TAIWANESE JOURNAL OF MATHEMATICS
Vol. 23, No. 3, pp. 727–749, June 2019
DOI: 10.11650/tjm/180902

LDG Methods for Reaction-diffusion Systems with Application of Krylov Implicit Integration Factor Methods

Na An*, Chaobao Huang and Xijun Yu

Abstract. In this paper, we present an efficient fully-discrete local discontinuous Galerkin (LDG) method for nonlinear reaction-diffusion systems, which are often used as mathematical models for many physical and biological applications. We can derive numerical approximations not only for solutions but also for their gradients at the same time, while most of methods derive numerical solutions only. And due to the strict time-step restriction ($\Delta t = O(h_{\min}^2)$) of explicit schemes for stability, we introduce the implicit integration factor (IIF) method based on Krylov subspace approximation, in which the time step can be taken as $\Delta t = O(h_{\min})$. Moreover, the method allows us to compute element by element and avoid solving a global system of nonlinear algebraic equations as the standard implicit schemes do, which can reduce the computational cost greatly. Numerical experiments about the reaction-diffusion equations with exact solutions and the well-studied Schnakenberg system are conducted to illustrate the accuracy, capability and advantages of the method.

1. Introduction

Mathematical models for many physical and biological applications are of the following form

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbb{B}\Delta\mathbf{u} + \mathbf{F}(\mathbf{u}), \tag{1.1}$$

where $\mathbf{u} \in \mathbf{R}^2$ represents a group of physical or biological species, $\mathbb{B} = \left(\begin{smallmatrix} \beta_u & 0 \\ 0 & \beta_v \end{smallmatrix}\right)$ is the diffusion constant matrix, $\mathbf{F}(\mathbf{u})$ describes the chemical or biological reactions and Δ is the Laplacian operator. Let $\Omega \subset \mathbf{R}^2$ be an open, bounded domain and the system is defined on $\Omega \times [0,T]$. Turing-type models [30] are the typical examples, which include the Schnakenberg model [28], the chloride-iodide-malonic acid reactive model [19], the Gierer-Meinhardt model [9], the Gray-Scott model [11] and so on.

In order to obtain accurate numerical solutions to the nonlinear reaction-diffusion system (1.1), which are usually highly stiff, many numerical schemes have been studied.

Received January 4, 2018; Accepted September 9, 2018.

Communicated by Suh-Yuh Yang.

2010 *Mathematics Subject Classification.* 35K57.

Key words and phrases. local discontinuous Galerkin, implicit integration factor, Krylov subspace approximation, unstructured meshes, nonlinear reaction-diffusion systems.

*Corresponding author.

These include the finite difference methods, spectral methods and finite element methods. Due to the complex shapes of actual problems, the system (1.1) are often defined on irregular geometrical domains. However, the extension of the finite difference or spectral methods to complicated and irregular domains is not at all trivial. On the contrary, finite element methods on unstructured meshes are powerful means for handling the complicated domain geometries [17]. In the references [13, 22, 26, 29], continuous Galerkin (CG) finite element methods were used to solve the reaction-diffusion systems on complex domains.

Recently, discontinuous Galerkin (DG) methods have attracted more attention since they possess several advantages: the flexibility for easy hp-adaptivity including changes of approximation orders between adjacent elements and allowing the existence of hanging nodes, the compactness and efficient parallel implementation. The Cheng-Shu DG method proposed in [6] was used to solve the system (1.1) in both [33] with the application of Strang type symmetrical operator splitting technique and [5] with Krylov implicit integration factor (IIF) method for temporal discretization. And Zhang et al. [32] introduced the direct discontinuous Galerkin (DDG) method proposed in [21] to discrete the space-derivative of the system (1.1). However, the above two kinds of DG methods both only derived the numerical solutions.

While the gradients of solutions are usually very important in practical problems, for example, the fluxes in fluid flow. And it is expected to derive the numerical approximations for both solutions and their gradients at the same time. So, in this paper, we choose to pursue another kind of DG method, the local discontinuous Galerkin (LDG) method. The LDG method is a generalization of the DG method which was designed for solving hyperbolic conservation laws containing only first order spatial derivatives in [25]. The LDG method aims at solving equations containing higher than first order spatial derivatives. It was proposed by Cockburn and Shu [8] to solve a convection-diffusion equation containing a second order spatial derivative, motivated by the successful numerical experiments of Bassi and Rebay [1] for the compressible Navier-Stokes equations. And the reason why it is called *local* DG methods is illustrated in [8]: It is to emphasize the difference between this method which uses discontinuous finite elements for space discretizations with the so-called DG method for parabolic problems introduced by Jamet [15] which has approximate solution discontinuous only in time, not in space. The LDG method need to rewrite the equations with higher order derivatives as a first order system by introducing an auxiliary variable (gradient), which can be locally solved in terms of the original variable (solution) at element level. By a careful choice of the rewriting, nonlinear stability can be obtained even without slope limiters, just like the DG method in the purely hyperbolic case [16]. The analysis of the LDG method can be see in [8, 31], which shows sub-optimal $O(h^k)$ L^2 error estimates for general triangulations and piecewise polynomials of degree k. However, in the DG method, approximations of the derivatives of the discontinuous approximate solution are not obtained directly and need to use a projection into suitable finite elements spaces. The projection requires the inversion of global mass matrices, which in [3, 4] are "lumped" in order to maintain the high parallelizability of the method. And the mass lumping needed to enforce the full parallelizability of the method could cause a degradation of the formal order of accuracy if polynomials of higher than one degree were used. But it is not the case for the LDG method, in which the original idea of the DG method is applied to both the solution and its gradient that are now independent unknowns. And the resulting LDG method is a highly parallelizable method with high-order accuracy.

The temporal discretization can be carried out using the following schemes: explicit, semi-implicit, implicit-explicit schemes, the exponential time-differencing scheme [18] and more recently the Krylov IIF method [5] which is based on the IIF method [24]. However, considering of the the large number of degrees of freedom caused by LDG spatial discretization which introduces another auxiliary variable, the computational cost can be significant for implicit methods which needs to solve a global system of nonlinear equations. It is known to all that, like any other DG methods, with an appropriate temporal discretization such as the explicit schemes and IIF schemes, the LDG discrete formulation can be solved element by element at every time step. But explicit methods require a severe time-step restriction ($\Delta t = O(h_{\min}^2)$). In this paper we adopt the Krylov IIF method which is an IIF method based on Krylov subspace approximations. Using the IIF method to discretize time can relax the time-step to $\Delta t = O(h_{\min})$ and maintain the property of LDG method that the resulting algebraic system can be solved element by element. However, by applying the IIF method, we need to evaluate the product of the matrix exponential and a vector. And the Krylov subspace method can approximate it efficiently. Therefore, we adopt the Krylov IIF method for temporal discretization. Recently, the semi-implicit spectral deferred correction method is proposed in [12], by using which can achieve the same effect as the IIF method, and we will study it in the future.

The rest of this paper is organized as follows. In Section 2, we present the LDG formulation for spacial discretization, eliminate the auxiliary variable $\mathbf{q}_h$ at the element level, and then apply the Krylov IIF methods to discretize the resulting ordinary differential equations (ODEs) which has only u_h as unknown. In Section 3, numerical experiments are conducted to confirm that the method has its advantages and is efficient for not only the linear and nonlinear test equations with exact solutions but also for the reaction-diffusion systems, for example, the classic Schnakenberg model.

2. The LDG formulation and application of Krylov IIF methods

In this section, we take the scalar case of (1.1) as an example to illustrate the process of combining LDG methods with Krylov IIF schemes to solve the nonlinear reaction-diffusion systems.

2.1. The LDG methods for spatial discretization

For the sake of simplicity and convenience, in the following analysis, we consider the scalar case of (1.1):

$$\frac{\partial u}{\partial t} = \beta \Delta u + F(u), \tag{2.1}$$

and now $\mathbb{B} = \beta$ is a diffusion constant. We take the boundary condition as

$$\frac{\partial u(x,y,t)}{\partial \mathbf{n}} = \mathbf{g}\cdot\mathbf{n} \quad \text{in } \partial\Omega \times (0,T) \tag{2.2}$$

with an appropriate initial condition

$$u(x,y,0) = u_0(x,y) \quad \text{in } \Omega,$$

where $\mathbf{n}$ is the outward unit normal to $\partial\Omega$. The numerical formulation for the scalar case can be straight forwardly extended to solve the system case component by component.

To develop a LDG formulation, we first rewrite the problem (2.1) as a system of first order differential equations

$$\mathbf{q} - \beta\nabla u = 0, \quad \frac{\partial u}{\partial t} - \nabla\cdot\mathbf{q} = F(u).$$

Let $\mathcal{T}_h = \{K\}$ be a regular triangulation of Ω with $h = \max\{h_K \mid h_K \text{ is the diameter of } K, \forall K \in \mathcal{T}_h\}$ the mesh size. $\mathcal{E}_h$ denotes the collection of all edges in $\mathcal{T}_h$. $\mathcal{E}_h^\circ$ and $\mathcal{E}_h^b$ are the sets of interior edges and boundary edges, respectively.

