Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Computational Physics 214 (2006) 239–263
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Time-stepping schemes for moving grid finite elements applied to reaction–diffusion systems on fixed and growing domains

Anotida Madzvamuse *

Department of Mathematics and Statistics, Auburn University, 316A Parker Hall, Auburn, AL 36849, United States

Received 19 May 2005; received in revised form 20 September 2005; accepted 20 September 2005
Available online 15 November 2005

Abstract

In this paper, we illustrate the application of time-stepping schemes to reaction–diffusion systems on fixed and continuously growing domains by use of finite element and moving grid finite element methods. We present two schemes for our studies, namely a first-order backward Euler finite differentiation formula coupled with a special form of linearisation of the nonlinear reaction terms (1-SBEM) and a second-order semi-implicit backward finite differentiation formula (2-SBDF) with no linearisation of the reaction terms. Our results conclude that for the type of reaction–diffusion systems considered in this paper, the 1-SBEM is more stable than the 2-SBDF scheme and that the 1-SBEM scheme has a larger region of stability (at least by a factor of 10) than that of the 2-SBDF scheme. As a result, the 1-SBEM scheme becomes a natural choice when solving reaction–diffusion problems on continuously deforming domains.

Keywords: Implicit–explicit schemes; Time-stepping methods; Finite elements; Moving meshes; Moving grid finite elements; Reaction–diffusion systems; Schnakenberg model

1. Introduction

Many problems in biology and bio-medicine involve growth and shape changes. The growth and spread of tumours, the initiation of limb buds in the developing foetus and the patterns formed on the skin of growing creatures are just some of the many such examples. Such growth, especially when it is irregular, presents a challenge for those building mathematical and computational models for simulation. In the physical sciences, problems such as melting and freezing can lead to moving (growing or shrinking) domains, but in biology, problems which are naturally described by partial differential equations on growing domains are much more widespread. A number of models used in biology, ecology and biochemistry comprise reaction of "species" in the presence of diffusion: hence reaction–diffusion systems arise. Turing [34] proposed that, under certain conditions, a chemical reaction in the presence of diffusion could produce spatial patterns of the chemical concentration. Well known examples of reaction–diffusion systems include the Gierer and Meinhardt [10], the

* Tel.: +1 334 844 5884; fax: +1 334 844 6555.
E-mail address: madzva1@auburn.edu.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.09.012

Thomas [33], the Schnakenberg [32], and the Gray and Scott [11,12] models. In biology and bio-medicine reaction–diffusion systems are used frequently to model the emergence of pattern formation, wound healing, cancer and angiogenesis. Domain growth has been observed experimentally to be a crucial factor in developmental biology [17]. In order to obtain accurate numerical solutions to these nonlinear reaction–diffusion systems (most of which are stiff), novel discretisation schemes have to be applied both in space and time. Although we focus on the Turing models in this paper, we believe that our results may be extended with modifications to models which include nonlinear diffusion, those that include chemotaxis terms (i.e., movement in response to the gradients of other species in the system) [28], others that include advection and even to those models with multiple species.

A variety of time-stepping methods have been used to approximate the time-derivative of reaction–diffusion models on fixed domains. These include explicit, semi-implicit, implicit–explicit (IMEX) schemes and more recently the exponential time-differencing scheme [16]. Examples of such schemes are the Forward Euler [4,15], Gear's method [2], predictor–corrector methods and the alternating-direction implicit (ADI) scheme of Peaceman and Rachford [25,8]. Implicit–explicit (IMEX) schemes use an implicit scheme to approximate the diffusion term and an explicit scheme to approximate the reaction terms. The spatial discretisation can be carried out using the following methods: the finite difference, spectral methods, finite elements or finite volume. In this paper, we choose to pursue the finite element methodology for reasons outlined below and during the course of this paper. The extension of the finite difference or spectral methods to complicated, irregular and sometimes continuously growing domains is not at all trivial, hence the choice of the finite element method. The applicability of the finite element methods to complicated domains is well known and the growing realisation that continuously changing boundaries can be readily handled by moving grid implementations with few changes to the finite element methodology.

In this article, we limit ourselves to two time-stepping schemes, a first-order backward Euler finite difference scheme coupled with a special form of linearisation of the reaction terms (1-SBEM) and a second-order semi-implicit backward differentiation formula (2-SBDF). The 2-SBDF scheme is one of the IMEX schemes that has been recommended for most problems when solving reaction–diffusion systems with high frequencies [29]. Such schemes do not linearise the reaction terms but compute the fully nonlinear reaction terms at previous time steps by simply evaluating the reaction terms at the previous known approximate solutions. Other IMEX schemes that have been used but not really recommended are the first-order semi-implicit backward differentiation formula (1-SBDF), a second-order Crank–Nicolson, Adams–Bashforth (CNAB), a modified Crank–Nicolson, Adams–Bashforth (MCNAB), and a third-order semi-implicit backward differentiation formula (3-SBDF). For more details on IMEX schemes refer to the papers by Ruuth [29,1]. Most first-order schemes are not recommended because of the severe restriction on the time-step. Other schemes are not recommended because they give weak damping of high frequency spatial modes [29].

The manner in which our scheme (1-SBEM) is implemented differs substantially from the other first-order semi-implicit schemes. We use an implicit scheme to approximate the diffusion term. The linear terms in the reaction terms are approximated implicitly while the nonlinear terms are linearised by assuming that there are small changes in the approximate solution from one time step to the next. This way we avoid computing the Jacobian of the nonlinear reaction terms. This kind of treatment can be thought of as fully implicit for the diffusion and linear terms but semi-implicit for the nonlinear reaction terms. This is the novelty of our numerical scheme. We compare results from the 1-SBEM scheme to those obtained by using the 2-SBDF scheme and show that the 1-SBEM scheme produces comparable results in most cases except in the case of very highly oscillatory solutions. However, on growing domains the selection process is very much influenced by domain growth and our scheme performs as well as the 2-SBDF for relatively highly oscillatory solutions. Moreover, the 1-SBEM scheme allows for larger time-steps compared to the 2-SBDF scheme, hence the scheme is less expensive and faster, and relatively as accurate.

In Section 2, we present a generalised reaction–diffusion model considered in this paper. The methodology of this paper focuses on presenting the time discretisation first as shown in Section 3 for the outlined model equations. In this section we present the 1-SBEM and the 2-SBDF schemes applied to reaction–diffusion systems on continuously growing domains. Then, we apply the spatial discretisation, this is achieved using the moving grid finite element method described in Section 4. Here, we derive the weak form of the reaction–diffusion system and describe the Galerkin moving grid finite element approximation on a continuously

deforming grid. Numerical experiments are presented in Section 5 on fixed and continuously deforming one- and two-dimensional domains. Lastly, in Section 6, we present conclusions and discuss the implications of the time-stepping schemes when solving these types of problems in biological systems. We also provide numerical software and documentation for the methods described in this paper which can be downloaded freely from: http://www.auburn.edu/~madzva1.

2. Model equations

In this paper, we focus on the generalised reaction–diffusion system

$$\mathbf{u}_t + \nabla \cdot (\mathbf{a} : \mathbf{u}) = \mathbf{R}(\mathbf{u}) + \mathbf{D}\nabla^2\mathbf{u} \quad \text{in } \Omega(t) \tag{2.1}$$

with $\Omega(t)$ representing a time-dependent domain. We define

$$\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} f(u,v) \\ g(u,v) \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} D_u & 0 \\ 0 & D_v \end{pmatrix}, \quad \text{and} \quad \mathbf{x} = (x(t), y(t)),$$

where u, v are the two chemical concentrations under investigation, f and g are reaction kinetics. $\mathbf{D}$ is the diffusion matrix (D_u and D_v are constant diffusion parameters) (27). The notation ($\mathbf{a}$:$\mathbf{u}$) represents ($\mathbf{a}u$, $\mathbf{a}v$,)$^{\mathrm{T}}$ with $\mathbf{a}$ representing the field flow velocity. In Madzvamuse et al. [19] typical classical reaction kinetics are presented, including the Gierer–Meinhardt [10], the Thomas [33], the 32 models and a generalised bivariate cubic polynomial reaction model. In this paper, we use only the Schnakenberg reaction kinetics given by

$$f(u,v) = \gamma(a - u + u^2 v), \tag{2.2}$$

$$g(u,v) = \gamma(b - u^2 v), \tag{2.3}$$

where γ is a scaling parameter, a and b are fixed positive parameters. Boundary conditions can either be Neumann (homogeneous: which describe zero-flux of u (or v) out of the boundary) or Dirichlet or periodic boundary conditions. Initial conditions are prescribed as small random perturbations about the uniform homogeneous steady state of the corresponding reaction systems.

3. Time-stepping schemes

Several time-stepping schemes have been widely used to compute solutions to reaction–diffusion systems on fixed domains. These include the Forward Euler's method, Gear's method, a modified Euler predictor–corrector method, Gourlay's method [5], a semi-implicit Rosenbock integrator [5], and Runge–Kutta schemes [13]. Most of these are inadequate because of the stiffness of the diffusive term. Fully explicit methods require excessively small time-steps resulting in computations that are prohibitively expensive especially in multi-dimensions. Fully implicit methods can be used, however such methods normally require calculating the Jacobian of the nonlinear reaction terms at every time step. In some cases, the Jacobian could be singular which makes it impossible to apply fast iterative methods. More recently IMEX schemes have been widely used to solve reaction–diffusion problems on fixed one-dimensional domains [1,29]. The key essence of these schemes is that an implicit scheme is applied to approximate the diffusive term and an explicit scheme is used to approximate the reaction kinetics, hence the name IMEX. Particular examples of IMEX schemes are the Crank–Nicolson–Adams–Bashforth scheme (CNAB) or its modified version (MCNAB), the second-order semi-implicit backward differentiation scheme (2-SBDF) and other modified schemes.

In this section, we present two time-stepping schemes, the 1-SBEM and 2-SBDF schemes. For the first-order semi-implicit backward Euler differentiation formula, we treat the diffusive term implicitly, linear reaction terms are also treated implicitly, nonlinear reaction terms are then approximated by some special form of linearisation – in a semi-implicit fashion. This method avoids computing the Jacobian. Numerical computations will show that the 1-SBEM scheme is more stable than the 2-SBDF scheme and that its region of stability is greater than that of the second-order scheme by a factor of approximately 10. Both schemes give rise to symmetric and positive definite matrices which can be readily solved by using fast iterative schemes [30,31].

3.1. First-order backward Euler finite difference scheme: 1-SBEM

For illustrative purposes, let us consider the Schnakenberg reaction kinetics (2.2) and (2.3) on a continuously deforming domain

$$u_t + \nabla \cdot (\mathbf{a}u) = \gamma(a - u + u^2 v) + D_u \nabla^2 u, \tag{3.1}$$

$$v_t + \nabla \cdot (\mathbf{a}v) = \gamma(b - u^2 v) + D_v \nabla^2 v. \tag{3.2}$$

Applying the backward Euler finite difference scheme to these equations gives

$$\frac{u^{m+1} - u^m}{\Delta t} + \nabla \cdot (\mathbf{a}u^{m+1}) = \gamma(a - u^{m+1} + u^m u^{m+1} v^m) + D_u \nabla^2 u^{m+1}, \tag{3.3}$$

$$\frac{v^{m+1} - v^m}{\Delta t} + \nabla \cdot (\mathbf{a}v^{m+1}) = \gamma(b - u^m u^m v^{m+1}) + D_v \nabla^2 v^{m+1}. \tag{3.4}$$

Observe that the diffusion, growth and linear reaction terms are treated implicitly, exploiting as much as we can the fully implicit scheme. On the other hand, the nonlinear term $u^2 v$ in (3.1) has been linearised as $u^m u^{m+1} v^m$ where u^m is valued at the previous time step m and u^{m+1} is valued at the present time step $m+1$ (see Eq. (3.3)). The assumption here is that, for example on fixed domains, the two successive approximate solutions at successive time steps become closer and closer with time. On continuously deforming domains, domain growth takes place slowly and therefore this assumption still holds. This is the essence of our scheme, that linear and diffusive terms are treated using a fully implicit scheme while the nonlinear terms are treated semi-implicitly. This form of linearisation is advantageous in that we avoid computing the Jacobian, for example, had we considered a fully implicit scheme for the nonlinear reaction terms. Moreover, this kind of numerical treatment yields symmetric positive definite matrices which makes the scheme more attractive as it can be solved efficiently and fast by use of iterative techniques [30,31].

