Moving Finite Element Simulations for Reaction-Diffusion Systems

Guanghui Hu[1], Zhonghua Qiao[2,*], and Tao Tang[3]

[1] *Department of Mathematics, Michigan State University, East Lansing, MI 48824-1027, United States*
[2] *Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*
[3] *Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong*

Abstract. This work is concerned with the numerical simulations for two reaction-diffusion systems, i.e., the Brusselator model and the Gray-Scott model. The numerical algorithm is based upon a moving finite element method which helps to resolve large solution gradients. High quality meshes are obtained for both the spot replication and the moving wave along boundaries by using proper monitor functions. Unlike [33], this work finds out the importance of the boundary grid redistribution which is particularly important for a class of problems for the Brusselator model. Several ways for verifying the quality of the numerical solutions are also proposed, which may be of important use for comparisons.

1 Introduction

The generation of spatial pattern of tissue structures is one of the elementary processes in morphogenesis. Since the pioneering work of Turing [28] in 1952, there have been many studies on two-component reaction-diffusion systems for the formation of spatially complex patterns, see, e.g., [4, 7–9, 21]. A two-component reaction-diffusion system with general reaction terms f and g has the following form:

$$\frac{\partial u}{\partial t} - D_u \Delta u = f(u,v) \qquad \text{in} \quad \Omega \times (0,\infty), \tag{1.1a}$$

$$\frac{\partial v}{\partial t} - D_v \Delta v = g(u,v) \qquad \text{in} \quad \Omega \times (0,\infty), \tag{1.1b}$$

subject to the no-flux boundary conditions and initial conditions

$$\frac{\partial u}{\partial \nu} = \frac{\partial v}{\partial \nu} = 0 \qquad \text{on} \quad \partial\Omega \times (0,\infty), \tag{1.2a}$$

$$u(x,y,0) = u^0(x,y), \quad v(x,y,0) = v^0(x,y) \qquad \text{in} \quad \Omega. \tag{1.2b}$$

Based on many specific forms of the reaction terms f and g, various models have been proposed for the pattern formation, such as Gray-Scott model [5], Schnakenberg model [25], Brusselator model [22] and Gierer-Meinhardt (GM) model [3]. These models have been widely

*Corresponding author. *Email addresses:* `ghhu@math.msu.edu` (G. Hu), `zqiao@inet.polyu.edu.hk` (Z. Qiao), `ttang@math.hkbu.edu.hk` (T. Tang)

used to model localization processes in nature, such as cell differentiation and morphogenesis [7, 20], and the formation of sea-shell patterns [21].

Asymptotic and analytical methodologies for the analysis of these reaction-diffusion systems mainly concentrated on the one-dimensional model, see, e.g., [11, 12, 27]. Recently, for the semi-strong interaction regime $D_u \ll 1$ with $D_v = \mathcal{O}(1)$, several asymptotic methodologies have been developed in two space dimensions, see, e.g., [30, 31]. However, for the weak interaction regime, where $D_v = \mathcal{O}(D_u) \ll 1$, there are only a few works, see, e.g., [32]. Note that in [32], the case $D_u \ll 1$ and $D_v \gg 1$ is also considered. A survey of the asymptotic methods for reaction-diffusion systems is given in [29].

Numerical simulations play an important role in studying pattern formations, especially when there is difficulty in using the asymptotic and analytical approaches. In this paper, we will focus on the numerical studies on the two-dimensional Gray-Scott model and Brusselator model in the weak interaction regime. The Gray-Scott model is given by:

$$u_t = D_u \Delta u - uv^2 + \gamma(1-u), \tag{1.3a}$$
$$v_t = D_v \Delta v + uv^2 - (\gamma + \kappa)v, \tag{1.3b}$$

and the Brusselator model is given as follows:

$$u_t = D_u \Delta u + A + uv^2 - (B+1)u, \tag{1.4a}$$
$$v_t = D_v \Delta v + Bu - uv^2. \tag{1.4b}$$

Here, the diffusion coefficients $D_u, D_v \ll 1$ and f, k, A, B are chemical reaction parameters.

The common aspect of these pattern formation problems is the subtle interplay between diffusion and reaction. Many of the numerically computed patterns for these two models have shown that the smaller the diffusion parameters D_u and D_v are chosen, the steeper the solutions become [22, 32]. These patterns include spots, labyrinthine stripes and mixed spots and stripes. Furthermore, these spots and stripes also have some dynamical behaviors, such as long-time oscillation, annihilation, replication, and so on. Consequently, very fine meshes over the spatial extension of the spots and stripes are needed in numerical simulations. Although it is possible to solve the problems using a very large number of equidistantly spaced spatial mesh points dynamics in one space dimension, it is computationally inefficient for two-dimensional simulations; and it is extremely challenging for three-dimensional simulations. As a result, adaptive grid methods seem necessary in resolving various pattern formulations with small diffusion parameters. One of such adaptive methods is the moving mesh method. In past years, several moving mesh strategies are proposed to solve (1.1). In [26], One-dimensional GM and Schnakenberg models were solved using the moving mesh method. One-dimensional Gray-Scott model was solved in [2]. Two-dimensional Gray-Scott and Brusselator model have been solved by a moving mesh method in [33]. In [23], simulations of various of spikes dynamics of the two-dimensional GM model has been demonstrated. The study of the pattern formation of Turing-type reaction-diffusion systems on growing domains has attracted a lot of attention recently. It is not feasible to obtain meaningful analytical solutions due to the nonlinear nature of the reaction-diffusion system that is coupled with the domain growth. In [16–19], a moving grid finite element method is used to study the pattern formation on growing domains in one and two-dimensions.

In this paper, we will employ a moving finite element method, introduced in [14, 15], to solve the Gray-Scott model and the Brusselator model in two space dimension. This moving finite element method is based on a harmonic mapping framework. One of the primary features of this method is that the mesh redistribution part and the PDE evolution part are separated. It allows us to use any appropriate PDE solver for solving (1.3) or (1.4) which is independent on mesh redistribution strategy.

For the Gray-Scott model, the soution with large gradient is far away from the boundary even after a long time. So by using moving mesh method, boundary points redistribution seems less important. However, this is not the case for the Brusselator model whose solution with large gradient may appear on the boundary. In [33], numerical results are also shown for both models. Results for Gray-Scott model look very good: the algorithm detected the replication procedure successfully, and such procedure is represented by the high quality of the mesh produced. For the Brusselator model, however, as to be demonstrated in our numerical experiments there are large phase errors along the boundary using the approach in [33]. We will show that this is due to the use of the fixed boundary grid points.

This paper is organized as follows. In section 2, we present a moving finite element method to solve the two reaction-diffusion systems (1.3) and (1.4) numerically. In section 3, numerical results are given to demonstrate the importance of the use of small time stepping and the boundary mesh redistribution. With correct use of the time stepping and boundary mesh redistribution, we can obtain satisfied numerical simulations efficiently. We end this paper with some concluding remarks in section 4.

2 Numerical Methods

2.1 PDE solver

In this work, we will employ standard finite element methods to discretize the governing equations spatially. To resolve the large variation of the solution accurately, the moving mesh technique is used. For consideration of numerical accuracy on the temporal discretization, the fourth-order Runge-Kutta (RK4) scheme is adopted.

