APPLICATION OF DISCONTINUOUS GALERKIN FINITE ELEMENT
METHODS FOR VERTEBRATE LIMB PATTERN FORMATION

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Jianfeng Zhu

______________________________
Mark Alber, Co-Director

______________________________
Yongtao Zhang, Co-Director

Graduate Program in Mathematics
Notre Dame, Indiana
April 2010
APPLICATION OF DISCONTINUOUS GALERKIN FINITE ELEMENT METHODS FOR VERTEBRATE LIMB PATTERN FORMATION

Abstract

by

Jianfeng Zhu

Major outstanding questions regarding vertebrate limb development are how the numbers of skeletal elements along the proximodistal (P-D) and anteroposterior (A-P) axes are determined and how the shape of a growing limb affects skeletal element formation. A mechanism based on local autoregulation of a molecular activator of cell aggregation coupled to a laterally acting inhibitor (a LALI system), is consistent with in vivo and in vitro experimental results and provides qualitative interpretations of several genetic anomalies affecting limb development. Nonlinear reaction-diffusion systems are often employed in mathematical modeling to study the activator-inhibitor subnetwork in developmental biology. These systems are usually highly stiff in both diffusion and reaction terms and are typically considered on multidimensional complex geometrical domains because of complex shapes of embryos. Using an empirically based mathematical representation of such reaction-diffusion mechanism and combining discontinuous Galerkin (DG) finite element methods with Strang type symmetrical operator splitting technique that permits simulation of LALI systems in domains of varying shape and size, we show that major aspects of the limb pattern, including those of aberrant and evolutionary transitional forms, emerge in a robust fashion from the inher-
Jianfeng Zhu

ent self-organizing properties of a core skeletal patterning mechanism in different geometric settings, without a requirement for positional information.

**Key Words:** discontinuous Galerkin finite element methods, reaction-diffusion equations, operator splitting, triangular meshes, moving domain, complex geometry, limb development
Dedicated to my family

Father: Zhu Longhua
Mother: Zhu XianLian
Brother: Zhu Jingming
CHAPTER 5: MOVING GRIDS AND VERTEBRATE LIMB PATTERN

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Cellular and molecular dynamics of limb formation</td>
<td>106</td>
</tr>
<tr>
<td>5.2</td>
<td>Computation methods</td>
<td>110</td>
</tr>
<tr>
<td>5.3</td>
<td>Computation results</td>
<td>112</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Normal development</td>
<td>112</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Anteroposteriorly expanded limb buds</td>
<td>117</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Fossils</td>
<td>122</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY | 143
FIGURES

2.1 Typical stencil ................................................. 22
2.2 Error as a function of cells $N$ for three different approaches for Example 2.1, 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. ............ 25
2.3 Error as a function of CPU time for three different approaches for Example 2.1, 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. ............ 26
2.4 Error as a function of cells $N$ for three different approaches for Example 2.2, 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. ............ 29
2.5 Error as a function of CPU time for three different approaches for Example 2.2, 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. ............ 30
2.6 Error as a function of number of cells $N$ for the DG- trapezoidal OS for Example 2.3. ......................... 34
2.7 Error as a function of $h_{\min}$ for three different approaches for Example 2.4, 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. ............ 37
2.8 Error as a function of CPU time for three different approaches for Example 2.4, 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. ............ 38
2.9 Error as a function of $h_{\text{min}}$ for three different approaches for Example 2.5, 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.

2.10 Error as a function of CPU time for three different approaches for Example 2.5, 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.

2.11 Error as a function of $h_{\text{min}}$ for the DG-trapezoidal OS for Example 2.6.

2.12 Error as a function of $h_{\text{min}}$ for three different approaches for Example 2.7, 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.

2.13 Error as a function of CPU time for three different approaches for Example 2.7, 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.

2.14 Error as a function of $h_{\text{min}}$ for the DG-trapezoidal OS for Example 2.8.

2.15 Error as a function of $h_{\text{min}}$ for three different approaches for Example 2.9, 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.

2.16 Error as a function of CPU time for three different approaches for Example 2.9, 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.

2.17 Error as a function of $h_{\text{min}}$ for three different approaches for Example 2.10, 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.

2.18 Error as a function of CPU time for three different approaches for Example 2.10, 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.

2.19 Error as a function of $h_{\text{min}}$ for the DG-trapezoidal OS for Example 2.11.
2.20 The coarsest triangular mesh of 44 cells ........................................ 67

3.1 Triangular mesh for Schnakenberg model of 1024 cells ................. 71

3.2 Numerical solution of the Schnakenberg model on \((0,1)^2\). Contour plots of time evolution of the concentration of the activator \(Ca\). .......................... 72

3.3 Numerical solution of the Schnakenberg model on circular domain \(\{(x,y)|x^2 + y^2 < 1\}\). Contour plots of time evolution of the concentration of the activator \(Ca\)................................. 73

3.4 Numerical solution of the Schnakenberg model on circular domain \(\{(x,y)|(x-0.5)^2 + (y-0.5)^2 < 0.5^2\}\). Contour plots of time evolution of the activator \(Ca\) ......... 74

3.5 Numerical solution of the Schnakenberg model on elliptic domain \(\{(x,y)|(x-0.5)^2 + y^2 < 1\}\). Contour plots of time evolution of the activator \(Ca\) ....................... 75

3.6 Numerical solution of the Schnakenberg model on elliptic domain \(\{(x,y)|(\frac{x}{0.5})^2 + (\frac{y}{0.5})^2 < 1\}\). Contour plots of time evolution of the activator \(Ca\) ....................... 76

3.7 Numerical solution of the Schnakenberg model on elliptic domain \(\{(x,y)|(\frac{x-0.5}{0.1})^2 + (\frac{y-0.5}{0.5})^2 < 1\}\). Contour plots of time evolution of the activator \(Ca\) ....................... 77

3.8 Numerical solution of the Schnakenberg model on \((0,0.05) \times (0,1)\). Contour plots of time evolution of the concentration of the activator \(Ca\) ................................. 78

3.9 Numerical solution of the Schnakenberg model on \((0,0.1) \times (0,1)\). Contour plots of time evolution of the concentration of the activator \(Ca\) ................................. 79

3.10 Numerical solution of the model 3.6 - 3.7 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 \(Ca\) at time \(T = 1.0\) (close to the steady-state) ........................................ 80

3.11 Numerical solution of the model 3.6 - 3.7 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 \(Ca\) at time \(T = 1.0\) (close to the steady-state) .......................... 81
3.12 Numerical solution of the model 3.6 - 3.7 on domains with irregular shapes, with $\lambda = 3900$. (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). ... 85

3.13 Numerical solution of the model 3.6 - 3.7 on domains with irregular shapes, with $\lambda = 6900$. (a), (b), (c), (d): domains with different shapes and their meshes; (e), (f), (g), (h): Contour plots of the concentration of the activator $C_a$ at time $T = 1.0$ (close to the steady-state). ... 86

5.1 Initial Meshes ................................. 112
5.2 FGF concentration .............................. 114
5.3 Relationship between core cartilage patterning network and bare bones framework for limb development. ...................... 130
5.4 Developmental progression of chicken forelimb between days 3 and 7 of development (indicated by the corresponding Hamburger-Hamilton stages). Early cartilage, including precartilage condensations, shown in light blue; definitive cartilage shown in darker blue [80]. ... 131
5.5 A sequence of snapshots from the simulation of normal limb development(1). Color legend: black/red/violet corresponds to 6.0, white/blue to 0.0, green to the median level 3.0. .................. 132
5.6 A sequence of snapshots from the simulation of normal limb development(2). Color legend: black/red/violet corresponds to 6.0, white/blue to 0.0, green to the median level 3.0. .................. 133
5.7 Dependence of the number of stripes on the kinetic parameters $\lambda$ and $\delta$ during the first phase of normal development in Figure 5.5. 134
5.8 Simulations on a shrinking domain with curved apical zone. ... 134
5.9 LALI zone fixed, other parameter changes as normal development in Figure 5.5. Color legend: black corresponds to 5.6, white to 0.0. 135
5.10 LALI zone narrows as Figure 5.5 but $\lambda$ and $\delta$ remain unchanged. Color legend: black corresponds to 5.6, white to 0.0. ................. 135
5.11 Number of parallel stripes in the model limb bud increases with expansion of the A-P ($y$) direction length of the LALI zone. Color legend: black corresponds to 5.5, white to 0.0. ................. 136
5.12 Top: an intact chicken wing bud at an early stage of development and the limb skeleton that it generates. Middle: a wing bud at the same early stage with the AER removed, and the resulting limb skeleton, which attains a normal size but is truncated beginning at the elbow. Bottom: a later stage wing bud whose AER has been removed. The resulting skeleton is truncated from the wrist onward.

5.13 Effect of limb bud distal expansion

5.14 Simulations of limb bud distal expansion. Color legend: for (a), (b), (c), black/red/violet corresponds to 6.1, white/blue to 0.0, green to the median level 3.05; for (d), (e), black/red/violet corresponds to 6.5, white/blue to 0.0, green to the median level 3.25.

5.15 Fossil limb skeletons

5.16 Simulations of fossil limb skeletons. Color legend: black/red/violet corresponds to 6.2, white/blue to 0.0, green to the median level 3.1.
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.1</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.2</td>
<td>28</td>
</tr>
<tr>
<td>2.3</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY OF THE DG-TRAPEZOIDAL OS FOR EXAMPLE 2.3</td>
<td>32</td>
</tr>
<tr>
<td>2.4</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.4</td>
<td>36</td>
</tr>
<tr>
<td>2.5</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.5</td>
<td>40</td>
</tr>
<tr>
<td>2.6</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY OF THE DG-TRAPEZOIDAL OS FOR EXAMPLE 2.6</td>
<td>44</td>
</tr>
<tr>
<td>2.7</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.7</td>
<td>48</td>
</tr>
<tr>
<td>2.8</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY OF THE DG-TRAPEZOIDAL OS FOR EXAMPLE 2.8</td>
<td>52</td>
</tr>
<tr>
<td>2.9</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.9</td>
<td>56</td>
</tr>
<tr>
<td>2.10</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.10</td>
<td>60</td>
</tr>
<tr>
<td>2.11</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY OF THE DG-TRAPEZOIDAL OS FOR EXAMPLE 2.11</td>
<td>64</td>
</tr>
<tr>
<td>3.1</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR DG-TRAPEZOIDAL OS APPLIED TO THE SCHNAKENBERG MODEL</td>
<td>71</td>
</tr>
<tr>
<td>4.1</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 4.1</td>
<td>100</td>
</tr>
<tr>
<td>4.2</td>
<td>CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 4.2</td>
<td>102</td>
</tr>
</tbody>
</table>
4.3 CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 4.3 ........................................... 104
4.4 CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 4.4 ........................................... 105
5.1 PARAMETER VALUES ........................................... 116
ACKNOWLEDGMENTS

First, I would like to thank my two advisors, Professor Mark Alber and Professor Yongtao Zhang at the University of Notre Dame for their continuous mentoring and constant encouragement for my research. It is also a pleasure to thank Professor Stuart Newman from the New York Medical College for his strong support and dedicated help in developmental biology. Without them, I could not have completed this project.

Next, I would like to thank Professor Bei Hu, Professor Andrew Sommese, and Professor Zhiliang Xu for being my dissertation readers. I really appreciate their help in my research and suggestions in my dissertation.

My group members including Nan Chen, Matt Rissler, Richard Gejji, Yuan Liu helped me a lot in the C and C++ codes which I used for this project. Other people who helped out were the administrators, secretaries of the Mathematics department, administrators of the Center for Research Computing (CRC) computer clusters. I used the CRC computers for all the mathematical computations and numerical simulations. The Center of Applied Mathematics deserves recognition for supporting my research.

At last, I would like to thank my family who always supporting my education.
CHAPTER 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

\[
\frac{\partial u}{\partial t} = D \nabla^2 u + F(u),
\]

where \( u \in \mathbb{R}^p \) 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.), \( D \in \mathbb{R}^{p \times p} \) 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 [104] such as the Gierer-Meinhardt model [38], the Schnakenberg model [92], the Thomas model [101], the Gray and Scott model [40, 41] and others described in [5, 6, 8, 9, 75]. Related systems include the models to study the robustness of gene network [30, 31, 55, 71, 93, 107, 120], and Fisher’s equation [14, 33] with various applications including tissue engineering (e.g. [65]) and gene propagation (e.g. [33]). Although equation 1.1 has a linear diffusion part, the nonlinear reaction part is usually complicated. Efficient and accurate numerical methods for equation 1.1 type of semi-linear PDEs are essential when we carry out the parameter studies and computational analysis for systems.
in developmental biology.