3.2. Second-order semi-implicit finite differentiation formula: 2-SBDF

We apply the second-order semi-implicit backward finite differentiation formula (2-SBDF) to Eqs. (3.1) and (3.2) to obtain

$$\frac{3u^{m+1} - 4u^m + u^{m-1}}{2\Delta t} = 2S_u(u^m, v^m) - S_u(u^{m-1}, v^{m-1}) + D_u \nabla^2 u^{m+1}, \tag{3.5}$$

$$\frac{3v^{m+1} - 4v^m + v^{m-1}}{2\Delta t} = 2S_v(u^m, v^m) - S_v(u^{m-1}, v^{m-1}) + D_v \nabla^2 v^{m+1}, \tag{3.6}$$

where

$$S_u(u^m, v^m) = \gamma(a - u^m + u^m u^m v^m) - \nabla \cdot (\mathbf{a}u^m)$$

and

$$S_v(u^m, v^m) = \gamma(b - u^m u^m v^m) - \nabla \cdot (\mathbf{a}v^m).$$

Note that $S_i(u^m,v^m)$ and $S_i(u^{m-1},v^{m-1})$, $i = u,v$, represent the nonlinear reaction terms plus terms from the domain growth. These terms are evaluated at time steps m and $m-1$, respectively. No form of linearisation is carried out on the $S_i(u^m,v^m)$ and $S_i(u^{m-1},v^{m-1})$ reaction terms. Observe that the diffusion terms are treated implicitly.

4. The moving grid finite element method

We formulate the moving grid finite element method (MGFEM) applied to the 1-SBEM and 2-SBDF schemes outlined in Sections 3.1 and 3.2, respectively. For example, let us apply the MGFEM to the 1-SBEM scheme. Following Madzvamuse et al. [19], let $w \in H^1(\Omega(t))$ be a test function. Multiplying Eqs. (3.3) and (3.4) by w leads to the following problem:

$$\left(\frac{u^{m+1}-u^m}{\Delta t},w\right)+(\nabla\cdot(\mathbf{a}u^{m+1}),w)=(\gamma a,w)-(\gamma u^{m+1},w)+(\gamma u^m u^{m+1}v^m,w)+D_u(\nabla^2u^{m+1},w), \tag{4.1}$$

$$\left(\frac{v^{m+1}-v^m}{\Delta t},w\right)+(\nabla\cdot(\mathbf{a}v^{m+1}),w)=(\gamma b,w)-(\gamma u^m u^m v^{m+1},w)+D_v(\nabla^2v^{m+1},w), \tag{4.2}$$

for all $w\in H^1(\Omega(t))$, where

$$(u,w)=\int\int_{\Omega(t)}uw\,\mathrm{d}\mathbf{x} \tag{4.3}$$

is the L_2 – inner product. Assuming that homogeneous Neumann boundary conditions are used, by Green's Theorem solving Eqs. (4.1) and (4.2) reduce to finding $u,v\in H^1(\Omega(t))$ such that

$$\left(\frac{u^{m+1}-u^m}{\Delta t},w\right)+(\nabla\cdot(\mathbf{a}u^{m+1}),w)=(\gamma a,w)-(\gamma u^{m+1},w)+(\gamma u^m u^{m+1}v^m,w)-D_u\big(\nabla u^{m+1},\nabla w\big),$$

$$\left(\frac{v^{m+1}-v^m}{\Delta t},w\right)+(\nabla\cdot(\mathbf{a}v^{m+1}),w)=(\gamma b,w)-\big(\gamma u^m u^m v^{m+1},w\big)-D_v\big(\nabla v^{m+1},\nabla w\big)$$

for all $w\in H^1(\Omega(t))$. Here, $\mathbf{a}=(\dot{x},\dot{y})^{\mathrm{T}}$ represents the grid velocity. Therefore we seek to find a solution u and $v\in H^1(\Omega(t))$ such that

$$\begin{aligned}&\left(\frac{u^{m+1}-u^m}{\Delta t},w\right)+(\dot{x}u_x^{m+1}+\dot{y}u_y^{m+1},w)+(\nabla\cdot\mathbf{a}u_h^{m+1},w)\\&\quad=(\gamma a,w)-(\gamma u^{m+1},w)+(\gamma u^m u^{m+1}v^m,w)-D_u(\nabla u^{m+1},\nabla w),\end{aligned} \tag{4.4}$$

$$\begin{aligned}&\left(\frac{v^{m+1}-v^m}{\Delta t},w\right)+(\dot{x}v_x^{m+1}+\dot{y}v_y^{m+1},w)+(\nabla\cdot\mathbf{a}v_h^{m+1},w)\\&\quad=(\gamma b,w)-(\gamma u^m u^m v^{m+1},w)-D_v(\nabla v^{m+1},\nabla w),\end{aligned} \tag{4.5}$$

for all $w\in H^1(\Omega(t))$. Observe that the terms $\nabla\cdot(\mathbf{a}u^{m+1})$ and $\nabla\cdot(\mathbf{a}v^{m+1})$ have been expanded appropriately as shown above. Let u_h^m (or similarly v_h^m) be a moving grid finite element approximation to u (or v) defined by

$$u_h^m=\sum_{j=1}^{n}U_j^m(t)\phi_j(\mathbf{x},\xi(t)), \tag{4.6}$$

where $\xi(t)$ represents the finite element moving grid and n is the number of unknowns.

The time-derivative of u (and similarly v) in Eq. (3.1) (and similarly (3.2)) can be expressed in two dimensions as [14,6,7]

$$\frac{\partial u_h^m}{\partial t}=\sum_{j=1}^{n}\left(\frac{\mathrm{d}U_j^m}{\mathrm{d}t}-\dot{x}_j u_{h_x}^m-\dot{y}_j u_{h_y}^m\right)\phi_j(\mathbf{x},\xi(t)) \tag{4.7}$$

and therefore the time-discretisation needs to be re-adjusted to take into account the extra terms from the MGFEM as shown in Eqs. (4.8) and (4.9). Therefore, the moving grid finite element approximation seeks to find u_h^m and $v_h^m\in V^h\subset H^1$ such that

$$\begin{aligned}&\left(\frac{u_h^{m+1}-u_h^m}{\Delta t},w\right)+(\dot{x}u_{h_x}^{m+1}+\dot{y}u_{h_y}^{m+1},w)+(\nabla\cdot\mathbf{a}u_h^{m+1},w)\\&\quad=(\gamma a,w)-(\gamma u_h^{m+1},w)+(\gamma u_h^m u_h^{m+1}v_h^m,w)-D_u(\nabla u_h^{m+1},\nabla w),\\&\left(\frac{v_h^{m+1}-v_h^m}{\Delta t},w\right)+(\dot{x}v_{h_x}^{m+1}+\dot{y}v_{h_y}^{m+1},w)+(\nabla\cdot\mathbf{a}v_h^{m+1},w)=(\gamma b,w)-(\gamma u_h^m u_h^m v_h^{m+1},w)-D_v(\nabla v_h^{m+1},\nabla w),\end{aligned}$$

for all $w\in V^h\subset H^1(\Omega(t))$. Without loss of generality, taking the test function to be $w=\phi_i$, $i=1,\ldots,n$, the above equations can be written as follows where the extra terms from the time derivative are taken into account:

$$\sum_{j=1}^{n}\left(\frac{U_j^{m+1}(\phi_j,\phi_i)-U_j^{m}(\phi_j,\phi_i)}{\Delta t}\right)-\sum_{j=1}^{n}(\dot{x}_j u_{h_x}^{m+1}\phi_j+\dot{y}_j u_{h_y}^{m+1}\phi_j,\phi_i)+\sum_{j=1}^{n}[(\dot{x}_j u_{h_x}^{m+1}+\dot{y}_j u_{h_y}^{m+1},\phi_i)$$
$$+\nabla\cdot\mathbf{a}(\phi_j,\phi_i)U_j^{m+1}]=\sum_{j=1}^{n}[(\gamma a,\phi_i)-\gamma U_j^{m+1}(\phi_j,\phi_i)]+\gamma\left(\sum_{j=1}^{n}U_j^{m}\phi_j\sum_{k=1}^{n}U_k^{m+1}\phi_k\sum_{l=1}^{n}V_l^{m}\phi_l,\phi_i\right)$$
$$-D_u\sum_{j=1}^{n}U_j^{m+1}(\nabla\phi_j,\nabla\phi_i), \tag{4.8}$$

and

$$\sum_{j=1}^{n}\left(\frac{V_j^{m+1}(\phi_j,\phi_i)-V_j^{m}(\phi_j,\phi_i)}{\Delta t}\right)-\sum_{j=1}^{n}(\dot{x}_j v_{h_x}^{m+1}\phi_j+\dot{y}_j v_{h_y}^{m+1}\phi_j,\phi_i)+\sum_{j=1}^{n}[(\dot{x}_j v_{h_x}^{m+1}+\dot{y}_j v_{h_y}^{m+1},\phi_i)$$
$$+\nabla\cdot\mathbf{a}(\phi_j,\phi_i)V_j^{m+1}]=\sum_{j=1}^{n}(\gamma b,\phi_i)-\gamma\left(\sum_{j=1}^{n}U_j^{m}\phi_j\sum_{k=1}^{n}U_k^{m}\phi_k\sum_{l=1}^{n}V_l^{m+1}\phi_l,\phi_i\right)$$
$$-D_v\sum_{j=1}^{n}V_j^{m+1}(\nabla\phi_j,\nabla\phi_i) \tag{4.9}$$

for all $j,k,l=1,2,\ldots,n$. Integrating over the whole domain gives rise to the following system of linear algebraic equations

$$M\frac{\mathbf{U}^{m+1}-\mathbf{U}^{m}}{\Delta t}-(P+Q)\mathbf{U}^{m+1}+(\bar{U}_x+\bar{U}_y)\mathbf{U}^{m+1}+\nabla\cdot\mathbf{a}M\mathbf{U}^{m+1}$$
$$=\gamma[a\mathbf{F}-M\mathbf{U}^{m+1}+C(\mathbf{U}^{m},\mathbf{V}^{m})\mathbf{U}^{m+1}]-D_uK\mathbf{U}^{m+1}, \tag{4.10}$$
$$M\frac{\mathbf{V}^{m+1}-\mathbf{V}^{m}}{\Delta t}-(P+Q)\mathbf{V}^{m+1}+(\bar{V}_x+\bar{V}_y)\mathbf{V}^{m+1}+\nabla\cdot\mathbf{a}M\mathbf{V}^{m+1}$$
$$=\gamma[b\mathbf{F}-C(\mathbf{U}^{m},\mathbf{U}^{m})\mathbf{V}^{m+1}]-D_vK\mathbf{V}^{m+1}, \tag{4.11}$$

where M is the mass matrix, K is the stiffness matrix, and P, Q, $\bar{U}_x$, $\bar{U}_y$ $\bar{V}_x$, $\bar{V}_y$ are resultant matrices from domain growth, and $C(\mathbf{U},\mathbf{V})$ is the linearised matrix corresponding to the term u^2v. More details on the derivations of these matrices can be found in Madzvamuse [21].

In all our simulations $\nabla\cdot\mathbf{a}$ is calculated from plausible growth functions or those derived from biological experiments and therefore is a known quantity (for example, see Sections 5.2 and 5.4 or [20] for specific details). Finally, we can write the system of linear algebraic equations in compact form as follows:

$$A_u(\mathbf{U}^{m},\mathbf{V}^{m})\mathbf{U}^{m+1}=b_u(\mathbf{U}^{m}), \tag{4.12}$$
$$A_v(\mathbf{U}^{m},\mathbf{V}^{m})\mathbf{V}^{m+1}=b_v(\mathbf{V}^{m}), \tag{4.13}$$

where the matrices

$$A_u(\mathbf{U}^{m},\mathbf{V}^{m})=M+D_u\Delta tK-\Delta t(P+Q-\bar{U}_x-\bar{U}_y-\nabla\cdot\mathbf{a}M)+\gamma\Delta tM-\gamma\Delta tC(\mathbf{U}^{m},\mathbf{V}^{m}),$$
$$A_v(\mathbf{U}^{m},\mathbf{V}^{m})=M+D_v\Delta tK-\Delta t(P+Q-\bar{V}_x-\bar{V}_y-\nabla\cdot\mathbf{a}M)-\gamma\Delta tC(\mathbf{U}^{m},\mathbf{U}^{m}),$$

are functions of $\mathbf{U}^m$ and $\mathbf{V}^m$ and the right-hand side vectors are given by

$$b_u(\mathbf{U}^{m})=M\mathbf{U}^{m}+\gamma a\Delta t\mathbf{F},$$
$$b_v(\mathbf{V}^{m})=M\mathbf{V}^{m}+\gamma b\Delta t\mathbf{F}.$$

Similarly for the 2-SBDF scheme the following system of linear algebraic equations can be derived

$$A_u\mathbf{U}^{m+1}=4M\mathbf{U}^{m}-M\mathbf{U}^{m-1}+4\Delta tb_u(\mathbf{U}^{m},\mathbf{V}^{m})-2\Delta tb_u(\mathbf{U}^{m-1},\mathbf{V}^{m-1}), \tag{4.14}$$
$$A_v\mathbf{V}^{m+1}=4M\mathbf{V}^{m}-M\mathbf{V}^{m-1}+4\Delta tb_v(\mathbf{U}^{m},\mathbf{V}^{m})-2\Delta tb_v(\mathbf{U}^{m-1},\mathbf{V}^{m-1}), \tag{4.15}$$

where the matrices

$$A_u = 3M + 2D_u\Delta tK,$$
$$A_u = 3M + 2D_v\Delta tK,$$

are independent of the previous solution values $\mathbf{U}^m$ and $\mathbf{V}^m$. The right-hand side vectors are indeed functions of the solutions and are given by

$$b_u(\mathbf{U}^m, \mathbf{V}^m) = (P + Q - \bar{U}_x - \bar{U}_y - \nabla \cdot \mathbf{a}M)\mathbf{U}^m + \gamma[a\mathbf{F} - M\mathbf{U}^m + \bar{\mathbf{C}}(\mathbf{U}^m, \mathbf{V}^m)],$$
$$b_v(\mathbf{U}^m, \mathbf{V}^m) = (P + Q - \bar{V}_x - \bar{V}_y - \nabla \cdot \mathbf{a}M)\mathbf{V}^m + \gamma[b\mathbf{F} - \bar{\mathbf{C}}(\mathbf{U}^m, \mathbf{V}^m)].$$

Note that $\bar{\mathbf{C}}(\mathbf{U}^m, \mathbf{V}^m)$ represents the non-linearised forcing vector corresponding to the nonlinear term $u^2 v$. In both the first and second-order schemes the matrices A_u and A_v are symmetric and positive definite. In one dimension, these are tri-diagonal matrices. The Thomas algorithm is used to solve the system of algebraic equations [25]. In two dimensions we use an iterative scheme to solve the system of algebraic equations, namely a preconditioned generalised minimum residual method [30,31].

