we have used an approach in which the numerical solution of the physical PDEs and the adaptive PDEs are decoupled. Further, to avoid solving large nonlinear systems, a second-order implicit–explicit time-integration method in combination with the iterative method Bi-CGSTAB is applied in the method-of-lines procedure. Numerical examples are given in one and two space dimensions.

MSC: 65C20; 65M20; 35K22; 35K57

Keywords: Method of lines; Adaptive mesh refinement; Finite differences; Moving mesh; Reaction–diffusion equations; Pattern formation

1. Introduction

Many unsteady models governed by reaction–diffusion partial differential equation systems have solutions with regions of high spatial variation such as emerging and splitting pulses, boundary layers or moving wave fronts. The partial differential equation (PDE) models often describe chemical reaction and diffusion processes with interesting spatial patterns. The reaction kinetics can be represented with a nonlinear source term and the diffusion by the Laplacian operator, respectively. From a computational point of view it is known that the use of uniform fixed spatial meshes is highly

* Corresponding author.
E-mail addresses: zegeling@math.uu.nl (P.A. Zegeling), h.p.kok@amc.uva.nl (H.P. Kok).

[1] Present address: Academic Medical Center, University of Amsterdam, The Netherlands.

0377-0427/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2003.06.013

inefficient for resolving the steep moving parts of the solution in such PDE systems. In those situations adaptive meshes play an increasingly growing role (see [1,2,6,10,15] and references therein). In this paper we describe an adaptive moving mesh technique that is based on a minimization of a so-called mesh-energy integral. The corresponding Euler–Lagrange equations then define a set of adaptive mesh PDEs. In one space dimension this reduces to the widely used equidistribution principle; in two space dimensions it is related to harmonic mapping theory supplied by a monitor matrix to detect the steep transitions in the solution. The adaptive mesh method can be seen as a discretization of the physical model in a transformed coordinate system coupled with the mesh equations [18]. The method of lines (MOL) is used to numerically approximate the solution of the PDE model. For the integration in time, 2-SBDF is applied, which is a second-order implicit–explicit method. The implicit part of this method deals with the (linear) diffusion, whereas the explicit part takes care of the (nonlinear) reaction terms in the model [13], thereby avoiding the use of a nonlinear system solver, like Newton's method. Additionally, the discretization of the PDE model and the moving mesh is decoupled. For solving the linear (nonsymmetric) system behind the adaptive moving mesh equations, the iterative method Bi-CGSTAB (see [16]) is used. A simple filtering technique is used to cope with less smooth transitions between higher and lower mesh concentrations in the domain. The chosen combination of techniques makes the full procedure of interest for accurately computing numerical solutions of reaction–diffusion systems with mildly stiff reaction-terms. Numerical results are shown for two different applications: the Gray–Scott model with complex pattern formation that possesses solutions with splitting pulses which describes irreversible chemical reactions with an inert product [12,4], and the Brusselator model [11] having periodic solutions with steep moving layers.

2. The adaptive moving mesh method

2.1. A coordinate transformation

Consider the following scalar reaction–diffusion model:

$$u_t = \varepsilon \Delta u + s(u, x, y, t), \tag{1}$$

where $(x, y) \in [x_l, x_r] \times [y_l, y_u] \subset \mathbb{R}^2$, $t \in [0, T]$, $0 < \varepsilon$ is the diffusion coefficient, and s a nonlinear sourceterm. It is common and useful in structured adaptive mesh methods to first apply a coordinate transformation to the physical PDE model (1). The adaptive mesh can then be seen as a uniform discretization of this mapping in the new variables. Applying the transformation

$$\xi = \xi(x, y, t), \quad \eta = \eta(x, y, t), \ \ \theta = t, \tag{2}$$

to Eq. (1) gives (a similar derivation can be made for a system of PDEs, and the one-dimensional case is obtained by freezing the second space direction)

$$u_\theta + \frac{1}{J}[u_\xi[y_\theta x_\eta - x_\theta y_\eta] + u_\eta[x_\theta y_\xi - y_\theta x_\xi]]$$

$$=s(u,x,y,\theta)+\frac{\varepsilon}{J}\left[\left[\frac{x_\eta^2+y_\eta^2}{J}\,u_\xi\right]_\xi-\left[\frac{y_\xi y_\eta+x_\xi x_\eta}{J}\,u_\eta\right]_\xi\right.$$

$$\left.-\left[\frac{y_\xi y_\eta+x_\xi x_\eta}{J}\,u_\xi\right]_\eta+\left[\frac{x_\xi^2+y_\xi^2}{J}\,u_\eta\right]_\eta\right], \tag{3}$$

where $J=x_\xi y_\eta-x_\eta y_\xi$ is the Jacobian of the inverse transformation.