Spatial Discretization

To describe the PDE solver, for simplicity, we rewrite the systems (1.3) or (1.4) into the compact form

$$M\mathbf{u}_t = K\Delta\mathbf{u} + \mathbf{f}(\mathbf{u}), \tag{2.1}$$

where $\mathbf{u} = (u, v)$,

$$M := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad K := \begin{pmatrix} D_u & 0 \\ 0 & D_v \end{pmatrix}. \tag{2.2}$$

For the Brusslator model,

$$\mathbf{f}(\mathbf{u}) := \begin{pmatrix} A + uv^2 - (B+1)u \\ Bu - uv^2 \end{pmatrix}; \tag{2.3}$$

and for the Gray-Scott model,

$$\mathbf{f}(\mathbf{u}) := \begin{pmatrix} -uv^2 + \gamma(1-u) \\ uv^2 - (\gamma+\kappa)v \end{pmatrix}. \tag{2.4}$$

For the spatial discretization, we use the linear finite element approximation. Let $\mathcal{T} = \{\mathcal{K}\}$ be a finite element partition of Ω into triangular element $\mathcal{K}$. Now we define the linear finite element space

$$S_h = \{w \in C(\Omega) : w|_{\mathcal{K}} \in P_1(\mathcal{K}), \forall \mathcal{K} \in \mathcal{T}\},$$

where $P_1(\mathcal{K})$ denotes the space of polynomials of degree 1 over $\mathcal{K}$. Then the finite element scheme of (2.1) is: Find $\mathbf{u}_h \in (S_h)^2$ such that

$$\left(M\frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{w}_h \right) + K(\nabla \mathbf{u}_h, \nabla \mathbf{w}_h) = (\mathbf{f}, \mathbf{w}_h), \quad \forall \mathbf{w}_h \in (S_h)^2, \tag{2.5a}$$

$$(\mathbf{u}_h(\mathbf{x}, 0), \mathbf{w}_h) = (\mathbf{u}_h^0, \mathbf{w}_h), \quad \forall \mathbf{w}_h \in (S_h)^2, \tag{2.5b}$$

where

$$\mathbf{u}_h^0(\mathbf{x}) := \left(u_h^0(x, y), v_h^0(x, y) \right)^T \in (S_h)^2$$

is the finite element approximation to $\mathbf{u}^0(\mathbf{x}) := \left(u^0(x, y), v^0(x, y) \right)^T$.

Temporal Discretization

So far we have only considered spatial discretization, leaving the time variable continuous. In this section, we give the full discretization scheme of (2.5).

The popular method to discretize the ODE system (2.5) is the semi-implicit time integration method. That is, the diffusive term is dealt with implicitly and the reaction term explicitly. The main advantage is that solving a nonlinear system, with for instance the Newton's method, can be avoided, while still having reasonable stability properties, at least for mildly stiff equations [33].

The simplest discretization method is the semi-implicit Euler scheme. The advantage of Euler method is its low requirement of storage. For each time level, just the variables at the previous time level are needed, and no more extra variable is needed. But it does not mean that it is an efficient way. Since it is a first order scheme, very small length of time step is needed to keep the accuracy. However, the time step can not be too small, since the accumulation of rounding error also may affect the accuracy of numerical solutions. So the numerical results after a long time calculation become not reliable, which results in that the Euler method is not suitable for the application.

Based on the consideration above, in our simulation, the semi-implicit fourth order Runge-Kutta method (RK4) is used to discretize the ODE system (2.5), and the full discretization scheme is given as follows:

$$\frac{1}{\Delta t}(M\mathbf{u}_h^{n+1}, \mathbf{w}_h) + K(\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{w}_h) = \frac{1}{\Delta t}(M\mathbf{u}_h^n, \mathbf{w}_h)$$
$$+ \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \forall \mathbf{w}_h \in (S_h)^2, \tag{2.6a}$$

$$(\mathbf{u}_h(\mathbf{x}, 0), \mathbf{w}_h) = (\mathbf{u}_h^0, \mathbf{w}_h), \quad \forall \mathbf{w}_h \in (S_h)^2, \tag{2.6b}$$

where

$$\begin{aligned}
k_1 &= (f(\mathbf{u}_h^n), \mathbf{w}_h), \\
k_2 &= (f(\mathbf{u}_h^n + \frac{k_1}{2}), \mathbf{w}_h), \\
k_3 &= (f(\mathbf{u}_h^n + \frac{k_2}{2}), \mathbf{w}_h), \\
k_4 &= (f(\mathbf{u}_h^n + k_3), \mathbf{w}_h),
\end{aligned}$$

By using the above RK4 method, at each time level, not only the extra storage is needed, but also total eight linear systems are solved (four for each variable). However, first, the results obtained from RK4 is much more accurate than that obtained from the Euler method. From the numerical experiment in the last section, the RK4 method can depict the phase of numerical solution even after a long time calculation, while the Euler method will introduce big phase

error with the same length of time step. Second, the RK4 method has larger stable region than the Euler method does, that means much larger length of time step can be adopted for the RK4 method. So in the simulations, the semi-implicit RK4 method is used.

2.2 Mesh-redistribution

In Sec. 3, two reaction-diffusion models will be simulated, the Gray-Scott model and the Brusselator model. In the simulations of both models, there are always solutions with large gradient of the solutions in the domain. The adaptive methods become necessary to resolve these solutions effectively. In this work, the moving mesh method will be used for this aim.

Note that there are differences between the numerical results obtained from two models. For the Gray-Scott model which describes the splitting and replication of spots in the domain, the solutions with large gradient appear in regions which are far away from the boundary. However, this is not the case for the Brusselator model, which the solutions with large gradient move along the boundary with the evolution of the time. Based on the above observation, one can imagine that a moving mesh method which can redistribute the internal grid points and boundary grid points simultaneously is needed to guarantee the numerical accuracy of both models.

So far, there are several ways proposed in the literature to redistribute the grid points on the boundary. Before [14, 15], the classical way is to solve a $(N-1)$-dimensional problem on the boundary (suppose the original problem is N-dimensional), and then move the mesh according to the solution of this $(N-1)$-dimensional problem. This method is always used for 2-dimensional problem. For 3-dimensional problem, it is very complicated to use this method.

In [14, 15], Li et al. proposed a uniform way to redistribute all grid points simultaneously. Instead of solving a Poisson equation with Dirichlet boundary condition to construct the harmonic mapping, an optimization problem is solved. With such method, both 2-dimensional and 3-dimensional problems can be handled well. For details of this method, we refer to [14,15] and references therein. In the following, we just give a brief summarization.

First, we need to construct an initial logical mesh. This can be done by solving the following optimization problem

$$\min \sum_k \int_\Omega \sum_i \left(\frac{\partial \xi^k}{\partial x^i} \right) d\vec{x},$$
$$s.t. \quad \vec{\xi}\,|_{\partial\Omega} = \xi_b \in \mathbf{K},$$

where $\vec{x}$ and $\vec{\xi}$ are the local coordinates of two compact Riemannian manifolds Ω and Ω_c respectively, and $\mathbf{K}$ is a mapping set from $\partial\Omega$ to $\partial\Omega_c$. During the whole simulation, this initial logical mesh is unchanged, and is used to measure the movement of the grid point in the logical domain.

Now assume that we have obtained a finite element solution (u_h^n, v_h^n) at $t = t_n$ on the mesh $\mathcal{T}^n$ using the methods described in section 2.1. The outline of the moving mesh algorithm is given in Algorithm 1.

According to Algorithm 1, when $||\,\vec{\xi}^* - \vec{\xi}^0\,||_{L^\infty}$ is small enough, we get a new mesh $\mathcal{T}^{n+1}$ for $t = t_{n+1}$. Then we can solve the PDE on the new mesh following the algorithm described in section 2.1. The detail in the above iteration can be found in [14, 15] and the codes for all the computations below are based on the adaptive finite element package AFEPack [13].