To obtain the weak formulation, we multiply the above two equations by arbitrary, smooth test functions $\mathbf{w}$, v, respectively, integrate over each element $K \in \mathcal{T}_h$, and apply divergence and Green's theorems to obtain

$$\begin{aligned}\int_K \mathbf{q}\cdot\mathbf{w}\,d\mathbf{x} + \int_K u\nabla\cdot(\beta\mathbf{w})\,d\mathbf{x} - \int_{\partial K} u\beta\mathbf{w}\cdot\mathbf{n}_K\,ds = 0,\\ \int_K \frac{\partial u}{\partial t} v\,d\mathbf{x} + \int_K \mathbf{q}\cdot\nabla v\,d\mathbf{x} - \int_{\partial K}\mathbf{q}\cdot\mathbf{n}_K v\,ds = \int_K F(u)v\,d\mathbf{x},\end{aligned}$$

where $\mathbf{n}_K$ is the outward unite normal to ∂K. Note that the above equations are well defined for any functions $(\mathbf{q}, u)$ and $(\mathbf{w}, v)$ in $\mathbf{W}\times V$, where

$$\begin{aligned}\mathbf{W} &= \{\mathbf{w}\in (L^2(\Omega))^2 : \mathbf{w}|_K \in (H^1(K))^2, \forall K\in\mathcal{T}_h\},\\ V &= \{v \in L^2(\Omega) : v|_K \in H^1(K), \forall K \in \mathcal{T}_h\}.\end{aligned}$$

Define the finite element space as follows:

$$\begin{aligned} V_h &= \{v_h \in L^2(\Omega) : v_h|_K \in P^1(K), \forall\, K \in \mathcal{T}_h\}, \\ \mathbf{W}_h &= \{\mathbf{w}_h \in (L^2(\Omega))^2 : \mathbf{w}_h|_K \in (P^1(K))^2, \forall\, K \in \mathcal{T}_h\}, \end{aligned}$$

where $P^1(K)$ denotes the linear polynomials on K. Then the LDG formulation can be defined as: finding $(\mathbf{q}_h, u_h) \in \mathbf{W}_h \times V_h$ such that, for $(\mathbf{w}_h, v_h) \in \mathbf{W}_h \times V_h$ and $K \in \mathcal{T}_h$,

$$\begin{aligned} &\int_K \mathbf{q}_h \cdot \mathbf{w}_h \, d\mathbf{x} + \int_K u_h \nabla \cdot (\beta \mathbf{w}_h)\, d\mathbf{x} - \int_{\partial K} \widehat{u_h} \beta \mathbf{w}_h \cdot \mathbf{n}_K \, ds = 0, \\ &\int_K \frac{\partial u_h}{\partial t} v_h \, d\mathbf{x} + \int_K \mathbf{q}_h \cdot \nabla v_h \, d\mathbf{x} - \int_{\partial K} \widehat{\mathbf{q}_h} \cdot \mathbf{n}_K \, v_h \, ds = \int_K F(u_h) v_h \, d\mathbf{x}, \end{aligned} \tag{2.3}$$

where the quantities $\widehat{u_h}$ and $\widehat{\mathbf{q}_h}$ are the so-called numerical fluxes and chosen as [2],

$$\forall\, e \in \mathcal{E}_h^\circ, \quad \widehat{u_h}|_e = \{u_h\} + \mathbf{C}_{12} \cdot [u_h] \quad \text{and} \quad \widehat{\mathbf{q}_h}|_e = \{\mathbf{q}_h\} - C_{11}[u_h] - \mathbf{C}_{12}[\mathbf{q}_h].$$

The stability parameter $C_{11} > 0$ and is taken to be $O(h_e^{-1})$ to enhance the accuracy of the LDG method. The auxiliary vector parameter $\mathbf{C}_{12}$ are generally chosen as $\mathbf{C}_{12} \cdot \mathbf{n}_e = O(1)$ on each edge e. The choice of these two parameters is stated in the reference [2,7]. We shall emphasize here that the jump $[\,\cdot\,]$ and the mean value $\{\,\cdot\,\}$ take the standard definitions in discontinuous Galerkin methods, that is, if we assume $e = \partial K_1 \cap \partial K_2$, then $[v]$ and $[\mathbf{w}]$ on e are defined by

$$[v] = v_1 \mathbf{n}_1 + v_2 \mathbf{n}_2, \quad [\mathbf{w}] = \mathbf{w}_1 \cdot \mathbf{n}_1 + \mathbf{w}_2 \cdot \mathbf{n}_2,$$

the mean value $\{\,\cdot\,\}$ are defined by

$$\{v\} = \frac{1}{2}(v_1 + v_2), \quad \{\mathbf{w}\} = \frac{1}{2}(\mathbf{w}_1 + \mathbf{w}_2),$$

where $(v_i, \mathbf{w}_i)$ are the traces of $(v, \mathbf{w})$ on e from the interior of K_i, $i = 1, 2$ and $\mathbf{n}_i$ is the outward unit normal to ∂K_i.

The boundary conditions (2.2) are imposed through the following definition of the numerical fluxes

$$\forall\, e \in \mathcal{E}_h^b, \quad \widehat{u_h}|_e = u_h \quad \text{and} \quad \widehat{\mathbf{q}_h}|_e = \mathbf{g}.$$

By use of basis functions, we can express $\mathbf{q}_h = (q_{h,x}, q_{h,y})$ and u_h as

$$q_{h,l} = \sum_{i=1}^{3} q_{l,i} \phi_i = \Phi^T \overline{\mathbf{q}_l}, \quad l = x, y, \quad u_h = \sum_{i=1}^{3} u_i \phi_i = \Phi^T \overline{\mathbf{u}},$$

where $\Phi = (\phi_1, \phi_2, \phi_3)^T$ are basis functions, $\overline{\mathbf{q}_l} = (q_{l,1}, q_{l,2}, q_{l,3})^T$, $l = x, y$ and $\overline{\mathbf{u}} = (u_1, u_2, u_3)^T$ are degrees of freedom.

By the similar means as shown on p. 117 and p. 134 in [20], we derive that, for an interior element K,

$$\begin{aligned}
&\left(\int_K \Phi\Phi^T\,d\mathbf{x}\right)\overline{\mathbf{q}}+\left(\int_K \nabla(\beta\phi)\phi^T\,d\mathbf{x}\right)\overline{\mathbf{u}}-\left(\int_{\partial K}\beta(\tfrac{1}{2}+C_{12})\Phi\Phi^T\mathbf{n}\,ds\right)\overline{\mathbf{u}}\\
&\quad-\left(\int_{\partial K}\beta(\tfrac{1}{2}-C_{12})\Phi\Phi^T_{(NB)}\mathbf{n}\,ds\right)\overline{\mathbf{u}}_{(NB)}=0,\\
&\left(\int_K \Phi\Phi^T\,d\mathbf{x}\right)\frac{\partial\overline{\mathbf{u}}}{\partial t}+\left(\int_K (\nabla\phi)\phi^T\,d\mathbf{x}\right)\overline{\mathbf{q}}-\left(\int_{\partial K}(\tfrac{1}{2}-C_{12})\Phi\Phi^T\mathbf{n}\,ds\right)\overline{\mathbf{q}}\\
&\quad-\left(\int_{\partial K}(\tfrac{1}{2}+C_{12})\Phi\Phi^T_{(NB)}\mathbf{n}\,ds\right)\overline{\mathbf{q}}_{(NB)}+\left(\int_{\partial K}C_{11}\Phi\Phi^T\,ds\right)\overline{\mathbf{u}}\\
&\quad-\left(\int_{\partial K}C_{11}\Phi\Phi^T_{(NB)}\,ds\right)\overline{\mathbf{u}}_{(NB)}=\int_K F(u_h)\Phi\,d\mathbf{x},
\end{aligned}$$

where $\mathbf{n}=\mathbf{n}_K$ and $\mathbf{n}=(n_x,n_y)$ to simplify notations; $\overline{\mathbf{q}}=(\overline{\mathbf{q}_x},\overline{\mathbf{q}_y})$ and subscript (NB) denotes the quantities belonging to the adjacent elements, see Figure 2.1. We also use the relations that $C_{12}=\mathbf{C}_{12}\cdot\mathbf{n}$ and $\mathbf{n}_{(NB)}=-\mathbf{n}$.

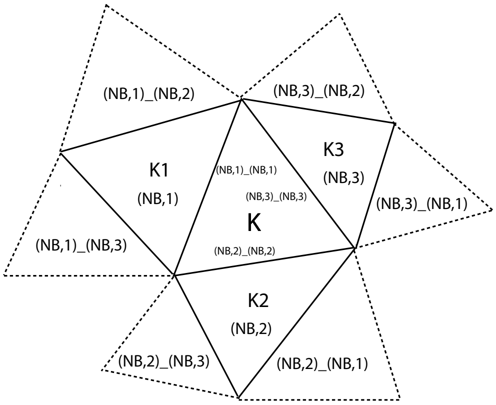


Figure 2.1: The sketch of triangular element K and its neighbor elements.