5. Numerical simulations

5.1. 1-D Numerical simulations on fixed domains

Most results in the literature have been illustrated on fixed domains. For comparison's sake we will show results on fixed domains and compare some of these to those already published. All numerical computations correspond to the Schnakenberg reaction model given in Eqs. (3.1) and (3.2). In all our simulations the diffusion coefficients have values $D_u = 1.0$ and $D_v = 10.0$. The rest of the parameters, a, b and γ vary as shown in the examples. The criterion for convergence to the spatially inhomogeneous steady state and hence a stopping criterion (for the $\mathbf{U}$ component solution for example) is given by

$$\sqrt{\frac{\sum |\mathbf{U}^{m+1} - \mathbf{U}^m|^2}{\sum |\mathbf{U}^{m+1}|^2}} \leqslant \epsilon. \tag{5.1}$$

We will call this the relative error. Some of the examples illustrated in this paper are taken from the paper by Ruuth [29] on implicit–explicit methods for reaction–diffusion problems in pattern formation. In this paper, numerical solutions are computed by use of finite difference schemes and in some cases by use of spectral methods on fixed domains. We compare some of their results to those computed by our numerical scheme. In our case, we do not compute numerical results using explicit methods as these have been shown [1,29] to be unable to compute realistic solutions for large values of the scaling parameter γ. Fully explicit methods are not suitable because of severe time-step restriction, these types of schemes have very small regions of stability for the time-step size and in some cases fail to accurately represent the fastest growing mode for large time-steps. Hence, computationally they are expensive and undesirable. Neither do we compute using first-order fully semi-implicit backward differentiation formulae, as these are not recommended because of the possibilities of the Jacobian being singular. In this paper results will be illustrated for the two time-stepping schemes: the first-order semi-implicit backward Euler finite differentiation scheme with linearisation (1-SBEM) and a second-order backward differentiation formula (2-SBDF) applied to finite element and moving grid finite element spatial discretisation methods for reaction–diffusion systems. The second-order backward differentiation formula has been recommended since the scheme strongly damps high frequency errors in the case of highly oscillatory solutions. In all our one-dimensional results we take 1500 elements over the interval $[l_1, l_2]$ with $l_1 < l_2$. Homogeneous Neumann boundary conditions are applied in all our simulations unless otherwise stated.

Example 1. Let us take $a = 0.126779$, $b = 0.792366$ and $\gamma = 10^4$ and compute solutions to Eqs. (3.1) and (3.2) on the interval $[0, 1]$. For illustrative purposes we compute solutions up to final time $t_F = 2.5$. The time-step Δt is varied until the method chosen fails to converge to a meaningful result. Initial conditions are taken as small random perturbations around the homogeneous steady state $(0.919145, 0.937903)$ and homogeneous Neumann boundary conditions are applied at both ends of the interval. Table 1 shows the region of stability

Table 1
The 1-SBEM is stable for Δt less than or equal to approximately 6.2×10^{-4} as illustrated by the relative errors of the numerical solution

Time-step Δt	No. of time steps	Relative errors
5×10^{-6}	500,000	1.575×10^{-16}
5×10^{-5}	50,000	2.1281×10^{-15}
5×10^{-4}	5000	8.185×10^{-14}
6×10^{-4}	4166	4.073×10^{-13}
6.1×10^{-4}	4098	4.5952×10^{-13}
6.2×10^{-4}	4032	1.2464×10^{-12}
6.3×10^{-4}	3968	0.231741

The second column shows the number of time steps taken to reach time $t_F = 2.5$.

of the time-step size Δt for the 1-SBEM scheme. Also included are the number of time steps taken to reach the final time $t_F = 2.5$, and the relative errors obtained from solving the equations using this scheme. The 1-SBEM scheme is stable and converges to a meaningful result for time-step sizes given by approximately $\Delta t < 6.2 \times 10^{-4}$.

Similarly, we solve the model Eqs. (3.1) and (3.2) using a 2-SBDF scheme with exactly the same model and numerical parameter values used for the 1-SBEM scheme on the interval $[0, 1]$. A variety of Δt's are used and the results of these computations are shown in Table 2. The region of stability for this scheme is given by approximately $\Delta t < 5.71 \times 10^{-5}$. This region of stability is smaller by a factor of ten to that of the 1-SBEM scheme. Crucially this factor amplifies the number of time steps by tenfold to those required by the 1-SBEM scheme. The relative errors for the 2-SBDF are much smaller than those of the 1-SBEM scheme, indicating that the second-order scheme produces more accurate solutions than the first-order scheme. Fig. 1 shows the regions of stability for the two methods whereby the values in Tables 1 and 2 have been used. The values of the time-steps when the plots turn vertically upward correspond to where the numerical scheme fails to converge to a meaningful result. Clearly the 1-SBEM (solid line) scheme allows for larger time-steps as compared to those required for the 2-SBDF (dashed line) scheme. This is true for all the computations that we have carried out in this paper. It has been observed that the first-order explicit methods require finer time-steps than those required by the 2-SBDF or other higher order semi-implicit methods [29] but this is not the case. In our next example, we show that the 1-SBEM scheme can compute solutions as accurate as those computed using the 2-SBDF scheme for relatively large values of the scaling parameter γ. Also the 1-SBEM scheme allows for larger time-steps Δt as compared to the 2-SBDF scheme (see Fig. 2).

Example 2. In this example, we fix all other parameter values except for the parameter value γ. Taking $a = 0.1$ and $b = 0.9$ we compute solutions on the interval $[-1, 1]$ for (a) $\gamma = 114$, (b) $\gamma = 1000$, (c) $\gamma = 5000$, (d) $\gamma = 10000$ and (e) $\gamma = 50000$ and compare the accuracy of the two methods. Let us take $\Delta t = 2.7 \times 10^{-3}$ for the 1-SBEM scheme and $\Delta t = 5 \times 10^{-4}$ for the 2-SBDF scheme and compute solutions to final time $t_F = 20$. Note that for the 1-SBEM scheme, we use a larger time-step size than that used for the 2-SBDF scheme. The results of these computations are shown in Fig. 3(a)–(e). The numerical results for the 1-SBEM (red) and the 2-SBDF (blue) schemes overlap, they are identical for increasing values of γ. The time-step size for the 1-SBEM scheme is 10 times bigger that of the 2-SBDF scheme, requiring ten times less the number of

Table 2
The region of stability of the time-steps for which the 2-SBDF scheme is stable

Time-step Δt	No. of time steps	Relative errors
5×10^{-6}	500,000	1.5×10^{-18}
5×10^{-5}	50,000	1.506×10^{-17}
5.5×10^{-5}	45,454	3.3415×10^{-17}
5.6×10^{-5}	44,642	3.7562×10^{-17}
5.7×10^{-5}	43,859	2.8×10^{-17}
5.71×10^{-5}	43,782	5.72×10^{-17}
5.8×10^{-5}	43,103	0.183827

Numerical solutions are computed to final time $t_F = 2.5$.

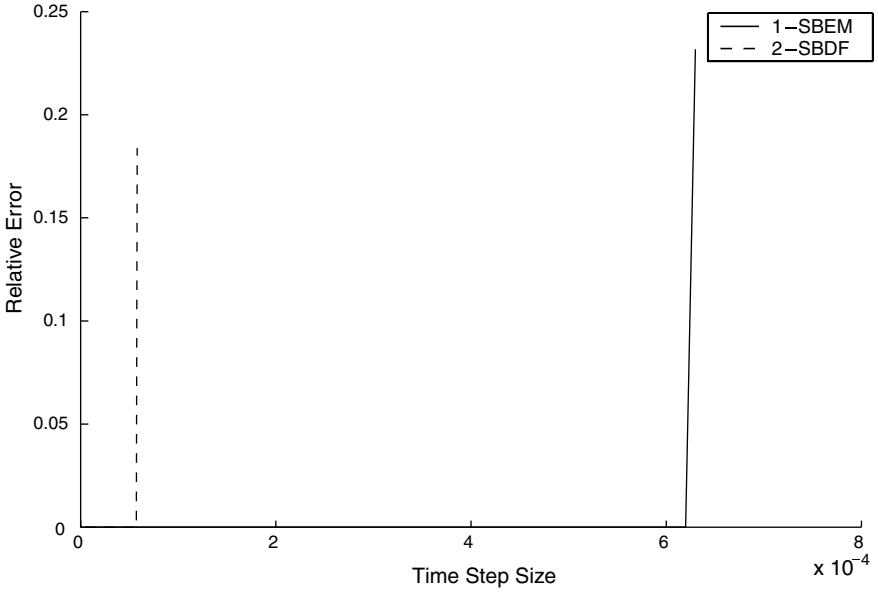


Fig. 1. Illustration of the time-step sizes required for the 1-SBEM and the 2-SBDF schemes and their respective relative errors.

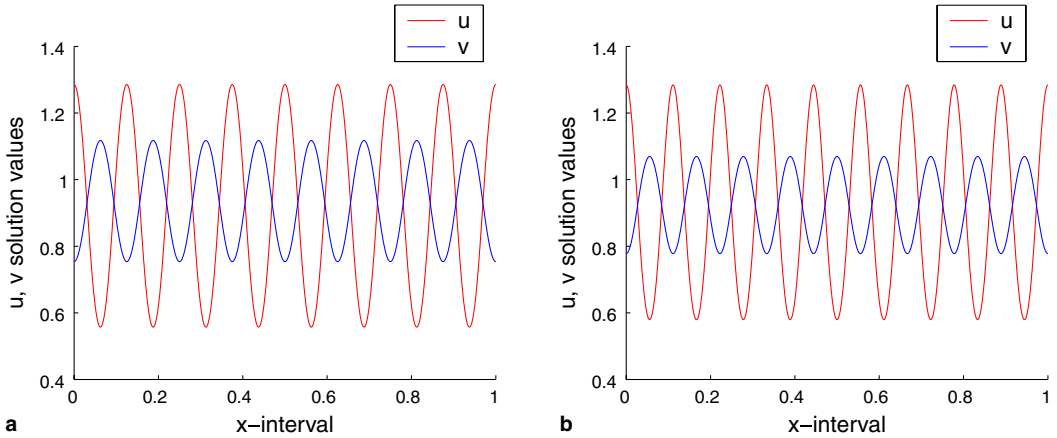


Fig. 2. Numerical results of applying the (a) 1-SBEM and (b) 2-SBDF schemes to the model equations with parameter values shown in Example 1. The solution profile of the u concentration is 180° out of phase with that of v.

computational time steps. However, as γ is increased to say 50000, the first-order 1-SBEM scheme fails to reproduce solutions as accurately as the 2-SBDF scheme. Even refining the mesh or time-step Δt for the first-order scheme does not improve the result. In this case, the solution is highly oscillatory and has high frequencies (see Fig. 3(e)). It is therefore recommended that in the case of highly oscillatory solutions, the 2-SBDF scheme be used. However, if lower values of the scaling parameter are used, the 1-SBEM scheme computes solutions equally as accurate as the second-order scheme. The choice of the scheme really depends on whether one is computing stationary solutions or highly oscillatory solutions.