For the choice of the monitor function $\mathcal{M}$, a popular one has the following form

$$\mathcal{M} = \sqrt{1 + \alpha\,|\nabla u|^2}, \tag{2.7}$$

where $\alpha > 0$ is a parameter which controls the adaptivity of the moving mesh method. Larger α means larger adaptivity, while $\alpha = 0$ results in no adaptivity. (2.7) can satisfy a large amount

---

Algorithm 1: Flow Chart of Moving Mesh Method

- Step 1 : Solving the following optimization problem

$$\min \sum_{k} \int_{\Omega} G^{ij} \frac{\partial \xi^k}{\partial x^i} \frac{\partial \xi^k}{\partial x^j} d\vec{x},$$
$$s.t. \quad \xi \mid_{\partial \Omega} = \xi_b \in \mathbf{K},$$

  to obtain a logical mesh $\vec{\xi}^*$, where $G = (G^{ij}) = \mathcal{M}^{-1}$. $\mathcal{M}$ is called a monitor function.

- Step 2 : Judge if $||\vec{\xi}^* - \vec{\xi}^0||_{L^\infty}$ is small enough, where $\vec{\xi}^0$ is the fixed initial logical mesh. If yes, the iteration is over. Otherwise, do the following procedure.

- Step 3 : Using the difference $\vec{\xi}^* - \vec{\xi}^0$ to compute the mesh-moving vector $\delta \vec{x}$ in the physical domain. Then move the grid points in the physical domain by using $\delta \vec{x}$.

- Step 4 : Update the numerical solutions from the old physical mesh to the new physical mesh.

- Step 5 : Go to step 1.

---

of problems. Most recently, this arc-length type monitor function was used in [24] for the time adaptivity in the simulation of the thin-film epitaxial growth. However, there is no universal criterion for the selection of α because it is deeply problem-dependent. In the simulation, we will use (2.7) as the monitor function, and use different α for two different models according to our numerical experience. This monitor function may not be optimal to steer the mesh adaptation. A new balancing method was proposed in [1] for the monitor functions, which may be one possible way to avoid this ad hoc approach.

3   Numerical Experiments

In this section, numerical simulations of the Brusselator model are given first. There is no analytical solution of this model, so we calculate a reference solution for the convergence test of our algorithm. Then the convergence of our moving finite element method is checked. The advantage of moving mesh method and the importance of movement of grid points on the boundary are demonstrated respectively. Finally, the numerical results of simulations of Gray-Scott model are presented.

For the simulation of Brusselator model, we use $\alpha = 1$ in (2.7). To describe the procedure of the spots replication accurately for Gray-Scott model, more adaptivity, say, larger α is expected. However, with the number of spots increases in the domain, the area of the region with large solution variation also increases. That means we can not use too large α, otherwise there will be not sufficient grid points to resolve after certain time. According to our numerical experience, we use $\alpha = 5$ in the simulation of Gray-Scott model.

3.1 The Brusselator model

We consider the two-dimensional Brusselator model with the following parameters:

$$D_u = D_v = 2 \times 10^{-3}, \quad A = 1 \quad \text{and} \quad B = 3.4.$$

Neumann boundary conditions are imposed on the boundary of the square domain $\Omega = [0,1] \times [0,1]$:

$$\frac{\partial u}{\partial n} = \frac{\partial v}{\partial n} = 0 \quad \text{on} \quad \partial \Omega.$$

The initial conditions read

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

With these boundary and initial conditions, a wave will move along the boundaries periodically. This model has also been studied in [6,33].

We first find out reliable reference solutions of Brusselator model.

3.1.1 Reference results on the fine uniform mesh

Several numerical tests are carried out to give reliable reference results. First, we use an 100×100 mesh, and different lengths of time step are tested such as $\Delta t = 1.0e-02$, $2.5e-03$, $6.25e-04$ and $1.5625e-04$. According to the comparisons of results obtained with these time steps, we find that the difference between the results obtained with $\Delta t = 6.25e-04$ and $\Delta t = 1.5625e-04$ is very small. So we fix the length of time step to be $\Delta t = 6.25e-04$ in our computations.

In the following, different mesh sizes will be used to test the numerical convergence of the algorithm when $\Delta t = 6.25e-04$. The mesh size $\Delta h = 1.6667e-02$, $\Delta h = 1.25e-02$, $\Delta h = 1.0e-02$ and $\Delta h = 8.3333e-03$ are used. According to the results, we find that the difference between the results obtained with $\Delta h = 1.0e-02$ and $8.3333e-03$ meshes is very small. So we fixed the mesh size as $\Delta h = 1.0e-02$.

According to the above discussion, numerical results which are obtained with $\Delta t = 6.25e-04$ and mesh size $\Delta h = 1.0e-02$(corresponds to 11833 degrees of freedom) are used as reference results for the Brusselator model in our work.

3.1.2 Numerical results with moving mesh method

In this subsection, we first check the numerical convergence of the moving finite element method according to the reference results described in Sect 3.1.1.

Numerical convergence

We mention again that $\alpha = 1$ is used in (2.7), and the time step $\Delta t = 6.25e-04$ is adopted. In simulations, three different meshes with 506 degree of freedom (DOFs), 1941 DOFs, and 4396 DOFs are used, and numerical results are given in Fig. 1 and Fig. 2 when $t = 11$ and $t = 19$ respectively. The numerical convergence is demonstrated clearly in the figures. That is, with the increment of number of DOFs, the difference between the numerical results and the reference results becomes smaller.