By calculation, we have the final matrix equation for the interior element K,

$$\begin{aligned}
&\begin{pmatrix}0&0&0\\0&0&0\\0&0&\mathbb{M}\end{pmatrix}\begin{pmatrix}(\overline{\mathbf{q}_x})_t\\(\overline{\mathbf{q}_y})_t\\(\overline{\mathbf{u}})_t\end{pmatrix}+\begin{pmatrix}\mathbb{M}&0&\mathbb{H}_x\\0&\mathbb{M}&\mathbb{H}_y\\\mathbb{J}_x&\mathbb{J}_y&0\end{pmatrix}\begin{pmatrix}\overline{\mathbf{q}_x}\\\overline{\mathbf{q}_y}\\\overline{\mathbf{u}}\end{pmatrix}+\sum_{i=1}^{3}\begin{pmatrix}0&0&\mathbb{H}_{x,i}\\0&0&\mathbb{H}_{y,i}\\\mathbb{J}_{x,i}&\mathbb{J}_{y,i}&\mathbb{G}_{u,i}\end{pmatrix}\begin{pmatrix}\overline{\mathbf{q}_x}\\\overline{\mathbf{q}_y}\\\overline{\mathbf{u}}\end{pmatrix}\\
&+\sum_{i=1}^{3}\begin{pmatrix}0&0&\mathbb{H}_{x,B,i}\\0&0&\mathbb{H}_{y,B,i}\\\mathbb{J}_{x,B,i}&\mathbb{J}_{y,B,i}&\mathbb{G}_{u,B,i}\end{pmatrix}\begin{pmatrix}\overline{\mathbf{q}_x}\\\overline{\mathbf{q}_y}\\\overline{\mathbf{u}}\end{pmatrix}_{(NB,i)}=\begin{pmatrix}\mathbf{0}\\\mathbf{0}\\\mathbf{S}_u\end{pmatrix},
\end{aligned}$$

where $(\overline{\mathbf{u}})_t = \partial\overline{\mathbf{u}}/\partial t$ denotes time derivative and the matrices are calculated as follows: for $m, n = 1, 2, 3$,

$$\begin{aligned}
\mathbb{H}_{l,mn} &= \int_K \frac{\partial(\beta\phi_m)}{\partial l}\phi_n\,d\mathbf{x}, & \mathbb{J}_{l,mn} &= \int_K \frac{\partial\phi_m}{\partial l}\phi_n\,d\mathbf{x},\ l = x, y,\\
\mathbb{M}_{mn} &= \int_K \phi_m\phi_n\,d\mathbf{x}, & \mathbf{S}_{u,m} &= \int_K F(u_h)\phi_m\,d\mathbf{x},\\
\mathbb{G}_{u,i,mn} &= \int_{(\partial K)_i} C_{11}\phi_m\phi_n\,d\mathbf{x}, & \mathbb{G}_{u,B,i,mn} &= -\int_{(\partial K)_i} C_{11}\phi_m(\phi_n)_{(NB,i)}\,d\mathbf{x},\ i = 1, 2, 3,\\
\mathbb{H}_{l,i,mn} &= -\int_{(\partial K)_i} (\tfrac{1}{2} + C_{12})\beta\phi_m\phi_n n_l\,ds, & \mathbb{H}_{l,B,i,mn} &= -\int_{(\partial K)_i} (\tfrac{1}{2} - C_{12})\beta\phi_m(\phi_n)_{(NB,i)}n_l\,ds,\\
\mathbb{J}_{l,i,mn} &= -\int_{(\partial K)_i} (\tfrac{1}{2} - C_{12})\phi_m\phi_n n_l\,ds, & \mathbb{J}_{l,B,i,mn} &= -\int_{(\partial K)_i} (\tfrac{1}{2} + C_{12})\phi_m(\phi_n)_{(NB,i)}n_l\,ds,
\end{aligned}$$

with $\partial K = \bigcup_{i=1}^3 (\partial K)_i$, $(\partial K)_i$ denoting the common edge between the element K and its adjacent element K_i. The notation $\mathbb{H}_{l,mn}$ denotes the (mth row, nth column) element of the matrix $\mathbb{H}_l$, and $\mathbb{H}_{l,B,i,mn}$ is the (mth row, nth column) element of the matrix $\mathbb{H}_{l,B,i}$. The meanings of other notations are the same.

In our numerical experiments, the above integrations are carried out by numerical integration formulas. For the integrations on the element K, we employ the numerical integration formula with midpoints at three edges of the triangular element

$$\int_K \phi\,d\mathbf{x} \approx \frac{|K|}{3}(\phi_{M_1} + \phi_{M_2} + \phi_{M_3}),$$

where ϕ_{M_1}, ϕ_{M_2}, ϕ_{M_3} are values of midpoints at three edges of the element K and $|K|$ is the area of K. Its algebraic accuracy is 2, which is precise for quadratic polynomials, linear polynomials and constants. For the integrations on the edge ∂K, we employ the Gauss-Legendre quadrature formula with three Gauss points, whose algebraic accuracy is 5.

The above derivation is aimed at interior elements, where we assume that K is an interior element. Now, we consider the boundary elements. If K is an boundary element, there exists at least one boundary edge. If the edge $(\partial K)_i$ sharing by K and K_i is a boundary edge, that is to say, K_i does not exist, then the above matrices related to the edge $(\partial K)_i$ can have some differences and be more simple. We only need to insert the numerical fluxes on boundary edges instead of the numerical fluxes on interior edges into (2.3). Then, by the same way as that derived the matrices for the interior edges, we can derive the matrices for the boundary edge $(\partial K)_i$,

$$\mathbb{J}_{l,i,mn} = 0, \quad \mathbb{G}_{u,i,mn} = 0, \quad \mathbb{H}_{l,i,mn} = -\int_{(\partial K)_i} \beta\phi_m\phi_n n_l\,ds, \quad l = x, y,$$

where we assume that the problem has the no-flux boundary condition. In addition, for an boundary edge $(\partial K)_i$, the quantities $\mathbb{G}_{u,B,i}$, $\mathbb{H}_{l,B,i}$, $\mathbb{J}_{l,B,i}$, $l = x, y$, do not exist and are not needed.

To facilitate computations, we rewrite the above matrix form into two separate matrix equations

$$\begin{pmatrix}\mathbb{M} & 0\\ 0 & \mathbb{M}\end{pmatrix}\begin{pmatrix}\overline{\mathbf{q}_x}\\ \overline{\mathbf{q}_y}\end{pmatrix}+\begin{pmatrix}\mathbb{H}_x+\mathbb{H}_X\\ \mathbb{H}_y+\mathbb{H}_Y\end{pmatrix}\overline{\mathbf{u}}+\sum_{i=1}^{3}\begin{pmatrix}\mathbb{H}_{x,B,i}\\ \mathbb{H}_{y,B,i}\end{pmatrix}\overline{\mathbf{u}}_{(NB,i)}=\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \tag{2.4}$$

$$\begin{aligned}&\mathbb{M}\frac{d\overline{\mathbf{u}}}{dt}+\mathbb{G}_U\overline{\mathbf{u}}+\sum_{i=1}^{3}\mathbb{G}_{u,B,i}\overline{\mathbf{u}}_{(NB,i)}+(\mathbb{J}_x+\mathbb{J}_X)\overline{\mathbf{q}_x}+(\mathbb{J}_y+\mathbb{J}_Y)\overline{\mathbf{q}_y}\\ &\quad+\sum_{i=1}^{3}\sum_{l=x,y}\mathbb{J}_{l,B,i}(\overline{\mathbf{q}_l})_{(NB,i)}=\mathbf{S}_u,\end{aligned} \tag{2.5}$$

where we use the notifications

$$\mathbb{H}_X=\sum_{i=1}^{3}\mathbb{H}_{x,i},\quad \mathbb{H}_Y=\sum_{i=1}^{3}\mathbb{H}_{y,i},\quad \mathbb{G}_U=\sum_{i=1}^{3}\mathbb{G}_{u,i},\quad \mathbb{J}_X=\sum_{i=1}^{3}\mathbb{J}_{x,i},\quad \mathbb{J}_Y=\sum_{i=1}^{3}\mathbb{J}_{y,i}.$$

If the degrees of freedom are taken as the values of midpoints at three edges of K, then $\mathbb{M}\approx\frac{|K|}{3}\mathbb{I}$ with $|K|$ the area of the element K and $\mathbb{I}$ the unit matrix. $\mathbf{S}_u\approx\frac{|K|}{3}\mathbf{F}(\overline{\mathbf{u}})$ with $\mathbf{F}(\overline{\mathbf{u}})=\begin{pmatrix}F(u_1)\\ F(u_2)\\ F(u_3)\end{pmatrix}$. At this moment, (2.4) can be rewritten as, $\forall K\in\mathcal{T}_h$,

$$\begin{aligned}\overline{\mathbf{q}_x}&=-\frac{3}{|K|}\left((\mathbb{H}_x+\mathbb{H}_X)\overline{\mathbf{u}}+\sum_{i=1}^{3}\mathbb{H}_{x,B,i}\overline{\mathbf{u}}_{(NB,i)}\right)\\ \overline{\mathbf{q}_y}&=-\frac{3}{|K|}\left((\mathbb{H}_y+\mathbb{H}_Y)\overline{\mathbf{u}}+\sum_{i=1}^{3}\mathbb{H}_{y,B,i}\overline{\mathbf{u}}_{(NB,i)}\right),\end{aligned} \tag{2.6}$$

which is an advantage of LDG method that the auxiliary variable can be expressed by original variable locally.