5.2. 1-D Moving grid finite element simulations

We illustrate some results obtained by solving the reaction–diffusion systems on a continuously growing interval given some growth function. Let $x(t) = X(0)r(t)$ define the grid movement where $X(0)$ represents the initial x-coordinates at time $t = 0$. The function $r(t)$ specifies the rate of growth of the initial grid and sat-

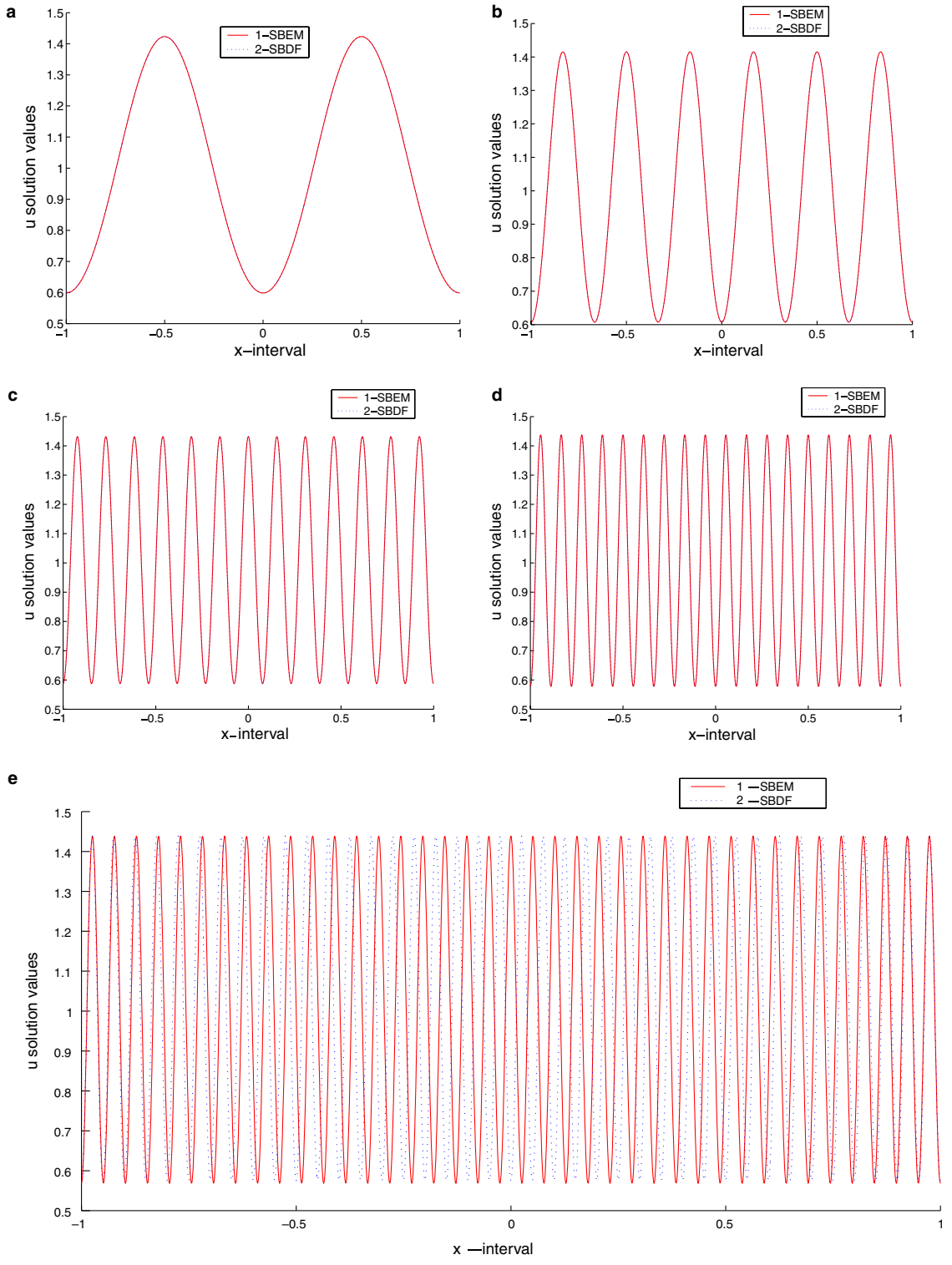


Fig. 3. Numerical results of applying the 1-SBEM (red) and 2-SBDF (blue) schemes to the model equations on a fixed interval with (a) $\gamma = 114.0$, (b) $\gamma = 1000.0$, (c) $\gamma = 5000.0$, (d) $\gamma = 10000.0$ and (e) $\gamma = 50000.0$. For smaller and relatively large values of the scaling parameter γ the two methods compute identical solutions. However for very large values of γ the 1-SBEM scheme fails to capture the highly oscillatory solutions as shown in (e). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Time-step size Δt versus the number of time steps required for the interval $[-1,1]$ to grow to twice this size given that $\rho = 10^{-5}$ and $\rho = 10^{-3}$, respectively

Time-step Δt	No. of time steps ($\rho = 10^{-5}$)	No. of time steps ($\rho = 10^{-3}$)
5×10^{-3}	1.3862943×10^{7}	1.3862943×10^{5}
5×10^{-4}	1.3862943×10^{8}	1.3862943×10^{6}
5×10^{-5}	1.3862943×10^{9}	1.3862943×10^{7}
5×10^{-6}	1.3862943×10^{10}	1.3862943×10^{8}
10^{-6}	6.9314718×10^{10}	6.9314718×10^{8}

isfies that $r(0) = 1$ and $r(t) > 0$ for all positive time. Let us consider that the initial interval $[-1,1]$ grows according to an exponential growth of the form $r(t) = e^{\rho t}$, where ρ represents the growth rate. In Table 3, we illustrate the number of time steps required to grow the interval to say, twice the initial interval given various values of the time-step Δt. The fact that for the 1-SBEM scheme large time-steps can be taken (larger by a factor of ten) as compared to the 2-SBDF scheme makes a huge difference to the number of time steps when solving reaction–diffusion systems on continuously growing domains. As shown in Table 3 large time-steps coupled with a reasonably small growth rate ρ give rise to lower order numbers of time steps. We have seen in previous simulations that the 2-SBDF requires time-step sizes of orders less than approximately 5×10^{-4}, thereby requiring huge numbers of time steps when one is solving the model equations on a continuously growing interval.

For illustrative purposes we solve the model equations on a continuously deforming growing interval $[-1,1]$ by use of the 1-SBEM and 2-SBDF schemes. We take the growth rate to be $\rho = 10^{-3}$ and compute solutions until the interval reaches twice its original size. Let us take $\Delta t = 5 \times 10^{-3}$ for both schemes. With these numerical parameters we require 138,629 number of time steps to grow the interval to twice its original size. We choose to take the parameter values $a = 0.1$, $b = 0.9$ and $\gamma = 29$. Fig. 4(a) and (b) show the numerical results obtained by solving the model equations using the 1-SBEM and 2-SBDF schemes. The relative errors for the 1-SBEM and 2-SBDF schemes are 1.8962×10^{-6} and 1.8951×10^{-6}, respectively. The results obtained from the two schemes when growth reaches the final interval $[-2,2]$ are identical as can be seen from Fig. 4(c). Fig. 4(d) and (e) illustrate results of computing the model equations with $\gamma = 1000$ using the two schemes but with different time-steps. For the 1-SBEM scheme, $\Delta t = 5 \times 10^{-3}$ is taken and the method converges. However the 2-SBDF fails to converge and we are forced to take a smaller time-step. We take $\Delta t = 5 \times 10^{-4}$ and the method converges. The number of time steps required by the 1-SBEM scheme for the domain to double in size is 138 629, but the 2-SBDF requires 10 times more, that is, 1,386,294 time steps. On the other hand the relative error obtained from the 1-SBEM scheme is 3.644×10^{-6} while for the 2-SBDF the relative error is 3.646×10^{-7}. Fig. 4(f) shows the transient solutions when the interval reaches domain size $[-2,2]$. The 1-SBEM and 2-SBDF schemes produce qualitatively similar results even though the two schemes use different time-steps Δt. Clearly the 1-SBEM scheme performs as well as the 2-SBDF taking less number of time steps and larger time-steps. The solution profile represents relatively highly oscillatory solutions. In Example 2 (Fig. 3(e)) we observed that the 1-SBEM scheme failed to compute highly oscillatory solutions as accurately as those computed by the 2-SBDF scheme on fixed intervals. This is still the case even on continuously growing domains as illustrated in Fig. 4(g). Here, we take $\gamma = 15{,}000$ and observe that the two schemes really provide different solution profiles along the boundaries as the domain grows to twice its original size. As the domain grows, there is a lot of activity going on along the boundaries in terms of peak splitting and insertion of the transient solutions. In the middle, the schemes seem to give more or less similar results. We believe that this may be due to domain growth as it has been observed that domain growth appears to select certain patterns at the expense of others [8,19,18]. We can not say much more than this as the phenomenon can only be observed from a computational point of view. Note that we did not plot the transient solutions in the case of $\gamma = 15{,}000$ as in the other cases (a)–(b) and (d)–(e). This is because the transient solutions correspond to highly oscillatory solutions, we were forced to take a small time-step $\Delta t = 5 \times 10^{-6}$ with growth rate $\rho = 0.1$ for both our time-stepping schemes: the 1-SBEM and 2-SBDF. In this scenario, 1 386 294 time steps were required to grow the interval to twice its original size.

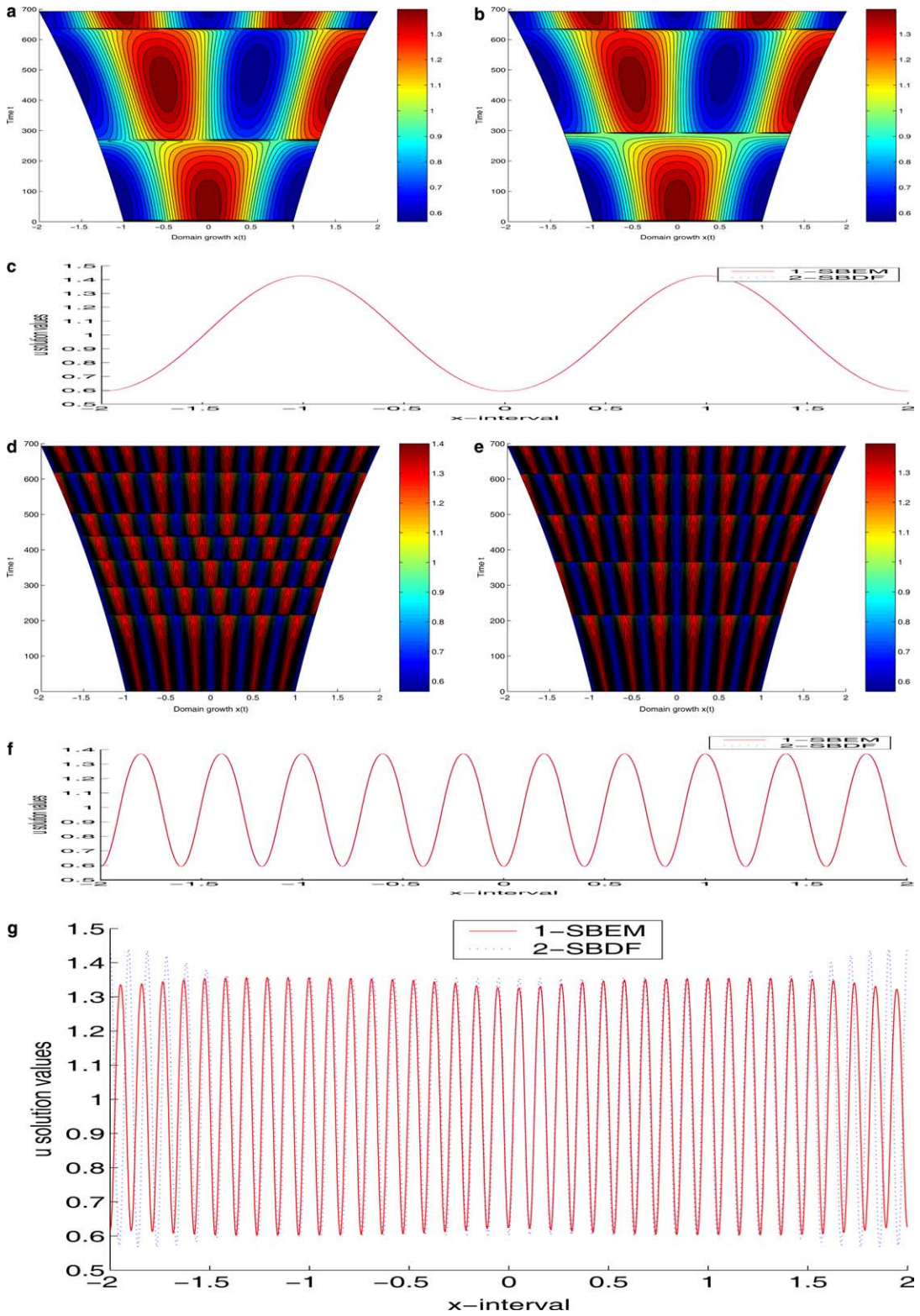


Fig. 4. Moving grid finite element results computed by the 1-SBEM and 2-SBDF schemes for $\gamma = 29$ (a)–(c), $\gamma = 1000$ (d)–(f) and $\gamma = 15{,}000$ (g). Solutions (c) and (f) are plots of the u values when the interval reaches domain length $[-2, 2]$. The results of the two schemes are identical at the final interval except for (g) which represents highly oscillatory solutions.