2.2. The adaptive mesh PDEs

The transformation, in other words, the adaptive mesh, is prescribed by the so-called adaptive moving mesh PDEs.

Let $\vec{x}=(x_1,\ldots,x_d)\in\Omega_p$ and $\vec{\xi}=(\xi_1,\ldots,\xi_d)\in\Omega_c:=[0,1]^d$ be the physical and computational coordinates, respectively (with $d\geqslant 1$ the spatial dimension). The general transformation is then given by: $\vec{\xi}=\vec{\xi}(\vec{x})$, $\vec{x}\in\Omega_p$, with inverse $\vec{x}=\vec{x}(\vec{\xi})$, $\vec{\xi}\in\Omega_c$. In a variational setting, the 'mesh-energy' functional is defined by

$$E(\vec{\xi})=\frac{1}{2}\sum_{k=1}^{d}\int_{\Omega_p}\nabla\xi_k^{\mathrm{T}}M_k^{-1}\nabla\xi_k\,\mathrm{d}\vec{x},$$

where $\nabla:=(\partial_{x_1},\partial_{x_2},\ldots,\partial_{x_d})^{\mathrm{T}}$ and M_k are given monitor matrices. In this context, the functional may be considered as the energy of a system of (virtual) springs connecting the mesh points ('mass points') in the discretized domain. The mesh, i.e. the transformation, is determined by minimizing the energy functional via the Euler–Lagrange equations: $\nabla\cdot(M_k^{-1}\nabla\xi_k)=0,\ 1\leqslant k\leqslant d$ (see [8] or [9] for more details on this specific type of functionals).

A simple choice for the monitor function is $M_k=\omega I, 1\leqslant k\leqslant d$, where I is the identity matrix and ω a positive weight function. With this choice we obtain Winslow's variable diffusion method [17]

$$\nabla\cdot\left(\frac{1}{\omega}\nabla\xi_k\right)=0,\quad 1\leqslant k\leqslant d. \tag{4}$$

In one space dimension the Euler–Lagrange equations reduce to $(\omega^{-1}\xi_x)_x=0$, which gives the *equidistribution principle*: $\omega^{-1}\xi_x=c$(onstant) $\Leftrightarrow$ $\omega x_\xi=\tilde{c}$(onstant), or equivalently the boundary-value problem: $(\omega x_\xi)_\xi=0$, with boundary conditions $x(0)=x_l$, $x(1)=x_r$. For this case, an explicit formula for the inverse transformation $\xi(x)$ can be derived. Note first that

$$1=\xi(x_r)-\xi(x_l)=\int_{x_l}^{x_r}\xi_x\,\mathrm{d}x=c\int_{x_l}^{x_r}\omega\,\mathrm{d}\bar{x}\Rightarrow\xi_x=\frac{\omega(x)}{\int_{x_l}^{x_r}\omega\,\mathrm{d}\bar{x}},$$

from which follows that the 1D transformation is nonsingular. Integrating once more gives $\xi(x)=\int_{x_l}^{x}\omega(\bar{x})\,\mathrm{d}\bar{x}/\int_{x_l}^{x_r}\omega(\bar{x})\,\mathrm{d}\bar{x}$.