Because of the complex shapes of embryos and their parts, the reaction-diffusion equation 1.1 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. [4, 28, 61, 76, 121]). Finite element numerical methods on unstructured meshes are powerful means for handling the complicated domain geometries [49]. In [44, 62–64, 89, 95], continuous Galerkin (CG) finite element methods were used to solve examples of the reaction-diffusion equation 1.1 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 [86], 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 [22–26], 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 h-p adaptivity (combinations of refining/unrefining elements (h-adaptivity) and changing order of base function (p-adaptivity)) including changes of approximation orders between neighboring elements and allowing 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 DG method and its recent development include the survey paper [27], and other papers in the same Springer volume, as well as several review articles [19, 20]. 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 [21], motivated by the successful numerical experiments of Bassi and Rebay [10] 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 in [57, 109, 114]. For alternative DG methods for diffusion problems, see [7, 11, 59, 83, 87, 88].

In a recent paper [16], Cheng and Shu developed a new DG method for solving time dependent PDEs with higher order spatial derivatives, based on [1, 37, 105]. 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 save storage and computational cost [16]. In this thesis, we have adopted the DG approaches of [16] for the spatial discretization of reaction-diffusion equation 1.1 on both one and two dimensional meshes.

Another computational challenge comes from the stiffness of reaction-diffusion
equation 1.1 and the DG spatial discretization operator, which would require efficient time discretization techniques. On one hand, standard explicit methods are highly inefficient because of their severe stability constraint, hence stabilized explicit methods were developed (e.g. [32, 106]). On the other hand, the implicit methods (e.g. [13, 43, 60]) 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. [89, 96, 97, 115]). We adopted the Strang type second-order symmetrical OS schemes [47, 97] to split the diffusion from the reaction terms of 1.1. 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 1.1 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 methods.

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 [89]. 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. [46, 98, 119]). 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 [27]. 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 the next chapter, we describe in details the DG spatial discretization combined with Strang type operator splitting and the Crank-Nicholson temporal discretization to solve the reaction-diffusion systems 1.1 on fixed meshes. We test the numerical methods for reaction-diffusion equations with exact solutions, and show its good stability and accuracy. We compare the cases with operator splitting and without operator splitting to demonstrate the computational efficiency enhancement due to the operator splitting technique. In chapter 3, we apply the methods to two reaction-diffusion models used in developmental biology, the classical Schnakenberg model [92] and a model for skeletal pattern formation during chicken limb development [4]. In chapter 4, we describe the Strang type symmetrical operator splitting on moving meshes and apply it to the skeletal pattern formation model in chapter 5. We show by various simulations of these models the effect on resulting patterns of varying domain sizes and shapes.
CHAPTER 2
NUMERICAL METHODS ON FIXED GRIDS

2.1 The DG spatial discretization
2.1.1 One dimension

We follow the same notations of the DG method as in [16]. For a given interval
$I = [a, b]$, divide it into $N$ cells as follows,

\[
a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = b. \tag{2.1}\]

Denote

\[I_m = (x_{m-\frac{1}{2}}, x_{m+\frac{1}{2}}), \quad x_m = \frac{x_{m-\frac{1}{2}} + x_{m+\frac{1}{2}}}{2}, \tag{2.2}\]

and

\[h_m = x_{m+\frac{1}{2}} - x_{m-\frac{1}{2}}, \quad h = \min_m h_m. \tag{2.3}\]

Define the finite element space

\[V_h^k = \{v : v|_{I_m} \in P^k(I_m), m = 1, \cdots, N\}, \tag{2.4}\]

where $P^k(I_m)$ denotes the set of all polynomials of degree at most $k$ on $I_m$.

We apply the DG formulation [16] to discretize the reaction-diffusion equation
\[1.1\] in the spatial direction, but keep the time variable continuous. The semi-
discrete scheme is: find $u \in V_h^k$, such that

$$
\int_{I_m} u_t v dx - D \int_{I_m} u v_{x} dx + D(\hat{u}v^-_{x})_{m+\frac{1}{2}} - D(\hat{u}v^+_{x})_{m-\frac{1}{2}}
-D(\bar{u}v^-_{x})_{m+\frac{1}{2}} + D(\bar{u}v^+_{x})_{m-\frac{1}{2}} = \int_{I_m} F(u)v dx
$$

(2.5)

holds true for any $v \in V_h^k$ and $m = 1, \ldots, N$. The numerical fluxes $\hat{u}$ and $\bar{u}_x$ are chosen to be

$$
\hat{u} = \frac{u^+ + u^-}{2},
$$

(2.6)

$$
\bar{u}_x = \frac{u^+_x + u^-_x}{2} + \beta [u],
$$

(2.7)

where the jump term

$$
[u] = u^+ - u^-.
$$

(2.8)

$u^+_{m+\frac{1}{2}}, u^-_{m+\frac{1}{2}}$ are the limit of value $u$ at $x_{m+\frac{1}{2}}$ from the left cell $I_m$ and the right cell $I_{m+1}$, respectively. $\beta$ is a positive constant that is of the order $O(h^{-1})$. The choice of numerical fluxes [2.6] - [2.8] is crucial for the stability and convergence of the DG scheme [2.5]. See [16, 21] for more discussions about the choice of numerical fluxes.

Here 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. To simplify notations in the following presentation, we just use $I_i$ and $I_j$ to represent the two neighboring elements $I_{m-1}$ and $I_{m+1}$ of element $I_m$, respectively. The liner polynomial on $I_m$ is represent by

$$
u(x, t) = a_m(t) + b_m(t)\xi_m,$$

(2.9)
where
\[ \xi_m = 2 \frac{x - x_m}{h_m}. \] (2.10)

\( \xi_m \) takes 1 at the right ending point \( x_{m+\frac{1}{2}} \) and \(-1\) at the left ending point \( x_{m-\frac{1}{2}} \). By taking \( v = 1 \), \( \xi_m \) on \( I_m \) and \( v = 0 \) elsewhere, the DG formulation (2.5) is converted from the integral form to the following system, for \( m = 1, \cdots, N \):

\[ q_{11} a_m'(t) + q_{12} b_m'(t) = D \left\{ w_{am1} a_m(t) + w_{bm1} b_m(t) + \sum_{i,j} [w_{al1} a_i(t) + w_{bl1} b_i(t)] \right\} + \left( \frac{q_{11}}{2} \right) \sum_{i=1,2} F(u(x_{m,i})), \] (2.11)

\[ q_{21} a_m'(t) + q_{22} b_m'(t) = D \left\{ w_{am2} a_m(t) + w_{bm2} b_m(t) + \sum_{i,j} [w_{al2} a_i(t) + w_{bl2} b_i(t)] \right\} + \left( \frac{q_{11}}{2} \right) \sum_{i=1,2} F(u(x_{m,i})) \xi_m(x_{m,i}), \] (2.12)

where the coefficients \( \{q_{rs}\}_{r,s=1}^2, \{w_{alr}\}_{r=1}^2, \{w_{blr}\}_{r=1}^2 \}_{l=m,i,j} \) are constants which depend on the local geometry of the mesh (i.e., cell \( I_m \) and its neighboring cells \( I_i, I_j \)), the local basis functions \( 1, \{\xi_l\}_{l=m,i,j} \), and the constant \( \beta \). \( \{x_{m,i}\}_{l=1,2} \) are the Gaussian quadrature points for the integral involving the nonlinear reaction terms in equation (2.5) and take

\[ x_{m,1} = \frac{\sqrt{3} + 1}{2\sqrt{3}} x_{m-\frac{1}{2}} + \frac{\sqrt{3} - 1}{2\sqrt{3}} x_{m+\frac{1}{2}}, \] (2.13)

\[ x_{m,2} = \frac{\sqrt{3} - 1}{2\sqrt{3}} x_{m-\frac{1}{2}} + \frac{\sqrt{3} + 1}{2\sqrt{3}} x_{m+\frac{1}{2}}. \] (2.14)

These mesh-dependent constants are pre-calculated before the time evolution since they don’t depend on the numerical solution \( u \). Rewrite equations (2.11) (2.12) to
the matrix-vector form

\[ Q_m \vec{V}_m'(t) = D \sum_{l=m,i,j} W_l \vec{V}_l(t) + \vec{F}_m(\vec{V}_m) \] (2.15)

where

\[
Q_m = \begin{pmatrix}
q_{11} & q_{12} \\
q_{21} & q_{22}
\end{pmatrix} = \begin{pmatrix}
\int_{I_m} dx & \int_{I_m} \xi_m dx \\
\int_{I_m} \xi_m dx & \int_{I_m} \xi_m^2 dx
\end{pmatrix} = \begin{pmatrix}
h_m & 0 \\
0 & \frac{h_m}{3}
\end{pmatrix}
\] (2.16)

\[
W_m = \begin{pmatrix}
w_{am1} & w_{bm1} \\
w_{am2} & w_{bm2}
\end{pmatrix} = \begin{pmatrix}
-2\beta & 0 \\
0 & -2\beta
\end{pmatrix}
\] (2.17)

\[
W_i = \begin{pmatrix}
w_{ai1} & w_{bi1} \\
w_{ai2} & w_{bi2}
\end{pmatrix} = \begin{pmatrix}
\beta & \beta - \frac{1}{h_{m-1}} \\
-\beta + \frac{1}{h_m} & -\beta + \frac{1}{h_{m-1}} + \frac{1}{h_m}
\end{pmatrix}
\] (2.18)

\[
W_j = \begin{pmatrix}
w_{aj1} & w_{bj1} \\
w_{aj2} & w_{bj2}
\end{pmatrix} = \begin{pmatrix}
\beta - \frac{1}{h_m} & -\beta + \frac{1}{h_{m+1}} \\
-\beta + \frac{1}{h_{m+1}} + \frac{1}{h_m}
\end{pmatrix}
\] (2.19)

\[
\vec{V}_m = \begin{pmatrix}
a_m(t) \\
b_m(t)
\end{pmatrix}, \vec{V}_i = \begin{pmatrix}
a_{m-1}(t) \\
b_{m-1}(t)
\end{pmatrix}, \vec{V}_j = \begin{pmatrix}
a_{m+1}(t) \\
b_{m+1}(t)
\end{pmatrix}
\] (2.20)

\[
\vec{F}_m(\vec{V}_m) = \begin{pmatrix}
(q_{11}/2) \sum_{l=1,2} F(u(x_{m,l})) \\
(q_{11}/2) \sum_{l=1,2} F(u(x_{m,l})) \xi_m(x_{m,l})
\end{pmatrix}
\] (2.21)

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

\[ \vec{V}_m'(t) = D \sum_{l=m,i,j} Q_m^{-1} W_l \vec{V}_l(t) + Q_m^{-1} \vec{F}_m(\vec{V}_m), \quad m = 1, \ldots, N \] (2.22)

Again, these mesh-dependent data \( Q_m^{-1} \) and \( W_l \) are pre-calculated and stored before the time evolution since they don’t depend on the numerical solution \( u \).
No-flux boundary conditions are applied to the two boundary points \( a = x_\frac{1}{2} \) and \( b = x_{N+\frac{1}{2}} \), we have

\[
\begin{align*}
(u^+)_a &= (u^-)_a, \quad (u^+)_x a = (u^-)_x a = 0, \quad (2.23) \\
(u^+)_b &= (u^-)_b, \quad (u^+)_x b = (u^-)_x b = 0. \quad (2.24)
\end{align*}
\]

After applying equations 2.23, 2.24 to equation 2.5, we need to change equation 2.17 to

\[
W_m = \begin{pmatrix}
w_{am1} & w_{bm1} \\
w_{am2} & w_{bm2}
\end{pmatrix} = \begin{pmatrix}
-\beta & -\beta + \frac{1}{h_m} \\
-\beta + \frac{1}{h_m} & -\beta - \frac{2}{h_m}
\end{pmatrix} \quad (2.25)
\]

for \( m = 1 \) and to

\[
W_m = \begin{pmatrix}
w_{am1} & w_{bm1} \\
w_{am2} & w_{bm2}
\end{pmatrix} = \begin{pmatrix}
-\beta & \beta - \frac{1}{h_m} \\
\beta - \frac{1}{h_m} & -\beta - \frac{2}{h_m}
\end{pmatrix} \quad (2.26)
\]

for \( m = N \). By taking \( W_i = 0 \) for \( m = 1 \) and \( W_j = 0 \) for \( m = N \), equations 2.15 and 2.22 still works on both boundary elements \( m = 1 \) and \( m = N \).