5.3. 2-D simulations on fixed domains

We solve the model equations on a fixed unit square domain discretized into unstructured triangular elements by a delaunay mesh generator [26]. In all our simulations the unit square domain is discretized into 3664 unstructured triangular elements with 1913 nodes. We shade according to the following: if the values of the u concentration are lower than 1.0 shade yellow otherwise red, this is known as constant threshold shading. The value 1.0 is chosen arbitrarily, any reasonable value will do. We plot results corresponding to the u chemical concentration, those that correspond to v are 180° out of phase with u (see for example, solutions in Fig. 2).

Example 3. Let us consider the model equations with parameter values given by $a = 0.1$, $b = 0.9$, $\gamma = 114.0$, $D_u = 1.0$ and $D_v = 10.0$. We compute solutions to final time $t_F = 2.5$ for various values of Δt. Homogeneous Neumann boundary conditions are used. We illustrate in Fig. 5 the regions of stability of the time-step Δt for 1-SBEM and 2-SBDF schemes, whereby we vary the time-step Δt. The 2-SBDF scheme has a smaller region of stability (by a factor of 10) as compared to the 1-SBEM scheme. The vertical lines indicate where the solution becomes unstable and fails to converge to a meaningful result.

Next we fix all other parameter values as follows: $a = 0.1, b = 0.9, D_u = 1.0, D_v = 10.0$ and vary γ. Let the time-step be $\Delta t = 5.0 \times 10^{-4}$ and compute solutions to final time $t_F = 5.0$. The effect of increasing γ is illustrated in Figs. 6 and 7. Clearly as γ increases, the complexity of the pattern solution increases. The 1-SBEM and 2-SBDF schemes give qualitatively similar results. The 2-SBDF scheme fails to converge when $\Delta t = 5.0 \times 10^{-4}$ for $\gamma = 5000$. Instead, we take a smaller step size $\Delta t = 10^{-4}$. On the other hand, the 1-SBEM scheme converges.

Example 4. Our next example is taken from the paper by Ruuth [29], Example 5. Let the parameter values be given by $a = 0.126779$, $b = 0.792366$, $\gamma = 1000.0$, $D_u = 1.0$ and $D_v = 10.0$. We compute solutions on a unit square with homogeneous Neumann boundary conditions, given initial conditions

$$u(\mathbf{x}, 0) = 0.919145 + 0.0016\cos(2\pi(x + y)) + 0.01\sum_{j=1}^{8}\cos(2\pi jx), \tag{5.2}$$

$$v(\mathbf{x}, 0) = 0.937903 + 0.0016\cos(2\pi(x + y)) + 0.01\sum_{j=1}^{8}\cos(2\pi jx). \tag{5.3}$$

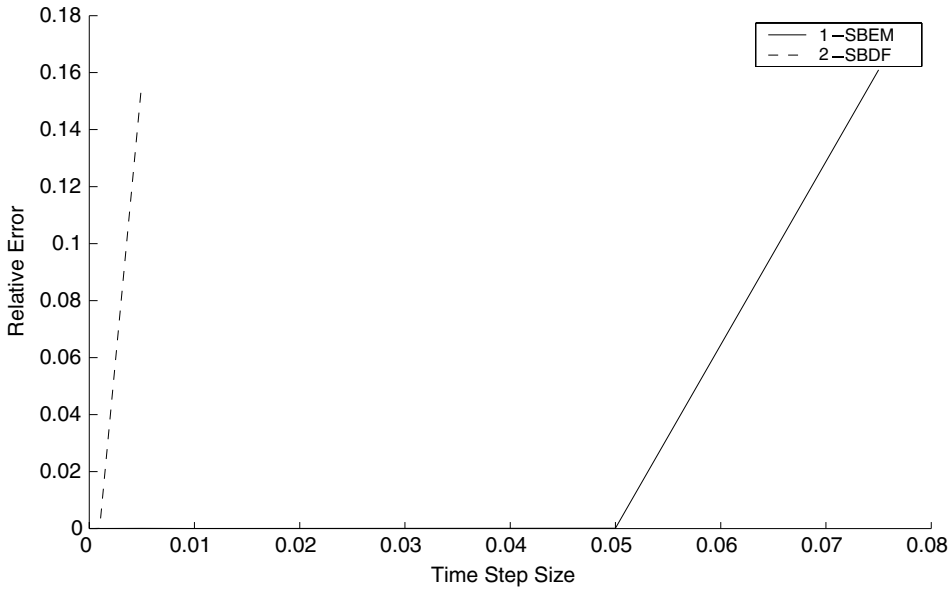


Fig. 5. Regions of stability for the 1-SBEM (solid) and 2-SBDF (dashed) schemes applied to the model equations. The 1-SBEM scheme has a larger region of stability than that of the 2-SBDF scheme.

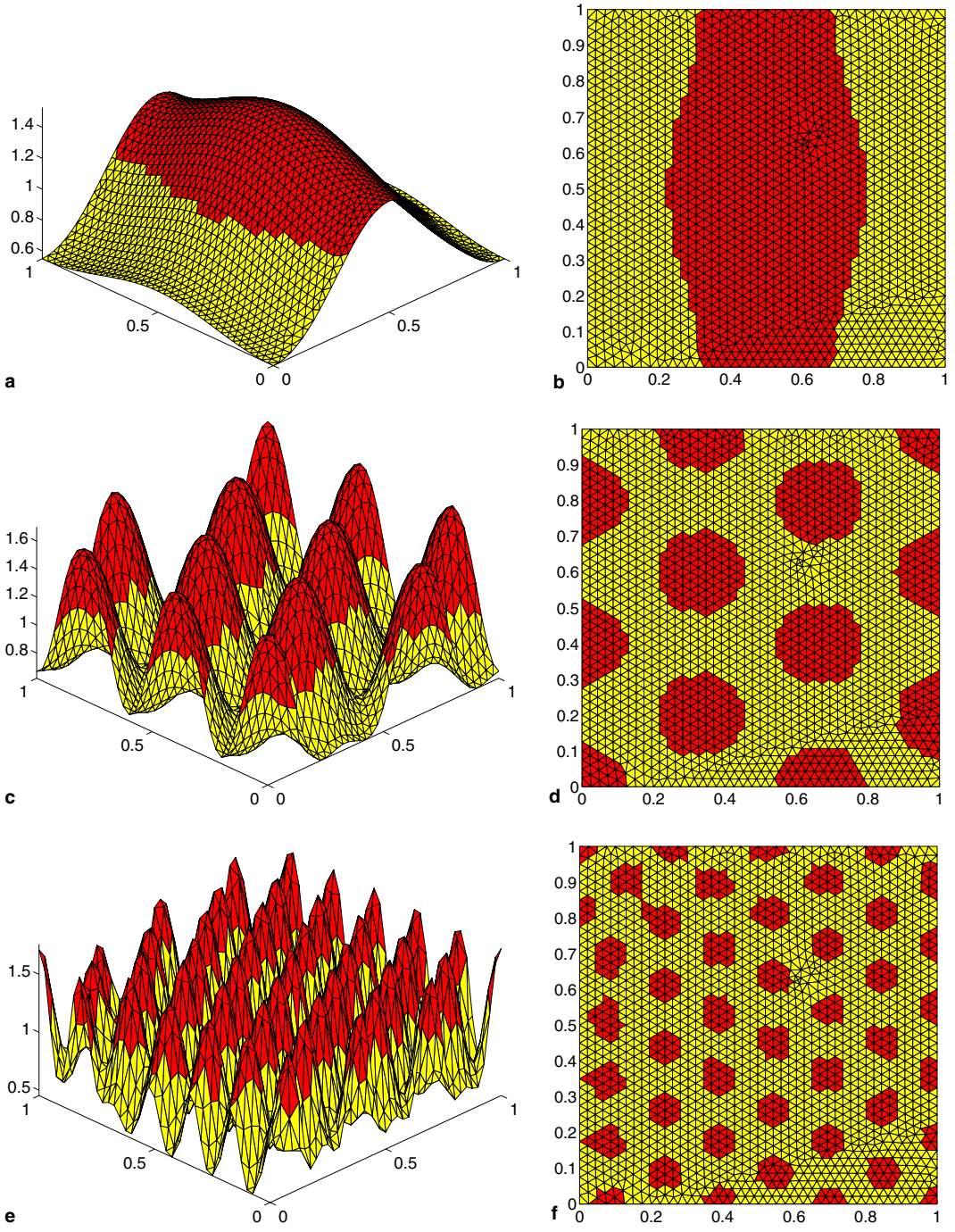


Fig. 6. Numerical results of applying the 1-SBEM scheme to the model equations with $\gamma = 114.0$, $\gamma = 1000.0$ and $\gamma = 5000.0$ with time-step $\Delta t = 5.0 \times 10^{-4}$ in all simulations. Simulations carried out to final time $t_F = 5.0$.

Numerical solutions to the model equations are computed to final time $t_F = 5$, with time-step $\Delta t = 10^{-4}$ by use of both the 1-SBEM and 2-SBDF schemes. The concentration values of u at times $t = 0.025$, $t = 0.125$, $t = 0.15$, $t = 0.2$, $t = 0.225$ and $t = 5.0$ are selected and illustrated. Figs. 8 and 9 show results of computing the model equations using the 1-SBEM and 2-SBDF schemes respectively with identical parameter and