It can also observed from Figs. 1 and 2 that the phase error of numerical results becomes large with time evolution. In the following comparisons, it will be shown that the moving finite element method has advantage on reducing the phase error.
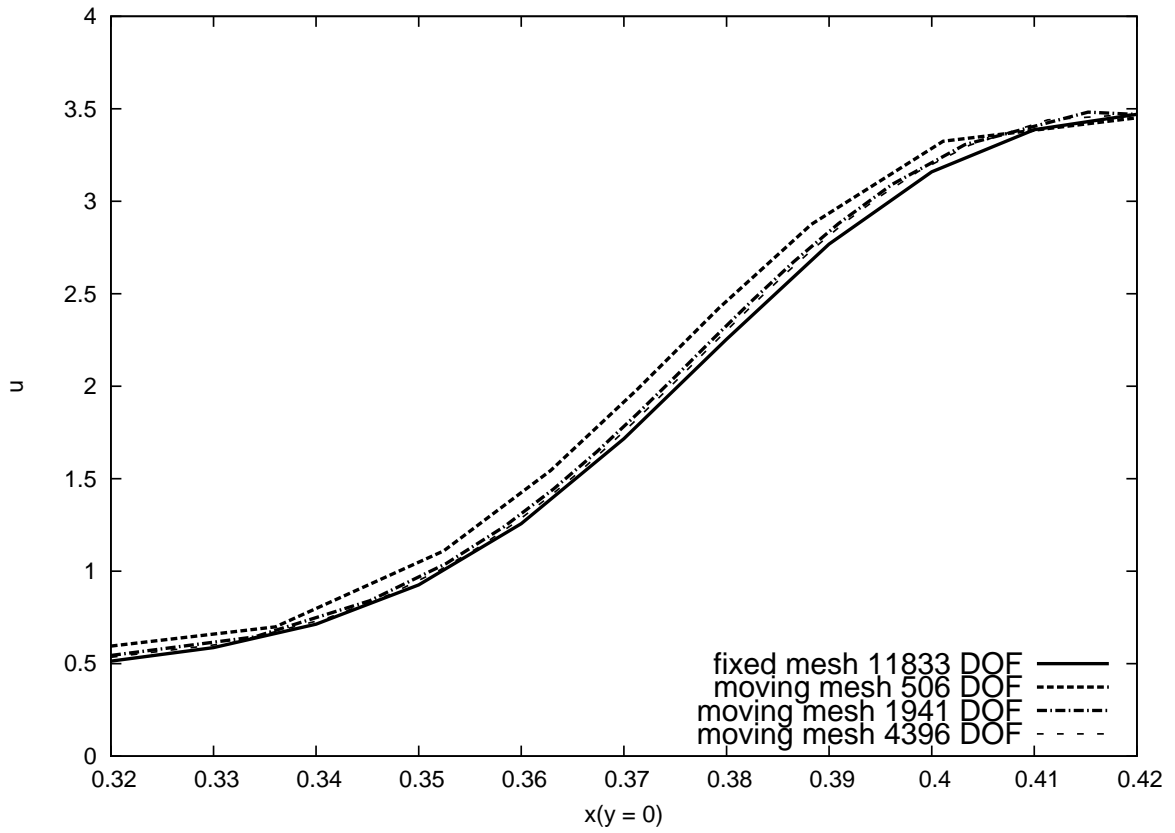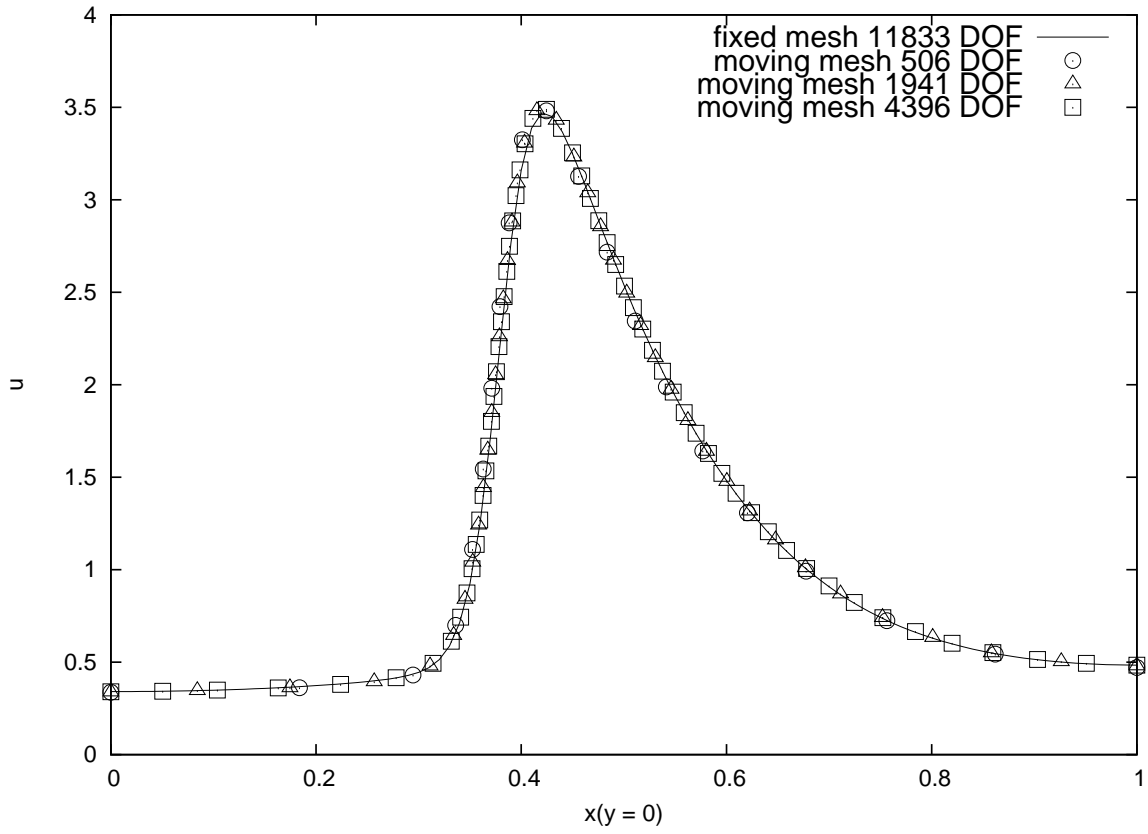
7

Figure 1: Numerical results of moving mesh method with different mesh sizes on $y=0$ when $t=11$. Top: numerical solutions of u on the whole line $y=0$; Bottom: numerical solutions on the range $x \in [0.32, 0.42]$.
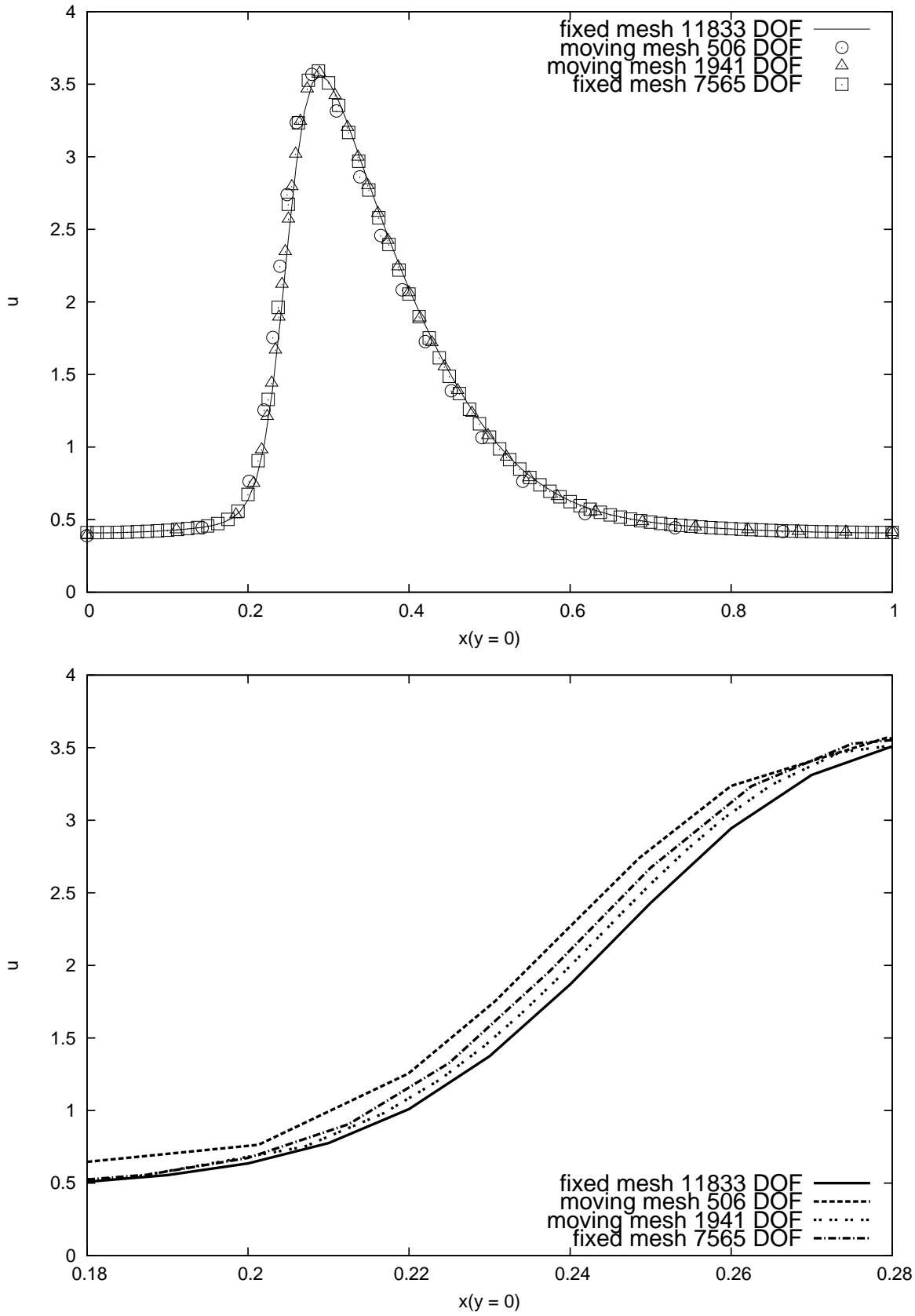
Figure 2: Numerical results of moving mesh method and fixed mesh with different mesh sizes on $y=0$ when $t=19$. Top: numerical solutions of u on the whole line $y=0$; Bottom: numerical solutions on the range $x \in [0.18,0.28]$.