2.1.2 Two dimensions with triangular mesh

Let \( \Omega \) be an open, bounded domain on which the reaction-diffusion system 1.1 is defined. We consider a triangulation \( \Omega_h \) of \( \Omega \) which consists of non-overlapping triangles \( \{\triangle_m\}_{m=1}^N \). Let \( h_{\text{min}} = \min_{1 \leq m \leq N} \rho_m \), where \( \rho_m \) is the diameter of the inscribed circle of the triangle \( \triangle_m \).

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

We apply the DG formulation [16] to discretize the reaction-diffusion equation
1.1 in the spatial direction, but keep the time variable continuous. The semi-
discrete scheme is: find $u \in V^k$, such that

$$
\int_{\Delta_m} u_t v dx - D \int_{\Delta_m} u \nabla^2 v dx + \int_{\partial\Delta_m} \tilde{u} \nabla v \cdot \tilde{n}_{\partial\Delta_m} dS - D \int_{\partial\Delta_m} v \tilde{\nabla} u \cdot \tilde{n}_{\partial\Delta_m} dS
= \int_{\Delta_m} F(u) v dx
$$

(2.27)

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

$$
\hat{u} = \frac{u^{in} + u^{ext}}{2},
$$

(2.28)

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

(2.29)

where the jump term

$$
[u] = (u^{ext} - u^{in})|_{\partial\Delta_m} \cdot \tilde{n}_{\partial\Delta_m}.
$$

(2.30)

$u^{in}$ and $u^{ext}$ are the limits of $u$ at $x \in \partial\Delta_m$ taken from the interior and the
exterior of $\Delta_m$ respectively, $\tilde{n}_{\partial\Delta_m}$ is the outward unit normal to the element $\Delta_m$
at $x \in \partial\Delta_m$, and $\beta$ is a positive constant that is of the order $O(h_{\min}^{-1})$. In all of
computations of this paper, we take $\beta = 10/h_{\min}$. The choice of numerical fluxes
2.28 - 2.30 is crucial for the stability and convergence of the DG scheme 2.27.

See [16, 21] for more discussions about the choice of numerical fluxes.

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 $\Delta_m$, 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 \(\triangle_m\), as shown in Figure 2.1a. The linear polynomial on \(\triangle_m\) is represented by

\[
 u(x, y, t) = a_m(t) + b_m(t)\xi_m + c_m(t)\eta_m, \\
\]

where

\[
\xi_m = \frac{x - x_m}{h_m}, \quad \eta_m = \frac{y - y_m}{h_m}, \\
\]

and \((x_m, y_m)\) is the barycenter of the element \(\triangle_m\), \(h_m = \sqrt{|\triangle_m|}\) with \(|\triangle_m|\) denoting the area of \(\triangle_m\). By taking \(v = 1, \xi_m, \eta_m\) on \(\triangle_m\) and \(v = 0\) elsewhere, the DG formulation 2.27 is converted from the integral form to the following system, for \(m = 1, \cdots, N\):

\[
 q_11 a_m'(t) + q_12 b_m'(t) + q_13 c_m'(t) = D \sum_{l=m,i,j,k} \left[ w_{al1} a_l(t) + w_{bl1} b_l(t) + w_{cl1} c_l(t) \right] \\
+ \left( q_{11}/3 \right) \sum_{l=i,j,k} F(u(x_m,l, y_m,l)) \xi_m(x_m,l, y_m,l), \\
\]

\[
 q_21 a_m'(t) + q_22 b_m'(t) + q_23 c_m'(t) = D \sum_{l=m,i,j,k} \left[ w_{al2} a_l(t) + w_{bl2} b_l(t) + w_{cl2} c_l(t) \right] \\
+ \left( q_{11}/3 \right) \sum_{l=i,j,k} F(u(x_m,l, y_m,l)) \xi_m(x_m,l, y_m,l), \\
\]

\[
 q_31 a_m'(t) + q_32 b_m'(t) + q_33 c_m'(t) = D \sum_{l=m,i,j,k} \left[ w_{al3} a_l(t) + w_{bl3} b_l(t) + w_{cl3} c_l(t) \right] \\
+ \left( q_{11}/3 \right) \sum_{l=i,j,k} F(u(x_m,l, y_m,l)) \eta_m(x_m,l, y_m,l), \\
\]

where the coefficients \(\{q_{rs}\}_{r,s=1}^3\), \(\{w_{a1r}\}_{r=1}^3\), \(\{w_{b1r}\}_{r=1}^3\), \(\{w_{c1r}\}_{r=1}^3\) \(l=m,i,j,k\) are con-
stants which depend on the local geometry of the mesh (i.e., triangle $\triangle_m$ and its
neighboring cells $i, j, k$ and $\vec{n}_{\partial \Delta_m}$ as shown in Figure 2.1a), the local basis func-
tions $1, \{\xi_l, \eta_l\}_{l=m,i,j,k}$, and the constant $\beta$. \{(x_m,t, y_m,t)\}_{l=i,j,k}$ are the mid-points
of the three edges $\{e_l\}_{l=i,j,k}$ of $\triangle_m$ which serve as Gaussian quadrature points for
the integral involving the nonlinear reaction terms in equation 2.27. In our im-
plementation, these mesh-dependent constants are pre-calculated before the time
evolution since they don’t depend on the numerical solution $u$. Rewrite equations
2.33 - 2.35 to the matrix-vector form

$$Q_m \vec{V}'_m(t) = D \sum_{l=m,i,j,k} W_l \vec{V}_l(t) + \vec{F}_m(\vec{V}_m), \quad (2.36)$$

where

$$Q_m = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix} = \begin{pmatrix} \int_{\triangle_m} dx & \int_{\triangle_m} \xi_m dx & \int_{\triangle_m} \eta_m dx \\ \int_{\triangle_m} \xi_m dx & \int_{\triangle_m} \xi^2_m dx & \int_{\triangle_m} \xi_m \eta_m dx \\ \int_{\triangle_m} \eta_m dx & \int_{\triangle_m} \xi_m \eta_m dx & \int_{\triangle_m} \eta^2_m dx \end{pmatrix} \quad (2.37)$$

$$W_m = \begin{pmatrix} w_{am1} & w_{bm1} & w_{cm1} \\ w_{am2} & w_{bm2} & w_{cm2} \\ w_{am3} & w_{bm3} & w_{cm3} \end{pmatrix} = \beta \sum_{l=i,j,k} \begin{pmatrix} r_{1l} & r_{2lm} & r_{3lm} \\ r_{2lm} & s_{1ml} & s_{2ml} \\ r_{3lm} & s_{2ml} & s_{3ml} \end{pmatrix}$$

$$+ \sum_{l=i,j,k} \begin{pmatrix} 0 & \frac{r_{1l} n_{l,x}}{2h_m} & \frac{r_{1l} n_{l,y}}{2h_m} \\ -\frac{r_{1l} n_{l,x}}{2h_m} & 0 & \frac{r_{2lm} n_{l,y} - r_{3lm} n_{l,x}}{2h_m} \\ -\frac{r_{1l} n_{l,y}}{2h_m} & \frac{r_{3lm} n_{l,x} - r_{2lm} n_{l,y}}{2h_m} & 0 \end{pmatrix}, \quad (2.38)$$
\[
W_l = \begin{pmatrix}
w_{al1} & w_{bl1} & w_{cl1} \\
w_{al2} & w_{bl2} & w_{cl2} \\
w_{al3} & w_{bl3} & w_{cl3}
\end{pmatrix} = \beta \begin{pmatrix}
r_{1l} & r_{2l} & r_{3l} \\
r_{2lm} & s_{1lm} & s_{2lm} \\
r_{3lm} & s_{3lm} & s_{4lm}
\end{pmatrix}
\]
\[
+ \begin{pmatrix}
0 & \frac{r_{11} n_{i,x}}{2h_l} & \frac{r_{11} n_{i,y}}{2h_l} \\
-\frac{r_{11} n_{i,x}}{2h_m} & \frac{r_{21} n_{i,x}}{2h_l} & \frac{r_{21} n_{i,y}}{2h_l} & \frac{r_{31} n_{i,x}}{2h_l} & \frac{r_{31} n_{i,y}}{2h_l} & \frac{r_{41} n_{i,x}}{2h_l} & \frac{r_{41} n_{i,y}}{2h_l}
\end{pmatrix},
\]
(2.39)

\[
\vec{V}_m = \begin{pmatrix}
a_m(t) \\
b_m(t) \\
c_m(t)
\end{pmatrix}, \quad \vec{V}_l = \begin{pmatrix}
a_l(t) \\
b_l(t) \\
c_l(t)
\end{pmatrix},
\]
(2.40)

\[
\tilde{F}_m(\vec{V}_m) = \begin{pmatrix}
(q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l})) \\
(q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l})) \xi_m(x_{m,l}, y_{m,l}) \\
(q_{11}/3) \sum_{l=i,j,k} F(u(x_{m,l}, y_{m,l})) \eta_m(x_{m,l}, y_{m,l})
\end{pmatrix},
\]
(2.41)

and

\[
r_{1l} = \int_{e_l} ds, \quad r_{2l} = \int_{e_l} \xi_l ds, \quad r_{3l} = \int_{e_l} \eta_l ds,
\]
\[
r_{2lm} = \int_{e_l} \xi_m ds, \quad r_{3lm} = \int_{e_l} \eta_m ds,
\]
\[
s_{1lm} = \int_{e_l} \xi_m \xi_l ds, \quad s_{2lm} = \int_{e_l} \xi_m \eta_l ds,
\]
\[
s_{3lm} = \int_{e_l} \xi_l \eta_m ds, \quad s_{4lm} = \int_{e_l} \eta_m \eta_l ds,
\]
(2.42)

\[
s_{1mml} = \int_{e_l} \xi_m^2 ds, \quad s_{2mml} = \int_{e_l} \xi_m \eta_m ds, \quad s_{3mml} = \int_{e_l} \eta_m^2 ds.
\]
Finally we have the ODE system resulting from the DG spatial discretization:

\[
\ddot{\mathbf{V}}_m(t) = D \sum_{l=m,i,j,k} Q_{-1}^{m} \mathbf{W}_l \ddot{\mathbf{V}}_l(t) + Q_{-1}^{m} \ddot{\mathbf{F}}_m(\mathbf{V}_m), \quad m = 1, \ldots, N, \quad (2.43)
\]

Again, these mesh-dependent data \(Q_{-1}^{m}\) and \(\mathbf{W}_l\) are pre-calculated and stored before the time evolution since they don’t depend on the numerical solution \(u\).

