

Application of Discontinuous Galerkin Methods for Reaction-Diffusion Systems in Developmental Biology

Jianfeng Zhu · Yong-Tao Zhang · Stuart A. Newman · Mark Alber

Received: 18 April 2008 / Revised: 17 June 2008 / Accepted: 24 June 2008
© Springer Science+Business Media, LLC 2008

Abstract Nonlinear reaction-diffusion systems which are often employed in mathematical modeling in developmental biology are usually highly stiff in both diffusion and reaction terms. Moreover, they are typically considered on multidimensional complex geometrical domains because of complex shapes of embryos. We overcome these computational challenges by combining discontinuous Galerkin (DG) finite element methods with Strang type symmetrical operator splitting technique, on triangular meshes. This allows us to avoid directly solving a coupled nonlinear system, as is necessary with the standard implicit schemes. Numerical solutions of two reaction-diffusion systems, the well-studied Schnakenberg model, which has been applied to several problems in developmental biology, and a new biologically based system for skeletal pattern formation in the vertebrate limb, are presented to demonstrate effects of various domain geometries on the resulting biological patterns.

Keywords Discontinuous Galerkin finite element methods · Reaction-diffusion equations · Operator splitting · Triangular meshes · Complex geometry · Developmental biology

1 Introduction

Mathematical simulation of systems in developmental biology has given rise to a variety of models which account for spatial-temporal patterning phenomena. Many of these mathematical models are reaction-diffusion systems which have the general form

$$\frac{\partial u}{\partial t} = D\nabla^2 u + F(u), \quad (1)$$

J. Zhu · Y.-T. Zhang (✉) · M. Alber
Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-4618, USA
e-mail: y Zhang10@nd.edu

M. Alber
e-mail: malber@nd.edu

S.A. Newman
Department of Cell Biology and Anatomy, Basic Science Building, New York Medical College,
Valhalla, NY 10595, USA

where $u \in \mathbf{R}^p$ represent concentrations of a group of biochemical molecules (p is the number of PDEs in the system and it can be 1, 2, 3, etc.), $D \in \mathbf{R}^{p \times p}$ is the diffusion constant matrix, $\nabla^2 u$ is the Laplacian associated with the diffusion of the molecule whose concentration is u , and $F(u)$ describes the biochemical reactions. Examples include Turing-type models [75] such as the Gierer-Meinhardt model [33], the Schnakenberg model [66], the Thomas model [73], the Gray-Scott model [31, 32] and others described in [4, 5, 7, 8, 54]. Related systems include models to study the robustness of gene networks [27, 28, 44, 53, 67, 78, 88], and Fisher's equation [13, 29] with various applications including tissue engineering (e.g. [52]) and gene propagation (e.g. [29]). Although (1) has a linear diffusion part, the nonlinear reaction part is usually complicated. Efficient and accurate numerical methods for the type of semilinear PDEs represented by (1) are essential when we carry out the parameter variation studies and computational analysis for systems in developmental biology.

Because of the complex shapes of embryos and their parts, the reaction-diffusion equations (1) are often applied to high dimensional irregular geometrical domains, particularly when the shape and size of embryos play important roles in the studied biological systems (e.g. [3, 25, 48, 55, 90]). Finite element numerical methods on unstructured meshes are powerful means for handling the complicated domain geometries [42]. In [36, 49–51, 63, 68], continuous Galerkin (CG) finite element methods were used to solve examples of the reaction-diffusion equations (1) on complex domains. Recently, discontinuous Galerkin (DG) finite element methods have become increasingly popular to solve various PDEs. The DG methods use a completely discontinuous piecewise polynomial space for the numerical solution and the test functions. The first DG method was introduced by Reed and Hill [60], in relation to the problem of neutron transport. A major development of the DG method was carried out by Cockburn, Shu et al. in a series of papers [20–24] in which they established a framework for solving nonlinear time dependent hyperbolic conservation laws.

DG methods confer several advantages that make them attractive for applications. These include their ability for readily addressing complicated geometry and boundary conditions (an advantage shared by all finite element methods), their flexibility for easy hp-adaptivity (combinations of refining/unrefining elements (h-adaptivity) and changing order of base functions (p-adaptivity)) including changes of approximation orders between neighboring elements and allowing the use of general meshes with hanging nodes, their compactness and efficient parallel implementation [12], and their easy coordination with finite volume techniques for computing problems with discontinuous or sharp gradient solutions.

The DG method has found applications in many diverse areas. Good references for the method and its recent development include a survey paper [17] and other papers in the same Springer volume, as well as several review articles [16, 18]. Besides its success in solving first order hyperbolic conservation laws, the DG method has been generalized to solve time dependent PDEs containing higher spatial derivatives. It has been adapted to solve a convection diffusion equation (containing second derivatives) by Cockburn and Shu [19], motivated by the successful numerical experiments of Bassi and Rebay [9] for the compressible Navier-Stokes equations. This method is termed the local discontinuous Galerkin (LDG) method because the auxiliary variables introduced to approximate spatial derivatives can be eliminated locally. Later, LDG methods were developed to solve various nonlinear time dependent PDEs with higher order derivatives [45, 80–85]. For alternative DG methods for diffusion problems, see [6, 10, 46, 59, 61, 62].

In a recent paper [14], Cheng and Shu developed a new DG method for solving time dependent PDEs with higher order spatial derivatives, based on [1, 30, 76]. The scheme is formulated by repeated integration by parts of the original equation and then replacing the interface values of the solution by carefully chosen numerical fluxes. Compared to the

LDG method, this new DG method can be applied without introducing any auxiliary variables or rewriting the original equation in the form of a larger system, hence it is easier to formulate and implement, has a smaller effective stencil, and may reduce storage and computational cost [14]. In this paper, we have adopted the DG approaches of [14] for the spatial discretization of reaction-diffusion equations (1), on two dimensional triangular meshes.

Another computational challenge comes from the stiffness of reaction-diffusion equations (1) and the DG spatial discretization operator, which would require efficient time discretization techniques. On the one hand, standard explicit methods are highly inefficient because of their severe stability constraint, hence stabilized explicit methods were developed (e.g. [26, 77]). On the other hand, the implicit methods (e.g. [11, 37, 47]) are more popularly used due to their larger stability region. But the fully implicit Runge-Kutta or Backward Difference Formula methods require the solution of typically nonlinear coupled system of equations, and the computational cost can be significant. One popular strategy to avoid solving the completely coupled nonlinear system is to use the Operator Splitting (OS) approach (see e.g. [63, 69, 70, 86]). We adopted the Strang type second-order symmetrical OS schemes [39, 70] to split the diffusion from the reaction terms of (1). Since the whole problem is thus broken down into smaller parts, we can solve the linear diffusion problem and the nonlinear reaction problem individually by implicit temporal schemes. By means of this approach, at each time step, the coupled system resulting from the DG spatial discretization of the diffusion term is linear and sparse and we can solve it efficiently by a sparse system solver. On each element, we only need to solve a small nonlinear system, which has the same size as the product of the number of the PDEs in the system (1) and the degrees of freedom of the approximation polynomial. The local nonlinear system can be solved efficiently by an iterative procedure such as Newton's method.

We note that an alternative way to solve reaction-diffusion systems efficiently on complicated geometrical domains is to use continuous Galerkin (CG) finite element methods together with an operator splitting technique [63]. CG and DG methods each have their own advantages. CG methods have fewer degrees of freedom, especially for the high spatial dimensional problems. However for the reaction-diffusion systems in developmental biology, sharp gradients are often formed. Adaptive methods are especially efficient for resolving the structure of sharp gradients in the solution (e.g. [41, 71, 87]). DG methods can easily handle adaptivity strategies since refinement or unrefinement of the grid can be achieved without dealing with the continuity restrictions typical of conforming finite element methods. Moreover, the degree of the approximating polynomial can be easily changed from one element to the other, and the use of general meshes with hanging nodes is allowed [17]. Based on these favorable properties of DG methods, DG methods should prove to be highly appropriate for designing adaptive methods for reaction-diffusion systems in developmental biology.

In Sect. 2, we describe in detail the DG spatial discretization combined with Strang's operator splitting and the Crank-Nicholson temporal discretization to solve the reaction-diffusion systems (1), on 2D triangular meshes. In Sect. 3, we test the numerical methods for reaction-diffusion equations with exact solutions, and demonstrate its good stability and accuracy. We compare the cases with and without OS to demonstrate the computational efficiency enhancement due to the operator splitting technique. In Sect. 4, we apply the methods to two reaction-diffusion models used in developmental biology, the classical Schnakenberg model [66] and a new model for skeletal pattern formation during vertebrate limb development [3]. We show by various simulations of these models the effect on resulting patterns of varying domain size and shape. Conclusions are given in Sect. 5.

2 Numerical Methods

For purposes of clear presentation, we first describe the numerical methods for the scalar case ($p = 1$) of (1), in Sects. 2.1 and 2.2. In Sect. 2.3, we straightforwardly extend the algorithm to solve the system case ($p > 1$) of (1).

2.1 The DG Spatial Discretization

Let Ω be an open, bounded domain on which the reaction-diffusion system (1) is defined. We consider a triangulation Ω_h of Ω which consists of nonoverlapping triangles $\{\Delta_m\}_{m=1}^N$. Let $h_{\min} = \min_{1 \leq m \leq N} \rho_m$, where ρ_m is the diameter of the inscribed circle of the triangle Δ_m .

Define the finite element space $V_h^k = \{v : v|_{\Delta_m} \in P^k(\Delta_m), m = 1, \dots, N\}$, where $P^k(\Delta_m)$ denotes the set of all polynomials of degree at most k on Δ_m .

We apply the DG formulation [14] to discretize the reaction-diffusion equations (1) in the spatial direction, but keep the time variable continuous. The semi-discrete scheme is: find $u \in V_h^k$, such that

$$\begin{aligned} & \int_{\Delta_m} u_t v dx - D \int_{\Delta_m} u \nabla^2 v dx + D \int_{\partial \Delta_m} \widehat{u} \nabla v \cdot \vec{n}_{\partial \Delta_m} dS - D \int_{\partial \Delta_m} v \widetilde{\nabla} u \cdot \vec{n}_{\partial \Delta_m} dS \\ & = \int_{\Delta_m} F(u) v dx \end{aligned} \tag{2}$$

holds true for any $v \in V_h^k$ and $m = 1, \dots, N$. The numerical fluxes on the element edges $\partial \Delta_m$ are chosen as

$$\widehat{u} = \frac{u^{in} + u^{ext}}{2}, \tag{3}$$

$$\widetilde{\nabla} u = \frac{(\nabla u)^{in} + (\nabla u)^{ext}}{2} + \beta[u], \tag{4}$$

where the jump term

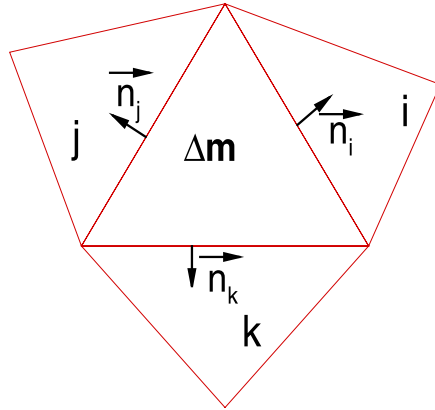
$$[u] = (u^{ext} - u^{in})|_{\partial \Delta_m} \cdot \vec{n}_{\partial \Delta_m}, \tag{5}$$

u^{in} and u^{ext} are the limits of u at $x \in \partial \Delta_m$ taken from the interior and the exterior of Δ_m respectively, $\vec{n}_{\partial \Delta_m}$ is the outward unit normal to the element Δ_m at $x \in \partial \Delta_m$, and β is a positive constant that is of the order $\mathcal{O}(h_{\min}^{-1})$. In all of computations of this paper, we take $\beta = 10/h_{\min}$. The choice of numerical fluxes (3)–(5) is crucial for the stability and convergence of the DG scheme (2). See [14, 19] for more discussions about the choice of numerical fluxes.