Compare with results obtained from the fixed mesh

It is very important for an algorithm to resolve the sharp phase of the solution as correct as possible, otherwise the numerical results will be not reliable with time evolution.

As we can see that by using moving finite element method, the phase error of numerical results when $t=19$ is larger than that when $t=11$. However, it should be noted that compared with fixed mesh case, the moving finite element method is much better on resolving the sharp phase. In Fig. 2, the numerical results obtained from fixed mesh with 7565 DOFs are also demonstrated. It is obvious that the phase error of moving finite element method with 1941 DOFs is even smaller than that obtained from fixed mesh with 7565 DOFs. That means we can use less DOFs to get more accurate results. In fact, the fixed mesh (7565 DOFs) is obtained by globally refining the mesh (1941 DOFs) one time.

Besides the moving mesh method, we also would like to mention that the high-order temporal discretization is also important for accurately depicting the sharp phase of the solutions.

As we mentioned before, in the Brusselator model, the sharp phase of solutions moves along the boundary. We have demonstrated that moving finite element method had advantage on resolving the sharp phase on the boundary $y=0$. It can be imagined that moving finite element methods also give much more accurate results than those obtained from fixed mesh inside the domain. This is shown by Fig. 3 which demonstrates the three isolines ($u=0.5, 1.0$, and 1.5) when $t=19$. From this figure, we can see that compared with the results obtained from the fixed mesh with 7565 DOFs, the results obtained from the moving mesh method with 4396 DOFs are closer to the reference results (especially see Fig. 3, right column).

Based on the above discussion, we can claim that the moving finite element method proposed in this paper is more efficient than the uniform mesh approach in simulating the Brusselator model. As the end of this subsection, we give a full view of numerical results on the domain. Fig. 4 shows the numerical results for u by using moving mesh (1941 DOFs) when $t=1, 15$, and 19. From the mesh configurations which are shown on the right column, we can see that our moving mesh method resolves the large solution variations successfully.

As we mentioned before, the Brusselator model is a special case because the solution with large gradient appears along the boundary. So it is very important that the grid points on the boundary can move together with the internal grid points at the same time, and its importance will be shown in the next numerical experiment.

Importance of movement of boundary points

The results of moving interior points but fixing the boundary points are shown in Fig. 5. As shown by the figure, the results obtained using 4396 DOFs are unacceptable, because there is a large phase error on the boundary $y=0$. Such results are even worse than that obtained from our moving mesh method with 506 DOFs. It can be expected that with such large phase error on the boundary, the internal solutions are also not reliable. This can be seen from Fig. 6 where the numerical results nearby the boundary are demonstrated. It confirms our discussion above that the moving mesh method without redistributing the boundary points is not suitable for problems having solutions with large gradient along the boundary. With time evolution, the error will become bigger and bigger. Fig. 7 shows the 2D view of solutions obtained using moving mesh method with fixed boundary points (4396 DOFs).

In [33], authors use a moving mesh method with fixed boundary. Based on our numerical experience, it seems that moving interior and boundary points simultaneously is important for the simulation.
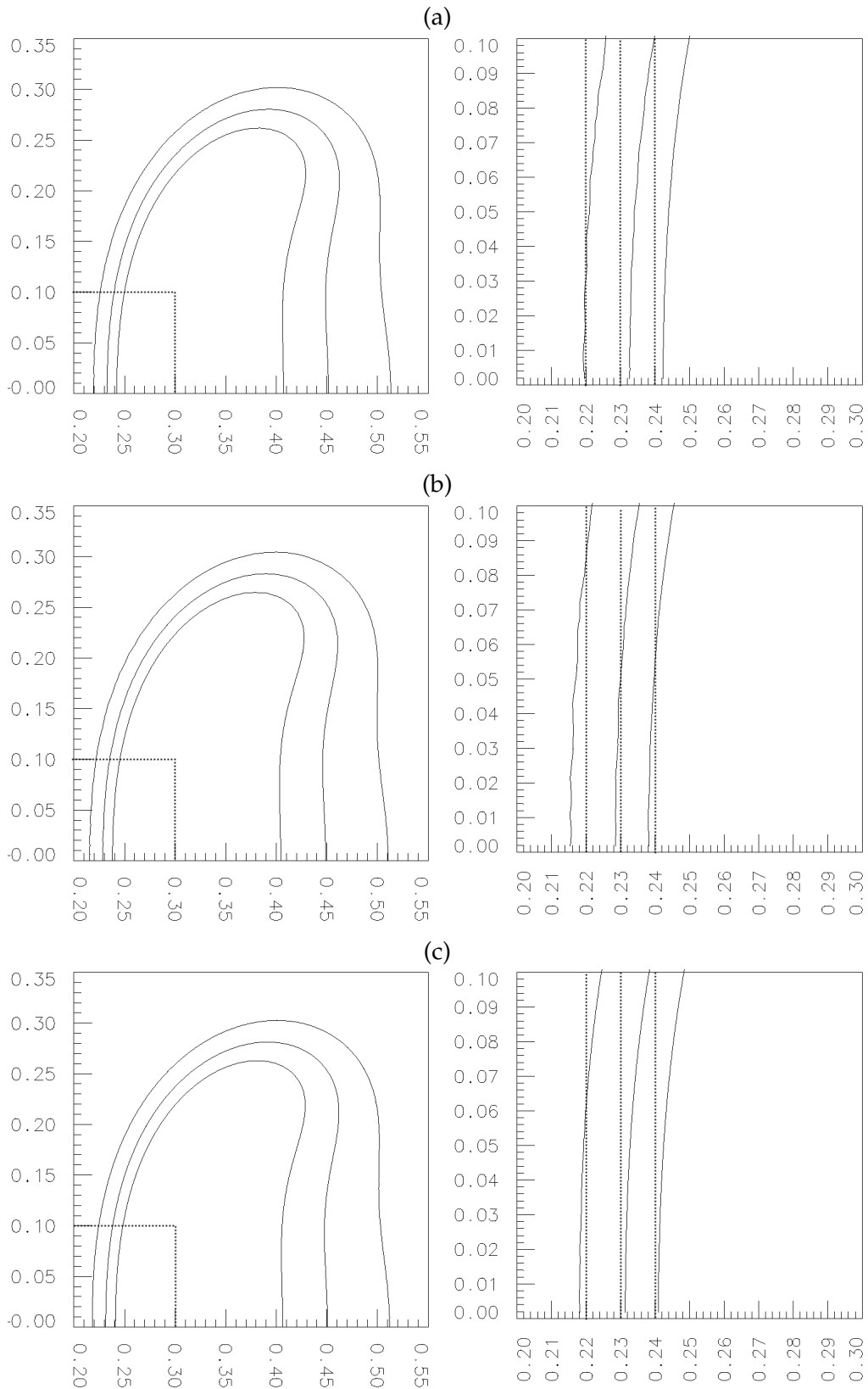
10