In two space dimensions (writing $x_1=x,\ x_2=y,\ \xi_1=\xi,\ \xi_2=\eta$) Eqs. (4) becomes

$$(\omega^{-1}\xi_x)_x+(\omega^{-1}\xi_y)_y=0,$$

$$(\omega^{-1}\eta_x)_x+(\omega^{-1}\eta_y)_y=0. \tag{5}$$

In practice, the physical domain may have very complex geometry and as a result directly solving the elliptic system (5) on structured meshes is unrealistic. Therefore, usually the corresponding mesh generation equations on the computational domain are solved by interchanging the dependent and independent variables in (5). However, the resulting system is much more complicated than the original equations (5), which requires more computational effort in obtaining numerical approximations. If the physical domain is convex, an alternative approach (see also [14]) is to consider the energy functional in the computational domain. Unlike the re-written classical equations (5) a very simple structure is maintained. In 2D the functional now reads:

$$\tilde{E}(x,y) = \tfrac{1}{2}\int_{\Omega_c} (\tilde{\nabla}^{\mathrm{T}} x M_1 \tilde{\nabla} x + \tilde{\nabla}^{\mathrm{T}} y M_2 \tilde{\nabla} y)\,\mathrm{d}\xi\,\mathrm{d}\eta,$$

where $\tilde{\nabla} = (\partial_\xi, \partial_\eta)^{\mathrm{T}}$. The corresponding Euler–Lagrange equations are then of the form

$$\begin{aligned}&\partial_\xi(M_1\partial_\xi x) + \partial_\eta(M_1\partial_\eta x) = 0,\\ &\partial_\xi(M_2\partial_\xi y) + \partial_\eta(M_2\partial_\eta y) = 0.\end{aligned} \tag{6}$$

We choose the monitor functions $M_1 = M_2 = \omega I$ with $\omega = \sqrt{1 + \beta\tilde{\nabla}u \cdot \tilde{\nabla}u}$ to detect regions with high first-order spatial derivatives. Using theoretical results from differential geometry [3], it can be proved that the transformation as a solution of the elliptic PDEs (6) is nonsingular, and therefore that the adaptive mesh cannot collapse (at least in a continuous formulation). The parameter β is an 'adaptivity'-parameter which controls the amount of adaptivity. For $\beta = 0$, we get $\omega = 1$ and $M_1{=}M_2{=}I$. Eqs. (6) then yield a system of two Laplace equations for the mesh with trivial boundary conditions on the unit square. The solution of this system, obviously, is the identity transformation: $x(\xi,\eta) = x$, $y(\xi,\eta) = y$, representing a uniform mesh in both directions. Higher values of $\beta > 0$ allow, of course, for more adaptivity. A default value for this parameter is: $\beta = 1$. However, this choice may depend on the size of the domain and the range (maximum and minimum values) in the PDE solution.

2.3. Numerical solution of the complete PDE system

One approach would be to couple the discretized systems for the adaptive mesh PDEs and the physical PDEs. However, there are a number of disadvantages to this approach. First, the size of the resulting system would be large and even for moderate grid densities may be prohibitive. Second, this approach does not easily admit different convergence criteria for the mesh and physical solution. As noted in literature, it is not necessary to compute the mesh with the same level of accuracy as the physical solution. Finally, a user may wish to control over the discretization of the physical problem and such flexibility is severely restricted by coupling the unknowns together into one large nonlinear system of equations. We have therefore decoupled the numerical solution procedure for the physical and adaptive mesh PDEs, and integrate in time in an iterative manner, solving for the mesh and the physical solution alternately. Furthermore, instead of solving (6) we integrate in time the parabolic PDE system

$$\begin{aligned}x_\tau &= (\omega x_\xi)_\xi + (\omega x_\eta)_\eta,\\ y_\tau &= (\omega y_\xi)_\xi + (\omega y_\eta)_\eta,\end{aligned} \tag{7}$$

where τ is an artificial time variable within the time integration process. In the theoretical limit, $\tau \to \infty$, the mesh reaches the steady-state situation (6). Numerically this means that after a number of time steps the mesh will adjust to the physical PDE solution. The decoupled procedure, which is closely related to the alternate solution procedure in Ref. [7], is outlined in Algorithm 1.