In this paper, we consider the P^1 case such that the order of accuracy in the spatial direction is consistent with the splitting error order in the temporal direction, and they are both two. For each element Δ_m , denote its three neighboring elements by i_m , j_m , and k_m . To simplify notations in the following presentation, we will omit the subscript m and just use i, j, k to represent the neighboring cells of Δ_m , as shown in Fig. 1. The linear polynomial on Δ_m is represented by

$$u(x, y, t) = a_m(t) + b_m(t)\xi_m + c_m(t)\eta_m \tag{6}$$

Fig. 1 A typical stencil



where

$$\xi_m = \frac{x - x_m}{h_m}, \tag{7}$$

$$\eta_m = \frac{y - y_m}{h_m}, \tag{8}$$

and (x_m, y_m) is the barycenter of the element Δ_m , $h_m = \sqrt{|\Delta_m|}$ with $|\Delta_m|$ denoting the area of Δ_m . By taking $v = 1, \xi_m, \eta_m$ on Δ_m and $v = 0$ elsewhere, the DG formulation (2) is converted from the integral form to the following system, for $m = 1, \dots, N$:

$$\begin{aligned} & q_{11}a'_m(t) + q_{12}b'_m(t) + q_{13}c'_m(t) \\ &= D \left\{ w_{am1}a_m(t) + w_{bm1}b_m(t) + w_{cm1}c_m(t) \right. \\ & \quad \left. + \sum_{l=i,j,k} [w_{al1}a_l(t) + w_{bl1}b_l(t) + w_{cl1}c_l(t)] \right\} \\ & \quad + (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l})), \end{aligned} \tag{9}$$

$$\begin{aligned} & q_{21}a'_m(t) + q_{22}b'_m(t) + q_{23}c'_m(t) \\ &= D \left\{ w_{am2}a_m(t) + w_{bm2}b_m(t) + w_{cm2}c_m(t) \right. \\ & \quad \left. + \sum_{l=i,j,k} [w_{al2}a_l(t) + w_{bl2}b_l(t) + w_{cl2}c_l(t)] \right\} \\ & \quad + (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l}))\xi_m(x_{m,l}, y_{m,l}), \end{aligned} \tag{10}$$

$$\begin{aligned} & q_{31}a'_m(t) + q_{32}b'_m(t) + q_{33}c'_m(t) \\ &= D \left\{ w_{am3}a_m(t) + w_{bm3}b_m(t) + w_{cm3}c_m(t) \right. \end{aligned}$$

$$\begin{aligned}
 & + \sum_{l=i,j,k} \left\{ w_{al3}a_l(t) + w_{bl3}b_l(t) + w_{cl3}c_l(t) \right\} \\
 & + (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l}))\eta_m(x_{m,l}, y_{m,l}), \tag{11}
 \end{aligned}$$

where the coefficients $\{q_{rs}\}_{r,s=1}^3, \{\{w_{alr}\}_{r=1}^3, \{w_{blr}\}_{r=1}^3, \{w_{clr}\}_{r=1}^3\}_{l=m,i,j,k}$ are constants which depend on the local geometry of the mesh (i.e., triangle Δ_m and its neighboring cells i, j, k and $\vec{n}_{\partial\Delta_m}$ as shown in Fig. 1, the local basis functions $1, \{\xi_l, \eta_l\}_{l=m,i,j,k}$, and the constant β . $\{(x_{m,l}, y_{m,l})\}_{l=i,j,k}$ are the mid-points of the three edges $\{e_l\}_{l=i,j,k}$ of Δ_m which serve as Gaussian quadrature points for the integral involving the nonlinear reaction terms in (2). The detailed formulae for computing these constants are presented in the Appendix. In our implementation, these mesh-dependent constants are pre-calculated before the time evolution since they don't depend on the numerical solution u . Rewrite equations (9)–(11) to the matrix-vector form

$$\mathbf{Q}_m \vec{V}'_m(t) = D \sum_{l=m,i,j,k} \mathbf{W}_l \vec{V}_l(t) + \vec{F}_m(\vec{V}_m) \tag{12}$$

where

$$\begin{aligned}
 \mathbf{Q}_m &= \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix}, & \mathbf{W}_l &= \begin{pmatrix} w_{al1} & w_{bl1} & w_{cl1} \\ w_{al2} & w_{bl2} & w_{cl2} \\ w_{al3} & w_{bl3} & w_{cl3} \end{pmatrix}, & \vec{V}_m &= \begin{pmatrix} a_m(t) \\ b_m(t) \\ c_m(t) \end{pmatrix}, \\
 \vec{V}_l &= \begin{pmatrix} a_l(t) \\ b_l(t) \\ c_l(t) \end{pmatrix}, & \vec{F}_m(\vec{V}_m) &= \begin{pmatrix} (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l})) \\ (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l}))\xi_m(x_{m,l}, y_{m,l}) \\ (q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l}))\eta_m(x_{m,l}, y_{m,l}) \end{pmatrix}.
 \end{aligned}$$

Finally we have the ODE system resulting from the DG spatial discretization:

$$\vec{V}'_m(t) = D \sum_{l=m,i,j,k} \tilde{\mathbf{W}}_l \vec{V}_l(t) + \tilde{F}_m(\vec{V}_m), \quad m = 1, \dots, N, \tag{13}$$

where $\tilde{\mathbf{W}}_l = \mathbf{Q}_m^{-1} \mathbf{W}_l, \tilde{F}_m = \mathbf{Q}_m^{-1} \vec{F}_m$. Again, these mesh-dependent data \mathbf{Q}_m^{-1} and $\tilde{\mathbf{W}}_l$ are pre-calculated and stored before the time evolution since they don't depend on the numerical solution u .

2.2 Operator Splitting

The ODE system (13) has a linear term resulting from the diffusion and a nonlinear term coming from the reaction of (1). Both of terms can cause stiffness in the reaction-diffusion models arising in developmental biology (e.g. [3]). Hence we need to use fully implicit schemes to solve (13). In order to avoid solving a large coupled nonlinear system of equations at every time step, We adopted the popular Strang type second-order symmetrical OS schemes [39, 70] to split the diffusion from the reaction terms of (1). The large nonlinear problem is decoupled, hence we can solve the linear diffusion problem and the nonlinear reaction problem individually by implicit temporal schemes. The resulting nonlinear problems are local for each element hence they can be solved efficiently by an iterative method such as Newton's method.

Denote the numerical solution of the ODE system (13) at $t = t^n$ by \vec{V}_m^n . To evolve the system (13) from the time step t^n to t^{n+1} , the classical Strang symmetrical OS scheme [70] combined with the Crank-Nicholson method consists of the following three sub-steps:

Step 1 Apply Crank-Nicholson for the diffusion term at $[t^n, t^{n+\frac{1}{2}}]$:

$$\vec{v}_{0,m} = \vec{V}_m^n, \quad m = 1, \dots, N, \tag{14}$$

$$\vec{v}_{1,m} = \vec{v}_{0,m} + \frac{1}{4} \Delta t \left[D \sum_{l=m,i,j,k} \tilde{W}_l \vec{v}_{0,l} + D \sum_{l=m,i,j,k} \tilde{W}_l \vec{v}_{1,l} \right], \quad m = 1, \dots, N.$$

The sparse linear system (14) is solved by the sparse linear solver “lin_sol_gen_coordinate” of IMSL package.

Step 2 Apply Crank-Nicholson for the reaction term at $[t^n, t^{n+1}]$, with $\vec{v}_{1,m}$ as input data:

$$\vec{v}_{2,m} = \vec{v}_{1,m} + \frac{1}{2} \Delta t [\tilde{F}_m(\vec{v}_{1,m}) + \tilde{F}_m(\vec{v}_{2,m})]. \tag{15}$$

The local nonlinear system (15) on the element m is solved by Newton iterations, with the initial guess $\vec{v}_{1,m}$, for $m = 1, \dots, N$.

Step 3 Apply Crank-Nicholson for the diffusion term at $[t^{n+\frac{1}{2}}, t^{n+1}]$, with $\vec{v}_{2,m}$ as input data:

$$\vec{v}_{3,m} = \vec{v}_{2,m} + \frac{1}{4} \Delta t \left[D \sum_{l=m,i,j,k} \tilde{W}_l \vec{v}_{2,l} + D \sum_{l=m,i,j,k} \tilde{W}_l \vec{v}_{3,l} \right], \quad m = 1, \dots, N, \tag{16}$$

$$\vec{V}_m^{n+1} = \vec{v}_{3,m}, \quad m = 1, \dots, N.$$

Again, the sparse linear system (16) is solved by the solver “lin_sol_gen_coordinate”.

The Trapezoidal OS [39] scheme is a Strang type symmetrical operator splitting method. In stead of using Crank-Nicholson method in the steps 1 and 3, it uses the forward and backward Euler schemes hence is computationally cheaper, and still keeps second order accuracy. We also apply the Trapezoidal OS scheme for (13):

Step 1 Apply forward Euler for the diffusion term at $[t^n, t^{n+\frac{1}{2}}]$:

$$\vec{v}_{0,m} = \vec{V}_m^n, \quad m = 1, \dots, N, \tag{17}$$

$$\vec{v}_{1,m} = \vec{v}_{0,m} + \frac{1}{2} \Delta t \left[D \sum_{l=m,i,j,k} \tilde{W}_l \vec{v}_{0,l} \right], \quad m = 1, \dots, N.$$

Step 2 Apply Crank-Nicholson for the reaction term at $[t^n, t^{n+1}]$, with $\vec{v}_{1,m}$ as input data:

$$\vec{v}_{2,m} = \vec{v}_{1,m} + \frac{1}{2} \Delta t [\tilde{F}_m(\vec{v}_{1,m}) + \tilde{F}_m(\vec{v}_{2,m})]. \tag{18}$$

The local nonlinear system (18) on the element m is solved by Newton iterations, with the initial guess $\vec{v}_{1,m}$, for $m = 1, \dots, N$.

Step 3 Apply backward Euler for the diffusion term at $[t^{n+\frac{1}{2}}, t^{n+1}]$, with $\vec{v}_{2,m}$ as input data:

$$\vec{v}_{3,m} = \vec{v}_{2,m} + \frac{1}{2} \Delta t \left[D \sum_{l=m,i,j,k} \tilde{\mathbf{W}}_l \vec{v}_{3,l} \right], \quad m = 1, \dots, N, \tag{19}$$

$$\vec{V}_m^{n+1} = \vec{v}_{3,m}, \quad m = 1, \dots, N. \tag{20}$$

The sparse linear system (19) is solved by the solver “lin_sol_gen_coordinate”.

2.3 Reaction-Diffusion System

The numerical methods described in Sects. 2.1 and 2.2 can be straightforwardly extended to solve the case of a general system component by component. Consider the system case of (1). Let $u = (u_1, u_2, \dots, u_p)^T$, the diffusion matrix $D = (d_{i_1 i_2})_{p \times p}$, and $F = (F_1, F_2, \dots, F_p)^T$, then the reaction-diffusion system has the expanded form

$$\frac{\partial u_{i_1}}{\partial t} = \sum_{i_2=1}^p (d_{i_1 i_2} \nabla^2 u_{i_2}) + F_{i_1}(u_1, u_2, \dots, u_p), \quad i_1 = 1, 2, \dots, p. \tag{21}$$

Represent the numerical solution on Δ_m by

$$u_{i_1}(x, y, t) = a_{m,i_1}(t) + b_{m,i_1}(t) \xi_m + c_{m,i_1}(t) \eta_m, \quad i_1 = 1, 2, \dots, p. \tag{22}$$

We apply the DG spatial discretization in Sect. 2.1 to each equation of (21). Let

$$\vec{V}_{m,i_1} = \begin{pmatrix} a_{m,i_1}(t) \\ b_{m,i_1}(t) \\ c_{m,i_1}(t) \end{pmatrix}, \quad i_1 = 1, 2, \dots, p,$$

and we have the ODE system

$$\begin{aligned} \vec{V}'_{m,i_1}(t) &= \sum_{l=m,i,j,k} \tilde{\mathbf{W}}_l \left(\sum_{i_2=1}^p d_{i_1 i_2} \vec{V}_{l,i_2}(t) \right) + \tilde{F}_{m,i_1}(\vec{V}_{m,1}, \vec{V}_{m,2}, \dots, \vec{V}_{m,p}), \\ i_1 &= 1, 2, \dots, p, \quad m = 1, \dots, N. \end{aligned} \tag{23}$$

Then the operator splitting technique in Sect. 2.2 is applied to solve the ODE system (23).