As a special case of (2.6), for the adjacent elements K_i, $i=1,2,3$, we have

$$\begin{aligned}(\overline{\mathbf{q}_x})_{(NB,i)}&=-\frac{3}{|K_i|}\left((\mathbb{H}_x^{(NB,i)}+\mathbb{H}_X^{(NB,i)})\overline{\mathbf{u}}_{(NB,i)}+\sum_{j=1}^{3}\mathbb{H}_{x,B,j}^{(NB,i)}\overline{\mathbf{u}}_{(NB,i)_{(NB,j)}}\right)\\ (\overline{\mathbf{q}_y})_{(NB,i)}&=-\frac{3}{|K_i|}\left((\mathbb{H}_y^{(NB,i)}+\mathbb{H}_Y^{(NB,i)})\overline{\mathbf{u}}_{(NB,i)}+\sum_{j=1}^{3}\mathbb{H}_{y,B,j}^{(NB,i)}\overline{\mathbf{u}}_{(NB,i)_{(NB,j)}}\right),\end{aligned}$$

where (NB,i), $i=1,2,3$, denote the quantities belonging to K's adjacent elements K_i, see Figure 2.1. Similarly, $(NB,i)_{(NB,j)}$, $j=1,2,3$, denote the adjacent elements K_{ij} of K_i, respectively. And $\overline{\mathbf{u}}_{(NB,i)_{(NB,j)}}=\overline{\mathbf{u}}$ when $i=j$.

Substituting the above equations and (2.6) into (2.5), we derive a system including original variable only,

$$\frac{d\overline{\mathbf{u}}}{dt}=-\frac{3}{|K|}\left(\mathbb{N}\overline{\mathbf{u}}+\sum_{i=1}^{3}\overline{\mathbb{N}}_i\overline{\mathbf{u}}_{(NB,i)}+\sum_{i=1}^{3}\sum_{j=1}^{3}\widetilde{\mathbb{N}}_{ij}\overline{\mathbf{u}}_{(NB,i)_{(NB,j)}}\right)+\mathbf{F}(\overline{\mathbf{u}}), \tag{2.7}$$

where

$$\begin{aligned}\mathbb{N} &= \mathbb{G}_U - \frac{3}{|K|}\left((\mathbb{J}_x + \mathbb{J}_X)(\mathbb{H}_x + \mathbb{H}_X) + (\mathbb{J}_y + \mathbb{J}_Y)(\mathbb{H}_y + \mathbb{H}_Y)\right),\\ \overline{\mathbb{N}}_i &= \mathbb{G}_{u,B,i} - \frac{3}{|K|}\left((\mathbb{J}_x + \mathbb{J}_X)\mathbb{H}_{x,B,i} + (\mathbb{J}_y + \mathbb{J}_Y)\mathbb{H}_{y,B,i}\right)\\ &\quad - \frac{3}{|K_i|}\left(\mathbb{J}_{x,B,i}(\mathbb{H}_x^{(NB,i)} + \mathbb{H}_X^{(NB,i)}) + \mathbb{J}_{y,B,i}(\mathbb{H}_y^{(NB,i)} + \mathbb{H}_Y^{(NB,i)})\right),\\ \widetilde{\mathbb{N}}_{i,j} &= -\frac{3}{|K_i|}\left(\mathbb{J}_{x,B,i}\mathbb{H}_{x,B,j}^{(NB,i)} + \mathbb{J}_{y,B,i}\mathbb{H}_{y,B,j}^{(NB,i)}\right).\end{aligned}$$

2.2. IIF methods based on Krylov subspace approximation for time discretization

Assembling (2.7) over all of the elements in $\mathcal{T}_h$, we derive the global system of ODEs,

$$\frac{d\overline{\mathbf{U}}}{dt} = \mathbb{A}\overline{\mathbf{U}} + \mathbf{F}(\overline{\mathbf{U}}), \tag{2.8}$$

where $\overline{\mathbf{U}} = (\overline{\mathbf{u}}_1^T, \overline{\mathbf{u}}_2^T, \ldots, \overline{\mathbf{u}}_{N_e}^T)^T$, $\mathbf{F}(\overline{\mathbf{U}}) = (\mathbf{F}(\overline{\mathbf{u}}_1)^T, \mathbf{F}(\overline{\mathbf{u}}_2)^T, \ldots, \mathbf{F}(\overline{\mathbf{u}}_{N_e})^T)^T$, $\overline{\mathbf{u}}_j$ is the degrees of freedom on element K_j, $j = 1, 2, \ldots, N_e$, and here N_e denotes the number of triangular elements. The $3N_e \times 3N_e$ global matrix $\mathbb{A}$ is sparse and formulated element by element according to (2.7). Each element $K \in \mathcal{T}_h$ contributes to the global matrix $\mathbb{A}$ with no more than ten nonzero 3×3 block matrices at three corresponding rows.

Then we apply the second order IIF scheme proposed in [24] for the time evolution in the ODE system (2.8) and obtain the full discrete form

$$\overline{\mathbf{U}}^{n+1} = e^{\mathbb{A}\Delta t}\left(\overline{\mathbf{U}}^n + \frac{\Delta t}{2}\mathbf{F}(\overline{\mathbf{U}}^n)\right) + \frac{\Delta t}{2}\mathbf{F}(\overline{\mathbf{U}}^{n+1}), \tag{2.9}$$

where n is the time level, $t_{n+1} = t_n + \Delta t$ and $\overline{\mathbf{U}}^n = \overline{\mathbf{U}}(t_n)$.

Now, we given the specific process to derive the above second order IIF scheme. First, multiplying (2.8) by the integrating factor $e^{-\mathbb{A}t}$, we integrate the equation over one time step from t_n to $t_{n+1} \equiv t_n + \Delta t$ to get

$$\overline{\mathbf{U}}(t_{n+1}) = \overline{\mathbf{U}}(t_n)e^{\mathbb{A}\Delta t} + e^{\mathbb{A}\Delta t}\int_0^{\Delta t} e^{-\mathbb{A}\tau}\mathbf{F}(\overline{\mathbf{U}}(t_n + \tau))\, d\tau.$$

To construct a scheme of second order truncation, we define $\overline{\mathbf{U}}^n$ as the numerical solution for $\overline{\mathbf{U}}(t_n)$ and approximate $g(\tau) = e^{-\mathbb{A}\tau}\mathbf{F}(\overline{\mathbf{U}}(t_n + \tau))$ with a first order Lagrange polynomial $L_1(\tau)$ with interpolation points at t_{n+1}, t_n, i.e., $g(0) = \mathbf{F}(\overline{\mathbf{U}}^n)$, $g(\Delta t) = e^{-\mathbb{A}\Delta t}\mathbf{F}(\overline{\mathbf{U}}^{n+1})$, then the second order approximation to $g(\tau)$ is

$$\begin{aligned}L_1(\tau) &= \frac{1}{\Delta t}\left(\mathbf{F}(\overline{\mathbf{U}}^n)(\Delta t - \tau) + e^{-\mathbb{A}\Delta t}\mathbf{F}(\overline{\mathbf{U}}^{n+1})\tau\right)\\ &= \mathbf{F}(\overline{\mathbf{U}}^n) + \frac{1}{\Delta t}\left(e^{-\mathbb{A}\Delta t}\mathbf{F}(\overline{\mathbf{U}}^{n+1}) - \mathbf{F}(\overline{\mathbf{U}}^n)\right)\tau, \quad 0 \le \tau \le \Delta t.\end{aligned}$$

Now the above equation can be discretized as

$$\overline{\mathbf{U}}^{n+1} = e^{\mathbb{A}\Delta t}\overline{\mathbf{U}}^{n} + e^{\mathbb{A}\Delta t}\int_0^{\Delta t} L_1(\tau)\, d\tau$$

with

$$\begin{aligned}\int_0^{\Delta t} L_1(\tau)\, d\tau &= \mathbf{F}(\overline{\mathbf{U}}^{n})\Delta t + \frac{\Delta t}{2}\left(e^{-\mathbb{A}\Delta t}\mathbf{F}(\overline{\mathbf{U}}^{n+1}) - \mathbf{F}(\overline{\mathbf{U}}^{n})\right)\\ &= \frac{\Delta t}{2}\mathbf{F}(\overline{\mathbf{U}}^{n}) + \frac{\Delta t}{2}e^{-\mathbb{A}\Delta t}\mathbf{F}(\overline{\mathbf{U}}^{n+1}).\end{aligned}$$

So we obtain the second order discrete scheme is of the form (2.9).

In the full discrete form (2.9), when we compute $\overline{\mathbf{U}}^{n+1}$, the vector $\mathbf{Q} = e^{\mathbb{A}\Delta t}(\overline{\mathbf{U}}^{n} + \frac{\Delta t}{2}\mathbf{F}(\overline{\mathbf{U}}^{n}))$ is a known quantity related to the earlier time level and can be computed in advance. Although the matrix $\mathbb{A}$ is sparse, the exponential matrix $e^{\mathbb{A}\Delta t}$ is dense. Computing $e^{\mathbb{A}\Delta t}$ directly is not practical. And here, we do not need the the exponential matrix $e^{\mathbb{A}\Delta t}$ itself, but only the product of the matrix exponential and the vector $\overline{\mathbf{U}}^{n} + \frac{\Delta t}{2}\mathbf{F}(\overline{\mathbf{U}}^{n})$. So the Krylov subspace approximation shown in [5] is applied to evaluate it, which is a good choice in terms of both accuracy and efficiency.