Algorithm 1. The decoupled numerical PDE procedure

```
Given the physical solution $u^{(n)}$, the mesh $\vec{x}^{(n)}$ and the time stepsize $\Delta t$ at time $t = t_n$.
1. Calculate the new monitor function $M^{(n)} = M^{(n)}(t_n, \vec{x}^{(n)}, u^{(n)})$.
2. Calculate the new mesh $\vec{x}^{(n+1)}$ by integrating the MMPDEs from $t = t_n$ to $t = t_n + \Delta t$,
   using $\vec{x}^{(n)}$ as initial mesh and keeping the monitor function $M$ constant in time during
   the integration.
3. Calculate the physical solution $u^{(n+1)}$ by integrating the physical PDEs from $t = t_n$ to
   $t = t_n + \Delta t$, using the mesh $\vec{x}^{(n+1)}$ and mesh speed $\dot{\vec{x}}(t) = (\vec{x}^{(n+1)} - \vec{x}^{(n)})/\Delta t$.
```

Within the decoupled procedure we freeze the coefficients in system (6) and replace the spatial derivatives by second-order central difference operators. The resulting ODE system is solved by the implicit Euler method, and for the linearized system of equations the iterative method Bi-CGSTAB (see [16]) is applied with implicit diagonal preconditioning.

For the reaction–diffusion equation (1) in which $s(u,x,y,t)$ is a nonlinear source term in general, it is appropriate to make use of an *implicit–explicit* time-integration method (see [13] for more details). The main advantage is that solving a nonlinear system, with for instance Newton's method, can be avoided, while still having reasonable stability properties, at least for mildly stiff equations. It must be noted that the reaction-terms in the intended applications are indeed mildly stiff, which justifies their explicit treatment. For example, within this class of integrators the first-order method 1-SBDF yields: $u^{(n+1)} - \varepsilon\Delta t\Delta u^{(n+1)} = u^{(n)} + \Delta t s^{(n)}(u,x,y,t)$, where $\Delta u^{(n+1)}$ is the semi-discretized approximation of the four second-order derivative terms in Eq. (3). Unfortunately, as analyzed in [13], this method may perform poorly to reproduce the dominant wavenumber accurately. This type of error is undesirable because it may lead to incorrect modal growth and hence a plausible-looking, yet qualitatively wrong solution. A better option for numerically integrating (1) in time is to apply the second-order method 2-SBDF. This method is recommended in [13] after an intensive numerical study of reaction–diffusion systems. It allows relatively larger time steps and strongly damps high frequency errors. The 2-SBDF formula reads

$$\tfrac{3}{2}u^{(n+1)} - \varepsilon\Delta t\Delta u^{(n+1)} = 2\Delta t s^{(n)}(u,x,y,t) - \Delta t s^{(n-1)}(u,x,y,t) + 2u^{(n)} - \tfrac{1}{2}u^{(n-1)}. \tag{8}$$

Since at each time step both $u^{(n)}$ and $u^{(n-1)}$ are needed in 2-SBDF, we have applied at the first time step the one-step explicit–implicit Euler method. The linear system $Au^{(n+1)} = b$ behind (8) is again solved with the iterative method Bi-CGSTAB with implicit diagonal preconditioning. In order to obtain 'smoother' transitions in the mesh, rather than merely using Eqs. (6), an additional filter (see also [14]) is applied on the weight functions. Instead of working with M_{ij}, the averaged monitor values

$$\begin{aligned}\tilde{M}_{ij} = {} & \tfrac{1}{4}M_{ij} + \tfrac{1}{8}(M_{i+1,j} + M_{i-1,j} + M_{i,j+1} + M_{i,j-1}) \\ & + \tfrac{1}{16}(M_{i-1,j-1} + M_{i-1,j+1} + M_{i+1,j-1} + M_{i+1,j+1})\end{aligned}$$

are being used in the mesh equations. This weighted sum corresponds with averaging the influence of the first spatial derivatives on the mesh points and serves as a tool to decrease the possibility of *numerical* mesh distortion.