3 Numerical Tests on Simple Systems

In this section, we test the stability, accuracy and efficiency of the DG-Strang symmetrical OS scheme (14)–(16) and the DG-trapezoidal OS scheme (17)–(20), for simple parabolic PDEs with an exact solution. First we test the linear stability using both one-dimensional and two dimensional linear parabolic problems. By increasing the time step size successively, we observe the convergence behavior of the schemes. Numerical results for the DG-trapezoidal OS scheme will be presented. To save space, we omit the presentation of the results for the DG-Strang symmetrical OS scheme, which follow a similar stability pattern in our tests. Then using a nonlinear problem, we compare the numerical results for the approaches of DG

methods with OS and without OS. All numerical simulations of this and the next sections are performed on a 2.2 GHz, 8 GB RAM Linux PC.

All of the reaction-diffusion systems considered in the paper are subject to no-flux boundary conditions. If the element edge e_l of Δ_m is aligned with the domain boundary $\partial\Omega$, we take $u^{in}|_{e_l} = u^{ext}|_{e_l}$, and $(\nabla u)^{in}|_{e_l} \cdot \vec{n}_{e_l} = (\nabla u)^{ext}|_{e_l} \cdot \vec{n}_{e_l} = 0$ in the numerical fluxes (3)–(5). Hence we have

$$\widehat{u}|_{e_l} = u^{in}|_{e_l}, \quad \widetilde{\nabla}u|_{e_l} \cdot \vec{n}_{e_l} = 0$$

in the scheme (2).

Example 1 Consider the one-dimensional problem

$$\begin{cases} u_t = u_{xx} - u + \pi^2 e^{-t} \cos(\pi x), & 0 < x < 1, \\ u(x, 0) = \cos(\pi x), \end{cases} \quad (24)$$

with the no-flux boundary conditions. The exact solution is $u(x, t) = e^{-t} \cos(\pi x)$. The simulation is carried up to $T = 8.0$ at which the L^1 , L^2 and L^∞ errors are measured. The time step size is increased successively as $\Delta t = \Delta x, 2\Delta x, 4\Delta x, 8\Delta x$, and $16\Delta x$ where $\Delta x = 1/N$ and N is the number of spatial elements. CPU time, errors, and order of accuracy are reported in Table 1 for different ratios of Δt and Δx . From the Table 1, we can observe that the DG-trapezoidal OS scheme is stable even for a very large time step size, and second order accuracy is obtained. This is also illustrated in Fig. 2.

Example 2 Consider the two-dimensional problem

$$\begin{cases} u_t = u_{xx} + u_{yy} - u + 2\pi^2 e^{-t} \cos(\pi x) \cos(\pi y), & (x, y) \in (0, 1) \times (0, 1); \\ u(x, y, 0) = \cos(\pi x) \cos(\pi y), \end{cases} \quad (25)$$

with the no-flux boundary conditions. The exact solution is $u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y)$. We use triangular meshes, shown in Fig. 3 for the coarsest case of 44 cells, to perform the convergence study. The refinement of the meshes is done in a uniform way, namely by cutting each triangle into four smaller similar ones [89]. The simulation is carried up to $T = 8.0$ at which the L^1 , L^2 and L^∞ errors are measured. The time step size is increased successively as $\Delta t = h_{\min}, 2h_{\min}, 4h_{\min}, 8h_{\min}$, and $16h_{\min}$. CPU time, errors, and order of accuracy are reported in Table 2 for different ratios of Δt and h_{\min} . Similarly to the one-dimensional problem in Example 1, we can observe that the DG-trapezoidal OS scheme is stable even for a very large time step size, and second order accuracy is obtained, from both the results in Table 2 and the illustration Fig. 4.

Example 3 Consider the two-dimensional nonlinear problem

$$\begin{cases} u_t = u_{xx} + u_{yy} - u^2 + e^{-2t} \cos(\pi x)^2 \cos(\pi y)^2 + (2\pi^2 - 1)e^{-t} \cos(\pi x) \cos(\pi y), \\ (x, y) \in (0, 1) \times (0, 1), \\ u(x, y, 0) = \cos(\pi x) \cos(\pi y), \end{cases} \quad (26)$$

with no-flux boundary conditions. The exact solution is $u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y)$. We use the triangular meshes (Fig. 3) as that in Example 2. The simulation is carried up to

Table 1 CPU time, error, and order of accuracy of the DG method with trapezoidal OS for Example 1. Time step size is increased successively. Final time $T = 8.0$

$\Delta t = \Delta x$							
N	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10	0.03	1.23E-06	–	1.36E-06	–	1.93E-06	–
20	0.1	3.09E-07	1.99	3.44E-07	1.99	4.86E-07	1.99
40	0.43	7.75E-08	2.00	8.61E-08	2.00	1.22E-07	2.00
80	1.69	1.94E-08	2.00	2.15E-08	2.00	3.04E-08	2.00
160	6.87	4.85E-09	2.00	5.38E-09	2.00	7.61E-09	2.00
320	28.1	1.21E-09	2.00	1.35E-09	2.00	1.90E-09	2.00
$\Delta t = 2\Delta x$							
N	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10	0.02	3.20E-07	–	3.54E-07	–	4.99E-07	–
20	0.06	7.76E-08	2.04	8.61E-08	2.04	1.22E-07	2.04
40	0.2	1.92E-08	2.01	2.14E-08	2.01	3.02E-08	2.01
80	0.86	4.80E-09	2.00	5.33E-09	2.00	7.54E-09	2.00
160	3.42	1.20E-09	2.00	1.33E-09	2.00	1.88E-09	2.00
320	14.01	3.00E-10	2.00	3.33E-10	2.00	4.71E-10	2.00
$\Delta t = 4\Delta x$							
N	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10	0.01	6.66E-06	–	7.40E-06	–	1.04E-05	–
20	0.03	1.63E-06	2.03	1.82E-06	2.03	2.57E-06	2.02
40	0.11	4.07E-07	2.01	4.52E-07	2.01	6.39E-07	2.01
80	0.42	1.02E-07	2.00	1.13E-07	2.00	1.60E-07	2.00
160	1.71	2.54E-08	2.00	2.82E-08	2.00	3.99E-08	2.00
320	7.01	6.35E-09	2.00	7.05E-09	2.00	9.97E-09	2.00
$\Delta t = 8\Delta x$							
N	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10	0.	3.46E-05	–	3.85E-05	–	5.43E-05	–
20	0.02	8.01E-06	2.11	8.90E-06	2.11	1.26E-05	2.11
40	0.05	1.97E-06	2.03	2.18E-06	2.03	3.09E-06	2.03
80	0.22	4.89E-07	2.01	5.43E-07	2.01	7.68E-07	2.01
160	0.85	1.22E-07	2.00	1.36E-07	2.00	1.92E-07	2.00
320	3.51	3.05E-08	2.00	3.39E-08	2.00	4.80E-08	2.00
$\Delta t = 16\Delta x$							
N	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
10	0.	2.26E-04	–	2.51E-04	–	3.54E-04	–
20	0.	3.61E-05	2.64	4.01E-05	2.64	5.67E-05	2.64
40	0.02	8.36E-06	2.11	9.28E-06	2.11	1.31E-05	2.11
80	0.12	2.05E-06	2.03	2.28E-06	2.03	3.22E-06	2.03
160	0.42	5.10E-07	2.01	5.66E-07	2.01	8.01E-07	2.01
320	1.78	1.27E-07	2.00	1.41E-07	2.00	2.00E-07	2.00

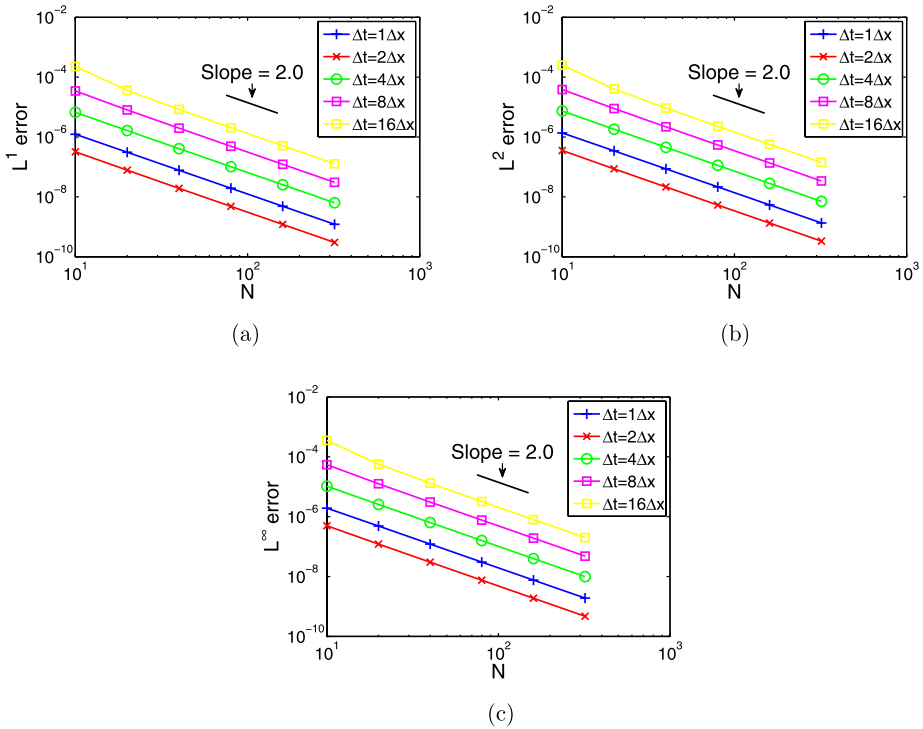
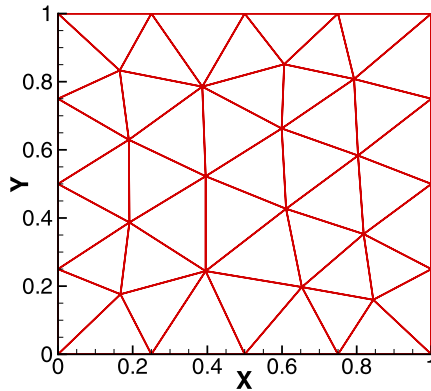


Fig. 2 Error as a function of number of cells N for the DG method with trapezoidal OS for Example 1. (a) N vs. L^1 error; (b) N vs. L^2 error; (c) N vs. L^∞ error

Fig. 3 Coarsest mesh with 44 triangles in the convergence study



$T = 1.0$ at which the L^1 , L^2 and L^∞ errors are measured. The time step size is taken to be $\Delta t = 0.1h_{\min}$. We use the three different approaches: the DG spatial discretization (13) with the Crank-Nicholson temporal discretization (without OS), the DG-Strang symmetrical OS scheme (14)–(16), and the DG-trapezoidal OS scheme (17)–(20). We list the CPU time, errors, and order of accuracy for the DG w/o OS in Table 3, for the DG-Strang symmetrical OS in Table 4, and for the DG-trapezoidal OS in Table 5. The second order accuracy is achieved for all of these three approaches. This is also illustrated in Fig. 5. In Fig. 6, we

Table 2 CPU time, error, and order of accuracy of the DG method with trapezoidal OS for Example 2. Final time $T = 8.0$