Figure 3: Numerical results of u when $t = 19$. (a): fixed mesh with 11833 DOFs. (b): fixed mesh with 7565 DOFs. (c): moving mesh with 4396 DOFs. Left: isolines with $u = 0.5, 1.0$ and 1.5. Right: the detailed results shown in the dashed box of the left figure.
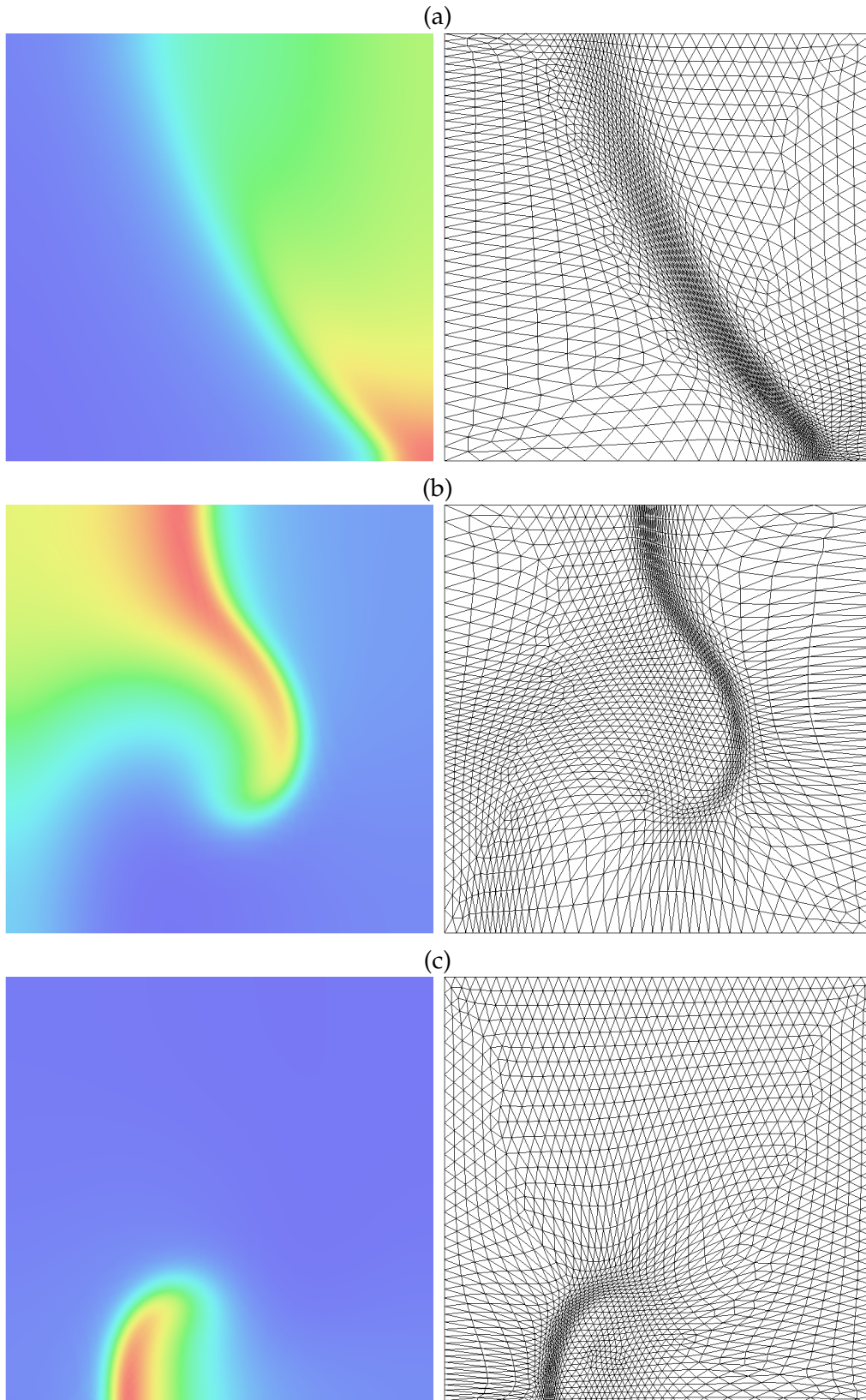
Figure 4: Numerical results for u (left column) obtained with moving mesh (1941 DOFs) when $t=1,15$, and 19. Right column shows the corresponding mesh configuration. (a): $t=1$, (b):$t=15$, and (c): $t=19$.
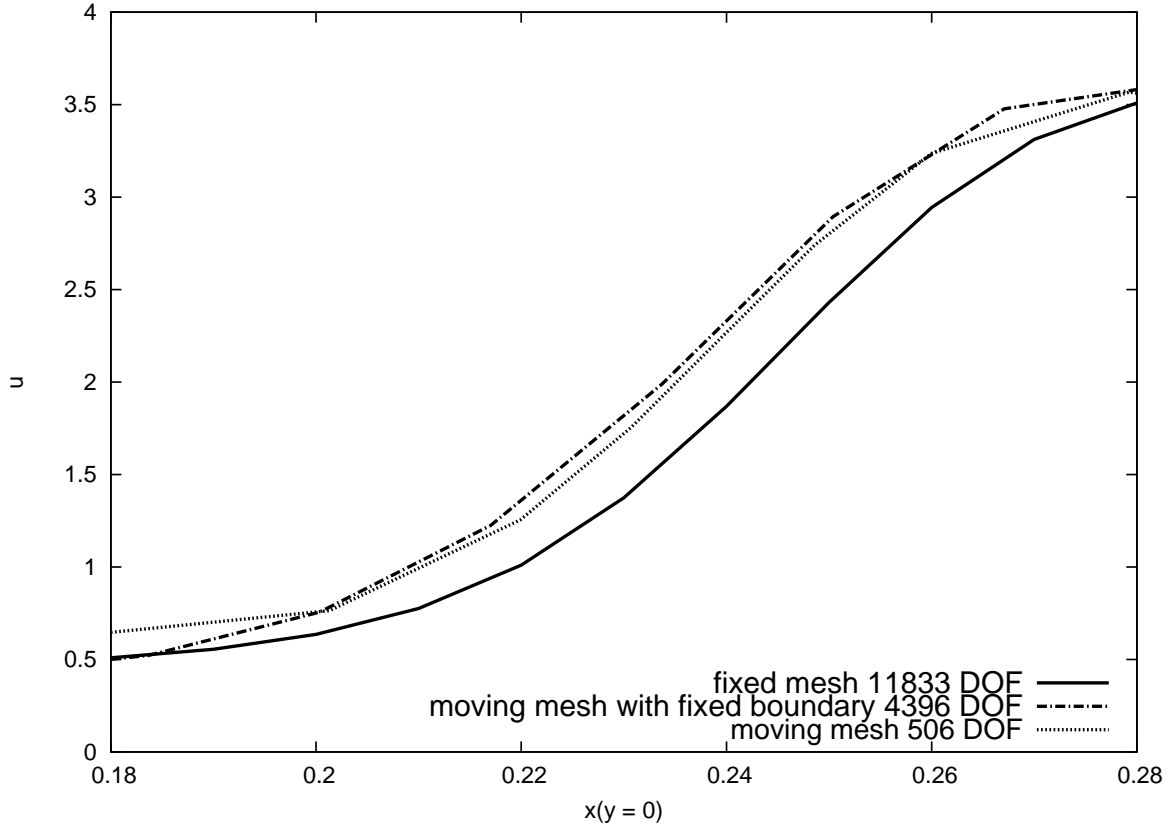
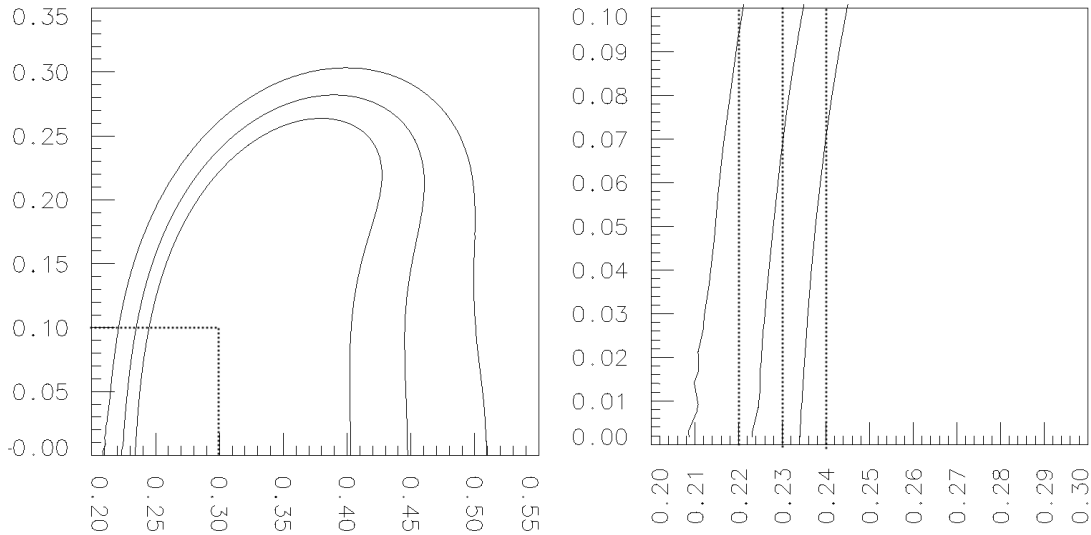Figure 5: Same to Fig. 2 except for that results are obtained from different methods.