If equation 1.1 is subject to no-flux boundary conditions and if the element edge \(e_l\) of \(\Delta_m\) is aligned with the domain boundary \(\partial\Delta_m\), we take

\[
u^{|e_l} = u^{ext}|_{e_l}, \quad (\nabla u)^{in}|_{e_l} \cdot \vec{n}_{e_l} = (\nabla u)^{ext}|_{e_l} \cdot \vec{n}_{e_l} = 0, \quad (2.44)
\]

in the numerical fluxes 2.28 - 2.30. Hence we have

\[
\hat{u}|_{e_l} = u^{in}|_{e_l}, \quad \hat{\nabla} u|_{e_l} \cdot \vec{n}_{e_l} = 0, \quad (2.45)
\]

in the scheme 2.27. Without loss of generality, take \(l = i\) in equations 2.44 and 2.45 and we need to change equation 2.38 to
\[
W_m = \begin{pmatrix}
w_{am1} & w_{bm1} & w_{cm1} \\
w_{am2} & w_{bm2} & w_{cm2} \\
w_{am3} & w_{bm3} & w_{cm3}
\end{pmatrix} = \begin{pmatrix}0 & 0 & 0 \\
-\frac{r_{1m_{l,x}}}{h_m} & -\frac{r_{2m_{l,x}}}{h_m} & -\frac{r_{3m_{l,x}}}{h_m} \\
-\frac{r_{1m_{l,y}}}{h_m} & -\frac{r_{2m_{l,y}}}{h_m} & -\frac{r_{3m_{l,y}}}{h_m}
\end{pmatrix} = l=i
\]

\[-\beta \sum_{l=j,k} \begin{pmatrix}r_{1l} & r_{2lm} & r_{3lm} \\
r_{2lm} & s_{1mml} & s_{2mml} \\
r_{3lm} & s_{2mml} & s_{3mml}
\end{pmatrix}
\]

\[+ \sum_{l=j,k} \begin{pmatrix}0 & \frac{r_{l1m_{l,x}}}{2h_m} & \frac{r_{l1m_{l,y}}}{2h_m} \\
\frac{r_{1m_{l,x}}}{2h_m} & 0 & \frac{r_{2m_{l,y}}}{2h_m} \frac{r_{3m_{l,x}}}{2h_m} \\
\frac{r_{1m_{l,y}}}{2h_m} & \frac{r_{2m_{l,x}}}{2h_m} & 0
\end{pmatrix}, \quad (2.46)
\]

and take \( W_i = 0 \) in equation 2.43.

### 2.1.3 Two dimensions with rectangular mesh

The DG spatial discretization in the rectangular mesh is very similar to that in the triangular mesh. For each element \( \square_m \), denote its four neighboring elements by \( h_m, i_m, j_m, \) and \( k_m \) and we just use \( h, i, j, \) and \( k \) to represent the neighboring cells of \( \square_m \), as shown in Figure 2.1b.

We have the same semi-discrete scheme 2.27 and the same fluxes 2.28 - 2.30.

Instead of equation 2.32, we take

\[
\xi_m = 2\frac{x - x_m}{h x_m}, \quad \eta_m = 2\frac{y - y_m}{h y_m}, \quad (2.47)
\]

where \( h x_m \) and \( h y_m \) are the sizes of element \( \square_m \) in the \( x \) and \( y \) directions.
Equation 2.36 is changed to

\[ Q_m \vec{V}_m'(t) = D \sum_{l=m,h,i,j,k} W_l \vec{V}_l(t) + \vec{F}_m(\vec{V}_m), \]  

(2.48)

where

\[ W_m = \begin{pmatrix} w_{am1} & w_{bm1} & w_{cm1} \\ w_{am2} & w_{bm2} & w_{cm2} \\ w_{am3} & w_{bm3} & w_{cm3} \end{pmatrix} = -\beta \sum_{l=h,i,j,k} \begin{pmatrix} r_{1l} & r_{2lm} & r_{3lm} \\ r_{2lm} & s_{1mml} & s_{2mml} \\ r_{3lm} & s_{2mml} & s_{3mml} \end{pmatrix}, \]  

(2.49)

\[ W_l = \begin{pmatrix} w_{al1} & w_{bl1} & w_{cl1} \\ w_{al2} & w_{bl2} & w_{cl2} \\ w_{al3} & w_{bl3} & w_{cl3} \end{pmatrix} = \beta \begin{pmatrix} r_{1l} & r_{2l} & r_{3l} \\ r_{2l} & s_{1lm} & s_{2lm} \\ r_{3l} & s_{3lm} & s_{4lm} \end{pmatrix}, \]  

(2.50)

\[ \vec{F}_m(\vec{V}_m) = \begin{pmatrix} (q_{11}/4) \sum_{l=h,i,j,k} F(u(x_{m,l}, y_{m,l})) \\ (q_{11}/4) \sum_{l=h,i,j,k} F(u(x_{m,l}, y_{m,l})) \xi_m(x_{m,l}, y_{m,l}) \\ (q_{11}/4) \sum_{l=h,i,j,k} F(u(x_{m,l}, y_{m,l})) \eta_m(x_{m,l}, y_{m,l}) \end{pmatrix}, \]  

(2.51)

and equation 2.42 keeps the same.
Finally we have the ODE system resulting from the DG spatial discretization:

\[
\dot{\vec{V}}_m(t) = D \sum_{l=m,h,i,j,k} Q_{l,m}^{-1} \vec{W}_l \dot{V}_l(t) + Q_{l,m}^{-1} \vec{F}_m(\vec{V}_m), \quad m = 1, \cdots, N, \quad (2.52)
\]

For the no-flux boundary conditions, the equations 2.44 and 2.45 still work. Similar, if the element edge \(e_i\) of \(\Box_m\) is aligned with the domain boundary \(\partial \Box_m\), we take

\[
W_m = \begin{pmatrix}
w_{am1} & w_{bm1} & w_{cm1} \\
w_{am2} & w_{bm2} & w_{cm2} \\
w_{am3} & w_{bm3} & w_{cm3}
\end{pmatrix} = \begin{pmatrix}
0 & 0 & 0 \\
-\frac{2r_{1i}n_{i,x}}{h_{x,m}} & \frac{2r_{2lm}n_{i,x}}{h_{x,m}} & -\frac{2r_{3lm}n_{i,x}}{h_{x,m}} \\
-\frac{2r_{1i}n_{i,y}}{h_{y,m}} & -\frac{2r_{2lm}n_{i,y}}{h_{y,m}} & -\frac{2r_{3lm}n_{i,y}}{h_{y,m}}
\end{pmatrix}_{l=i}
\]

\[
-\beta \sum_{l=h,j,k} \begin{pmatrix}
r_{1l} & r_{2lm} & r_{3lm} \\
r_{2lm} & s_{1mml} & s_{2mml} \\
r_{3lm} & s_{2mml} & s_{3mml}
\end{pmatrix}
\]

\[
+ \sum_{l=h,j,k} \begin{pmatrix}
0 & \frac{r_{1l}n_{i,x}}{2h_m} & \frac{r_{1l}n_{i,y}}{2h_m} \\
-\frac{r_{1i}n_{i,x}}{2h_m} & 0 & \frac{r_{2lm}n_{i,y} - r_{3lm}n_{l,x}}{2h_m} \\
-\frac{r_{1i}n_{i,y}}{2h_m} & \frac{r_{3lm}n_{l,x} - r_{2lm}n_{l,y}}{2h_m} & 0
\end{pmatrix}, \quad (2.53)
\]

and \(W_i = 0\) in equations 2.48 and 2.52.

2.2 Operator splitting

The ODE systems 2.22, 2.43, and 2.52 have very close formations and we use 2.43 to explain the operator splitting scheme. 2.43 have a linear term resulting from the diffusion and a nonlinear term coming from the reaction of 1.1. Both of terms can cause stiffness in the reaction-diffusion models arising in developmental biology [4]. Hence we need to use fully implicit schemes to solve 2.43. 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 \[47, 97\] to split the diffusion from the reaction terms of \[1.1\]. 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 methods.

Denote the numerical solution of the ODE system 2.43 at \( t = t^n \) by \( \vec{V}^n_m \).

To evolve the system 2.43 from the time step \( t^n \) to \( t^{n+1} \), the classical Strang symmetrical OS scheme \[97\] combined with the Crank Nicholson method consists of the following three sub-steps:

**Step 1** – apply Crank Nicholson for the diffusion term at \( [t^n, t^{n+\frac{1}{2}}] \):

\[
\vec{v}_{0,m} = \vec{V}^n_m, \quad m = 1, \cdots, N; \tag{2.54}
\]
\[
\vec{v}_{1,m} = \vec{v}_{0,m} + \frac{1}{4} \Delta t \left[ D \sum_{l=m,i,j,k} Q_m^{-1} W_l \vec{v}_{0,l} + D \sum_{l=m,i,j,k} Q_m^{-1} W_l \vec{v}_{1,l} \right], \quad m = 1, \cdots, N.
\]

The sparse linear system 2.54 is solved by the solver \texttt{“lin\_sol\_gen\_coordinate”} in the package IMSL.

**Step 2** – apply Crank Nicholson for the reaction term at \( [t^n, t^{n+1}] \), with \( \vec{v}_{1,m} \) as input data:

\[
\vec{v}_{2,m} = \vec{v}_{1,m} + \frac{1}{2} \Delta t \left[ Q_m^{-1} F_m(\vec{v}_{1,m}) + Q_m^{-1} F_m(\vec{v}_{2,m}) \right], \quad m = 1, \cdots, N. \tag{2.55}
\]

The local nonlinear system 2.55 on the element \( m \) is solved by Newton iterations, with the initial guess \( \vec{v}_{1,m} \), for \( m = 1, \cdots, N \).

**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:

\[
\tilde{v}_{3,m} = \tilde{v}_{2,m} + \frac{1}{4} \Delta t \left[ D \sum_{l=m,i,j,k} Q_{m}^{-1} \mathbf{W}_l \tilde{v}_{2,l} + D \sum_{l=m,i,j,k} Q_{m}^{-1} \mathbf{W}_l \tilde{v}_{3,l} \right], \ m = 1, \cdots, N;
\]

\[
\tilde{v}_{n+1}^m = \tilde{v}_{3,m}, \ m = 1, \cdots, N. \tag{2.56}
\]

Again, the sparse linear system \[2.56\] is solved by the solver “lin_sol_gen_coordinate” in the package IMSL.

Trapezoidal OS \[47\] scheme is a Strang type symmetrical operator splitting method. In stead of using Crank Nicholson method in the step 1 and 3, it uses the forward and backward Euler schemes hence is computationally cheaper, and still keeps second order accuracy. We apply the Trapezoidal OS scheme for equation \[2.43\]

**Step 1** – apply forward Euler for the diffusion term at \([t^n, t^{n+\frac{1}{2}}]\):

\[
\tilde{v}_{0,m} = \tilde{v}_{n}^m, \quad m = 1, \cdots, N; \tag{2.57}
\]

\[
\tilde{v}_{1,m} = \tilde{v}_{0,m} + \frac{1}{2} \Delta t \left[ D \sum_{l=m,i,j,k} Q_{m}^{-1} \mathbf{W}_l \tilde{v}_{0,l} \right], \quad m = 1, \cdots, N.
\]

**Step 2** – apply Crank Nicholson for the reaction term at \([t^n, t^{n+1}]\), with \(\tilde{v}_{1,m}\) as input data:

\[
\tilde{v}_{2,m} = \tilde{v}_{1,m} + \frac{1}{2} \Delta t [Q_{m}^{-1} \mathbf{F}_m(\tilde{v}_{1,m}) + Q_{m}^{-1} \mathbf{F}_m(\tilde{v}_{2,m})], \quad m = 1, \cdots, N. \tag{2.58}
\]

The local nonlinear system \[2.58\] on the element \(m\) is solved by Newton iterations, with the initial guess \(\tilde{v}_{1,m}\), for \(m = 1, \cdots, N\).

**Step 3** – apply backward Euler for the diffusion term at \([t^{n+\frac{1}{2}}, t^{n+1}]\), with \(\tilde{v}_{2,m}\) as
input data:

\[
\vec{v}_{3,m} = \vec{v}_{2,m} + \frac{1}{2} \Delta t \left[ D \sum_{l=m,i,j,k} Q_{-1}^{-1} W_{l} \vec{v}_{3,l} \right], \quad m = 1, \cdots, N;
\]

\[
\vec{v}_{m+1} = \vec{v}_{3,m}, \quad m = 1, \cdots, N. \tag{2.59}
\]

The sparse linear system 2.59 is solved by the solver “lin_sol_gen_coordinate” in the package IMSL.
Figure 2.1. Typical stencil
2.3 Numerical Tests

2.3.1 One dimension

**Example 2.1** Consider the one-dimensional problem

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

with the no flux boundary conditions. The exact solution is \( u(x, t) = e^{-t} \cos(\pi x) \).