3. Numerical results

3.1. The Gray–Scott model

Reaction–diffusion models of chemical species can produce a variety of patterns, reminiscent of those often seen in nature. The Gray–Scott equations model such a reaction. Numerical simulations of this model were performed in an attempt to find stationary lamellar patterns like those observed in earlier laboratory experiments on ferrocyanide–iodate–sulphite reactions (for more details on the background of the model and the chemical reactions we refer to [12]). The chemical reactions for this particular situation, taken from [12], are described by

$$U + 2V \rightarrow 3V,$$
$$V \rightarrow P,$$

in which U, V and P are chemical species. The system of reaction–diffusion equations for this situation is given by

$$\begin{aligned} u_t &= \varepsilon_1 \Delta u - uv^2 + f(1-u), \\ v_t &= \varepsilon_2 \Delta v + uv^2 - (f+k)v, \end{aligned} \tag{9}$$

where ε_1 and ε_2 are the diffusion rates in the process, k represents the rate of conversion of V to P, and f the rate of the process that feeds U and drains U, V and P.

In the 1D numerical experiments the following choices for the model parameters are made: $\varepsilon_1 = 10^{-4}$, $\varepsilon_2 = 10^{-6}$, $f = 0.024$, $k = 0.06$. The initial conditions are $u(x,0) = 1 - \frac{1}{2}\sin^{100}(\pi x)$, $v(x,0) = \frac{1}{4}\sin^{100}(\pi x)$, supplemented with Dirichlet boundary conditions on the domain $[0,1]$. We have used 300 mesh points on the spatial domain $[0,1]$ with a time step of $\Delta t = 0.01$. For the adaptivity parameter β in the weight function we took the value 0.01. The mesh and the solutions of the PDEs are determined with an accuracy (measured in terms of residual error) of 10^{-4} and 10^{-5}, respectively. With these conditions (see Ref. [4]) we expect a splitting of the initial pulse first into two and then into four pulses. Fig. 1 shows the trajectories of the adaptive mesh up to $t = 2000$ and the solutions at $t = 0$ and 2000 for u and v. The ability of the adaptive mesh to capture and follow the splitting process is clearly demonstrated in this figure. It is worthwhile to mention that application of a uniform mesh of similar size produces numerical solutions that lag far behind in the splitting process and finally even loose their symmetry.

In two space dimensions we start with two block functions

$$u(x,y,0) = \begin{cases} 0.5 & \text{if } 0.3 \leqslant x \leqslant 0.7 \text{ and } 0.3 \leqslant y \leqslant 0.7, \\ 1 & \text{elsewhere,} \end{cases}$$

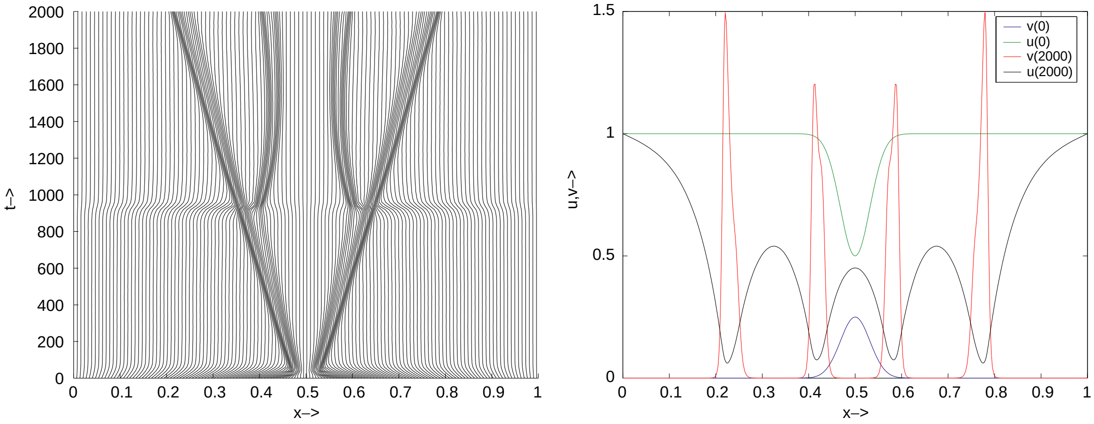

Fig. 1. Adaptive mesh solutions for the 1D Gray–Scott model.