Therefore, the nonlinear system at t_{n+1} is decoupled from the diffusion with a simple form

$$\overline{\mathbf{U}}^{n+1} = \mathbf{Q} + \frac{\Delta t}{2}\mathbf{F}(\overline{\mathbf{U}}^{n+1}),$$

which can be solved element by element. And the N_e 3×3 systems are independent of each other with every system of the same structure. Then we only need to solve the local algebraic system on every element K_j, $j = 1, 2, \ldots, N_e$,

$$\mathbf{R}(\overline{\mathbf{u}}_j^{n+1}) = 0$$

with

$$\overline{\mathbf{u}}_j^{n+1} = \begin{pmatrix} u_{j,1}^{n+1}\\ u_{j,2}^{n+1}\\ u_{j,3}^{n+1}\end{pmatrix}, \quad \mathbf{R}(\overline{\mathbf{u}}_j^{n+1}) = \begin{pmatrix} u_{j,1}^{n+1} - \mathbf{Q}_{3(j-1)+1} - \frac{\Delta t}{2}F(u_{j,1}^{n+1})\\ u_{j,2}^{n+1} - \mathbf{Q}_{3(j-1)+2} - \frac{\Delta t}{2}F(u_{j,2}^{n+1})\\ u_{j,3}^{n+1} - \mathbf{Q}_{3(j-1)+3} - \frac{\Delta t}{2}F(u_{j,3}^{n+1})\end{pmatrix},$$

where $\mathbf{Q}_{3(j-1)+m}$, $m = 1, 2, 3$, are the $[3(j-1)+m]$th elements of the vector $\mathbf{Q}$, respectively.

An iterative method such as the Newton method can be applied to implement for the above system. In the iterations to compute $\overline{\mathbf{u}}_j^{n+1}$, we use the numerical value $\overline{\mathbf{u}}_j^{n}$ at time t_n as the initial guess. And the threshold value for judging Newton iteration can be set small enough and we take 10^{-13} in the numerical examples.

3. Numerical experiments

In this section, numerical experiments are presented to demonstrate the validity and accuracy of the LDG method with Krylov IIF schemes for solving the reaction-diffusion systems on two-dimensional triangular meshes. First, we give two test examples with exact solutions to manifest the convergence accuracy of the method. Then we apply the method to a two reaction-diffusion system. And numerical results agree well with those in other references. In addition, we derive the gradients of solutions at the same time, which is not available in other methods.

All of the numerical examples considered in this section are subject to no-flux boundary conditions. The triangular partitions used here are Delaunay partitions got from EasyMesh. And the auxiliary parameters in the numerical fluxes are taken as

$$C_{11} = \frac{\widetilde{C}}{h_e}, \quad \mathbf{C}_{12} \cdot \mathbf{n}_e = \frac{1}{2}\,\mathrm{sign}(n_{e1} + n_{e2}), \quad \forall\, e \in \partial K \cap \mathcal{E}_h,$$

where h_e is the length of edge e, $\widetilde{C} > 0$ is the penalization parameter and set to be $\widetilde{C} = 1$ in the following computation. $\mathbf{n}_e = (n_{e1}, n_{e2})$ is the outward unit normal vector of K on e. The time step size is taken as $\Delta t = O(h_{\min})$ instead of $\Delta t = O(h_{\min}^2)$, where $h_{\min}$ is the length of the minimum edge in the triangular partition. And the dimension of the Krylov subspace, which was used when implement the Krylov algorithm, is chosen to be the same 25 for all the following numerical examples.

3.1. Numerical examples with exact solutions

In this subsection, we consider the linear and nonlinear parabolic problems defined on the unit square domain $\Omega = [0,1]^2$. The simulation is carried up to $t = 2.0$ at which the errors in L^2-norm and L^∞-norm are measured for both solutions and the gradients of solutions.

Example 3.1. We consider the linear reaction-diffusion problem defined by (2.1) with $\beta = 1$, $F(u) = f(x,y,t) - u$. The initial condition and right-hand side $f(x,y,t)$ are determined by the following exact solution

$$u(x,y,t) = \exp(-t)\cos(\pi x)\cos(\pi y).$$

We use LDG method for spatial discretization on various triangular meshes with the second order Krylov IIF temporal discretization to solve this example. And the time step size is taken as $\Delta t = 0.05 * h_{\min}$. CPU time, errors and order of accuracy for the method are presented in Table 3.1. We can see that numerical solutions possess the optimal approximation results in both L^2-norm and L^∞-norm for the exact solution

$$\|u - u_h\|_0 \approx O(h^2), \quad \|u - u_h\|_\infty \approx O(h^2),$$

where the two norms are defined by

$$\|u-u_h\|_0=\left(\sum_{K\in\mathcal{T}_h}\int_K (u(x,y,T)-u_h(x,y,T))^2\,dxdy\right)^{1/2},$$

$$\|u-u_h\|_\infty=\max_{K\in\mathcal{T}_h}\left(\max_{(x,y)\in K}|u(x,y,T)-u_h(x,y,T)|\right),$$

which are approximated by

$$\|u-u_h\|_0\approx\left(\sum_{K\in\mathcal{T}_h}(u(x_K,y_K,T)-u_h(x_K,y_K,T))^2|K|\right)^{1/2},$$

$$\|u-u_h\|_\infty\approx\max_{K\in\mathcal{T}_h}|u(x_K,y_K,T)-u_h(x_K,y_K,T)|$$

with (x_K,y_K) and $|K|$ the barycenter and area of the element K respectively.

N_e	CPU(s)	error	$\|\text{error}\|_0$	Order	$\|\text{error}\|_\infty$	Order
44	0.08	$u-u_h$	3.1624E-3	-	6.4101E-3	-
150	0.17		8.4925E-4	1.7733	2.0087E-3	1.5651
596	7.37		2.1449E-4	2.4124	5.6723E-4	2.2167
2388	53.98		5.3074E-5	1.6864	1.5677E-4	1.5528
9502	403.28		1.2932E-5	2.1290	4.1637E-5	1.9990
37930	3362.80		3.2233E-6	2.0088	1.1111E-5	1.9101
		$\mathbf{q}-\mathbf{q}_h$	1.6782E-2	-	3.1025E-2	-
			7.2733E-3	1.1277	1.5119E-2	0.9695
			3.4847E-3	1.2899	8.0486E-3	1.1053
			1.6869E-3	0.8760	3.0903E-3	1.1558
			8.3527E-4	1.0598	1.5482E-3	1.0421
			4.2083E-4	0.9912	7.8215E-4	0.9873

Table 3.1: Numerical results for Example 3.1 at $t=2.0$ discretized in time by the second order Krylov IIF method.

And the numerical approximation for the gradients have the suboptimal-order convergence accuracy in L^2-norm and L^∞-norm

$$\|\mathbf{q}-\mathbf{q}_h\|_0\approx O(h),\quad \|\mathbf{q}-\mathbf{q}_h\|_\infty\approx O(h),$$

where these two norms are approximated by

$$\|\mathbf{q}-\mathbf{q}_h\|_0 \approx \left\{ \sum_{K\in\mathcal{T}_h} \left(\sum_{i=x,y} q_i(x_K,y_K,T) - q_{hi}(x_K,y_K,T) \right)^2 |K| \right\}^{1/2},$$

$$\|\mathbf{q}-\mathbf{q}_h\|_\infty \approx \max_{K\in\mathcal{T}_h} \left(\max_{i=x,y} |q_i(x_K,y_K,T) - q_{hi}(x_K,y_K,T)| \right)$$

with $\mathbf{q} = (q_x, q_y)$, $\mathbf{q}_h = (q_{hx}, q_{hy})$. Here, T is the final time we have calculated and $T = 2$ in the numerical experiments.

The numerical results are consistent with the priori error analysis for the LDG method in [2].

N_e	CPU(s)	error	$\|\text{error}\|_0$	Order	$\|\text{error}\|_\infty$	Order
44	12.68	$u-u_h$	2.2154E-3	-	4.3794E-3	-
150	140.47		6.5091E-4	1.6520	1.4343E-3	1.5056
596	2408.69		1.7345E-4	2.3184	4.3966E-4	2.0728
2388	34535.88		4.3241E-5	1.6773	1.1163E-4	1.6553
		$\mathbf{q}-\mathbf{q}_h$	1.5516E-2	-	3.1418E-2	-
			7.0270E-3	1.0684	1.4929E-2	1.0036
			3.4099E-3	1.2676	8.0136E-3	1.0907
			1.6826E-3	0.8529	3.1344E-3	1.1335

Table 3.2: Numerical results for Example 3.1 at $t = 2.0$ discretized in time by the second order Runge-Kutta method.

Then we come to verify, compared with the explicit-LDG method, the IIF-LDG method can shorten the calculation time greatly. We discretized this example by the LDG method in space and the second order explicit Runge-Kutta method in time. Now, the time step size is taken to be $\Delta t = 0.01 * h^2_{\min}$, where the CFL number 0.01 is the largest value to remain stable. And we present the numerical results and CPU times cost in Table 3.2, where we only calculate to the fourth grid partition with 2388 elements because the cost time has been long. The data in Tables 3.1 and 3.2 show that these two methods both derive the expected accuracy in space. Compared the CPU times in these two tables, we can observe that the IIF-LDG method indeed improves the calculation efficiency and the smaller the grid size, the more obvious the advantage.

Example 3.2. Consider the nonlinear reaction-diffusion problem defined by (2.1) with $\beta = 1$, $F(u) = f(x, y, t) - u^2$. The initial condition and $f(x, y, t)$ are chosen such that the exact solution is also the form as the linear case.

First, we apply the LDG method and the second order Krylov IIF method to discretized this problem in space and in time respectively, where the time step size is taken as $\Delta t = 0.05 * h_{\min}$. Numerical results at $t = 2.0$ are given in Table 3.3, from which we can observe that, by using LDG method for spacial discretization, we derived the expected accuracy orders in both L^2-norm and L^∞-norm as the linear case above: second-order convergency for the solution u and first-order convergency for the gradient $\mathbf{q}$.