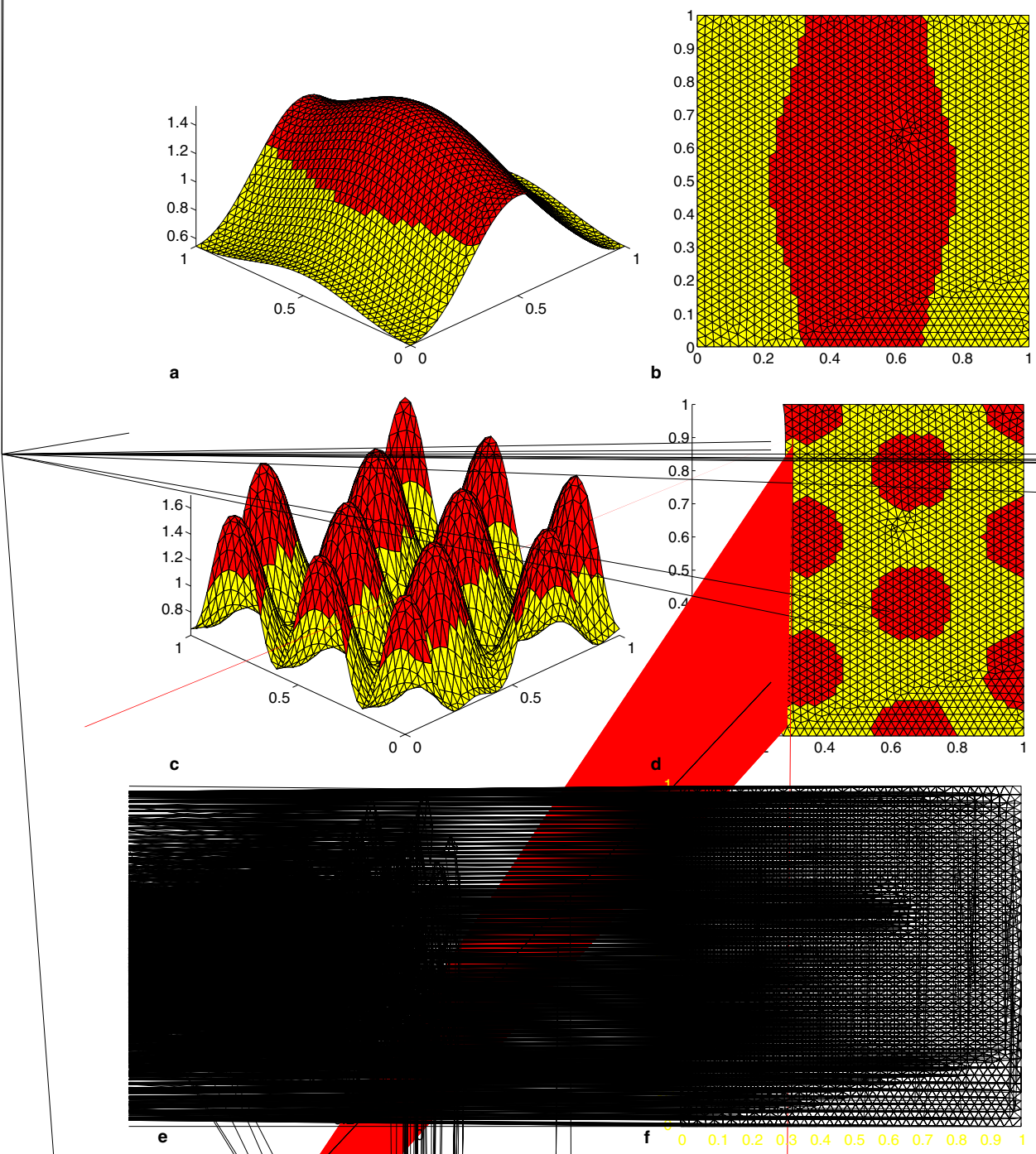
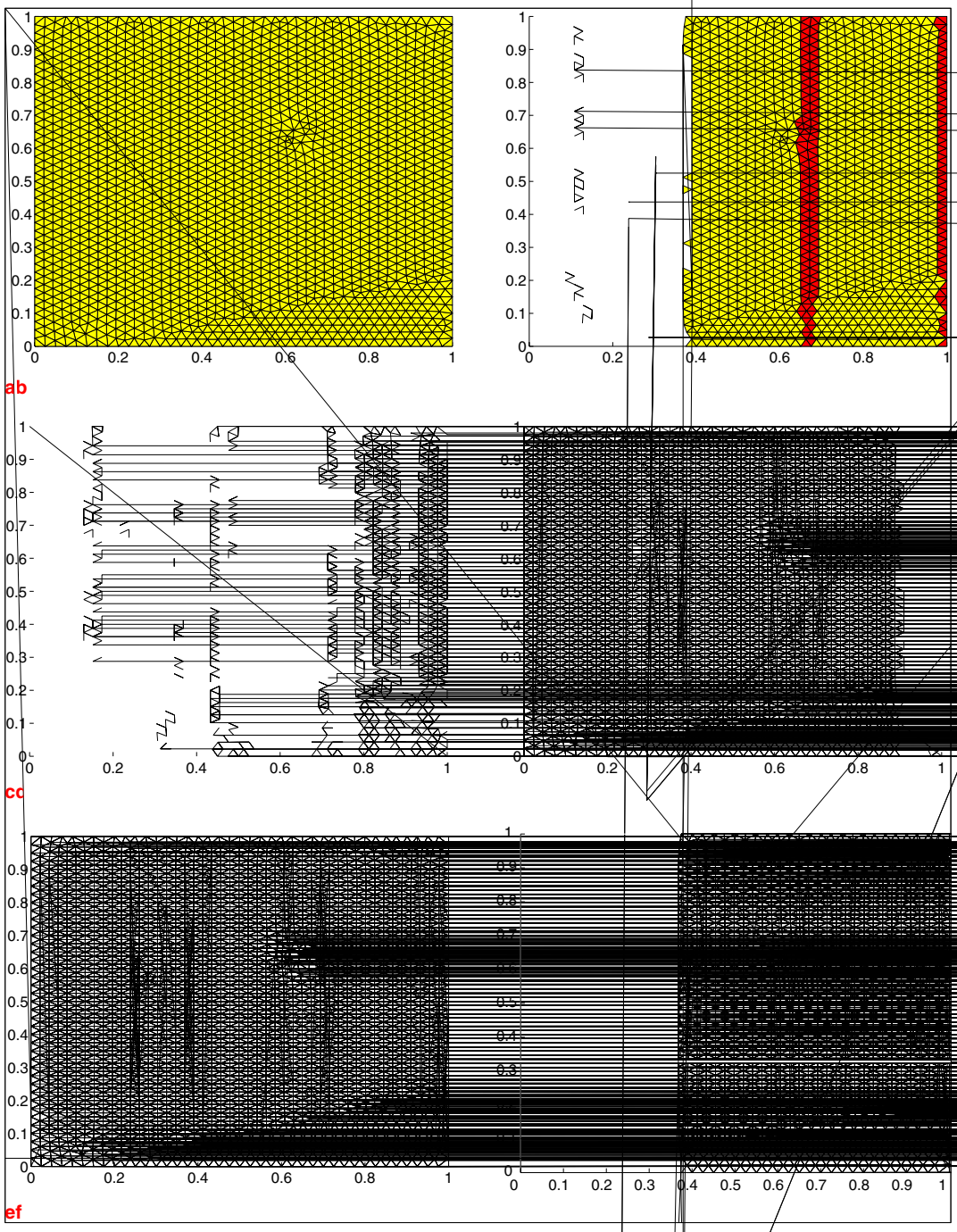


Fig. 7. Numerical results of applying the 2-SBDF scheme to the model equations with $\gamma = 114.0$, $\gamma = 1000.0$ and $\gamma = 5000.0$ with time-step $\Delta t = 5.0 \times 10^{-4}$ in simulations (a)–(d) and $\Delta t = 10^{-4}$ in simulation (e)–(f). Simulations carried out to final time $t_F = 5.0$.

numerical values [illegible] we have computed solutions using homogeneous Neumann boundary conditions. Ruuth [29] cons[illegible]d periodic boundary conditions of the form

$$u(x = 0, y, t) = u(x = 1, y, t), \quad \text{and} \quad u(x, y = 0, t) = u(x, y = 1, t), \tag{5.4}$$

$$v(x = 0, y, t) = v(x = 1, y, t), \quad \text{and} \quad v(x, y = 0, t) = v(x, y = 1, t), \tag{5.5}$$



given the same initial conditions (5.2) and (5.3). We solved the model equations using these conditions and the results of these computations are shown in Figs. 10 and 11. These were computed by using the 2-SBDF scheme. The 1-SBEM scheme yields qualitatively similar results with a relative error of 1.2161×10^{-6} while the 2-SBDF has error 1.1494×10^{-6}. The 1-SBEM scheme produces an error which is less than within 5.8% of the 2-SBDF scheme.

Our results differ completely from those computed and published in the paper by Ruut
the standard 5-point centred finite difference scheme to approximate the diffusion term in t
whereby a regular mesh of size $h = \frac{1}{128}$ was used. The time-step size was taken as $\Delta t =$
solution to the implicit equations was performed using $V(1,1)$ cycles with one iteration pe
were computed to final time $t_F = 0.1$. Our results obtained by using the 1-SBEM sch
scheme) differ substantially from those shown in their paper. Our results are inde
structure, regular or irregular structure give similar results. The results in Figs. 10 and

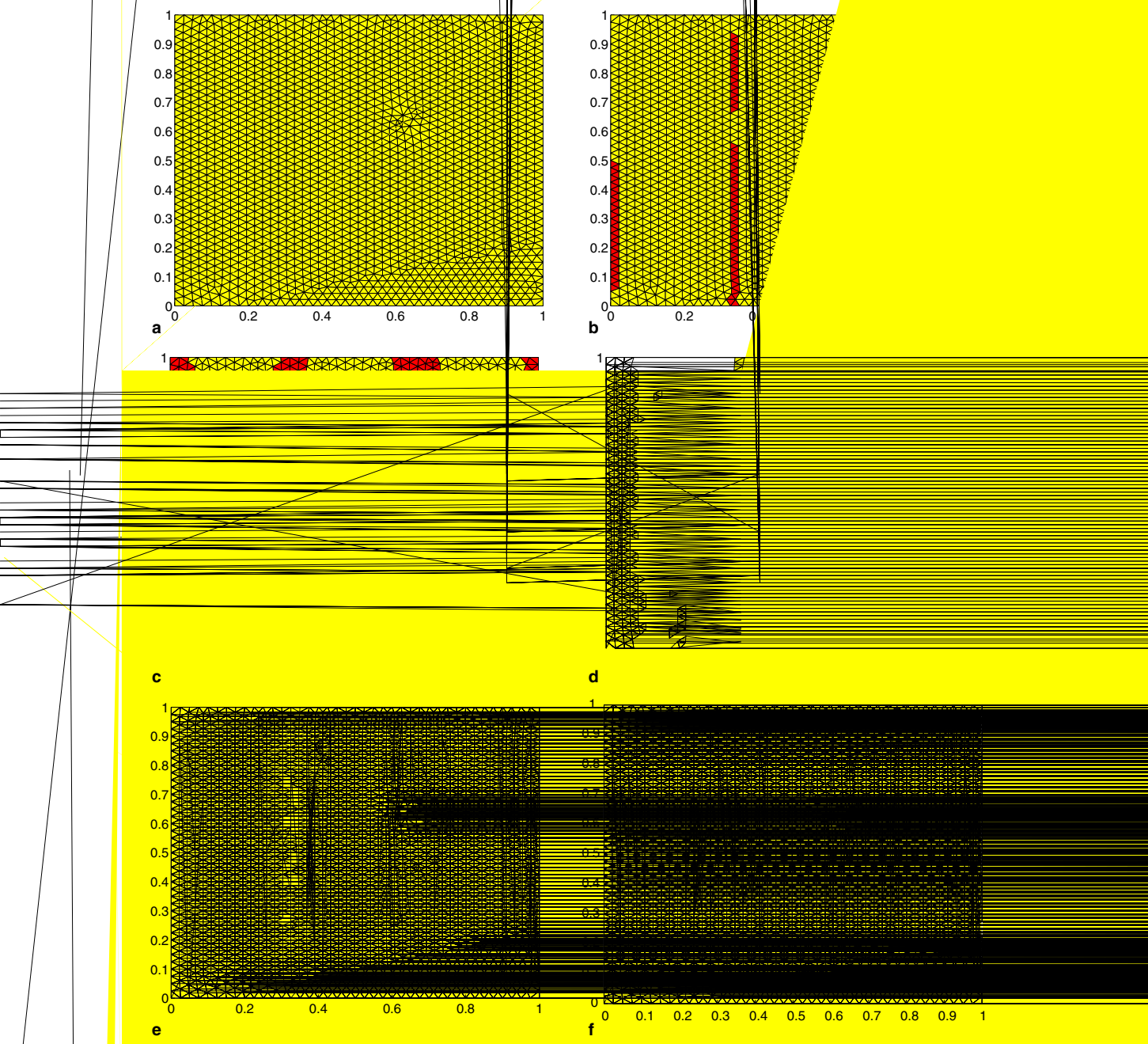
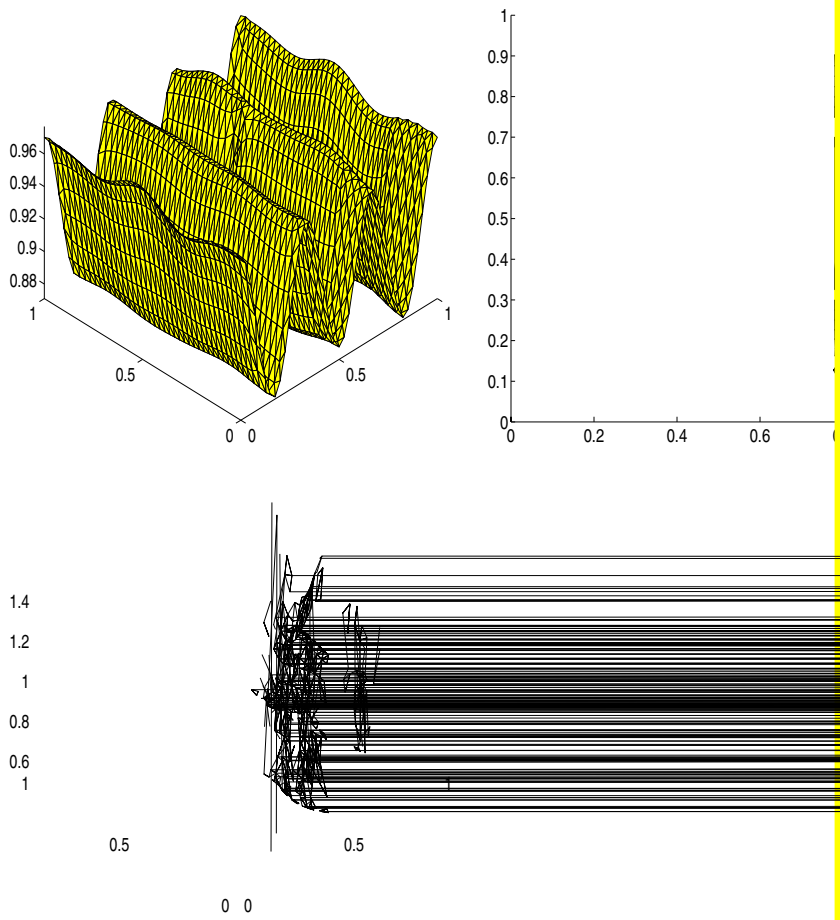


Fig. 9. Numerical results corresponding to the concentration u of the model equations at times $t = 0.025$, $t = 0.075$, $t = 0.15$, $t = 0.2$, $t = 0.225$ and $t = 5.0$. Solutions computed by use of the 2-SBDF scheme.

of spot patterns from initial conditions of stripe pattern form. If the concentration
organise (taking homogeneous Neumann boundary conditions), the development of t
different from those obtained when periodic boundary conditions are imposed as can
results in Figs. 8 and 9 to those shown in Figs. 10 and 11.