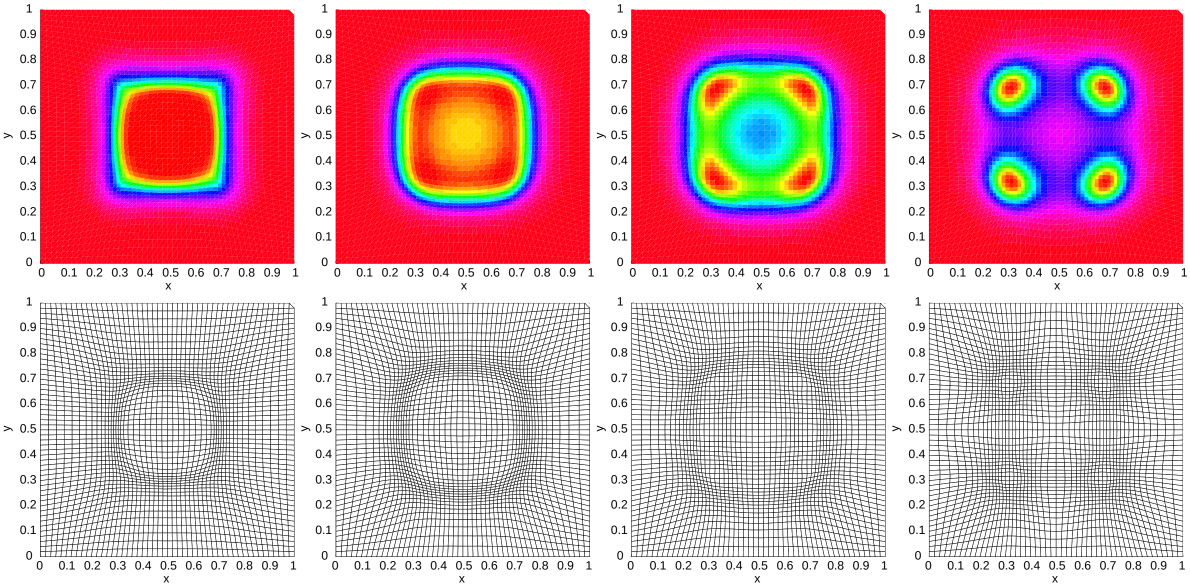

Fig. 2. Adaptive mesh solutions for the 2D Gray–Scott model at $t = 1, 50, 100,$ and 150.

$$v(x, y, 0) = \begin{cases} 0.25 & \text{if } 0.3 \leqslant x \leqslant 0.7 \text{ and } 0.3 \leqslant y \leqslant 0.7, \\ 0 & \text{elsewhere,} \end{cases}$$

on the domain $[0, 1] \times [0, 1]$. We choose the same time step, and tolerances as in 1D on a spatial mesh of 51×51 mesh points with Dirichlet boundary conditions and diffusion coefficients $\varepsilon_1 = 8 \cdot 10^{-5}$, $\varepsilon_2 = 4 \cdot 10^{-5}$. Fig. 2 depicts the numerical solutions for $\beta = 1$ at $t = 1, 50, 100,$ and 150. The 2D-splitting process of the initial block functions into four spots is illustrated in terms of the first component u and the adaptive mesh.

3.2. The Brusselator

The second model is the so-called Brusselator. This is a very well-studied model for a hypothetical tri-molecular reaction, which was introduced in Brussels in 1971 (for more details see [11]). The four single reactions are given by

$$\begin{aligned}
&A \to U, &\\
&B + U \to V + D, &\text{Bimolecular reaction.}\\
&2U + V \to 3U, &\text{Autocatalytic trimolecular reaction.}\\
&U \to E.&
\end{aligned}$$

Adding the rates of production and loss of the two intermediate species U and V in these reactions, and assuming all rate constants to be unity, leads to the following simplified time evolution equations for the concentration fields U and V:

$$\begin{aligned}
u_t &= \varepsilon_1 \Delta u + A + u^2 v - (B+1)u,\\
v_t &= \varepsilon_2 \Delta v + Bu - u^2 v,
\end{aligned} \tag{10}$$

where the diffusion coefficients are $\varepsilon_1 = \varepsilon_2 = 10^{-4}$ (in 1D) and $\varepsilon_1 = \varepsilon_2 = 2 \cdot 10^{-3}$ (in 2D), and the chemical parameters $A = 1$ and $B = 3.4$. This model has also been studied in Ref. [5]. Note that the smaller the ε_1- and ε_2-values are chosen, the steeper the solutions become, and that adaptive meshes become more and more important.