$\Delta t = h_{\min}$							
# of cells	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
44	0.17	9.55E-06	–	1.13E-05	–	2.81E-05	–
176	1.44	2.44E-06	1.97	2.91E-06	1.95	7.78E-06	1.85
704	16.39	6.14E-07	1.99	7.37E-07	1.98	2.06E-06	1.91
2816	274	1.53E-07	2.00	1.84E-07	2.00	5.29E-07	1.96
11264	17717	3.52E-08	2.12	4.31E-08	2.10	1.58E-07	1.74
$\Delta t = 2h_{\min}$							
# of cells	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
44	0.08	9.10E-06	–	1.08E-05	–	3.09E-05	–
176	0.76	2.29E-06	1.99	2.76E-06	1.97	8.55E-06	1.85
704	8.92	5.77E-07	2.00	6.95E-07	1.99	2.22E-06	1.94
2816	146	1.44E-07	2.00	1.74E-07	2.00	5.66E-07	1.97
11264	9788	3.54E-08	2.02	4.28E-08	2.02	1.64E-07	1.79
$\Delta t = 4h_{\min}$							
# of cells	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
44	0.04	6.23E-06	–	7.87E-06	–	3.21E-05	–
176	0.4	1.56E-06	2.00	2.01E-06	1.97	8.94E-06	1.84
704	5.02	3.88E-07	2.01	5.03E-07	2.00	2.53E-06	1.82
2816	81.67	9.65E-08	2.01	1.25E-07	2.01	7.31E-07	1.79
11264	5126	2.39E-08	2.03	3.09E-08	2.02	2.08E-07	1.81
$\Delta t = 8h_{\min}$							
# of cells	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
44	0.03	1.59E-05	–	1.96E-05	–	7.81E-05	–
176	0.23	3.54E-06	2.16	4.34E-06	2.20	1.92E-05	2.04
704	3.14	8.71E-07	2.02	1.06E-06	2.03	5.18E-06	1.89
2816	49.68	2.17E-07	2.00	2.65E-07	2.01	1.40E-06	1.89
11264	2803	5.48E-08	1.99	6.68E-08	1.99	3.77E-07	1.89
$\Delta t = 16h_{\min}$							
# of cells	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
44	0.02	1.13E-04	–	1.39E-04	–	2.74E-04	–
176	0.14	2.30E-05	2.30	2.85E-05	2.29	6.88E-05	1.99
704	2.14	4.86E-06	2.24	5.99E-06	2.25	1.54E-05	2.16
2816	33.47	1.19E-06	2.03	1.46E-06	2.03	3.91E-06	1.98
11264	1469	2.96E-07	2.01	3.65E-07	2.01	1.00E-06	1.96

compare the computational time of these three approaches by plotting errors as a function of CPU time, and it is obvious that in order to reach the same numerical error level, the DG-trapezoidal OS approach requires the smallest computational time, and the DG w/o OS

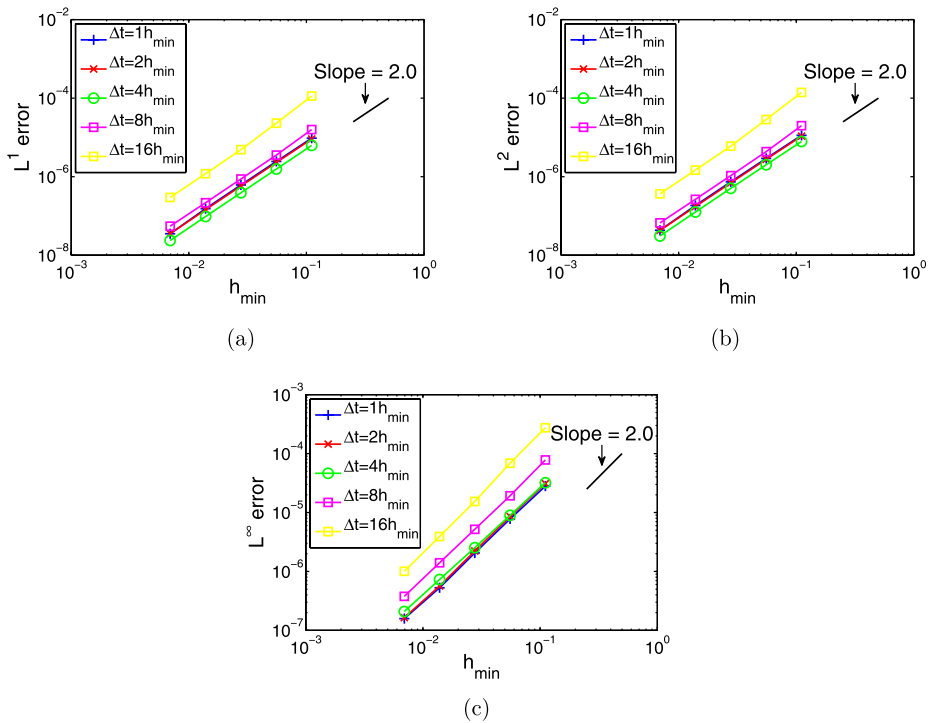


Fig. 4 Error as a function of h_{\min} for the DG method with trapezoidal OS for Example 2. (a) h_{\min} vs. L^1 error; (b) h_{\min} vs. L^2 error; (c) h_{\min} vs. L^∞ error

Table 3 CPU time, error, and order of accuracy of the DG method w/o OS for Example 3. Final time $T = 1.0$

# of cells	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
44	0.65	1.42E-02	–	1.74E-02	–	3.83E-02	–
176	9.28	3.74E-03	1.92	4.56E-03	1.93	1.04E-02	1.89
704	207	9.50E-04	1.98	1.16E-03	1.98	2.73E-03	1.92
2816	8371	2.39E-04	1.99	2.91E-04	1.99	7.41E-04	1.88
11264	809367	5.98E-05	2.00	7.30E-05	2.00	2.12E-04	1.81

Table 4 CPU time, error, and order of accuracy of the DG-Strang symmetrical OS scheme for Example 3. Final time $T = 1.0$

# of Cells	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
44	0.52	1.36E-02	–	1.67E-02	–	3.84E-02	–
176	3.95	3.80E-03	1.84	4.66E-03	1.84	1.29E-02	1.58
704	38	1.01E-03	1.91	1.24E-03	1.91	3.74E-03	1.78
2816	617	2.38E-04	2.09	2.94E-04	2.08	8.18E-04	2.19
11264	42705	6.31E-05	1.92	7.74E-05	1.92	2.34E-04	1.81

Table 5 CPU time, error, and order of accuracy of the DG-trapezoidal OS scheme for Example 3. Final time $T = 1.0$

# of cells	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
44	0.53	1.20E-02	–	1.46E-02	–	3.35E-02	–
176	3.91	3.04E-03	1.98	3.71E-03	1.98	8.93E-03	1.91
704	32	7.62E-04	1.99	9.34E-04	1.99	2.34E-03	1.93
2816	406	1.91E-04	2.00	2.33E-04	2.00	6.46E-04	1.86
11264	22563	4.77E-05	2.00	5.84E-05	2.00	1.88E-04	1.78

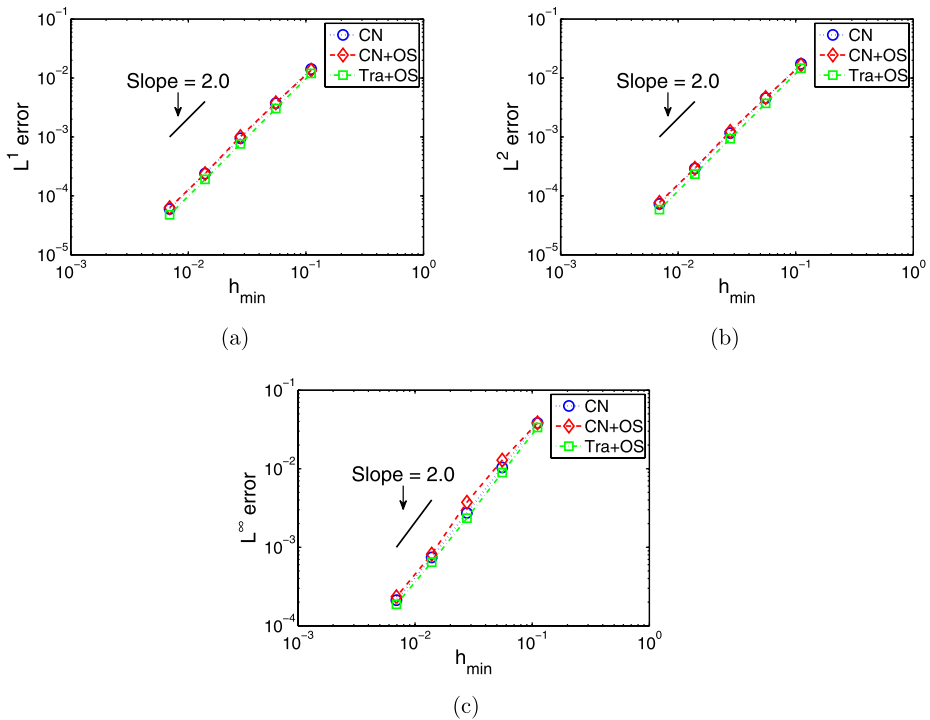


Fig. 5 Error as a function of h_{\min} for three different approaches for Example 3, dotted lines with circles: DG w/o OS; dashed lines with diamonds: DG-Strang symmetrical OS scheme; dash-dotted lines with squares: DG-trapezoidal OS scheme. (a) h_{\min} vs. L^1 error; (b) h_{\min} vs. L^2 error; (c) h_{\min} vs. L^∞ error

is the most expensive. The DG-Strang symmetrical OS approach falls between them. This is due to the fact that at every time step we do not need to solve coupled nonlinear systems by using OS approaches, and the DG-trapezoidal OS only requires solving the linear system once rather than twice as in the DG-Strang symmetrical OS approach.

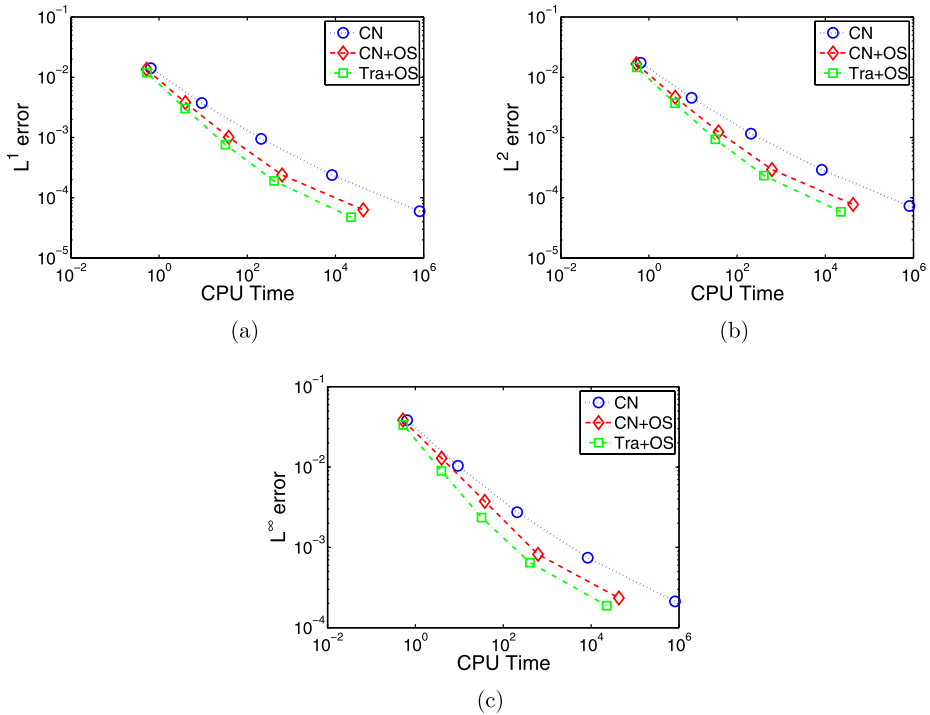


Fig. 6 Error as a function of CPU time for three different approaches for Example 3, *dotted lines with circles*: DG w/o OS; *dashed lines with diamonds*: DG-Strang symmetrical OS scheme; *dash-dotted lines with squares*: DG-trapezoidal OS scheme. (a) CPU time vs. L^1 error; (b) CPU time vs. L^2 error; (c) CPU time vs. L^∞ error

4 Applications to Two Models in Developmental Biology

For proper functioning of tissues, organs and embryos, each cell is required to differentiate appropriately for its position. Position-dependent cell differentiation is often controlled by concentration gradients of morphogens. Morphogens are signaling molecules that become distributed over a tissue domain or field, and when bound to cell receptors, assign different cell fates at different concentrations [72, 79]. Morphogen-based mechanisms have been widely proposed for tissue patterning for over half a century; but only recently have there been sufficient experimental data and adequate modeling for us to begin to understand how various morphogens interact with cells and patterns emerge [35, 44]. In this section, we apply the DG methods with trapezoidal operator splitting (17)–(20) to two reaction-diffusion systems that have been used for modelling morphogen systems in developmental biology. An important outcome of this analysis is a demonstration of the effects of different domain geometries on the patterns of these reaction-diffusion models.