Figure 6: Same to Fig. 3, except for that results are obtained moving mesh (4396 DOFs) where the grid points on the boundary are fixed.

3.2 The Gray-Scott model

The two-dimensional Gray-Scott model is considered in this section. We take the square domain $\Omega = [0,2.5] \times [0,2.5]$. At the boundaries, homogeneous Neumann boundary conditions are
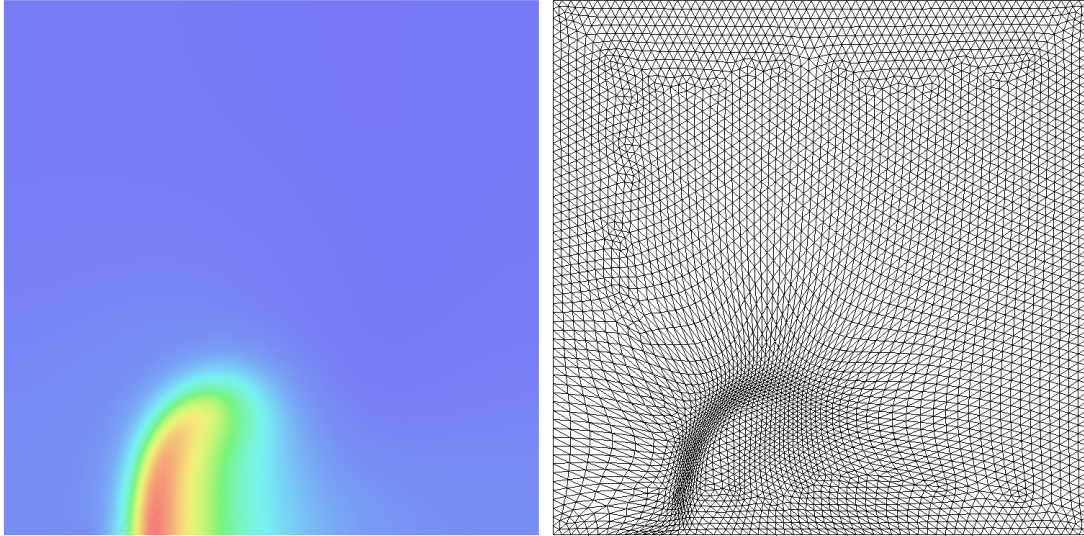
Figure 7: Same to Fig. 4 (c), except that the boundary points are not redistributed and 4396 DOFs are used when $t = 19$.

imposed both for u and v. The initial conditions are

$$
\begin{aligned}
u(x,y,0) &= 1-2v(x,y,0), \\
v(x,y,0) &= \begin{cases} \frac{1}{4}\sin^2(4\pi x)\sin^2(4\pi y), & \text{if} \quad 1 \leq x,y \leq 1.5, \\ \\ 0, & \text{elsewhere.} \end{cases}
\end{aligned}
$$

Thus the v-component consists initially of four spots near the center of the domain. In the simulation, the parameters are taken as $D_u = 8 \times 10^{-5}, D_v = 4 \times 10^{-5}, \gamma = 0.024$ and $\kappa = 0.06$ in the Gray-Scott equations (1.3). With these parameters and initial values, the initial spots will split, replicate, and form a spots pattern in the domain. This problem has also been studied in [10]. As we discussed at the beginning of this section, $\alpha = 5$ in (2.7) is used for simulation of the Gray-Scott model.

Fig. 8 shows the time evolution of the v-component by contour lines $v = 0.02 \times i, 1 \leq i \leq 14$ in the $(x,y)-$plane at various times with a time step $\Delta t = 0.01$. For clarity the solutions are only displayed for $0.5 \leq x,y \leq 2$. The moving mesh solutions agree well with those obtained with very fine uniform meshes shown in [10]. With the moving mesh strategy, the outlines of spot are depicted by grid points perfectly.

Other patterns, e.g., stripes patterns, can be obtained with different choices of parameters and initial values. More examples for the Gray-Scott model can be found in [27].

4 Conclusions

In this paper, we apply a moving finite element method to the two-dimensional Gray-Scott model and Brusselator model in weak regime for simulating the pattern formation. High quality meshes are obtained for both the spots replication and the moving wave along boundaries. Based on these high quality meshes, the two-dimensional Gray-Scott model and Brusselator model are resolved. Moreover, the importance of moving interior and boundary points simultaneously is discussed. The importance of the boundary grids redistribution is due to the solution itself. In case of the sharp layers interact with the boundary, the boundary grids is to be redistributed to improve the solution quality. Several ways of verifying the quality of
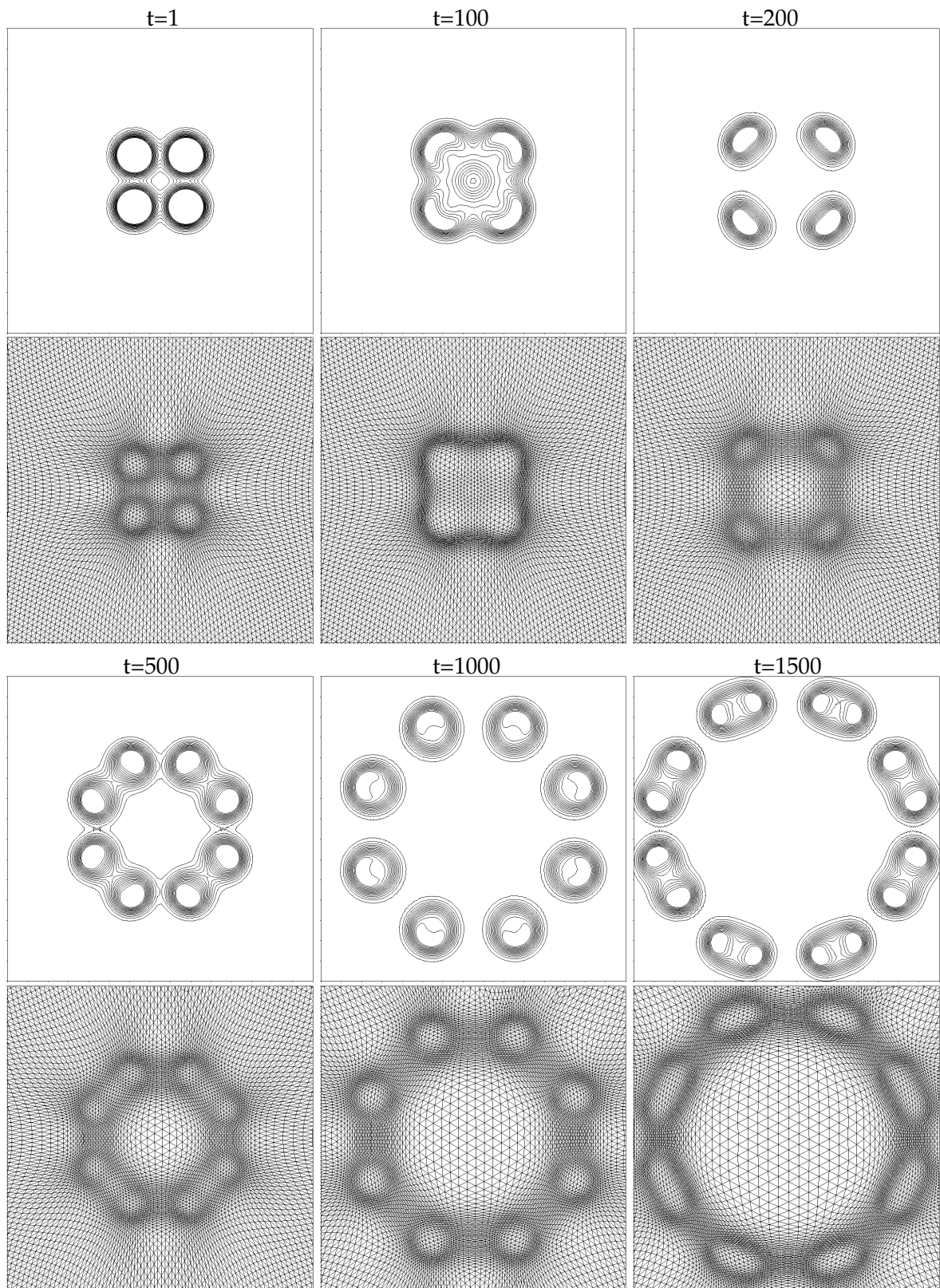
14