The simulation is carried up to \( T = 1.0 \) at which the \( L^1 \), \( L^2 \) and \( L^\infty \) errors are measured. The time step size is \( \Delta t = \Delta x \) where \( \Delta x = 1/N \) and \( N \) is the number of spatial elements. We use the three different approaches: the DG spatial discretization 2.22 with the Crank Nicholson temporal discretization (without OS), the DG-Strang symmetrical OS scheme 2.54 - 2.56, and the DG-trapezoidal OS scheme 2.57 - 2.59. We list the CPU time, errors, and order of accuracy for the DG w/o OS, the DG-Strang symmetrical OS, and the DG-trapezoidal OS in Table 2.1. The second order accuracy is achieved for all of these three approaches in Figure 2.2. In Figure 2.3, we compare the computational time of these three approaches by plotting errors as a function of CPU time. In order to reach the same numerical error level, the DG-trapezoidal OS requires smallest time when \( N = 320 \), and there is no big difference between the DG w/o OS and the DG-trapezoidal OS when cell number is small. The DG-Strang symmetrical OS is the most expensive in this case.
TABLE 2.1

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.1

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG method w/o OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>1.93E-03</td>
<td>-</td>
<td>2.15E-03</td>
<td>-</td>
<td>3.04E-03</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>4.86E-04</td>
<td>1.99</td>
<td>5.40E-04</td>
<td>2.00</td>
<td>7.63E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>40</td>
<td>0.02</td>
<td>1.22E-04</td>
<td>2.00</td>
<td>1.35E-04</td>
<td>2.00</td>
<td>1.91E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>80</td>
<td>0.1</td>
<td>3.04E-05</td>
<td>2.00</td>
<td>3.38E-05</td>
<td>2.00</td>
<td>4.78E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>0.55</td>
<td>7.60E-06</td>
<td>2.00</td>
<td>8.44E-06</td>
<td>2.00</td>
<td>1.19E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>3.19</td>
<td>1.90E-06</td>
<td>2.00</td>
<td>2.11E-06</td>
<td>2.00</td>
<td>2.98E-06</td>
<td>2.00</td>
</tr>
<tr>
<td>DG-Strang symmetrical OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>1.58E-02</td>
<td>-</td>
<td>1.75E-02</td>
<td>-</td>
<td>2.46E-02</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>3.92E-03</td>
<td>2.01</td>
<td>4.35E-03</td>
<td>2.01</td>
<td>6.14E-03</td>
<td>2.00</td>
</tr>
<tr>
<td>40</td>
<td>0.03</td>
<td>9.77E-04</td>
<td>2.00</td>
<td>1.08E-03</td>
<td>2.00</td>
<td>1.53E-03</td>
<td>2.00</td>
</tr>
<tr>
<td>80</td>
<td>0.12</td>
<td>2.44E-04</td>
<td>2.00</td>
<td>2.71E-04</td>
<td>2.00</td>
<td>3.83E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>0.54</td>
<td>6.10E-05</td>
<td>2.00</td>
<td>6.78E-05</td>
<td>2.00</td>
<td>9.58E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>2.32</td>
<td>1.53E-05</td>
<td>2.00</td>
<td>1.69E-05</td>
<td>2.00</td>
<td>2.40E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>DG-trapezoidal OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>1.35E-03</td>
<td>-</td>
<td>1.50E-03</td>
<td>-</td>
<td>2.11E-03</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>3.39E-04</td>
<td>1.99</td>
<td>3.77E-04</td>
<td>1.99</td>
<td>5.33E-04</td>
<td>1.99</td>
</tr>
<tr>
<td>40</td>
<td>0.04</td>
<td>8.50E-05</td>
<td>2.00</td>
<td>9.44E-05</td>
<td>2.00</td>
<td>1.33E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>80</td>
<td>0.11</td>
<td>2.13E-05</td>
<td>2.00</td>
<td>2.36E-05</td>
<td>2.00</td>
<td>3.34E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>0.49</td>
<td>5.31E-06</td>
<td>2.00</td>
<td>5.90E-06</td>
<td>2.00</td>
<td>8.35E-06</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>2.03</td>
<td>1.33E-06</td>
<td>2.00</td>
<td>1.48E-06</td>
<td>2.00</td>
<td>2.09E-06</td>
<td>2.00</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$.  

24
Figure 2.2. Error as a function of cells $N$ for three different approaches for Example 2.1, 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.
Figure 2.3. Error as a function of CPU time for three different approaches for Example 2.1, 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.
Example 2.2 Consider the one-dimensional problem

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

with the no flux boundary conditions. The exact solution is \( u(x, t) = e^{-t} \cos(\pi x) \).

The simulation is carried up to \( T = 1.0 \) at which the \( L^1 \), \( L^2 \) and \( L^\infty \) errors are measured. The time step size is \( \Delta t = \Delta x \) where \( \Delta x = 1/N \) and \( N \) is the number of spatial elements. We use the same three different approaches as in Example 2.1 and list the CPU time, errors, and order of accuracy for the DG w/o OS, the DG-Strang symmetrical OS, and the DG-trapezoidal OS in Table 2.2. The second order accuracy is achieved for all of these three approaches in Figure 2.4.

In Figure 2.5 we compare the computational time of these three approaches by plotting errors as a function of CPU time. In order to reach the same numerical error level, the DG w/o OS requires the smallest time, the DG-trapezoidal OS falls in the middle, and The DG-Strang symmetrical OS takes the most time.
TABLE 2.2
CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.2

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG method w/o OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>3.28E-03</td>
<td>-</td>
<td>4.10E-03</td>
<td>-</td>
<td>6.85E-03</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.02</td>
<td>8.27E-04</td>
<td>1.99</td>
<td>1.03E-03</td>
<td>1.99</td>
<td>1.73E-03</td>
<td>1.99</td>
</tr>
<tr>
<td>40</td>
<td>0.04</td>
<td>2.07E-04</td>
<td>2.00</td>
<td>2.59E-04</td>
<td>2.00</td>
<td>4.33E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>80</td>
<td>0.17</td>
<td>5.19E-05</td>
<td>2.00</td>
<td>6.48E-05</td>
<td>2.00</td>
<td>1.08E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>0.89</td>
<td>1.30E-05</td>
<td>2.00</td>
<td>1.62E-05</td>
<td>2.00</td>
<td>2.71E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>5.3</td>
<td>3.24E-06</td>
<td>2.00</td>
<td>4.05E-06</td>
<td>2.00</td>
<td>6.78E-06</td>
<td>2.00</td>
</tr>
<tr>
<td>DG-Strang symmetrical OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>6.99E-02</td>
<td>-</td>
<td>7.11E-02</td>
<td>-</td>
<td>8.78E-02</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.02</td>
<td>1.65E-02</td>
<td>2.08</td>
<td>1.68E-02</td>
<td>2.08</td>
<td>2.11E-02</td>
<td>2.05</td>
</tr>
<tr>
<td>40</td>
<td>0.06</td>
<td>4.08E-03</td>
<td>2.02</td>
<td>4.16E-03</td>
<td>2.02</td>
<td>5.24E-03</td>
<td>2.01</td>
</tr>
<tr>
<td>80</td>
<td>0.17</td>
<td>1.02E-03</td>
<td>2.00</td>
<td>1.04E-03</td>
<td>2.00</td>
<td>1.31E-03</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>0.73</td>
<td>2.54E-04</td>
<td>2.00</td>
<td>2.59E-04</td>
<td>2.00</td>
<td>3.27E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>3.18</td>
<td>6.35E-05</td>
<td>2.00</td>
<td>6.47E-05</td>
<td>2.00</td>
<td>8.17E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>DG-trapezoidal OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>7.21E-02</td>
<td>-</td>
<td>7.23E-02</td>
<td>-</td>
<td>7.90E-02</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.02</td>
<td>9.80E-03</td>
<td>2.88</td>
<td>9.80E-03</td>
<td>2.88</td>
<td>1.01E-02</td>
<td>2.97</td>
</tr>
<tr>
<td>40</td>
<td>0.06</td>
<td>1.98E-03</td>
<td>2.30</td>
<td>1.98E-03</td>
<td>2.30</td>
<td>2.04E-03</td>
<td>2.31</td>
</tr>
<tr>
<td>80</td>
<td>0.17</td>
<td>4.67E-04</td>
<td>2.09</td>
<td>4.67E-04</td>
<td>2.09</td>
<td>4.86E-04</td>
<td>2.07</td>
</tr>
<tr>
<td>160</td>
<td>0.73</td>
<td>1.15E-04</td>
<td>2.02</td>
<td>1.15E-04</td>
<td>2.02</td>
<td>1.20E-04</td>
<td>2.02</td>
</tr>
<tr>
<td>320</td>
<td>2.96</td>
<td>2.86E-05</td>
<td>2.01</td>
<td>2.86E-05</td>
<td>2.01</td>
<td>2.99E-05</td>
<td>2.00</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$. 28
Figure 2.4. Error as a function of cells $N$ for three different approaches for Example 2.2, 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.
Figure 2.5. Error as a function of CPU time for three different approaches for Example 2.2, 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.
Example 2.3 Consider the one-dimensional problem the same as equation \(2.60\)

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

(2.62)

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 \(L^\infty\) 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 2.3 for different ratios of \(\Delta t\) and \(\Delta x\). From the Table 2.3, 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 in Figure 2.6.
<table>
<thead>
<tr>
<th>$N$</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta t = \Delta x$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.03</td>
<td>1.23E-06</td>
<td>-</td>
<td>1.36E-06</td>
<td>-</td>
<td>1.93E-06</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.06</td>
<td>3.09E-07</td>
<td>1.99</td>
<td>3.44E-07</td>
<td>1.99</td>
<td>4.86E-07</td>
<td>1.99</td>
</tr>
<tr>
<td>40</td>
<td>0.25</td>
<td>7.75E-08</td>
<td>2.00</td>
<td>8.61E-08</td>
<td>2.00</td>
<td>1.22E-07</td>
<td>2.00</td>
</tr>
<tr>
<td>80</td>
<td>0.96</td>
<td>1.94E-08</td>
<td>2.00</td>
<td>2.15E-08</td>
<td>2.00</td>
<td>3.04E-08</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>4.11</td>
<td>4.85E-09</td>
<td>2.00</td>
<td>5.38E-09</td>
<td>2.00</td>
<td>7.61E-09</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>16.18</td>
<td>1.21E-09</td>
<td>2.00</td>
<td>1.35E-09</td>
<td>2.00</td>
<td>1.90E-09</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta t = 2\Delta x$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>3.20E-07</td>
<td>-</td>
<td>3.54E-07</td>
<td>-</td>
<td>4.99E-07</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.03</td>
<td>7.76E-08</td>
<td>2.04</td>
<td>8.61E-08</td>
<td>2.04</td>
<td>1.22E-07</td>
<td>2.04</td>
</tr>
<tr>
<td>40</td>
<td>0.12</td>
<td>1.92E-08</td>
<td>2.01</td>
<td>2.14E-08</td>
<td>2.01</td>
<td>3.02E-08</td>
<td>2.01</td>
</tr>
<tr>
<td>80</td>
<td>0.47</td>
<td>4.80E-09</td>
<td>2.00</td>
<td>5.33E-09</td>
<td>2.00</td>
<td>7.54E-09</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>1.94</td>
<td>1.20E-09</td>
<td>2.00</td>
<td>1.33E-09</td>
<td>2.00</td>
<td>1.88E-09</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>8.26</td>
<td>3.00E-10</td>
<td>2.00</td>
<td>3.33E-10</td>
<td>2.00</td>
<td>4.71E-10</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta t = 4\Delta x$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>6.66E-06</td>
<td>-</td>
<td>7.40E-06</td>
<td>-</td>
<td>1.04E-05</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>1.63E-06</td>
<td>2.03</td>
<td>1.82E-06</td>
<td>2.03</td>
<td>2.57E-06</td>
<td>2.02</td>
</tr>
<tr>
<td>40</td>
<td>0.06</td>
<td>4.07E-07</td>
<td>2.01</td>
<td>4.52E-07</td>
<td>2.01</td>
<td>6.39E-07</td>
<td>2.01</td>
</tr>
<tr>
<td>80</td>
<td>0.24</td>
<td>1.02E-07</td>
<td>2.00</td>
<td>1.13E-07</td>
<td>2.00</td>
<td>1.60E-07</td>
<td>2.00</td>
</tr>
</tbody>
</table>

continued...
### TABLE 2.3: Continued

<table>
<thead>
<tr>
<th>$N$</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>160</td>
<td>0.98</td>
<td>2.54E-08</td>
<td>2.00</td>
<td>2.82E-08</td>
<td>2.00</td>
<td>3.99E-08</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>4.12</td>
<td>6.35E-09</td>
<td>2.00</td>
<td>7.05E-09</td>
<td>2.00</td>
<td>9.97E-09</td>
<td>2.00</td>
</tr>
</tbody>
</table>