4.1 The Schnakenberg Model

The Schnakenberg system [66] has been used to model the spatial distribution of a morphogen, e.g., the distribution of calcium in the hairs of the whorl in *Acetabularia* [34]. It is

Table 6 CPU time, error, and order of accuracy for DG methods with trapezoidal operator splitting applied to the Schnakenberg model, $T = 1$

Δt	CPU (s)	L^1 error	Order	L^2 error	Order	L^∞ error	Order
5e-4	3400	6.23E-03	–	1.21E-02	–	9.31E-02	–
2.5e-4	5846	1.69E-03	1.88	3.36E-03	1.85	2.58E-02	1.85
1.25e-4	10464	4.33E-04	1.97	8.63E-04	1.96	6.63E-03	1.96
6.25e-5	19018	1.09E-04	1.99	2.16E-04	1.99	1.67E-03	1.99

also a classical example for the testing of numerical methods for reaction-diffusion equations, e.g. [40, 49, 51, 64]. The Schnakenberg system has the form

$$\frac{\partial C_a}{\partial t} = D_1 \nabla^2 C_a + \kappa(a - C_a + C_a^2 C_i), \tag{27}$$

$$\frac{\partial C_i}{\partial t} = D_2 \nabla^2 C_i + \kappa(b - C_a^2 C_i), \tag{28}$$

where C_a and C_i denote the concentration of activator and inhibitor respectively, D_1 and D_2 are diffusion coefficients, κ , a and b are rate constants of the biochemical reactions. Following the setup in [40], we take the initial conditions as

$$C_a(x, y, 0) = a + b + 10^{-3} \exp^{-100((x-\frac{1}{3})^2 + (y-\frac{1}{2})^2)}, \tag{29}$$

$$C_i(x, y, 0) = \frac{b}{(a + b)^2}, \tag{30}$$

and the boundary conditions are taken as homogeneous Neumann. The parameters values are $\kappa = 100$, $a = 0.1305$, $b = 0.7695$, $D_1 = 0.05$, $D_2 = 1$.

First we compute (27)–(30) on the unit square domain $\Omega = (0, 1)^2$. To study the performance and convergence of the DG methods with trapezoidal operator splitting (17)–(20), we list in Table 6 the CPU time, error, and order of accuracy for simulations of the Schnakenberg model (27)–(28). In this case, the spatial resolution is fixed with 1024 elements, 553 nodes and 1576 sides, as shown in Fig. 7a. The error at Δt is measured as a difference between this solution, $C_{a,\Delta t}$, and the solution $C_{a,2\Delta t}$ for time step size $2\Delta t$ at $T = 1$, i.e.,

$$E_{\Delta t} = \|C_{a,\Delta t} - C_{a,2\Delta t}\|. \tag{31}$$

The DG method with trapezoidal operator splitting (17)–(20) clearly shows a second order of accuracy in time as expected. The time evolution of the concentration of activator C_a is shown in Fig. 7. We can observe that the initial perturbation in (29)–(30) is amplified and spreads, leading to formation of spot-like patterns.

Next we vary the shape and size of the domain, but keep all of parameters in the model (27)–(30) unchanged. The time evolution plots of concentrations of activator C_a are shown in Fig. 8 for a circular domain $\Omega = \{(x, y) | (x - 0.5)^2 + (y - 0.5)^2 < 0.5^2\}$, Fig. 9 for an elliptic domain $\Omega = \{(x, y) | (\frac{x-0.5}{0.25})^2 + (\frac{y-0.5}{0.5})^2 < 1\}$, and Fig. 10 for a narrow rectangular domain $\Omega = [0, 0.05] \times [0, 1]$. It is interesting to notice that spot-like patterns are formed on the circular and elliptic domains (Figs. 8, 9), but stripe-like patterns are formed on the narrow rectangular domain Fig. 10. These simulations of the Schnakenberg system on different domains provide an example of the sensitivity of patterns in reaction-diffusion systems with respect to the domain size and shape.

Fig. 7 Numerical solution of the Schnakenberg model on a square domain. Contour plots of time evolution of the concentration of the activator C_a

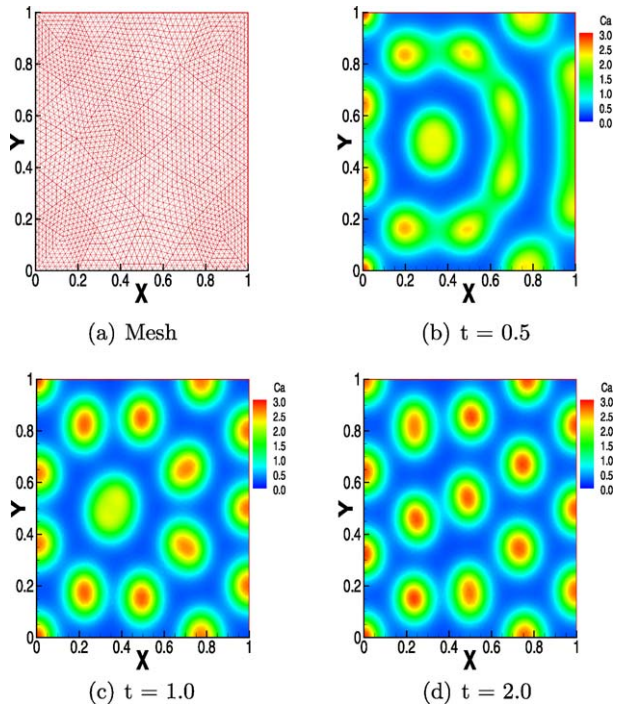


Fig. 8 Numerical solution of the Schnakenberg model on a circular domain. Contour plots of time evolution of the concentration of the activator C_a

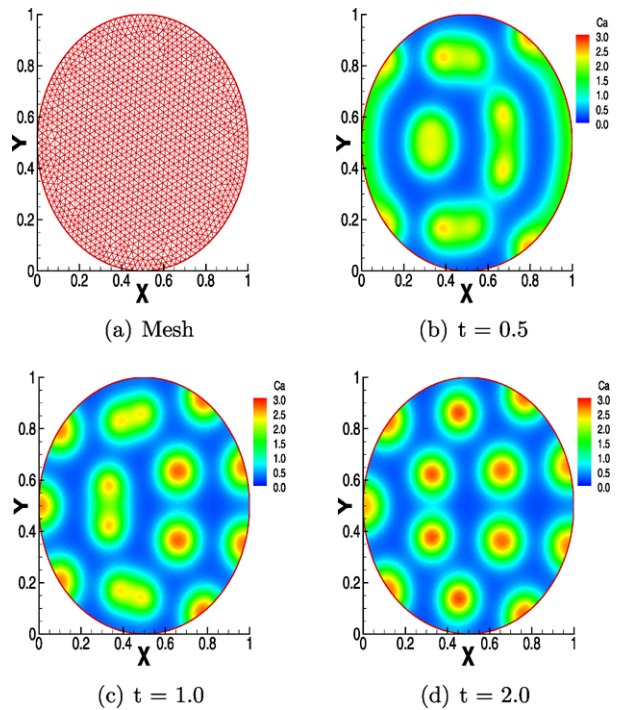


Fig. 9 Numerical solution of the Schnakenberg model on an elliptic domain. Contour plots of time evolution of the concentration of the activator C_a

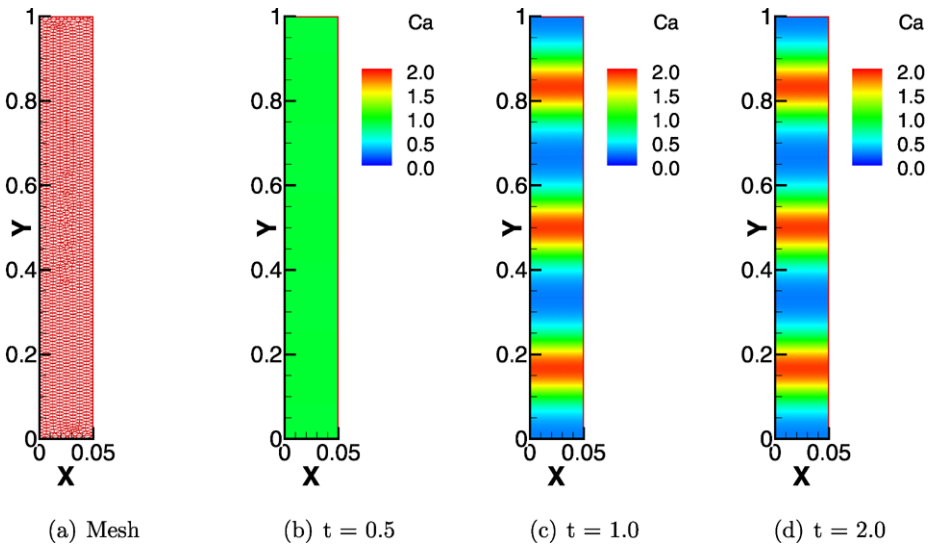
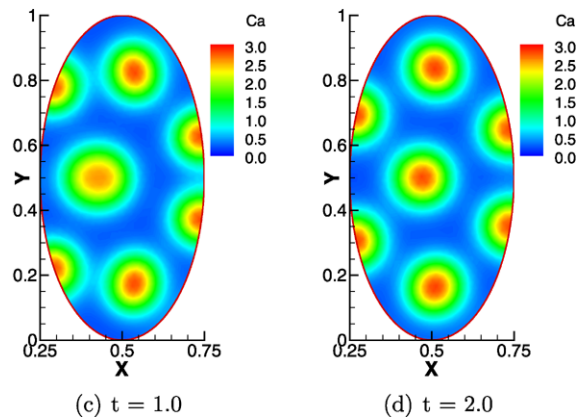
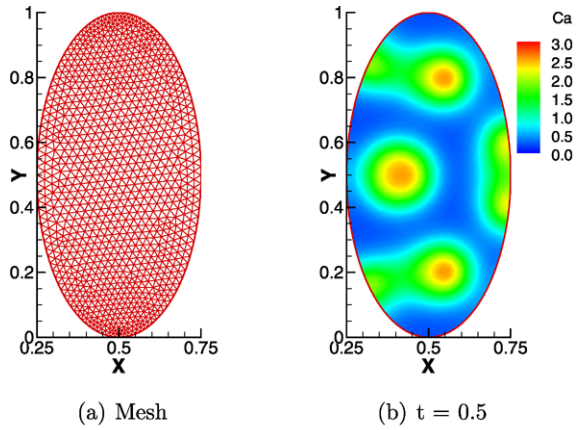


Fig. 10 Numerical solution of the Schnakenberg model on a narrow rectangular domain. Contour plots of time evolution of the concentration of the activator C_a

4.2 A Model for Skeletal Pattern Formation in the Vertebrate Limb

Skeletal patterning in the vertebrate limb, i.e., the spatiotemporal regulation of cartilage differentiation (“chondrogenesis”) during embryogenesis and regeneration, is one of the best studied examples of a multicellular organism developmental process [56, 74]. Limb morphogenesis involves subcellular, cellular and supracellular components that interact in a reliable fashion to produce functional skeletal structures. Since many of the components and interactions are also typical of other embryonic processes, understanding this phenomenon can provide insights into a variety of morphogenetic events in early development.

The limb skeleton consists of nodules and rods of cartilage or bone, arranged in tandem and parallel arrays [57, 58]. It thus lends itself to being modeled by systems like (1), which readily generate spot- and stripe-like patterns.