We have observed from various computational results that the finite difference scheme applied on a regular mesh imposes symmetry intrinsically to the solutions [9,29]. However, if the mesh regularity is tempered with, say, by introducing very few irregular meshes into the finite difference scheme, such solutions are unobservable. Most results shown in the literature whereby the finite difference scheme has been applied to reaction–diffusion systems are obtained using a regular mesh. There is a need to check the validity of the computational results by not only

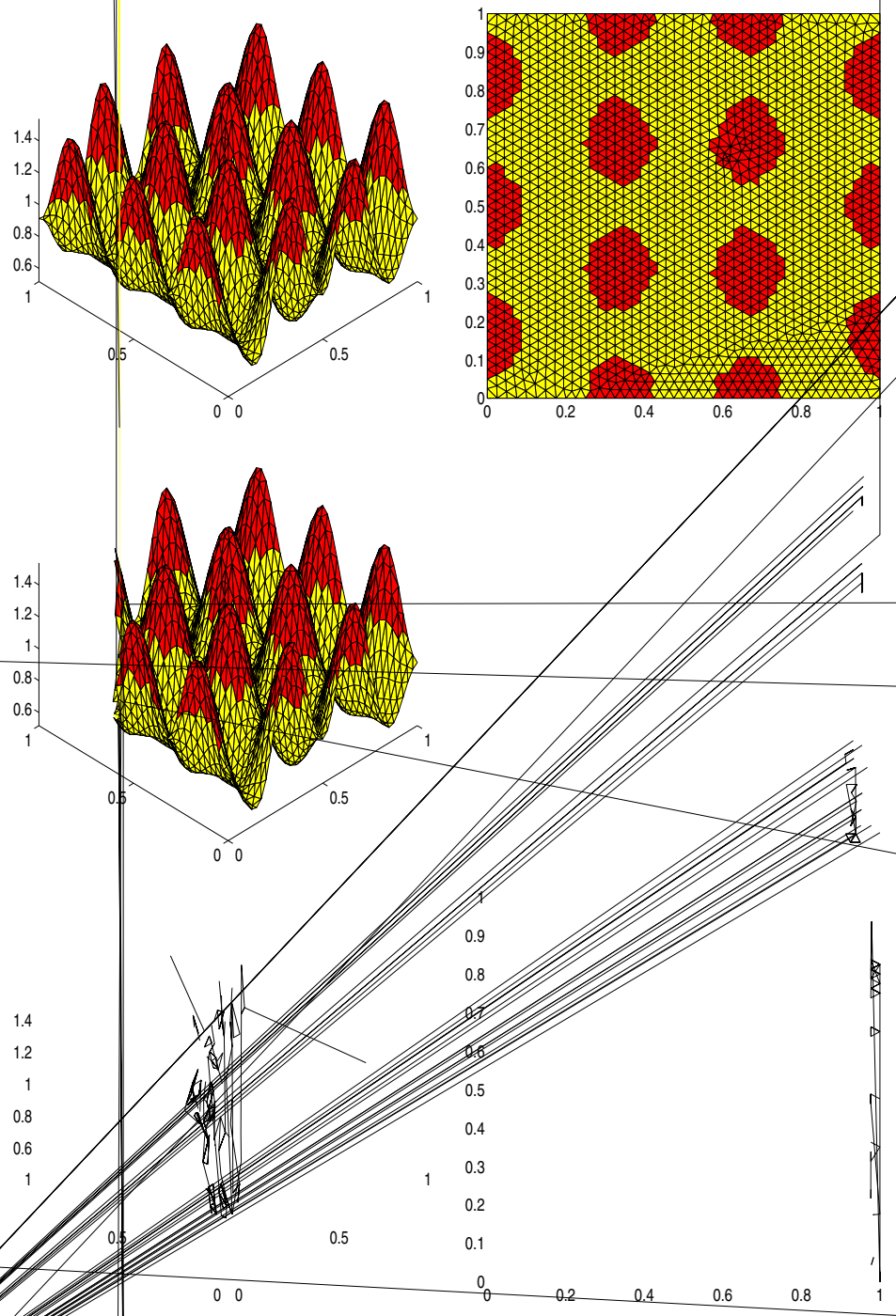


Fig. 1. Continued. The solution is saved at times $t = 0.2$, $t = 0.225$ and $t = 0.6$.

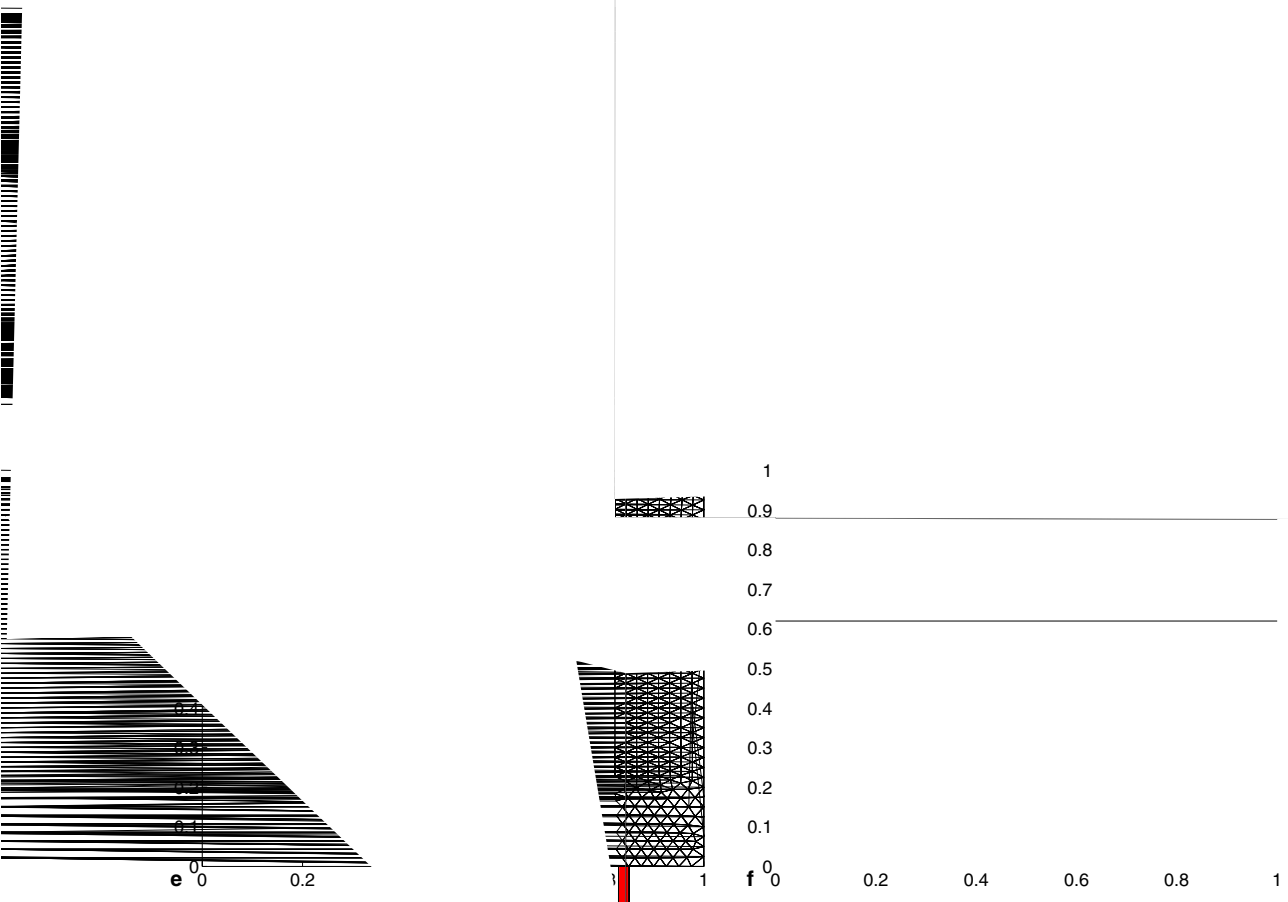
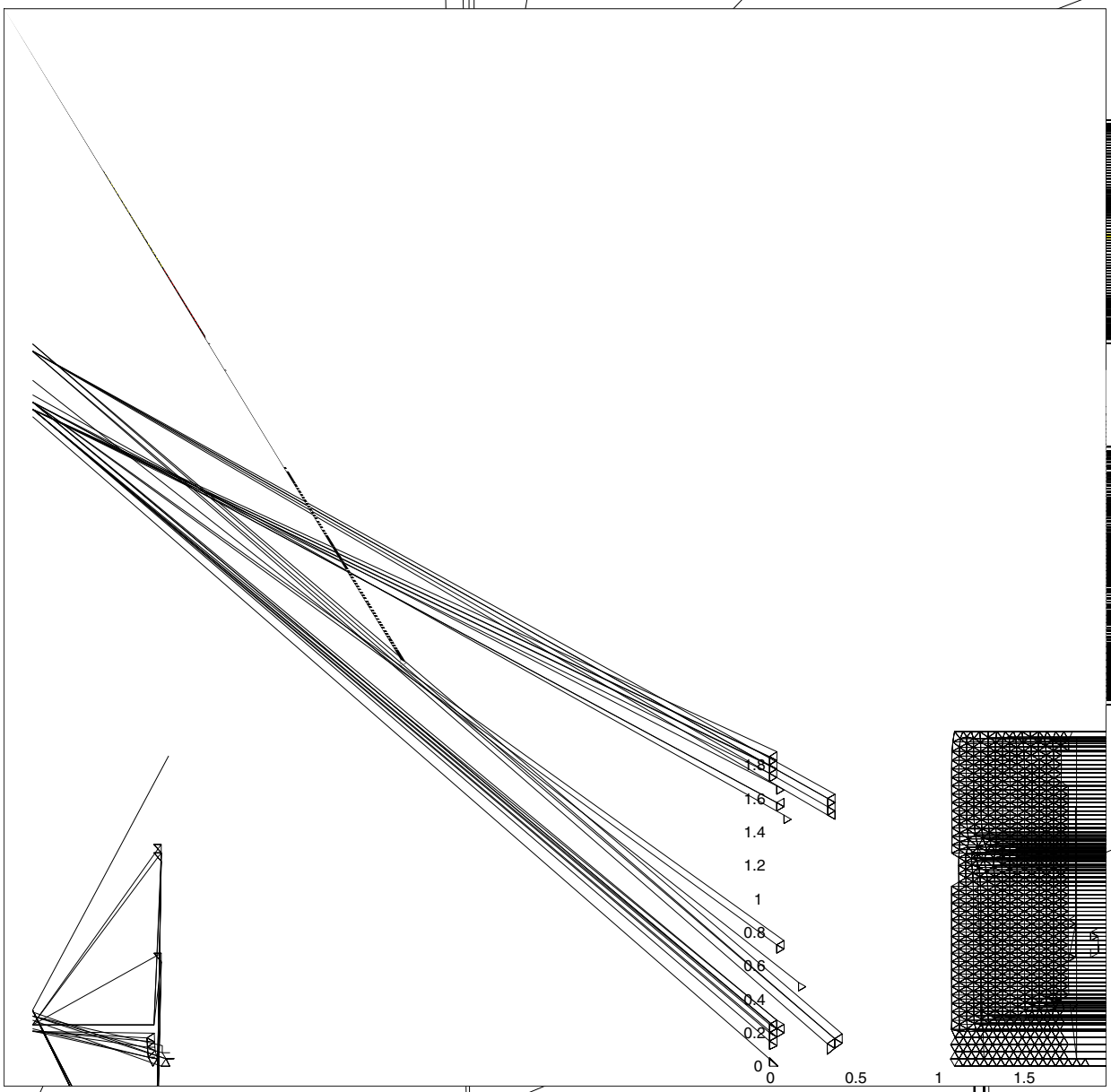


Fig. 12. Numerical results corresponding to the concentration u of the model equations at times $t = 0.00416$, $t = 0.03328$, $t = 0.04992$, $t = 0.06656$, $t = 0.0832$ and $t = 0.1$. Solutions computed by use of the 1-SBEM scheme. Model and numerical parameter values are described in Example 5. Homogeneous Neumann boundary conditions have been used.

$$u(\mathbf{x},0) = 1.886485 + 0.001\sum_{j=1}^{37}\frac{\cos(2\pi jx)}{j}, \tag{5.6}$$

$$v(\mathbf{x},0) = 0.779539 + 0.001\sum_{j=1}^{37}\frac{\cos(2\pi jx)}{j}, \tag{5.7}$$

given homogeneous Neumann conditions along the boundary. We compute solutions with numerical parameter values $\Delta t = 10^{-5}$ to time $t_F = 0.1$. These computations illustrate stripe patterns as opposed to spot patterns as shown clearly in Figs. 12 and 13.