At the boundary of the domain Neumann conditions are imposed

$$\left.\frac{\partial u}{\partial n}\right|_{\partial\Omega} = \left.\frac{\partial v}{\partial n}\right|_{\partial\Omega} = 0 \quad \text{with } \Omega = [0,1]^d.$$

The initial conditions in 1D and 2D read, respectively, as

$$u(x,0) = \frac{1}{2}, \quad v(x,0) = 1 + 5x; \quad u(x,y,0) = \frac{1}{2} + y, \quad v(x,y,0) = 1 + 5x. \tag{11}$$

With these boundary and initial conditions a periodically (with period ≈ 7) moving wave will be induced. In Figs. 3 and 4 (200 meshpoints) and (51 × 51 meshpoints) we observe this behaviour both in one- and two-space dimensions. The numerical parameters for these runs were $\Delta t = 0.01$, $\beta = 0.01$ in 1D and 1 in 2D, respectively.

4. Summary and future work

In this paper we have applied an adaptive moving mesh method to 1D and 2D reaction–diffusion models from chemistry. Although the numerical results are very promising there is still much room to improve the performance of the method. Two of the most important issues in this respect are the implementation of a variable timestep control and the use of more sophisticated preconditioners for

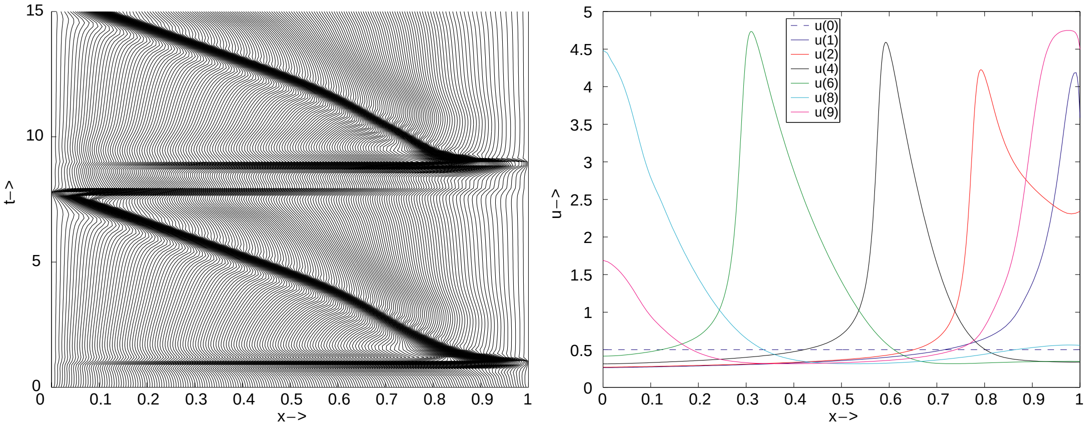

Fig. 3. Adaptive mesh solutions for the 1D Brusselator (mesh trajectories: left; solutions at different values between $t = 0$ and right; $t = 9$).