$\Delta t = 8\Delta x$

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.</td>
<td>3.46E-05</td>
<td>-</td>
<td>3.85E-05</td>
<td>-</td>
<td>5.43E-05</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>8.01E-06</td>
<td>2.11</td>
<td>8.90E-06</td>
<td>2.11</td>
<td>1.26E-05</td>
<td>2.11</td>
</tr>
<tr>
<td>40</td>
<td>0.03</td>
<td>1.97E-06</td>
<td>2.03</td>
<td>2.18E-06</td>
<td>2.03</td>
<td>3.09E-06</td>
<td>2.03</td>
</tr>
<tr>
<td>80</td>
<td>0.12</td>
<td>4.89E-07</td>
<td>2.01</td>
<td>5.43E-07</td>
<td>2.01</td>
<td>7.68E-07</td>
<td>2.01</td>
</tr>
<tr>
<td>160</td>
<td>0.49</td>
<td>1.22E-07</td>
<td>2.00</td>
<td>1.36E-07</td>
<td>2.00</td>
<td>1.92E-07</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>2.08</td>
<td>3.05E-08</td>
<td>2.00</td>
<td>3.39E-08</td>
<td>2.00</td>
<td>4.80E-08</td>
<td>2.00</td>
</tr>
</tbody>
</table>

$\Delta t = 16\Delta x$

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.</td>
<td>2.26E-04</td>
<td>-</td>
<td>2.51E-04</td>
<td>-</td>
<td>3.54E-04</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>3.61E-05</td>
<td>2.64</td>
<td>4.01E-05</td>
<td>2.64</td>
<td>5.67E-05</td>
<td>2.64</td>
</tr>
<tr>
<td>40</td>
<td>0.01</td>
<td>8.36E-06</td>
<td>2.11</td>
<td>9.28E-06</td>
<td>2.11</td>
<td>1.31E-05</td>
<td>2.11</td>
</tr>
<tr>
<td>80</td>
<td>0.06</td>
<td>2.05E-06</td>
<td>2.03</td>
<td>2.28E-06</td>
<td>2.03</td>
<td>3.22E-06</td>
<td>2.03</td>
</tr>
<tr>
<td>160</td>
<td>0.25</td>
<td>5.10E-07</td>
<td>2.01</td>
<td>5.66E-07</td>
<td>2.01</td>
<td>8.01E-07</td>
<td>2.01</td>
</tr>
<tr>
<td>320</td>
<td>1.04</td>
<td>1.27E-07</td>
<td>2.00</td>
<td>1.41E-07</td>
<td>2.00</td>
<td>2.00E-07</td>
<td>2.00</td>
</tr>
</tbody>
</table>

Final time $T = 8.0$. Time step size is increased successively.
Figure 2.6. Error as a function of number of cells $N$ for the DG-trapezoidal OS for Example 2.3.
2.3.2 Two dimensions

**Example 2.4** Consider the two-dimensional linear problem

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

(2.63)

with the no flux boundary conditions. The exact solution is

\[u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y).\]

We use rectangular meshes. The simulation is carried up to \(T = 1.0\) at which the \(L^1, L^2\) and \(L^\infty\) errors are measured. The time step size is \(\Delta t = h_{\text{min}}\) where \(h_{\text{min}} = 1/N\) and \(N \times N\) is the number of spatial elements. We use the three different approaches: the DG spatial discretization \(2.43\) with the Crank Nicholson temporal discretization (without OS), the DG-Strang symmetrical OS scheme \(2.54\) - \(2.56\) and the DG-trapezoidal OS scheme \(2.57\) - \(2.59\). We list the CPU time, errors, and order of accuracy for the DG w/o OS, the DG-Strang symmetrical OS, and the DG-trapezoidal OS in Table **2.4**. The second order accuracy is achieved for all of these three approaches in Figure **2.7**. In Figure **2.8**, we 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. The DG-Strang symmetrical OS approach is the most expensive and the DG w/o OS approach falls between them in this example.
### TABLE 2.4

**CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.4**

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DG method w/o OS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.15</td>
<td>1.13E-02</td>
<td>-</td>
<td>1.38E-02</td>
<td>-</td>
<td>2.74E-02</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>3.14</td>
<td>3.04E-03</td>
<td>1.90</td>
<td>3.71E-03</td>
<td>1.90</td>
<td>7.37E-03</td>
<td>1.89</td>
</tr>
<tr>
<td>40*40</td>
<td>96.65</td>
<td>7.74E-04</td>
<td>1.97</td>
<td>9.45E-04</td>
<td>1.97</td>
<td>1.88E-03</td>
<td>1.97</td>
</tr>
<tr>
<td>80*80</td>
<td>4596</td>
<td>1.94E-04</td>
<td>1.99</td>
<td>2.37E-04</td>
<td>1.99</td>
<td>4.72E-04</td>
<td>1.99</td>
</tr>
<tr>
<td>160*160</td>
<td>162756</td>
<td>4.87E-05</td>
<td>2.00</td>
<td>5.94E-05</td>
<td>2.00</td>
<td>1.18E-04</td>
<td>2.00</td>
</tr>
<tr>
<td><strong>DG-Strang symmetrical OS scheme</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.07</td>
<td>4.99E-02</td>
<td>-</td>
<td>6.19E-02</td>
<td>-</td>
<td>1.24E-02</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>0.64</td>
<td>1.22E-02</td>
<td>2.04</td>
<td>1.50E-02</td>
<td>2.04</td>
<td>3.00E-02</td>
<td>2.04</td>
</tr>
<tr>
<td>40*40</td>
<td>12.62</td>
<td>3.02E-03</td>
<td>2.01</td>
<td>3.73E-03</td>
<td>2.01</td>
<td>7.45E-03</td>
<td>2.01</td>
</tr>
<tr>
<td>80*80</td>
<td>488.24</td>
<td>7.55E-04</td>
<td>2.00</td>
<td>9.30E-04</td>
<td>2.00</td>
<td>1.86E-03</td>
<td>2.00</td>
</tr>
<tr>
<td>160*160</td>
<td>13921</td>
<td>1.89E-04</td>
<td>2.00</td>
<td>2.32E-04</td>
<td>2.00</td>
<td>4.65E-04</td>
<td>2.00</td>
</tr>
<tr>
<td><strong>DG-trapezoidal OS scheme</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.05</td>
<td>1.10E-02</td>
<td>-</td>
<td>1.34E-02</td>
<td>-</td>
<td>2.66E-02</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>0.59</td>
<td>2.95E-03</td>
<td>1.89</td>
<td>3.60E-03</td>
<td>1.90</td>
<td>7.15E-03</td>
<td>1.89</td>
</tr>
<tr>
<td>40*40</td>
<td>9.34</td>
<td>7.51E-04</td>
<td>1.97</td>
<td>9.16E-04</td>
<td>1.97</td>
<td>1.82E-03</td>
<td>1.97</td>
</tr>
<tr>
<td>80*80</td>
<td>323.98</td>
<td>1.89E-04</td>
<td>1.99</td>
<td>2.30E-04</td>
<td>1.99</td>
<td>4.58E-04</td>
<td>1.99</td>
</tr>
<tr>
<td>160*160</td>
<td>14210</td>
<td>4.72E-05</td>
<td>2.00</td>
<td>5.76E-05</td>
<td>2.00</td>
<td>1.15E-04</td>
<td>2.00</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$. 

36
Figure 2.7. Error as a function of $h_{min}$ for three different approaches for Example 2.4, 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.
Figure 2.8. Error as a function of CPU time for three different approaches for Example 2.4, 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.
**Example 2.5** Consider the two-dimensional nonlinear problem

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

(2.64)

with the no flux boundary conditions. The exact solution is

\[ u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y). \]

We still use rectangular meshes. The simulation is carried up to \( T = 1.0 \) at which the \( L^1, L^2 \) and \( L^\infty \) errors are measured. The time step size is \( \Delta t = h_{\text{min}} \) where \( h_{\text{min}} = 1/N \) and \( N \times N \) is the number of spatial elements. We use the same three different approaches as in Example 2.4 and list the CPU time, errors, and order of accuracy for the DG w/o OS, the DG-Strang symmetrical OS, and the DG-trapezoidal OS in Table 2.5. The second order accuracy is achieved for all of these three approaches in Figure 2.9. In Figure 2.10, we compare the computational time of these three approaches by plotting errors as a function of CPU time, and get similar conclusions as in Example 2.4.
## TABLE 2.5

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.5

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG method w/o OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.25</td>
<td>1.71E-02</td>
<td>-</td>
<td>2.07E-02</td>
<td>-</td>
<td>4.41E-02</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>4.19</td>
<td>4.70E-03</td>
<td>1.87</td>
<td>5.68E-03</td>
<td>1.87</td>
<td>1.21E-02</td>
<td>1.87</td>
</tr>
<tr>
<td>40*40</td>
<td>138.04</td>
<td>1.20E-03</td>
<td>1.97</td>
<td>1.45E-03</td>
<td>1.97</td>
<td>3.08E-03</td>
<td>1.97</td>
</tr>
<tr>
<td>80*80</td>
<td>7270</td>
<td>3.03E-04</td>
<td>1.99</td>
<td>3.66E-04</td>
<td>1.99</td>
<td>7.76E-04</td>
<td>1.99</td>
</tr>
<tr>
<td>160*160</td>
<td>246664</td>
<td>7.58E-05</td>
<td>2.00</td>
<td>9.15E-05</td>
<td>2.00</td>
<td>1.94E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>DG-Strang symmetrical OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.1</td>
<td>1.54E-01</td>
<td>-</td>
<td>1.65E-01</td>
<td>-</td>
<td>2.69E-01</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>0.93</td>
<td>3.37E-02</td>
<td>2.19</td>
<td>3.66E-02</td>
<td>2.17</td>
<td>6.23E-02</td>
<td>2.11</td>
</tr>
<tr>
<td>40*40</td>
<td>14.55</td>
<td>8.19E-03</td>
<td>2.04</td>
<td>8.90E-03</td>
<td>2.04</td>
<td>1.53E-03</td>
<td>2.02</td>
</tr>
<tr>
<td>80*80</td>
<td>499.97</td>
<td>2.03E-03</td>
<td>2.01</td>
<td>2.21E-03</td>
<td>2.01</td>
<td>3.82E-03</td>
<td>2.01</td>
</tr>
<tr>
<td>160*160</td>
<td>13625</td>
<td>5.07E-04</td>
<td>2.00</td>
<td>5.52E-04</td>
<td>2.00</td>
<td>9.53E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>DG-trapezoidal OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.1</td>
<td>1.07E-01</td>
<td>-</td>
<td>1.08E-01</td>
<td>-</td>
<td>1.31E-01</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>0.87</td>
<td>1.93E-02</td>
<td>2.47</td>
<td>1.96E-02</td>
<td>2.46</td>
<td>2.65E-02</td>
<td>2.31</td>
</tr>
<tr>
<td>40*40</td>
<td>12.93</td>
<td>4.40E-03</td>
<td>2.14</td>
<td>4.49E-03</td>
<td>2.13</td>
<td>6.28E-03</td>
<td>2.08</td>
</tr>
<tr>
<td>80*80</td>
<td>340.27</td>
<td>1.07E-03</td>
<td>2.04</td>
<td>1.10E-03</td>
<td>2.03</td>
<td>1.55E-03</td>
<td>2.02</td>
</tr>
<tr>
<td>160*160</td>
<td>14177</td>
<td>2.66E-04</td>
<td>2.01</td>
<td>2.72E-04</td>
<td>2.01</td>
<td>3.86E-04</td>
<td>2.01</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$.  