The most detailed model for vertebrate limb development presented thus far is that of [38], in which a system of eight PDEs was constructed largely on the basis of experimentally determined cellular-molecular interactions occurring in the avian and mouse limb buds. The full system has smooth solutions that exist globally in time but is difficult to handle [2] mathematically and computationally. Recently in [3], by analytically implementing the assumption that cell differentiation relaxes faster than the evolution of the overall cell density, a simplified two-equation system was extracted from the eight-equation system governing the interaction of two of the key morphogens: the activator and an activator-dependent inhibitor of precartilage condensation formation. The reduced reaction-diffusion system has the form

$$\frac{\partial C_a}{\partial t} = D_a \nabla^2 C_a + U(C_a) - k_a C_a C_i, \tag{32}$$

$$\frac{\partial C_i}{\partial t} = D_i \nabla^2 C_i + V(C_a) - k_a C_a C_i, \tag{33}$$

where C_a denotes the concentration of the activator TGF- β , C_i concentration of the inhibitor, D_a and D_i the diffusion constants for the activator and the inhibitor respectively, k_a the inhibitor-activator binding rate, U and V the production rates of C_a and C_i , respectively. The system is subject to no-flux boundary conditions and zero initial concentrations for C_a and C_i . The functions U and V are given by

$$\begin{aligned} U(C_a) &= [J_a^1 \alpha(C_a) + J_a(C_a) \beta(C_a)] R_{eq}, \\ V(C_a) &= J_i(C_a) \beta(C_a) R_{eq}, \end{aligned} \tag{34}$$

where $J_a(C_a) = J_{a \max} (C_a/s)^n / [1 + (C_a/s)^n]$, $J_i(C_a) = J_{i \max} (C_a/\delta)^q / [1 + (C_a/\delta)^q]$, and $\beta(C_a) = \beta_1 C_a / (\beta_2 + C_a)$. Following [3], the parameter values in the system are taken as $D_a = 1$, $D_i = 100.3$, $J_{a \max} = 6.0\lambda$, $J_{i \max} = 8.0\lambda$, $s = 4.0$, $k_a = \lambda$, $J_a^1 \alpha(C_a) = 0.05\lambda$, $\beta_1 = 0.693473$, $\beta_2 = 2.66294$, $R_{eq} = 2.0$, $n = q = 2$, where the value of λ is an important factor which will effect the pattern as shown in the following simulations.

Since the natural shape of a limb bud and its subdomains such as the apical and active zones [38] have non-standard geometries, it is important to study the effects of domain geometry on the pattern generated by the model (32)–(33). We use the DG-trapezoidal OS methods described in Sect. 2 to solve the system on two dimensional domains of different shapes. These domains represent cross-sections of what is actually a three-dimensional paddle-shaped tissue primordium termed the “apical zone”. This zone, which is the locus of

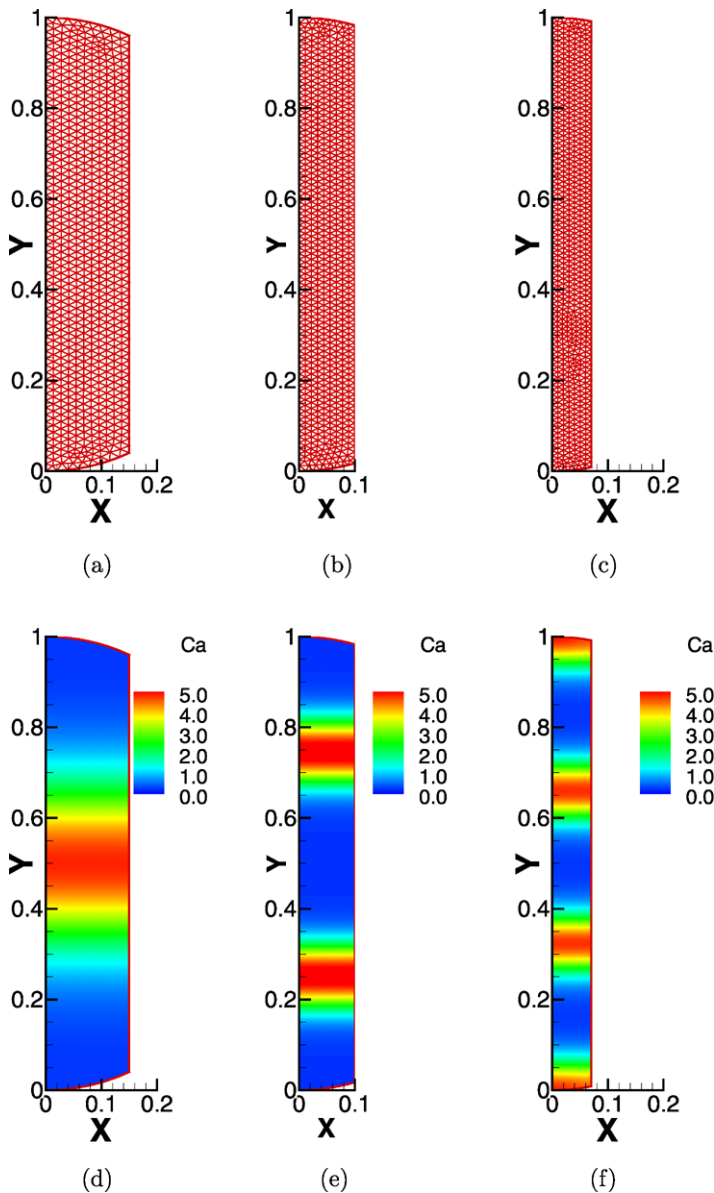


Fig. 11 Numerical solution of the model (32)–(33) on domains with curved *top* and *bottom* boundaries, which are part of the circles $x^2 + (y - 0.7)^2 = 0.3^2$ and $x^2 + (y - 0.3)^2 = 0.3^2$. (a), (b), (c): domains with successive decreasing width and their meshes; (d), (e), (f): Contour plots of the concentration of the activator C_a at time $T = 1.0$ (close to the steady-state)

activator-inhibitor dynamics, initially comprises the entire limb bud, but is increasingly confined to narrower bands of tissue at the limb bud's tip (reviewed in [57]). Triangular meshes are used to fit domains with irregular shapes. First we partition a domain with curved top and bottom boundaries by triangular mesh as shown in Fig. 11a. This domain has height

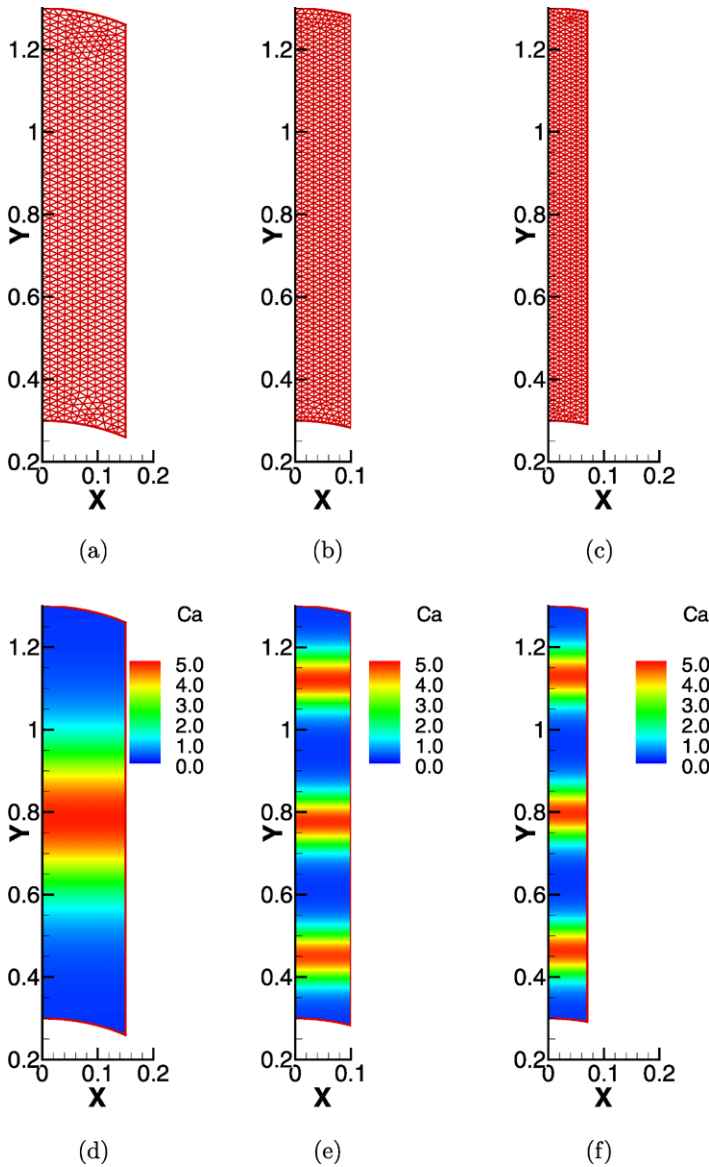


Fig. 12 Numerical solution of the model (32)–(33) on domains with curved top and bottom boundaries, which are part of the circles $x^2 + (y - 1.0)^2 = 0.3^2$ and $x^2 + y^2 = 0.3^2$. (a), (b), (c): domains with successive decreasing width and their meshes; (d), (e), (f): Contour plots of the concentration of the activator C_a at time $T = 1.0$ (close to the steady-state)

1.0 in the vertical direction, and width 0.15 in the horizontal direction. The top and bottom boundaries are parts of the circles $x^2 + (y - 0.7)^2 = 0.3^2$ and $x^2 + (y - 0.3)^2 = 0.3^2$, respectively. A flood contour plot of the concentration of the activator C_a at time $T = 1.0$ (close to the steady-state) is shown in Fig. 11d. The parameter $\lambda = 500$ in this case, and a one-stripe pattern is observed. Next we shrink the domain width in the horizontal direction

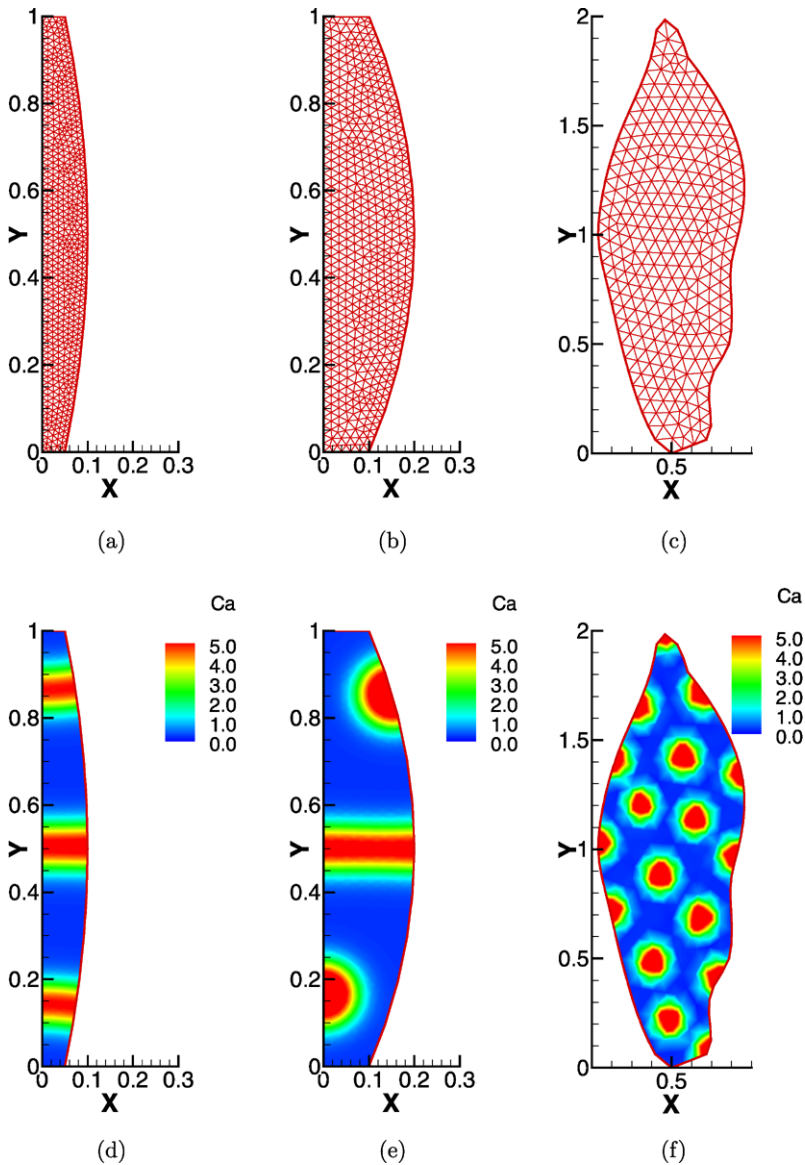


Fig. 13 Numerical solution of the model (32)–(33) on domains with irregular shapes, with same reaction parameters. (a), (b), (c): domains with different shapes and their meshes; (d), (e), (f): Contour plots of the concentration of the activator C_a at time $T = 1.0$ (close to the steady-state)

to be 0.1 and 0.07 respectively, shown in Fig. 11b and c. Two-stripe and three-stripe patterns can be obtained on these shrunken domains if we choose $\lambda = 3900$, shown in Fig. 11e and f. The transitions to higher numbers of parallel stripes of activator correspond to the proximodistal (i.e., from the body wall out to the limb tip) spatiotemporal order of development of increasing numbers of rod-like skeletal elements in most vertebrate limbs [56, 57].