N_e	CPU(s)	error	$\|\text{error}\|_0$	Order	$\|\text{error}\|_\infty$	Order
44	0.14	$u - u_h$	6.2293E-3	-	1.1794E-2	-
150	0.17		1.6289E-3	1.8092	3.1322E-3	1.7883
596	3.50		4.0714E-4	2.4306	8.2647E-4	2.3356
2388	63.39		1.0115E-4	1.6815	2.1224E-4	1.6415
9502	577.36		2.4315E-5	2.1493	5.3059E-5	2.0902
37930	4563.48		6.0484E-6	2.0117	1.3710E-5	1.9567
		$\mathbf{q} - \mathbf{q}_h$	1.7074E-2	-	3.1008E-2	-
			7.3242E-3	1.1416	1.5231E-2	0.9589
			3.4912E-3	1.2989	8.0777E-3	1.1118
			1.6887E-3	0.8770	3.0933E-3	1.1590
			8.3244E-4	1.0665	1.5635E-3	1.0288
			4.2086E-4	0.9862	7.8224E-4	1.0013

Table 3.3: Numerical results for Example 3.2 at $t = 2.0$ discretized in time by the second order Krylov IIF method.

Then we solve this example with the second order explicit Runge-Kutta method for temporal discretization. The time step size is taken as $\Delta t = 0.01 * h_{\min}^2$, and here 0.01 is the largest value to remain stable. Numerical results and CPU times cost are given in Table 3.4, where we only calculate to the fourth grid partition with 2388 elements also since the cost time has been long. The errors and order of accuracy in Table 3.4 are basically the same as Table 3.3. However, the CPU times are much bigger, which demonstrates that the IIF-LDG method can improve the calculation efficiency greatly compared with the explicit-LDG method.

N_e	CPU(s)	error	$\|\text{error}\|_0$	Order	$\|\text{error}\|_\infty$	Order
44	10.89	$u - u_h$	2.2236E-3	-	4.3858E-3	-
150	120.23		6.5389E-4	1.6509	1.4408E-3	1.5015
596	3102.86		1.7426E-4	2.3182	4.4077E-4	2.0763
2388	45712.23		4.3442E-5	1.6774	1.1172E-4	1.6573
		$\mathbf{q} - \mathbf{q}_h$	1.5551E-2	-	3.1384E-2	-
			7.0328E-3	1.0703	1.4937E-2	1.0015
			3.4106E-3	1.2686	8.0166E-3	1.0909
			1.6827E-3	0.8531	3.1347E-3	1.1338

Table 3.4: Numerical results for Example 3.2 at $t = 2.0$ discretized in time by the second order Runge-Kutta method.

We should point out that, although the CPU time in Tables 3.1 and 3.3 is respectively larger than that in Tables 6 and 8 of [5], we derive the approximations for the gradients at the same time here.

3.2. Application to a morphogenesis problem

In this subsection, we apply LDG method with Krylov IIF schemes to solve the Schnakenberg system.

Example 3.3. The Schnakenberg system proposed by Schnakenberg [28], which was used to model the spatial distribution of the morphogen, e.g., the distribution of calcium in the hairs of the whorl in Acetabularia [10], is also a classical example applied to test numerical methods for reaction-diffusion equations, such as [14,22,23,27,32,33]. In non-dimensional form the system is written as

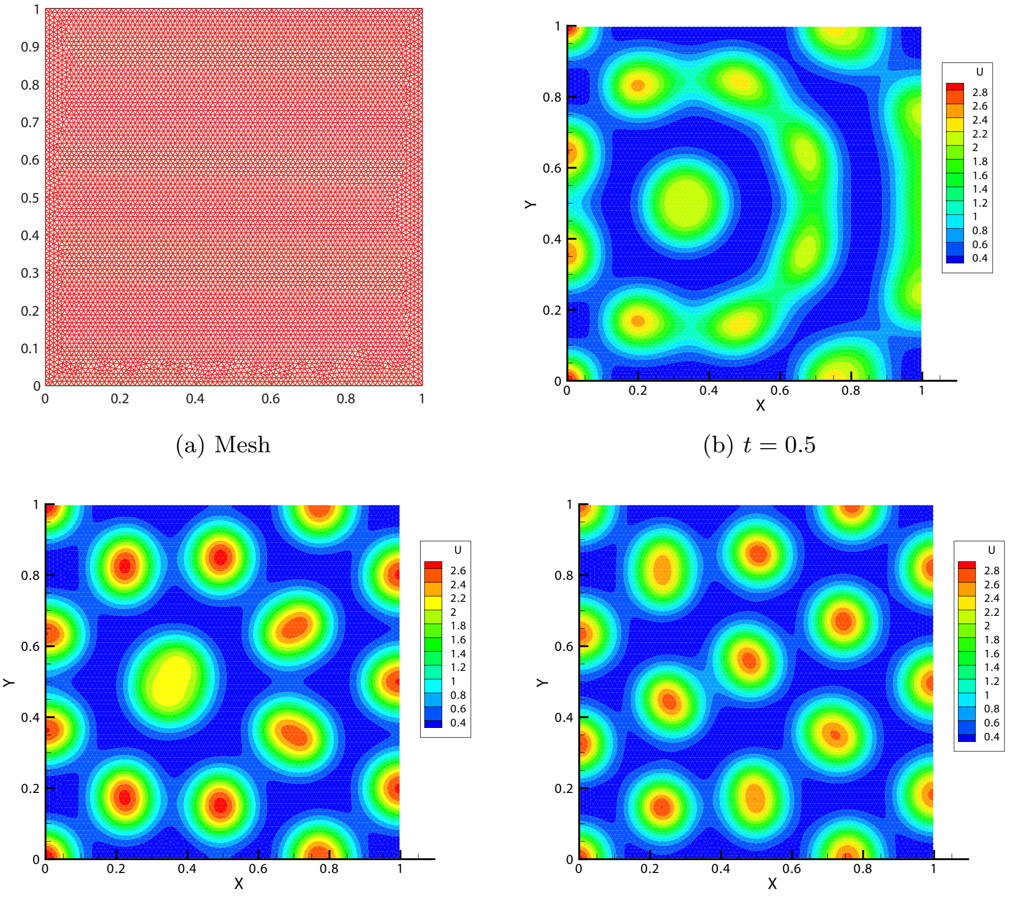
$$\frac{\partial u}{\partial t} = \beta_u \Delta u + \kappa(a - u + u^2 v), \quad \frac{\partial v}{\partial t} = \beta_v \Delta v + \kappa(b - u^2 v). \tag{3.1}$$

Following the setup in [14], we take the initial conditions as

$$u(x,y,0) = a + b + 10^{-3} \exp\left(-100((x - \tfrac{1}{3})^2 + (y - \tfrac{1}{2})^2)\right), \quad v(x,y,0) = \frac{b}{(a+b)^2} \tag{3.2}$$

with the parameters $\kappa = 100$, $a = 0.1305$, $b = 0.7695$, $\beta_u = 0.05$, $\beta_v = 1$.

In the computation, we take the time step size as $\Delta t = 0.01 * h_{\min}$. First, we compute the problem (3.1)–(3.2) on the square domain $\Omega = [0,1]^2$, which is divided into 14836 triangular elements, see Figure 3.1(a). And the other three pictures in Figure 3.1 are respectively the numerical approximations of u at $t = 0.5$, $t = 1.0$ and $t = 2.0$, which agree well with those in [32,33]. Here, we do not show the numerical solutions of v because they have the similar pattern with those of u. Moreover, we derive the approximations of $\mathbf{q} = \beta_u \nabla u$ and $\mathbf{p} = \beta_v \nabla v$ at the same time, which is not available by other methods. The two components of the numerical approximation for $\mathbf{q}$ are plotted in Figure 3.2.