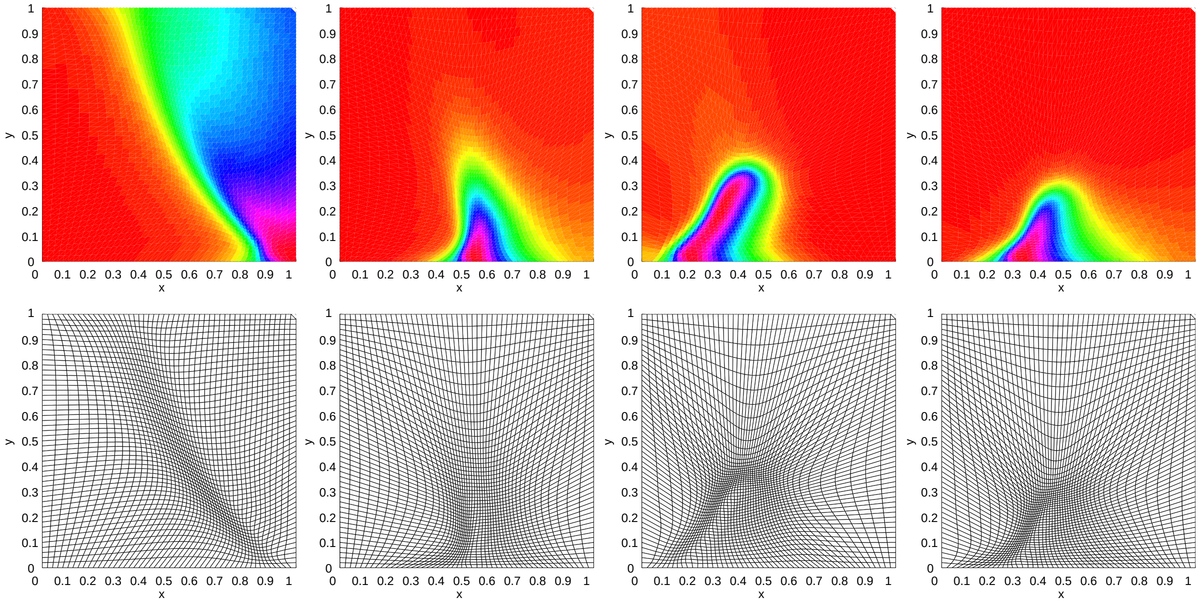

Fig. 4. Adaptive mesh solutions for the 2D Brusselator at $t = 1, 3, 5$, and 11.

the underlying linear systems. Other improvements could be, the development of a time-dependent and robust adaptivity parameter in the weight function, and a nonuniform evolutionary mesh distribution at the boundary of the domain (instead of fixing the grid there). Investigation of other weight functions and filtering operators for smoother mesh distributions will certainly improve the adaptive moving mesh as well, especially, if very steep layers and skew meshes are to be expected.

References

[1] G. Beckett, J.A. Mackenzie, A. Ramage, D.M. Sloan, Computational solution of two-dimensional unsteady PDEs using moving mesh methods, J. Comput. Phys. 182 (2002) 478–495.

[2] N. Carlson, K. Miller, Design and application of a gradient-weighted moving finite element code II: in two dimensions, SIAM J. Sci. Comput. 19 (1998) 766–798.

[3] Ph. Clement, R. Hagmeijer, G. Sweers, On the invertibility of mappings arising in 2D grid generation problems, Numer. Math. 73 (1996) 37–51.

[4] A. Doelman, T.J. Kaper, P.A. Zegeling, Pattern formation in the one-dimensional Gray–Scott model, Nonlinearity 10 (1997) 523–563.

[5] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations, I. Nonstiff Problems, Springer, Berlin, 1987.

[6] W. Huang, R.D. Russell, A high dimensional moving mesh strategy, Appl. Numer. Math. 26 (1997) 63–76.

[7] W. Huang, R.D. Russell, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, SIAM J. Sci. Comput. 3 (1999) 998–1015.

[8] P. Knupp, S. Steinberg, The Fundamentals of Grid Generation, CRC, Boca Raton, FL, 1993.

[9] V.D. Liseikin, Grid Generation Methods, Springer, Berlin, 1999.

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

[11] G. Nicolis, I. Prigogine, Exploring Complexity, Freeman, New York, 1989.

[12] J.A. Pearson, Complex patterns in a simple system, Science 261 (1993) 189–192.

[13] S.J. Ruuth, Implicit-explicit methods for reaction–diffusion problems in pattern formation, J. Math. Biol. 34 (1995) 148–176.

[14] H.Z. Tang, T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515.

[15] A. Vande Wouwer, Ph. Saucez, W.E. Schiesser (Eds.), Adaptive Method of Lines, CRC, Boca Raton, FL, 2001.

[16] 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. Statist. Comput. 3 (1992) 631–644.

[17] A.M. Winslow, Adaptive mesh zoning by the equipotential method, Technical Report UCID 19062, 1981.

[18] P.A. Zegeling, R. Keppens, Adaptive MOL for magneto-hydrodynamic PDE models, in: A.V. Wouwer, P. Saucez, W.E. Schiesser (Eds.), Adaptive Method of Lines, Chapman & Hall/CRC, London/Boca Raton, FL, 2001, pp. 117–137.