5.4. 2-D moving grid finite element simulations

In this section, we illustrate the application of the 1-SBEM scheme to reaction–diffusion systems (3.1) and (3.2) on continuously deforming two-dimensional domains. For simplicity, we consider a unit square domain that grows exponentially as illustrated in Section 5.2. The initial domain is discretized using a delaunay mesh generator [26] into 3664 unstructured triangular elements with 1913 nodes. The mesh connectivity remains constant with domain growth, there is no mesh refinement, see Fig. 14.

Let us take the following parameter values $a = 0.1$, $b = 0.9$, $D_u = 1.0$, $D_v = 10.0$ and $\gamma = 114.0$ subject to homogeneous Neumann boundary conditions given random initial conditions about the steady state. Take time-step $\Delta t = 5.0 \times 10^{-3}$ and growth rate $\rho = 10^{-3}$. Given these numerical and growth parameters, 277,258 time steps are required for the domain to reach twice its original size. Our results are saved every 11,552 time steps. The 2-SBDF scheme was not used to compute two-dimensional growing domain transient solutions as the scheme required time-step sizes of the order 5.0×10^{-6}. In other words 138,629,438 time steps are required for the domain to grow to twice its original size given such a small Δt, that is more than 500 times those required by the 1-SBDF scheme. The difference in the number of computer time steps between the two schemes is remarkable when solving the model equations on growing domains. The 2-SBDF scheme becomes extremely expensive and therefore computationally prohibitive when solving reaction–diffusion problems on growing domains.

6. Conclusion

There is a remarkable increase in the use of numerical computations of reaction–diffusion problems for pattern formation and their applications to wound healing, cancer and tumor growth. In most of these cases a simple first-order semi-implicit scheme is the first choice. In other cases a semi-implicit Crank–Nicolson is favoured (e.g. [8]). However, Ruuth [29] demonstrated both theoretically and computationally, in one dimension, that such schemes can not represent the fastest growing mode associated with these types of problems. Fully implicit methods are not recommended as they require computing the Jacobian of the nonlinear system, which may be singular in some cases. Of the various time-stepping schemes analysed, the second-order backward differentiation formula (2-SBDF) is recommended as it can resolve high frequencies allowing for relatively larger time-steps. We have taken this method as the benchmark and compared our first-order backward Euler method coupled with semi-implicit treatment of the nonlinear reaction terms (1-SBEM). The way this scheme is applied differs substantially from the usual IMEX schemes where an implicit scheme is used to approximate the diffusion term and an explicit scheme to approximate the reaction terms. Instead, the 1-SBEM scheme is characterised by approximating the diffusion term implicitly as in the IMEX schemes, however, the reaction terms are considered differently: linear terms are treated implicitly and nonlinear terms are linearised semi-implicitly. This is the novelty of our scheme and we believe that could account for the advantages observed in computational simulations. A more detailed theoretical analysis is warranted in this direction.

In this paper, we have illustrated the application of two time-stepping schemes: the first-order 1-SBEM and the second-order 2-SBDF schemes to solving reaction–diffusion problems by use of moving grid finite element methods on fixed and continuously deforming domains. Our numerical results are independent of the mesh structure: regular or irregular mesh yields similar results, with finer mesh providing better approximate solutions.

From various numerical computations we have shown that the 1-SBEM scheme allows for larger time-steps than those allowed for the 2-SBDF scheme, larger by a factor of ten, and computes solutions as relatively accurate as the second-order scheme for the kind of partial differential equations considered in this paper. The fact that our scheme allows for larger time-steps makes it more suitable when solving reaction–diffusion problems on continuously deforming domains. In this scenario, the number of time steps required by the 2-SBDF scheme is at least ten times more than that required by the 1-SBEM scheme. In most biological problems, domain growth takes place on a very slow time scale, hence the growth rate is very small. In multi-dimensions it becomes computationally prohibitive and too expensive as the 2-SBDF scheme requires time steps of the order of millions when domain growth is involved. We have shown through simulations that

the 1-SBEM scheme can compute transient solutions relatively as accurate as the 2-SBDF scheme in one and two dimensions. We have to point out, however that in the case of highly oscillatory solutions, our scheme fails to resolve the highest growing mode. In this case, the 2-SBDF out-performs the first-order scheme as expected. The choice of the scheme depends on the nature of the problem, if stationary solutions are sought, then the 1-SBEM scheme is recommended. However, in cases of highly oscillatory solutions, the 2-SBDF scheme should be used. We have illustrated the advantage of the first-order method from a computational point of view: we are currently carrying out analytical studies to verify these computational results although such studies are not trivial since we are dealing with nonlinear partial differential equations.

There are other types of reaction–diffusion problems, for example those that include nonlinear diffusion, or chemotaxis terms. For such problems it will be harder to implicitly treat the diffusion terms without further linearisation. Therefore the application of the IMEX schemes and the 1-SBEM scheme will have to be treated differently and modifications implemented accordingly to obtain appropriate numerical schemes. We have not addressed any of these in our studies.

Both these schemes have been incorporated into the software package that we are currently developing. This package is free and can be downloaded from the website: http://www.auburn.edu/~madzva1.

We have shown in previous papers [22,19,18] that patterns formed on a continuously growing domain are robustly selected due to domain growth and therefore insensitive to initial conditions as opposed to patterns obtained on fixed domains which are sensitive to initial conditions [27]. Domain growth has been shown to enhance the pattern selection process independent of the numerical scheme. Finite elements and finite difference schemes on unstructured mesh produce qualitatively similar results. We must point out however that the finite difference difference scheme on regular symmetrical mesh elements do influence the pattern selection process by imposing symmetry to numerical solutions (Madzvamuse, in preparation). Therefore, we recommend to always check numerical results by using unstructured mesh for centred finite difference schemes.

The moving grid finite element method assumes that domain deformation is calculated from plausible growth functions or those derived from biological experiments and therefore is a known quantity [8,20,19,18]. This assumption differs substantially from that of the classical moving finite element methods which assumes that both the nodal solution and nodal movement are unknown quantities to be determined simultaneously in the numerical process [3,7,23,24]. The work in this paper clearly extends the results obtained in our previous papers [19,18] and illustrates the advantages of using the first-order backward Euler scheme for the diffusion and nonlinear terms. The reaction kinetics used in this paper are one of the many suggested reaction kinetics in our previous papers. In all our simulations there is no mesh refinement since domain growth takes place at a very slow rate. However, we are developing discrete mesh refinement schemes in the case that domain grows at a faster rate.

Acknowledgements

A.M. acknowledge support from Auburn University. I thank Professor Philip K. Maini and Dr. Andrew J. Wathen, of Oxford University, for a critical reading of the manuscript.

References

[1] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit–explicit methods for time-dependent PDE's, SIAM, J. Numer. Anal. 32 (3) (1995) 797–823.
[2] P. Arcuri, J.D. Murray, Pattern sensitivity to boundary and initial conditions in reaction–diffusion models, J. Math. Biol. 24 (1986) 141–165.
[3] M.J. Baines, Moving Finite Elements. Monographs on Numerical Analysis, Clarendon Press, Oxford, 1994.
[4] P. Borkmans, A. DeWit, G. Dewel, Competition in ramped Turing structures, Physica A 188 (1992) 137–157.
[5] V. Dufiet, J. Boissonade, Conventional and unconventional turing patterns, J. Chem. Phys. 96 (1) (1992) 664–673.
[6] N.N. Carlson, K. Miller, Design and application of a gradient-weighted moving finite element code I: in one dimension, SIAM J. Sci. Comput. 19 (3) (1998) 728–765.
[7] N.N. Carlson, K. Miller, Design and application of a gradient-weighted moving finite element code II: in two dimensions, SIAM J. Sci. Comput. 19 (3) (1998) 766–798.
[8] E.J. Crampin, E.A. Gaffney, P.K. Maini, Reaction and diffusion on growing domains: scenarios for robust pattern formation, Bull. Math. Biol. 61 (2002) 1093–1120.

[9] E.J. Crampin, Reaction–diffusion patterns on growing domains. D Phil Thesis, University of Oxford, 2000.
[10] A. Gierer, H. Meinhardt, A theory of biological pattern formation, Kybernetik 12 (1972) 30–39.
[11] P. Gray, S.K. Scott, Autocatalytic reactions in the isothermal, continuous stirred tank reactor: isolas and other forms of multistability, Chem. Eng. Sci. 38 (1) (1983) 29–43.
[12] P. Gray, S.K. Scott, Autocatalytic reactions in the isothermal, continuous stirred tank reactor: oscillations and the instabilities in the system $A + 2B \rightarrow 3B$, $B \rightarrow C$, Chem. Eng. Sci. 39 (6) (1984) 1087–1097.
[13] E. Hairer, S.P. Norsett, G. Wanner, Solving ordinary differential equations I, Springer-Verlag, 1987.
[14] P.K. Jimack, A.J. Wathen, Temporal derivatives in the finite element method on continuously deforming grids, SIAM J. Numer. Anal. 28 (4) (1991) 990–1003.
[15] M.L. Kagan, R. Ksloff, O. Citri, D. Avnir, Chemical formation of spatial patterns induced by nonlinearity in a concentration-dependent diffusion coefficient, J. Phys. Chem. 93 (7) (2003) 2728–2731.
[16] A. Kassam, L.N. Trefethen, Fourth-order time stepping for stiff PDEs, SIAM J. Sci. Comp. 26 (4) (2005) 1214–1233.
[17] S. Kondo, R. Asai, A reaction–diffusion wave on the skin of the marine angelfish Pomacanthus, Nature 376 (1995) 765–768.
[18] A. Madzvamuse, P.K. Maini, A.J. Wathen, A moving grid finite element method for the simulation of pattern generation Turing models on growing domains, J. Sci. Comp. 24 (2) (2005) 247–262.
[19] A. Madzvamuse, P.K. Maini, A.J. Wathen, A moving grid finite element method applied to a model biological pattern generator, J. Comput. Phys. 190 (2003) 478–500.
[20] A. Madzvamuse, R.D.K. Thomas, P.K. Maini, A.J. Wathen, A numerical approach to the study of spatial pattern formation in the Ligaments of Arcoid Bivalves, Bull. Math. Biol. 64 (2002) 501–530.
[21] A. Madzvamuse, A numerical approach to the study of spatial pattern formation, D Phil Thesis, University of Oxford, 2000.
[22] P.K. Maini, E.J. Crampin, A. Madzvamuse, A.J. Wathen, R.D.K. Thomas, Implications of domain growth in morphogenesis, in: V. Capasso (Ed.), Mathematical Modeling and Computing in Biology and Medicine, 5th ESMTB Conference, Societa Editrice Esculapio, 2002, pp. 67–73.
[23] K. Miller, R.N. Miller, Moving finite elements. Part I, SIAM J. Numer. Anal. 18 (1981) 1019–1032.
[24] K. Miller, Moving finite elements. Part II, SIAM J. Numer. Anal. 18 (1981) 1033–1057.
[25] K.W. Morton, D.F. Mayers, Numerical Solution of Partial Differential Equations, Cambridge University Press, Cambridge, 1994.
[26] J.D. Müller, P.L. Roe, H. Deconinck, A frontal approach for internal node generation for Delaunay triangulation, Int. J. Numer. Meth. Fluids 17 (3) (1993) 241–256.
[27] J.D. Murray, Mathematical Biology. II: Spatial Models and Biomedical Applications, Springer, Heidelberg, NY, 2002.
[28] K.J. Painter, P.K. Maini, H.G. Othmer, Complex spatial patterns in a hybrid chemotaxis reaction–diffusion model, J. Math. Biol. 41 (4) (2000) 285–314.
[29] S. Ruuth, Implicit–explicit methods for reaction–diffusion problems in pattern-formation, J. Math. Biol. 34 (2) (1995) 148–176.
[30] Y. Saad, SPARSEKIT, a basic tool kit for sparse matrix computations, 1994. Available from: <http://www-users.cs.umn.edu/saad/>.
[31] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Co., 1996.
[32] J. Schnakenberg, Simple chemical reaction systems with limit cycle behavior, J. Theor. Biol. 81 (1979) 389–400.
[33] D. Thomas, Artificial enzyme membrane, transport, memory and oscillatory phenomena, in: D. Thomas, J.-P. Kervenez (Eds.), Analysis and Control of Immobilised Enzyme Systems, Springer, Berlin, 1975, pp. 115–150.
[34] A.M. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. B (237) (1952) 37–72.