40
Figure 2.9. Error as a function of $h_{\text{min}}$ for three different approaches for Example 2.5, 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.
Figure 2.10. Error as a function of CPU time for three different approaches for Example 2.5, 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.
Example 2.6 Consider the two-dimensional linear problem the same as equation \ref{example2.6:original}

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

with the no flux boundary conditions. The exact solution is

\[ u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y). \]

We use rectangular meshes to perform the convergence study. The refinement of the meshes is done in a uniform way, by cutting each rectangular into four smaller similar ones. The simulation is carried up to \(T = 8.0\) at which the \(L^1\), \(L^2\) and \(L^\infty\) errors are measured. The time step size is increased successively as \(\Delta t = h_{\text{min}}, 2h_{\text{min}}, 4h_{\text{min}}, 8h_{\text{min}}\), and \(16h_{\text{min}}\) where \(h_{\text{min}} = 1/N\) and \(N \times N\) is the number of spatial elements. CPU time, errors, and order of accuracy are reported in Table \ref{example2.6:table} for different ratios of \(\Delta t\) and \(h_{\text{min}}\). Similarly to the one-dimensional problem in Example 2.3, 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 in Table \ref{example2.6:table} and in Figure \ref{example2.6:figure}. 

43
TABLE 2.6: CPU TIME, ERROR, AND ORDER OF ACCURACY OF THE DG-TRAPEZOIDAL OS FOR EXAMPLE 2.6

<table>
<thead>
<tr>
<th># of cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t = h_{min}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.32</td>
<td>1.00E-05</td>
<td>-</td>
<td>1.22E-05</td>
<td>-</td>
<td>2.42E-05</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>3.48</td>
<td>2.69E-06</td>
<td>1.89</td>
<td>3.28E-06</td>
<td>1.90</td>
<td>6.52E-06</td>
<td>1.89</td>
</tr>
<tr>
<td>40*40</td>
<td>53.27</td>
<td>6.85E-07</td>
<td>1.97</td>
<td>8.36E-07</td>
<td>1.97</td>
<td>1.66E-06</td>
<td>1.97</td>
</tr>
<tr>
<td>80*80</td>
<td>1695</td>
<td>1.72E-07</td>
<td>1.99</td>
<td>2.10E-07</td>
<td>1.99</td>
<td>4.18E-07</td>
<td>1.99</td>
</tr>
<tr>
<td>160*160</td>
<td>55443</td>
<td>4.31E-08</td>
<td>2.00</td>
<td>5.25E-08</td>
<td>2.00</td>
<td>1.05E-07</td>
<td>2.00</td>
</tr>
<tr>
<td>$\Delta t = 2h_{min}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.17</td>
<td>9.08E-06</td>
<td>-</td>
<td>1.11E-05</td>
<td>-</td>
<td>2.19E-05</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>1.65</td>
<td>2.45E-06</td>
<td>1.89</td>
<td>2.98E-06</td>
<td>1.89</td>
<td>5.91E-06</td>
<td>1.89</td>
</tr>
<tr>
<td>40*40</td>
<td>26.36</td>
<td>6.23E-07</td>
<td>1.97</td>
<td>7.59E-07</td>
<td>1.97</td>
<td>1.51E-06</td>
<td>1.97</td>
</tr>
<tr>
<td>80*80</td>
<td>885.92</td>
<td>1.57E-07</td>
<td>1.99</td>
<td>1.91E-07</td>
<td>1.99</td>
<td>3.80E-07</td>
<td>1.99</td>
</tr>
<tr>
<td>160*160</td>
<td>25457</td>
<td>3.92E-08</td>
<td>2.00</td>
<td>4.77E-08</td>
<td>2.00</td>
<td>9.49E-08</td>
<td>2.00</td>
</tr>
<tr>
<td>$\Delta t = 4h_{min}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.09</td>
<td>5.34E-06</td>
<td>-</td>
<td>6.40E-06</td>
<td>-</td>
<td>1.25E-05</td>
<td>-</td>
</tr>
<tr>
<td>20*20</td>
<td>0.94</td>
<td>1.48E-06</td>
<td>1.86</td>
<td>1.77E-06</td>
<td>1.85</td>
<td>3.47E-06</td>
<td>1.85</td>
</tr>
<tr>
<td>40*40</td>
<td>15.02</td>
<td>3.79E-07</td>
<td>1.96</td>
<td>4.54E-07</td>
<td>1.96</td>
<td>8.92E-07</td>
<td>1.96</td>
</tr>
<tr>
<td>80*80</td>
<td>490.55</td>
<td>9.53E-08</td>
<td>1.99</td>
<td>1.14E-07</td>
<td>1.99</td>
<td>2.24E-07</td>
<td>1.99</td>
</tr>
<tr>
<td>160*160</td>
<td>13740</td>
<td>2.39E-08</td>
<td>2.00</td>
<td>2.86E-08</td>
<td>2.00</td>
<td>5.62E-08</td>
<td>2.00</td>
</tr>
</tbody>
</table>

continued...
<table>
<thead>
<tr>
<th># of cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t = 8h_{\text{min}}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.05</td>
<td>1.15E-05</td>
<td></td>
<td>1.40E-05</td>
<td></td>
<td>2.76E-05</td>
<td></td>
</tr>
<tr>
<td>20*20</td>
<td>0.56</td>
<td>2.70E-06</td>
<td>2.09</td>
<td>3.29E-06</td>
<td>2.09</td>
<td>6.54E-06</td>
<td>2.08</td>
</tr>
<tr>
<td>40*40</td>
<td>9.35</td>
<td>6.56E-07</td>
<td>2.04</td>
<td>7.99E-07</td>
<td>2.04</td>
<td>1.59E-06</td>
<td>2.04</td>
</tr>
<tr>
<td>80*80</td>
<td>355.55</td>
<td>1.63E-07</td>
<td>2.01</td>
<td>1.98E-07</td>
<td>2.01</td>
<td>3.94E-07</td>
<td>2.01</td>
</tr>
<tr>
<td>160*160</td>
<td>8344</td>
<td>4.06E-08</td>
<td>2.00</td>
<td>4.95E-08</td>
<td>2.00</td>
<td>9.84E-08</td>
<td>2.00</td>
</tr>
<tr>
<td>$\Delta t = 16h_{\text{min}}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10*10</td>
<td>0.03</td>
<td>1.32E-04</td>
<td></td>
<td>1.62E-04</td>
<td></td>
<td>3.23E-04</td>
<td></td>
</tr>
<tr>
<td>20*20</td>
<td>0.39</td>
<td>1.96E-05</td>
<td>2.74</td>
<td>2.42E-05</td>
<td>2.74</td>
<td>4.84E-05</td>
<td>2.74</td>
</tr>
<tr>
<td>40*40</td>
<td>6.72</td>
<td>4.77E-06</td>
<td>2.04</td>
<td>5.88E-06</td>
<td>2.04</td>
<td>1.18E-05</td>
<td>2.04</td>
</tr>
<tr>
<td>80*80</td>
<td>285.71</td>
<td>1.17E-06</td>
<td>2.03</td>
<td>1.44E-06</td>
<td>2.03</td>
<td>2.88E-06</td>
<td>2.03</td>
</tr>
<tr>
<td>160*160</td>
<td>5295</td>
<td>2.91E-07</td>
<td>2.01</td>
<td>3.59E-07</td>
<td>2.01</td>
<td>7.18E-07</td>
<td>2.01</td>
</tr>
</tbody>
</table>

Final time $T = 8.0$. Time step size is increased successively.
Figure 2.11. Error as a function of $h_{\text{min}}$ for the DG-trapezoidal OS for Example 2.6.
Example 2.7 Consider the two-dimensional linear problem the same as equation 2.63

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

(2.66)

with the no flux boundary conditions. The exact solution is

\[
u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y).
\]

We use uniform triangle meshes. The simulation is carried up to \( T = 1.0 \) at which the \( L^1 \), \( L^2 \) and \( L^\infty \) errors are measured. The time step size is \( \Delta t = h_{\text{min}} \) where 

\[
h_{\text{min}} = 1/N \quad \text{and} \quad 2 \times N \times N \quad \text{is the number of spatial elements.}
\]

We use the three different approaches the DG spatial discretization 2.43 with the Crank Nicholson temporal discretization (without OS), the DG-Strang symmetrical OS scheme 2.54 - 2.56, and the DG-trapezoidal OS scheme 2.57 - 2.59. We list the CPU time, errors, and order of accuracy for the DG w/o OS, the DG-Strang symmetrical OS, and the DG-trapezoidal OS in Table 2.7. The second order accuracy is achieved for all of these three approaches in Figure 2.12. In Figure 2.13, we 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. The DG-Strang symmetrical OS approach is the most expensive and the DG w/o OS approach falls between them in this example.
<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DG method w/o OS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2<em>5</em>5</td>
<td>0.02</td>
<td>1.65E-02</td>
<td>-</td>
<td>1.99E-02</td>
<td>-</td>
<td>5.11E-02</td>
<td>-</td>
</tr>
<tr>
<td>2<em>10</em>10</td>
<td>0.29</td>
<td>3.51E-03</td>
<td>2.24</td>
<td>4.18E-03</td>
<td>2.25</td>
<td>1.00E-02</td>
<td>2.35</td>
</tr>
<tr>
<td>2<em>20</em>20</td>
<td>7.19</td>
<td>8.99E-04</td>
<td>1.97</td>
<td>1.08E-03</td>
<td>1.95</td>
<td>3.13E-03</td>
<td>1.68</td>
</tr>
<tr>
<td>2<em>40</em>40</td>
<td>254.06</td>
<td>2.26E-04</td>
<td>1.99</td>
<td>2.72E-04</td>
<td>1.99</td>
<td>9.29E-04</td>
<td>1.75</td>
</tr>
<tr>
<td>2<em>80</em>80</td>
<td>10918</td>
<td>5.68E-05</td>
<td>2.00</td>
<td>6.82E-05</td>
<td>2.00</td>
<td>2.68E-04</td>
<td>1.79</td>
</tr>
<tr>
<td>2<em>160</em>160</td>
<td>379443</td>
<td>1.42E-05</td>
<td>2.00</td>
<td>1.71E-05</td>
<td>2.00</td>
<td>7.59E-05</td>
<td>1.82</td>
</tr>
<tr>
<td><strong>DG-Strang symmetrical OS scheme</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2<em>5</em>5</td>
<td>0.01</td>
<td>2.21E-01</td>
<td>-</td>
<td>2.60E-01</td>
<td>-</td>
<td>6.96E-01</td>
<td>-</td>
</tr>
<tr>
<td>2<em>10</em>10</td>
<td>0.13</td>
<td>5.29E-02</td>
<td>2.06</td>
<td>6.29E-02</td>
<td>2.05</td>
<td>1.89E-01</td>
<td>1.88</td>
</tr>
<tr>
<td>2<em>20</em>20</td>
<td>1.69</td>
<td>1.31E-02</td>
<td>2.02</td>
<td>1.56E-02</td>
<td>2.01</td>
<td>4.79E-02</td>
<td>1.98</td>
</tr>
<tr>
<td>2<em>40</em>40</td>
<td>45.9</td>
<td>3.26E-03</td>
<td>2.00</td>
<td>3.89E-03</td>
<td>2.00</td>
<td>1.20E-02</td>
<td>2.00</td>
</tr>
<tr>
<td>2<em>80</em>80</td>
<td>1545</td>
<td>8.14E-04</td>
<td>2.00</td>
<td>9.72E-04</td>
<td>2.00</td>
<td>3.00E-03</td>
<td>2.00</td>
</tr>
<tr>
<td>2<em>160</em>160</td>
<td>84117</td>
<td>2.03E-04</td>
<td>2.00</td>
<td>2.43E-04</td>
<td>2.00</td>
<td>7.51E-04</td>
<td>2.00</td>
</tr>
<tr>
<td><strong>DG-trapezoidal OS scheme</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2<em>5</em>5</td>
<td>0.01</td>
<td>1.23E-02</td>
<td>-</td>
<td>1.51E-02</td>
<td>-</td>
<td>3.67E-02</td>
<td>-</td>
</tr>
<tr>
<td>2<em>10</em>10</td>
<td>0.12</td>
<td>3.04E-03</td>
<td>2.02</td>
<td>3.65E-03</td>
<td>2.05</td>
<td>1.00E-02</td>
<td>1.88</td>
</tr>
<tr>
<td>2<em>20</em>20</td>
<td>1.39</td>
<td>7.78E-04</td>
<td>1.96</td>
<td>9.46E-04</td>
<td>1.95</td>
<td>3.10E-03</td>
<td>1.69</td>
</tr>
<tr>
<td>2<em>40</em>40</td>
<td>30.48</td>
<td>1.96E-04</td>
<td>1.99</td>
<td>2.39E-04</td>
<td>1.99</td>
<td>9.20E-04</td>
<td>1.75</td>
</tr>
<tr>
<td>2<em>80</em>80</td>
<td>893.67</td>
<td>4.92E-05</td>
<td>2.00</td>
<td>6.00E-05</td>
<td>2.00</td>
<td>2.66E-04</td>
<td>1.79</td>
</tr>
<tr>
<td>2<em>160</em>160</td>
<td>44042</td>
<td>1.23E-05</td>
<td>2.00</td>
<td>1.50E-05</td>
<td>2.00</td>
<td>7.54E-05</td>
<td>1.82</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$.  