We vary the shape of the domains by changing the top and bottom boundary curves to be part of the circles $x^2 + (y - 1.0)^2 = 0.3^2$ and $x^2 + y^2 = 0.3^2$ respectively, shown in Fig. 12a, b, and c. Again we obtain one, two, and three-stripe patterns on successive shrinking domains with $\lambda = 500, 1900, 3900$ respectively, shown in Fig. 12d, e, and f. This stability of the patterns, and transitions between them, corresponds well to the general robustness of developmental outcome to small changes in tissue shape, which may occur, for example, between related species.

In contrast, if we fix the value of the kinetic parameter λ , and only vary the shape of the domain, we observe the transition of stripe patterns to spot patterns. In the last set of numerical experiments, Fig. 13, the value of λ is fixed to be 6900. We constructed irregular domains with different shapes as shown in Fig. 13a, b, and c. We observe a transition from stripe to spot patterns, as shown in Fig. 13d, e, and f. Configurations of nodular skeletal elements arranged like the activator spots in Fig. 13f, are also found in portions of vertebrate limbs, where they form the wrist and ankle bones that intervene between tiers of rod-like elements.

5 Conclusions

In this paper discontinuous Galerkin (DG) finite element methods, coupled with Strang type symmetrical operator splitting methods, were used for solving reaction-diffusion systems in domains with complex geometry. The general approach for overcoming difficulties related to complex domain geometry was demonstrated on two-dimensional triangular meshes. We have shown improvement of computational efficiency due to the usage of operator splitting technique in a variety of examples.

The new numerical approach resulted in important extensions of two reaction-diffusion models with applications in developmental biology which were then used to demonstrate the effects of domain geometry on the pattern formation. Namely, the methods described here afford the possibility of treating reaction-diffusion and related pattern forming mechanisms in domains with non-standard geometries, including the natural shapes found in embryos and organ primordia. In particular, in the described limb development model described, the effects of varying domain size and shape correspond to those seen under conditions of normal growth and experimental manipulation. More generally, the DG methods provide a means for testing hypotheses for the interaction between mechanisms of pattern formation and changes in tissue geometry during development and evolution [65].

The numerical methods described in the paper can also be extended to numerically solve other reaction-diffusion-advection type systems in mathematical biology including computationally challenging systems modeling chemotaxis [15, 43]. These systems involve advection terms with hyperbolic properties which can be treated by using DG methods with upwind fluxes [18]. This will be studied in the future.

Acknowledgements The research of Y.-T. Zhang is partially supported by Oak Ridge Associated Universities (ORAU) Ralph E. Powe Junior Faculty Enhancement Award. SAN acknowledges support from the National Science Foundation (IBN-034467, EF-0526854). Mark Alber was partially supported by the NSF grant DMS 0719895.

Appendix: Formulae for Mesh-Dependent Constants in (12)

1. The matrix

$$\mathbf{Q}_m = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix} :$$

$$q_{11} = \int_{\Delta_m} dx,$$

$$q_{12} = q_{21} = \int_{\Delta_m} \xi_m dx,$$

$$q_{13} = q_{31} = \int_{\Delta_m} \eta_m dx,$$

$$q_{22} = \int_{\Delta_m} \xi_m^2 dx,$$

$$q_{23} = q_{32} = \int_{\Delta_m} \xi_m \eta_m dx,$$

$$q_{33} = \int_{\Delta_m} \eta_m^2 dx.$$

2. The matrix

$$\mathbf{W}_m = \begin{pmatrix} w_{am1} & w_{bm1} & w_{cm1} \\ w_{am2} & w_{bm2} & w_{cm2} \\ w_{am3} & w_{bm3} & w_{cm3} \end{pmatrix} :$$

$$w_{am1} = \sum_{l=i,j,k} (-\beta r_{1l}),$$

$$w_{bm1} = \sum_{l=i,j,k} \left(\frac{r_{1l} n_{l,x}}{2h_m} - \beta r_{2lm} \right),$$

$$w_{cm1} = \sum_{l=i,j,k} \left(\frac{r_{1l} n_{l,y}}{2h_m} - \beta r_{3lm} \right),$$

$$w_{am2} = \sum_{l=i,j,k} \left(-\frac{r_{1l} n_{l,x}}{2h_m} - \beta r_{2lm} \right),$$

$$w_{bm2} = \sum_{l=i,j,k} (-\beta s_{1mml}),$$

$$w_{cm2} = \sum_{l=i,j,k} \left(\frac{r_{2lm} n_{l,y} - r_{3lm} n_{l,x}}{2h_m} - \beta s_{2mml} \right),$$

$$w_{am3} = \sum_{l=i,j,k} \left(-\frac{r_{1l} n_{l,y}}{2h_m} - \beta r_{3lm} \right),$$

$$w_{bm3} = \sum_{l=i,j,k} \left(\frac{r_{3lm}n_{l,x} - r_{2lm}n_{l,y}}{2h_m} - \beta s_{2mml} \right).$$

$$w_{cm3} = \sum_{l=i,j,k} (-\beta s_{3mml});$$

and the matrix

$$\mathbf{W}_l = \begin{pmatrix} w_{al1} & w_{bl1} & w_{cl1} \\ w_{al2} & w_{bl2} & w_{cl2} \\ w_{al3} & w_{bl3} & w_{cl3} \end{pmatrix}, \quad l = i, j, k :$$

$$w_{al1} = \beta r_{1l},$$

$$w_{bl1} = \frac{r_{1l}n_{l,x}}{2h_l} + \beta r_{2l},$$

$$w_{cl1} = \frac{r_{1l}n_{l,y}}{2h_l} + \beta r_{3l},$$

$$w_{al2} = -\frac{r_{1l}n_{l,x}}{2h_m} + \beta r_{2lm},$$

$$w_{bl2} = \frac{r_{2lm}n_{l,x}}{2h_l} - \frac{r_{2l}n_{l,x}}{2h_m} + \beta s_{1lm},$$

$$w_{cl2} = \frac{r_{2lm}n_{l,y}}{2h_l} - \frac{r_{3l}n_{l,x}}{2h_m} + \beta s_{2lm},$$

$$w_{al3} = -\frac{r_{1l}n_{l,y}}{2h_m} + \beta r_{3lm},$$

$$w_{bl3} = \frac{r_{3lm}n_{l,x}}{2h_l} - \frac{r_{2l}n_{l,y}}{2h_m} + \beta s_{3lm},$$

$$w_{cl3} = \frac{r_{3lm}n_{l,y}}{2h_l} - \frac{r_{3l}n_{l,y}}{2h_m} + \beta s_{4lm},$$

where

$$r_{1l} = \int_{e_l} ds, r_{2l} = \int_{e_l} \xi_l ds, r_{3l} = \int_{e_l} \eta_l ds,$$

$$r_{2lm} = \int_{e_l} \xi_m ds, r_{3lm} = \int_{e_l} \eta_m ds,$$

$$s_{1lm} = \int_{e_l} \xi_m \xi_l ds, s_{2lm} = \int_{e_l} \xi_m \eta_l ds,$$

$$s_{3lm} = \int_{e_l} \xi_l \eta_m ds, s_{4lm} = \int_{e_l} \eta_m \eta_l ds,$$

$$s_{1mml} = \int_{e_l} \xi_m^2 ds, s_{2mml} = \int_{e_l} \xi_m \eta_m ds, s_{3mml} = \int_{e_l} \eta_m^2 ds.$$

References

1. Adjerid, S., Temimi, H.: A discontinuous Galerkin method for higher-order ordinary differential equations. *Comput. Methods Appl. Mech. Eng.* (2008, to appear)
2. Alber, M., Hentschel, H.G.E., Kazmierczak, B., Newman, S.A.: Existence of solutions to a new model of biological pattern formation. *J. Math. Anal. Appl.* **308**, 175–194 (2005)
3. Alber, M., Glimm, T., Hentschel, H.G.E., Kazmierczak, B., Zhang, Y.-T., Zhu, J., Newman, S.A.: The morphostatic limit for a model of skeletal pattern formation in the vertebrate limb. *Bull. Math. Biol.* **70**, 460–483 (2008)
4. Aragón, J.L., Torres, M., Gil, D., Barrio, R.A., Maini, P.K.: Turing patterns with pentagonal symmetry. *Phys. Rev. E* **65**, 1–9 (2002)
5. Aragón, J.L., Varea, C., Barrio, R.A., Maini, P.K.: Spatial patterning in modified Turing systems: application to pigmentation patterns on Marine fish. *Forma* **13**, 213–221 (1998)
6. Arnold, D.: An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.* **19**, 742–760 (1982)
7. Barrio, R.A., Varea, C., Aragón, J.L., Maini, P.K.: A two-dimensional numerical study of spatial pattern formation in interacting systems. *Bull. Math. Biol.* **61**, 483–505 (1999)
8. Barrio, R.A., Maini, P.K., Aragón, J.L., Torres, M.: Size-dependent symmetry breaking in models for morphogenesis. *Physica D* **290**, 1–12 (2002)
9. Bassi, F., Rebay, S.: A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.* **131**, 267–279 (1997)
10. Baumann, C.E., Oden, J.T.: A discontinuous hp finite element method for convection-diffusion problems. *Comput. Methods Appl. Mech. Eng.* **175**, 311–341 (1999)
11. Butcher, J.C.: High order A-stable numerical methods for stiff problems. *J. Sci. Comput.* **25**(1–2), 51–66 (2005)
12. Biswas, R., Devine, K.D., Flaherty, J.: Parallel, adaptive finite element methods for conservation laws. *Appl. Numer. Math.* **14**, 255–283 (1994)
13. Canosa, J.: On a nonlinear diffusion equation describing population growth. *IBM J. Res. Develop.* **17**(4), 307–313 (1973)
14. Cheng, Y., Shu, C.-W.: A discontinuous Galerkin finite element method for time dependent partial differential equations with higher order derivatives. *Math. Comput.* **77**, 699–730 (2008)
15. Childress, S., Percus, J.K.: Nonlinear aspects of chemotaxis. *Math. Biosci.* **56**, 217–237 (1981)
16. Cockburn, B.: Discontinuous Galerkin methods for convection-dominated problems. In: Barth, T.J., Deconinck, H. (eds.) *High-Order Methods for Comput. Phys. Lecture Notes in Computational Science and Engineering*, vol. 9, pp. 69–224. Springer, Berlin (1999)
17. Cockburn, B., Karniadakis, G., Shu, C.-W.: The development of discontinuous Galerkin methods. In: Cockburn, B., Karniadakis, G., Shu, C.-W. (eds.) *Discontinuous Galerkin Methods: Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering*, vol. 11, pp. 3–50. Springer, Berlin (2000), Part I: Overview
18. Cockburn, B., Shu, C.-W.: Runge-Kutta discontinuous Galerkin method for convection-dominated problems. *J. Sci. Comput.* **16**, 173–261 (2001)
19. Cockburn, B., Shu, C.-W.: The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.* **35**, 2440–2463 (1998)
20. Cockburn, B., Lin, S.-Y., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. *J. Comput. Phys.* **84**, 90–113 (1989)
21. Cockburn, B., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Math. Comput.* **52**, 411–435 (1989)
22. Cockburn, B., Hou, S., Shu, C.-W.: The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Math. Comput.* **54**, 545–581 (1990)
23. Cockburn, B., Shu, C.-W.: The Runge-Kutta local projection P^1 -discontinuous Galerkin finite element method for scalar conservation laws. *Math. Model. Numer. Anal.* **25**, 337–361 (1991)
24. Cockburn, B., Shu, C.-W.: The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *J. Comput. Phys.* **141**, 199–224 (1998)
25. Crampin, E.J., Hackborn, W.W., Maini, P.K.: Pattern formation in reaction-diffusion models with nonuniform domain growth. *Bull. Math. Biol.* **64**, 747–769 (2002)
26. Eriksson, K., Johnson, C., Logg, A.: Explicit time-stepping for stiff ODEs. *SIAM J. Sci. Comput.* **25**(4), 1142–1157 (2003)
27. Eldar, A., Dorfman, R., Weiss, D., Ashe, H., Shilo, B.Z., Barkai, N.: Robustness of the BMP morphogen gradient in *Drosophila* embryonic patterning. *Nature* **419**, 304–308 (2002)
28. Eldar, A., Rosin, D., Shilo, B.Z., Barkai, N.: Self-enhanced ligand degradation underlies robustness of morphogen gradients. *Dev. Cell* **5**, 635–646 (2003)