Figure 8: Time evolution (v-component) in the Gray-Scott model, together with the corresponding meshes. Region: $[0.5,2.0] \times [0.5,2.0]$

the numerical solutions, such as one-dimensional plot with large gradients (Figs. 1 and 2) and isolines (Figs. 3 and 6) are proposed in this work.

Acknowledgement

References

[1] A. Van Dam and P. A. Zegeling. Balanced monitoring of flow phenomena in moving mesh methods. *Commun. Comput. Phys.*, 7:138–170, 2010.

[2] A. Doelman, T. J. Kaper, and P. A. Zegeling. Pattern formation in the one-dimensional Gray-Scott model. *Nonlinearity*, 10:523–563, 1997.

[3] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12:30–39, 1972.

[4] M. I. Granero, A. Porati, and D. Zanacca. A bifurcation analysis of pattern formation in a diffusion governed morphogenetic field. *J. Math. Biol.*, 4:21–27, 1977.

[5] P. Gray and S. K. Scott. Sustained oscillations and other exotic patterns of behavior in isothermal reaction. *J. Phys. Chem.*, 59:22–32, 1985.

[6] E. Hairer, S. P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations, I. Nonstiff Problems*. Springer, Berlin, 1987.

[7] L. Harrison and D. Holloway. Order and localization in reaction-diffusion patter. *Phys. A.*, 222:210–233, 1995.

[8] D. Holloway. Reaction-diffusion theory of localized structures with applications to vertebrate organogenesis. *Ph. D. thesis, Department of Chemistry, University of British Columbia, Vancouver, Canada*, 1995.

[9] A. Hunding. Morphogen prepatterns during mitosis and cytokinesis in flattened cells: Three-dimensional Turing structures of reaction-diffusion systems in cylindrical coordinates. *J. Theoret. Biol.*, 114:571–588, 1985.

[10] W. H. Hundsdorfer and J. G. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*. Springer, 2003.

[11] D. Iron and M. J. Ward. The dynamics of multispike solutions to the one-dimensional Gierer-Meinhardt model. *SIAM J. Appl. Math.*, 62:1924–1951, 2002.

[12] D. Iron, M. J. Ward, and J. Wei. The stability of spike solutions to the one-dimensional Gierer-Meinhardt model. *Phys. D.*, 150:25–62, 2001.

[13] R. Li and W. Liu. http://dsec.pku.edu.cn/~rli.

[14] R. Li, T. Tang, and P. W. Zhang. Moving mesh methods in multiple dimensions based on harmonic maps. *J. Comput. Phys.*, 170:562–588, 2001.

[15] R. Li, T. Tang, and P. W. Zhang. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *J. Comput. Phys.*, 177:365–393, 2002.

[16] A. Madzvamuse. Time-stepping schemes for moving grid finite elements applied to reaction-diffusion systems on fixed and growing domains. *J. Comput. Phys.*, 214:239–263, 2006.

[17] A. Madzvamuse and P. K. Maini. Velocity-induced numerical solutions of reaction-diffusion systems on fixed and growing domains. *J. Comput. Phys.*, 225:100–119, 2007.

[18] A. Madzvamuse, P. K. Maini, and A. J. Wathen. A moving grid finite element method applied to a model biological pattern generator. *J. Comput. Phys.*, 190:478–500, 2003.

[19] A. Madzvamuse, P. K. Maini, and A. J. Wathen. A moving grid finite element method for the simulation of pattern generation by Turing models on growing domains. *J. Sci. Comput.*, 24:247–262, 2005.

[20] H. Meinhardt. *Models of biological pattern formation*. Academic Press, London, 1982.

[21] H. Meinhardt. *The algorithmic beauty of sea shells*. Springer-Verlag, Berlin, 1995.

[22] G. Nicolis and I. Prigogine. *Self-organization in Non-Equilibrium system: from dissipative structures to order through fluctuations*. Wiley, New York, 1977.

[23] Z. Qiao. Numerical investigations of the dynamical behaviors and instabilities for the Gierer-Meinhardt system. *Commun. Comput. Phys.*, 3:406–426, 2008.

[24] Z. Qiao, Z. Zhang, and T. Tang. An adaptive time-stepping strategy for the molecular beam epitaxy models. *SIAM J. Sci. Comput.*, 33:1395–1414, 2011.

[25] J. Schnakenberg. Simple chemical reaction systems with limit cycle behavior. *J. Theoret. Biol.*, 81:389–400, 1979.

[26] W. Sun, T. Tang, M. J. Ward, and J. Wei. Numerical challenges for resolving spike dynamics for two reaction-diffusion systems. *Studies in Applied Math.*, 111:41–84, 2003.

[27] W. Sun, M. J. Ward, and R. Russell. The slow dynamics of two-spike solutions for the Gray-Scott and Gierer-Meinhardt systems competition and oscillatory instabilities. *SIAM J. Appl. Dyn. Syst.*, 4:904–953, 2005.

[28] A. Turing. The chemical basis of morphogenesis. *Phil. Trans. R. Soc. B*, 237:32–72, 1952.

[29] M. J. Ward. Asymptotic methods for reaction-diffusion systems: past and present. *Bull. Math. Biol.*, 68:1151–1167, 2006.

[30] M. J. Ward, D. Mcinereny, and P. Houston. The dynamics and pinning o f a spike for a reaction-diffusion system. *SIAM J. Appl. Math.*, 62:1297–1328, 2002.

[31] J. Wei and M. Winter. On the two-dimensional Gierer-Meinhardt system with strong coupling. *SIAM J. Math. Anal.*, 30:1241–1263, 1999.

[32] J. Wei and M. Winter. Spikes for the two-dimensional Gierer-Meinhardt system: the weak coupling case. *J. Nonlinear Sci.*, 11:415–458, 2001.

[33] P. A. Zegeling and H. P. Kok. Adaptive moving mesh computations for reaction-diffusion systems. *J. Comput. Appl. Math.*, 168:519–528, 2004.