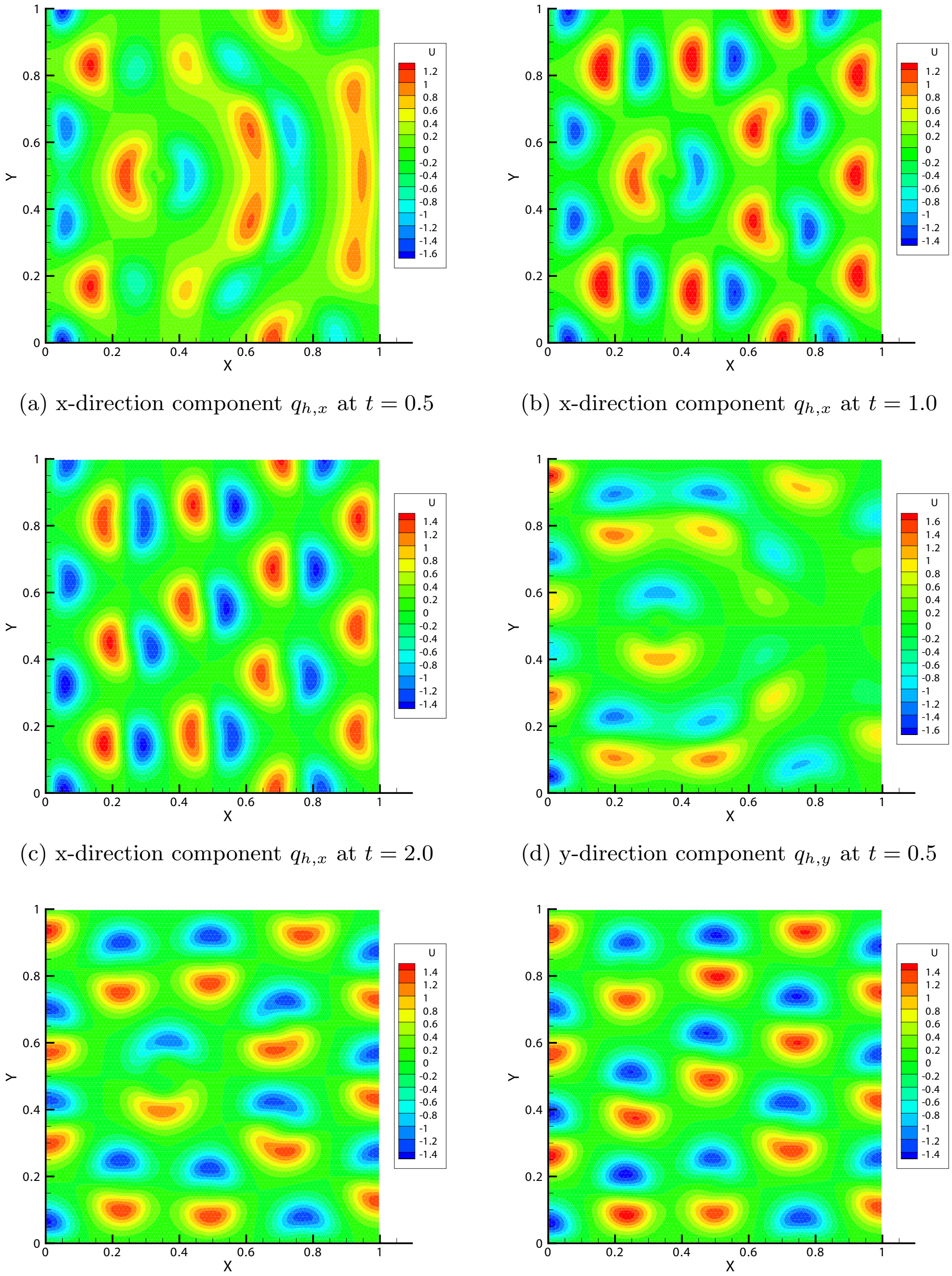
(a) Mesh

(b) $t = 0.5$

(c) $t = 1.0$

(d) $t = 2.0$

Figure 3.1: The time evolution of the numerical solution u_h on the square domain.



(a) x-direction component $q_{h,x}$ at $t = 0.5$

(b) x-direction component $q_{h,x}$ at $t = 1.0$

(c) x-direction component $q_{h,x}$ at $t = 2.0$

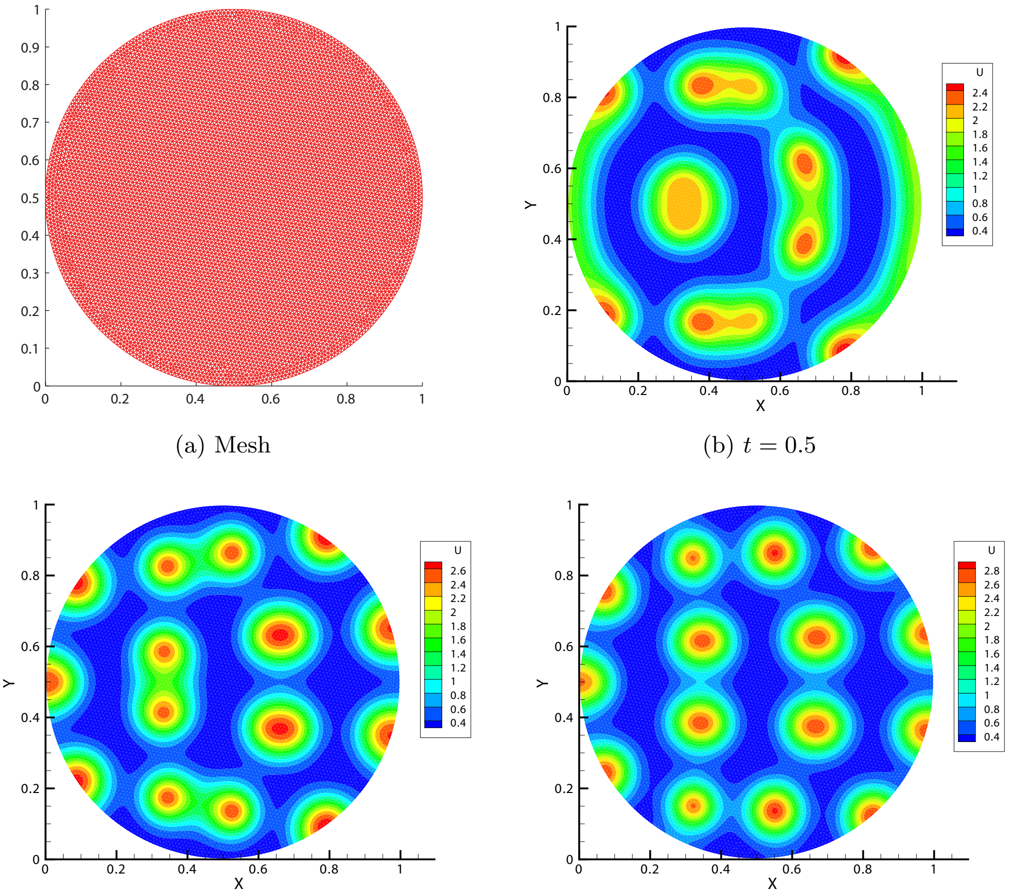
(d) y-direction component $q_{h,y}$ at $t = 0.5$

(e) y-direction component $q_{h,y}$ at $t = 1.0$

(f) y-direction component $q_{h,y}$ at $t = 2.0$

Figure 3.2: The two components of the numerical approximation for $\mathbf{q}$ on the square domain.

Then we compute the system on the circular domain $\Omega = \{(x, y) \mid (x - 0.5)^2 + (y - 0.5)^2 \leq 0.5^2\}$. And we use the Delaunay partitions with 16698 triangular elements, see Figure 3.3(a). The numerical solutions u_h at $t = 0.5$, $t = 1.0$ and $t = 2.0$ are presented in Figure 3.3(b-d). And they also agree well with those in [32, 33]. In addition, Figure 3.4 shows the two components of the numerical approximation for $\mathbf{q}$.



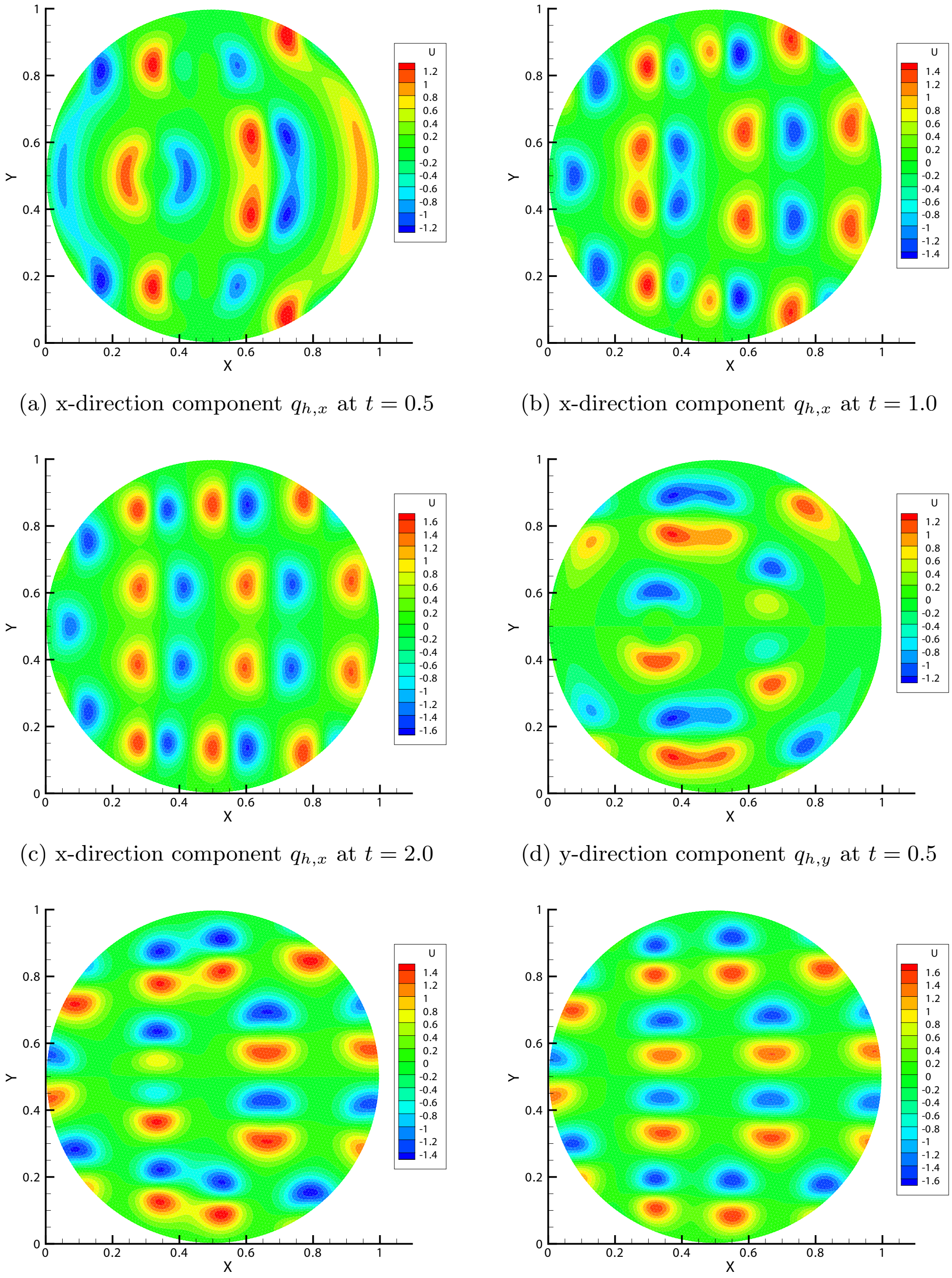
(a) Mesh

(b) $t = 0.5$

(c) $t = 1.0$

(d) $t = 2.0$

Figure 3.3: The time evolution of the numerical solution u_h on the circular domain.