29. Fisher, R.A.: The wave of advance of advantageous genes. *Ann. Eugenics* **7**, 353–369 (1937)
30. Gassner, G., Lorcher, F., Munz, C.-D.: A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *J. Comput. Phys.* **224**, 1049–1063 (2007)
31. Gray, P., Scott, S.K.: Autocatalytic reactions in the isothermal, continuous stirred tank reactor: isolas and other forms of multistability. *Chem. Eng. Sci.* **38**(1), 29–43 (1983)
32. Gray, P., Scott, S.K.: Autocatalytic reactions in the isothermal, continuous stirred tank reactor: oscillations and the instabilities in the system $A + 2B \rightarrow 3B$, $B \rightarrow X$. *Chem. Eng. Sci.* **39**(6), 1087–1097 (1984)
33. Gierer, A., Meinhardt, H.: A theory of biological pattern formation. *Kybernetik* **12**, 30–39 (1972)
34. Goodwin, B.C., Trainor, L.E.H.: Tip and whorl morphogenesis in *Acetabularia* by calcium-regulated strain fields. *J. Theor. Biol.* **117**, 79–106 (1985)
35. Gurdon, J.B., Bourillot, P.Y.: Morphogen gradient interpretation. *Nature* **413**, 797–803 (2001)
36. Hanhart, A.L., Gobbert, M.K., Izu, L.T.: A memory-efficient finite element method for systems of reaction-diffusion equations with non-smooth forcing. *J. Comput. Appl. Math.* **169**, 431–458 (2004)
37. Hairer, E., Wanner, G.: Stiff differential equations solved by Radau methods. *J. Comput. Appl. Math.* **111**, 93–111 (1999)
38. Hentschel, H.G.E., Glimm, T., Glazier, J.A., Newman, S.A.: Dynamical mechanisms for skeletal pattern formation in the vertebrate limb. *Proc. R. Soc. B* **271**, 1713–1722 (2004)
39. Hundsdorfer, W.: Trapezoidal and midpoint splittings for initial-boundary value problems. *Math. Comput.* **67**, 1047–1062 (1998)
40. Hundsdorfer, W., Verwer, J.: *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer, Berlin (2003)
41. Huang, W., Ma, J., Russell, R.D.: A study of moving mesh PDE methods for numerical simulation of blowup in reaction diffusion equations. *J. Comput. Phys.* **227**, 6532–6552 (2008)
42. Johnson, C.: *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge
43. Keller, E.F., Segel, L.A.: Model for chemotaxis. *J. Theor. Biol.* **30**, 225–234 (1971)
44. Lander, A., Nie, Q., Wan, F.: Do morphogen gradients arise by diffusion? *Dev. Cell* **2**(6), 785–796 (2002)
45. Levy, D., Shu, C.-W., Yan, J.: Local discontinuous Galerkin methods for nonlinear dispersive equations. *J. Comput. Phys.* **196**, 751–772 (2004)
46. Liu, H., Yan, J.: The direct discontinuous Galerkin (DDG) methods for diffusion problems. *Math. Comput.* (2008, submitted)
47. Lowrie, R.B.: A comparison of implicit time integration methods for nonlinear relaxation and diffusion. *J. Comput. Phys.* **196**, 566 (2004)
48. Lyons, M.J., Harrison, L.G.: Stripe selection: an intrinsic property of some pattern-forming models with nonlinear dynamics. *Dev. Dyn.* **195**, 201–215 (1992)
49. Madzvamuse, A., Wathen, A.J., Maini, P.K.: A moving grid finite element method applied to a model biological pattern generator. *J. Comput. Phys.* **190**, 478–500 (2003)
50. Madzvamuse, A., Maini, P.K., Wathen, A.J.: A moving grid finite element method for the simulation of pattern generation by Turing models on growing domains. *J. Sci. Comput.* **24**(2), 247–262 (2005)
51. Madzvamuse, A.: 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)
52. Maini, P.K., McElwain, D.L.S., Leavesley, D.: Travelling waves in a wound healing assay. *Appl. Math. Lett.* **17**, 575–580 (2004)
53. Mizutani, C.M., Nie, Q., Wan, F., Zhang, Y.-T., Vilmos, P., Sousa-Neves, R., Bier, E., Marsh, J.L., Lander, A.D.: Formation of the BMP activity gradient in the *Drosophila* embryo. *Dev. Cell* **8**(6), 915–924 (2005)
54. Murray, J.D.: *Mathematical Biology*, vol. II, 3rd edn. Springer, Berlin (2003)
55. Myerscough, M.R., Maini, P.K., Painter, K.J.: Pattern formation in a generalized chemotactic model. *Bull. Math. Biol.* **60**, 1–26 (1998)
56. Newman, S.A., Müller, G.B.: Origination and innovation in the vertebrate limb skeleton: an epigenetic perspective. *J. Exp. Zool. B Mol. Dev. Evol.* **304**, 593–609 (2005)
57. Newman, S.A., Bhat, R.: Activator-inhibitor dynamics of vertebrate limb pattern formation. *Birth Defects Res. C Embryo Today* **81**, 305–319 (2007)
58. Newman, S.A., Christley, S., Glimm, T., Hentschel, H.G.E., Kazmierczak, B., Zhang, Y.-T., Zhu, J., Alber, M.: Multiscale models for vertebrate limb development. *Curr. Top. Dev. Biol.* **81**, 311–340 (2008)
59. Oden, J.T., Babuska, I., Baumann, C.E.: A discontinuous hp finite element method for diffusion problems. *J. Comput. Phys.* **146**, 491–519 (1998)
60. Reed, W.H., Hill, T.R.: Triangular mesh methods for neutron transport equation. Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory (1973)
61. Riviere, B., Wheeler, M.F., Girault, V.: A priori error estimates for finite element methods based on discontinuous approximation spaces for elliptic problems. *SIAM J. Numer. Anal.* **39**(3), 902–931 (2001)

62. Romkes, A., Prudhomme, S., Oden, J.T.: A posteriori error estimation for a new stabilized discontinuous Galerkin method. *Appl. Math. Lett.* **16**(4), 447–452 (2003)
63. Ropp, D.L., Shadid, J.N., Ober, C.C.: Studies of the accuracy of time integration methods for reaction-diffusion equations. *J. Comput. Phys.* **194**, 544–574 (2004)
64. Ruuth, S.: Implicit-explicit methods for reaction-diffusion problems in pattern-formation. *J. Math. Biol.* **34**(2), 148–176 (1995)
65. Salazar-Ciudad, I., Jernvall, J., Newman, S.A.: Mechanisms of pattern formation in development and evolution. *Development* **130**, 2027–2037 (2003)
66. Schnakenberg, J.: Simple chemical reaction systems with limit cycle behavior. *J. Theor. Biol.* **81**, 389–400 (1979)
67. Shimmi, O., Umulis, D., Othmer, H., O'Connor, M.: Facilitated transport of a Dpp/Sew heterodimer by Sog/Tsg leads to robust patterning of the Drosophila blastoderm embryo. *Cell* **120**, 873–886 (2005)
68. Soane, A.M., Gobbert, M.K., Seidman, T.I.: Numerical exploration of a system of reaction-diffusion equations with internal and transient layers. *Nonlinear Anal. Real World Appl.* **6**, 914–934 (2005)
69. Sportisse, B.: An analysis of operating splitting techniques in the stiff case. *J. Comput. Phys.* **161**, 140–168 (2000)
70. Strang, G.: On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.* **8**(3), 506–517 (1968)
71. Sun, W., Tang, T., Ward, M.J., Wei, J.: Numerical challenges for resolving spike dynamics for two one-dimensional reaction-diffusion systems. *Stud. Appl. Math.* **111**, 41–84 (2003)
72. Teleman, A.A., Strigini, M., Cohen, S.M.: Shaping morphogen gradients. *Cell* **105**, 559–562 (2001)
73. Thomas, D.: Artificial enzyme membrane, transport, memory and oscillatory phenomena. In: Thomas, D., Kervenez, J.-P. (eds.) *Analysis and Control of Immobilised Enzyme Systems*, pp. 115–150. Springer, Berlin (1975)
74. Tickle, C.: Patterning systems—from one end of the limb to the other. *Dev. Cell* **4**, 449–458 (2003)
75. Turing, A.M.: The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond. B* **237**, 37–72 (1952)
76. van Leer, B., Nomura, S.: Discontinuous Galerkin for diffusion. In: 17th AIAA Computational Fluid Dynamics Conference, June 6–9, 2005. AIAA, Washington (2005). AIAA paper 2005-5108
77. Verwer, J.G., Sommeijer, B.P., Hundsdorfer, W.: RKC time-stepping for advection-diffusion-reaction problems. *J. Comput. Phys.* **201**, 61–79 (2004)
78. Wang, Y.-C., Ferguson, E.L.: Spatial bistability of Dpp-receptor interactions during Drosophila dorsal-ventral patterning. *Nature* **434**, 229–234 (2005)
79. Wolpert, L., Beddington, R., Brockes, J., Jessel, T., Lawrence, P., Meyerowitz, E.: *Principles of Development*. Oxford University Press, London (2002)
80. Xu, Y., Shu, C.-W.: Local discontinuous Galerkin methods for three classes of nonlinear wave equations. *J. Comput. Math.* **22**, 250–274 (2004)
81. Xu, Y., Shu, C.-W.: Local discontinuous Galerkin methods for nonlinear Schrodinger equations. *J. Comput. Phys.* **205**, 72–97 (2005)
82. Xu, Y., Shu, C.-W.: Local discontinuous Galerkin methods for two classes of two dimensional nonlinear wave equations. *Physica D* **208**, 21–58 (2005)
83. Xu, Y., Shu, C.-W.: Local discontinuous Galerkin methods for the Kuramoto-Sivashinsky equations and the Ito-type coupled KdV equations. *Comput. Methods Appl. Mech. Eng.* **195**, 3430–3447 (2006)
84. Yan, J., Shu, C.-W.: A local discontinuous Galerkin method for KdV type equations. *SIAM J. Numer. Anal.* **40**, 769–791 (2002)
85. Yan, J., Shu, C.-W.: Local discontinuous Galerkin methods for partial differential equations with higher order derivatives. *J. Sci. Comput.* **17**, 27–47 (2002)
86. Yanenko, N.N.: *The Method of Fractional Steps*. Springer, New York (1971)
87. Zegeling, P.A., Kok, H.P.: Adaptive moving mesh computations for reaction-diffusion systems. *J. Comput. Appl. Math.* **168**(1–2), 519–528 (2004)
88. Zhang, Y.-T., Lander, A., Nie, Q.: Computational analysis of BMP gradients in dorsal-ventral patterning of the zebrafish embryo. *J. Theor. Biol.* **248**(4), 579–589 (2007)
89. Zhang, Y.-T., Shu, C.-W.: High order WENO schemes for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comput.* **24**, 1005–1030 (2003)
90. Zykov, V., Engel, H.: Dynamics of spiral waves under global feedback in excitable domains of different shapes. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **70**, 016201 (2004)