(a) x-direction component $q_{h,x}$ at $t = 0.5$

(b) x-direction component $q_{h,x}$ at $t = 1.0$

(c) x-direction component $q_{h,x}$ at $t = 2.0$

(d) y-direction component $q_{h,y}$ at $t = 0.5$

(e) y-direction component $q_{h,y}$ at $t = 1.0$

(f) y-direction component $q_{h,y}$ at $t = 2.0$

Figure 3.4: The two components of the numerical approximation for $\mathbf{q}$ on the circular domain.

4. Conclusions

In this paper, we have developed the LDG method, coupled with the Krylov IIF time discretization, for nonlinear reaction-diffusion systems. We choose the values of midpoints at three edges of the triangular element as the degrees of freedom, which can simplify the form of the discrete system when we apply the Gauss numerical integration formula with midpoints at its three edges to approximate the integral on a triangular element. By using the LDG method for spacial discretization, we derive not only the numerical solutions but also the numerical approximations for the gradients at the same time. However, the DG method is designed for solving equations containing only first order spatial derivatives and the approximations of the gradients are not obtained directly which need to use a projection into suitable finite elements spaces. Furthermore, the important property of LDG method that the computation can proceed element by element can also be remained, which benefits from applying the IIF method for temporal discretization. And It also relaxes the strict time-step restriction that is necessary for explicit schemes. In addition, the Krylov subspace approximation is applied to efficiently compute the product of the matrix exponential and a vector, which is produced from IIF temporal discretization. Direct numerical simulations on two test equations with exact solutions, as well as the morphogenesis systems from developmental biology, show that the method is efficient, accurate and possesses its advantages. What is more, by comparing the CPU times to implement the first two examples with the second order Krylov IIF method and the second order explicit Runge-Kutta method, we verify that the Krylov IIF method, coupled with LDG method, can shorten the calculation time greatly.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 11801332, 11471196 and 11571002), the Science and Technology Development Foundation of CAEP (Grant No. 2015B0101021) and the Defense Industrial Technology Development Program (Grant No. B1520133015).

References

[1] F. Bassi and S. Rebay, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*, J. Comput. Phys. **131** (1997), no. 2, 267–279.

[2] P. Castillo, B. Cockburn, I. Perugia and D. Schötzau, *An a priori error analysis of the local discontinuous Galerkin method for elliptic problems*, SIAM J. Numer. Anal. **38** (2000), no. 5, 1676–1706.

[3] Z. Chen, B. Cockburn, C. Gardner and J. W. Jerome, *Quantum hydrodynamic simulation of hysteresis in the resonant tunneling diode*, J. Comput. Phys. **117** (1995), no. 2, 274–280.

[4] Z. Chen, B. Cockburn, J. W. Jerome and C.-W. Shu, *Mixed-RKDG finite element methods for the 2-D hydrodynamic model for semiconductor device simulation*, VLSI Design **3** (1995), no. 2, 145–158.

[5] S. Chen and Y.-T. Zhang, *Krylov implicit integration factor methods for spatial discretization on high dimensional unstructured meshes: application to discontinuous Galerkin methods*, J. Comput. Phys. **230** (2011), no. 11, 4336–4352.

[6] Y. Cheng and C.-W. Shu, *A discontinuous Galerkin finite element method for time dependent partial differential equations with higher order derivatives*, Math. Comp. **77** (2008), no. 262, 699–730.

[7] B. Cockburn, G. Kanschat, I. Perugia and D. Schötzau, *Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids*, SIAM J. Numer. Anal. **39** (2002), no. 1, 264–285.

[8] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal. **35** (1998), no. 6, 2440–2463.

[9] A. Gierer and H. Meinhardt, *A theory of biological pattern formation*, Kybernetik **12** (1972), no. 1, 30–39.

[10] B. C. Goodwin and L. E. H. Trainor, *Tip and whorl morphogenesis in Acetabularia by calcium-regulated strain fields*, J. Theoret. Biol. **117** (1985), no. 1, 79–106.

[11] P. Gray and S. K. Scott, *Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Isolas and other forms of multistability*, Chem. Eng. Sci. **38** (1983), no. 1, 29–43.

[12] R. Guo, Y. Xia and Y. Xu, *Semi-implicit spectral deferred correction methods for highly nonlinear partial differential equations*, J. Comput. Phys. **338** (2017), 269–284.

[13] A. L. Hanhart, M. K. Gobbert and L. T. Izu, *A memory-efficient finite element method for systems of reaction-diffusion equations with non-smooth forcing*, J. Comput. Appl. Math. **169** (2004), no. 2, 431–458.

[14] W. Hundsdorfer and J. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer Series in Computational Mathematics **33**, Springer-Verlag, Berlin, 2003.

[15] P. Jamet, *Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain*, SIAM J. Numer. Anal. **15** (1978), no. 5, 912–928.

[16] G. S. Jiang and C.-W. Shu, *On a cell entropy inequality for discontinuous Galerkin methods*, Math. Comp. **62** (1994), no. 206, 531–538.

[17] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, 1987.

[18] A. Kassam and L. N. Trefethen, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput. **26** (2005), no. 4, 1214–1233.

[19] I. Lengyel and I. R. Epstein, *Modeling of turing structures in the chlorite-iodide-malonic acid-starch reaction system*, Science **251** (1991), no. 4994, 650–652.

[20] B. Q. Li, *Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer*, Computational Fluid and Solid Mechanics, Springer-Verlag London, London, 2006.

[21] H. Liu and J. Yan, *The direct discontinuous Galerkin (DDG) methods for diffusion problems*, SIAM J. Numer. Anal. **47** (2009), no. 1, 675–698.

[22] A. Madzvamuse, *Time-stepping schemes for moving grid finite elements applied to reaction-diffusion systems on fixed and growing domains*, J. Comput. Phys. **214** (2006), no. 1, 239–263.

[23] A. Madzvamuse, A. J. Wathen and P. K. Maini, *A moving grid finite element method applied to a model biological pattern generator*, J. Comput. Phys. **190** (2003), no. 2, 478–500.

[24] Q. Nie, Y.-T. Zhang and R. Zhao, *Efficient semi-implicit schemes for stiff systems*, J. Comput. Phys. **214** (2006), no. 2, 512–537.

[25] W. H. Reed and T. R. Hill, *Triangular mesh methods for the neutron transport equation*, Los Alamos Scientific Laboratory report LA-UR-73-479, Los Alamos, NM, 1973.

[26] D. L. Ropp, J. N. Shadid and C. C. Ober, *Studies of the accuracy of time integration methods for reaction-diffusion equations*, J. Comput. Phys. **194** (2004), no. 2, 544–574.

[27] S. J. Ruuth, *Implicit-explicit methods for reaction-diffusion problems in pattern formation*, J. Math. Biol. **34** (1995), no. 2, 148–176.

[28] J. Schnakenberg, *Simple chemical reaction systems with limit cycle behaviour*, J. Theoret. Biol. **81** (1979), no. 3, 389–400.

[29] A. M. Soane, M. K. Gobbert and T. I. Seidman, *Numerical exploration of a system of reaction-diffusion equations with internal and transient layers*, Nonlinear Anal. Real World Appl. **6** (2005), no. 5, 914–934.

[30] A. M. Turing, *The chemical basis of morphogenesis*, Philos. Trans. Roy. Soc. London Ser. B **237** (1952), no. 641, 37–72.

[31] Y. Xu and C.-W. Shu, *Error estimates of the semi-discrete local discontinuous Galerkin method for nonlinear convection-diffusion and KdV equations*, Comput. Methods Appl. Mech. Engrg. **196** (2007), no. 37-40, 3805–3822.

[32] R. Zhang, X. Yu, J. Zhu and A. F. D. Loula, *Direct discontinuous Galerkin method for nonlinear reaction-diffusion systems in pattern formation*, Appl. Math. Model. **38** (2014), no. 5-6, 1612–1621.

[33] J. Zhu, Y.-T. Zhang, S. A. Newman and M. Alber, *Application of discontinuous Galerkin methods for reaction-diffusion systems in developmental biology*, J. Sci. Comput. **40** (2009), no. 1-3, 391–418.

Na An
School of Mathematics and Statistics, Shandong Normal University, Jinan 250014, China
E-mail address: annabetter@sdnu.edu.cn

Chaobao Huang
Applied and Computational Mathematics division, Beijing Computational Science
Research Center, Beijing 100193, China
E-mail address: huangcb@csrc.ac.cn

Xijun Yu
Laboratory of Computational Physics, Institute of Applied Physics and Computational
Mathematics, Beijing 100088, China
E-mail address: yuxj@iapcm.ac.cn