48
Figure 2.12. Error as a function of $h_{\text{min}}$ for three different approaches for Example 2.7, 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.
Figure 2.13. Error as a function of CPU time for three different approaches for Example 2.7, 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.
Example 2.8 Consider the two-dimensional linear problem the same as equation 2.63

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

(2.67)

with the no flux boundary conditions. The exact solution is

\[ u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y). \]

We use uniform triangular meshes to perform the convergence study. The refinement of the meshes is done in a uniform way, by cutting each triangular into four smaller similar ones. The simulation is carried up to \( T = 8.0 \) at which the \( L^1 \), \( L^2 \) and \( L^\infty \) errors are measured. The time step size is increased successively as \( \Delta t = h_{\text{min}}, 2h_{\text{min}}, 4h_{\text{min}}, 8h_{\text{min}}, \) and \( 16h_{\text{min}} \) where \( h_{\text{min}} = 1/N \) and \( 2*N*N \) is the number of spatial elements. CPU time, errors, and order of accuracy are reported in Table 2.8 for different ratios of \( \Delta t \) and \( h_{\text{min}} \). Similarly to the two-dimensional problem in Example 2.6, 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 2.8 and in Figure 2.14.
TABLE 2.8: CPU TIME, ERROR, AND ORDER OF ACCURACY OF THE DG-TRAPEZOIDAL OS FOR EXAMPLE 2.8

<table>
<thead>
<tr>
<th># of cells</th>
<th>CPU(s)</th>
<th>( L^1 ) error</th>
<th>order</th>
<th>( L^2 ) error</th>
<th>order</th>
<th>( L^\infty ) error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta t = h_{\text{min}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2<em>5</em>5</td>
<td>0.09</td>
<td>1.00E-05</td>
<td>-</td>
<td>1.22E-05</td>
<td>-</td>
<td>2.99E-05</td>
<td>-</td>
</tr>
<tr>
<td>2<em>10</em>10</td>
<td>0.67</td>
<td>2.77E-06</td>
<td>1.86</td>
<td>3.35E-06</td>
<td>1.86</td>
<td>9.84E-06</td>
<td>1.60</td>
</tr>
<tr>
<td>2<em>20</em>20</td>
<td>7.83</td>
<td>7.10E-07</td>
<td>1.96</td>
<td>8.63E-07</td>
<td>1.96</td>
<td>3.01E-06</td>
<td>1.71</td>
</tr>
<tr>
<td>2<em>40</em>40</td>
<td>171.44</td>
<td>1.79E-07</td>
<td>1.99</td>
<td>2.18E-07</td>
<td>1.99</td>
<td>8.84E-07</td>
<td>1.77</td>
</tr>
<tr>
<td>2<em>80</em>80</td>
<td>5745.65</td>
<td>4.48E-08</td>
<td>2.00</td>
<td>5.46E-08</td>
<td>2.00</td>
<td>2.54E-07</td>
<td>1.80</td>
</tr>
<tr>
<td>2<em>160</em>160</td>
<td>365318</td>
<td>1.12E-08</td>
<td>2.00</td>
<td>1.37E-08</td>
<td>2.00</td>
<td>7.15E-08</td>
<td>1.83</td>
</tr>
<tr>
<td>( \Delta t = 2h_{\text{min}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2<em>5</em>5</td>
<td>0.04</td>
<td>6.62E-06</td>
<td>-</td>
<td>8.15E-06</td>
<td>-</td>
<td>2.11E-05</td>
<td>-</td>
</tr>
<tr>
<td>2<em>10</em>10</td>
<td>0.36</td>
<td>1.93E-06</td>
<td>1.78</td>
<td>2.36E-06</td>
<td>1.79</td>
<td>7.10E-06</td>
<td>1.57</td>
</tr>
<tr>
<td>2<em>20</em>20</td>
<td>4.4</td>
<td>4.98E-07</td>
<td>1.96</td>
<td>6.15E-07</td>
<td>1.94</td>
<td>2.31E-06</td>
<td>1.62</td>
</tr>
<tr>
<td>2<em>40</em>40</td>
<td>97.98</td>
<td>1.26E-07</td>
<td>1.99</td>
<td>1.56E-07</td>
<td>1.98</td>
<td>7.07E-07</td>
<td>1.70</td>
</tr>
<tr>
<td>2<em>80</em>80</td>
<td>3106.74</td>
<td>3.15E-08</td>
<td>2.00</td>
<td>3.90E-08</td>
<td>1.99</td>
<td>2.09E-07</td>
<td>1.76</td>
</tr>
<tr>
<td>2<em>160</em>160</td>
<td>175272</td>
<td>7.89E-09</td>
<td>2.00</td>
<td>9.77E-09</td>
<td>2.00</td>
<td>6.05E-08</td>
<td>1.79</td>
</tr>
<tr>
<td>( \Delta t = 4h_{\text{min}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2<em>5</em>5</td>
<td>0.02</td>
<td>1.25E-05</td>
<td>-</td>
<td>1.52E-05</td>
<td>-</td>
<td>5.93E-05</td>
<td>-</td>
</tr>
<tr>
<td>2<em>10</em>10</td>
<td>0.19</td>
<td>3.05E-06</td>
<td>2.04</td>
<td>3.67E-06</td>
<td>2.05</td>
<td>1.56E-05</td>
<td>1.93</td>
</tr>
<tr>
<td>2<em>20</em>20</td>
<td>2.21</td>
<td>7.51E-07</td>
<td>2.02</td>
<td>9.05E-07</td>
<td>2.02</td>
<td>4.33E-06</td>
<td>1.85</td>
</tr>
<tr>
<td>2<em>40</em>40</td>
<td>48.88</td>
<td>1.86E-07</td>
<td>2.01</td>
<td>2.25E-07</td>
<td>2.01</td>
<td>1.21E-06</td>
<td>1.84</td>
</tr>
</tbody>
</table>

continued...
# TABLE 2.8: Continued

<table>
<thead>
<tr>
<th># of cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>2<em>80</em>80</td>
<td>1608.41</td>
<td>4.65E-08</td>
<td>2.00</td>
<td>5.62E-08</td>
<td>2.00</td>
<td>3.33E-07</td>
<td>1.85</td>
</tr>
<tr>
<td>2<em>160</em>160</td>
<td>92472.5</td>
<td>1.16E-08</td>
<td>2.00</td>
<td>1.40E-08</td>
<td>2.00</td>
<td>9.14E-08</td>
<td>1.87</td>
</tr>
</tbody>
</table>

$\Delta t = 8h_{\text{min}}$

<table>
<thead>
<tr>
<th># of cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>2<em>5</em>5</td>
<td>0.02</td>
<td>1.33E-04</td>
<td>-</td>
<td>1.63E-04</td>
<td>-</td>
<td>3.93E-04</td>
<td>-</td>
</tr>
<tr>
<td>2<em>10</em>10</td>
<td>0.1</td>
<td>1.97E-05</td>
<td>2.75</td>
<td>2.43E-05</td>
<td>2.75</td>
<td>5.86E-05</td>
<td>2.75</td>
</tr>
<tr>
<td>2<em>20</em>20</td>
<td>1.36</td>
<td>4.80E-06</td>
<td>2.04</td>
<td>5.90E-06</td>
<td>2.04</td>
<td>1.42E-05</td>
<td>2.04</td>
</tr>
<tr>
<td>2<em>40</em>40</td>
<td>30.12</td>
<td>1.18E-06</td>
<td>2.03</td>
<td>1.45E-06</td>
<td>2.03</td>
<td>3.59E-06</td>
<td>1.99</td>
</tr>
<tr>
<td>2<em>80</em>80</td>
<td>896.31</td>
<td>2.93E-07</td>
<td>2.01</td>
<td>3.60E-07</td>
<td>2.01</td>
<td>9.23E-07</td>
<td>1.96</td>
</tr>
<tr>
<td>2<em>160</em>160</td>
<td>42174.2</td>
<td>7.31E-08</td>
<td>2.00</td>
<td>8.99E-08</td>
<td>2.00</td>
<td>2.38E-07</td>
<td>1.95</td>
</tr>
</tbody>
</table>

$\Delta t = 16h_{\text{min}}$

<table>
<thead>
<tr>
<th># of cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>2<em>5</em>5</td>
<td>0.01</td>
<td>5.36E-03</td>
<td>-</td>
<td>6.62E-03</td>
<td>-</td>
<td>1.44E-02</td>
<td>-</td>
</tr>
<tr>
<td>2<em>10</em>10</td>
<td>0.08</td>
<td>1.47E-04</td>
<td>5.19</td>
<td>1.82E-04</td>
<td>5.19</td>
<td>3.86E-04</td>
<td>5.22</td>
</tr>
<tr>
<td>2<em>20</em>20</td>
<td>0.93</td>
<td>2.19E-05</td>
<td>2.75</td>
<td>2.71E-05</td>
<td>2.75</td>
<td>5.70E-05</td>
<td>2.76</td>
</tr>
<tr>
<td>2<em>40</em>40</td>
<td>21.78</td>
<td>5.31E-06</td>
<td>2.05</td>
<td>6.55E-06</td>
<td>2.05</td>
<td>1.37E-05</td>
<td>2.05</td>
</tr>
<tr>
<td>2<em>80</em>80</td>
<td>589.27</td>
<td>1.30E-06</td>
<td>2.03</td>
<td>1.61E-06</td>
<td>2.03</td>
<td>3.39E-06</td>
<td>2.02</td>
</tr>
<tr>
<td>2<em>160</em>160</td>
<td>27822.6</td>
<td>3.24E-07</td>
<td>2.01</td>
<td>3.99E-07</td>
<td>2.01</td>
<td>8.50E-07</td>
<td>1.99</td>
</tr>
</tbody>
</table>

Final time $T = 8.0$. Time step size is increased successively.
Figure 2.14. Error as a function of $h_{\text{min}}$ for the DG-trapezoidal OS for Example 2.8.
Example 2.9 Consider the two-dimensional linear problem the same as equation 2.63

\[
\begin{cases}
    u_t = u_{xx} + u_{yy} - u + 2\pi^2 e^{-t} \cos(\pi x) \cos(\pi y), & (x, y) \in (0, 1) \times (0, 1); \\
    u(x, y, 0) = \cos(\pi x) \cos(\pi y),
\end{cases}
\] (2.68)

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, show in Figure 2.20 for the coarsest case of 44 cells. The simulation is carried up to \( T = 1.0 \) at which the \( L^1 \), \( L^2 \) and \( L^\infty \) errors are measured. The time step size is \( \Delta t = h_{\text{min}} \). We use the three different approaches: the DG spatial discretization 2.43 with the Crank Nicholson temporal discretization (without OS), the DG-Strang symmetrical OS scheme 2.54 - 2.56, and the DG-trapezoidal OS scheme 2.57 - 2.59. We list the CPU time, errors, and order of accuracy for the DG w/o OS, the DG-Strang symmetrical OS, and the DG-trapezoidal OS in Table 2.9. The second order accuracy is achieved for all of these three approaches in Figure 2.15. In Figure 2.16, we 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.
TABLE 2.9

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.9

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG method w/o OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.03</td>
<td>9.85E-03</td>
<td>-</td>
<td>1.18E-02</td>
<td>-</td>
<td>2.52E-02</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>0.5</td>
<td>3.29E-03</td>
<td>1.58</td>
<td>3.92E-03</td>
<td>1.58</td>
<td>1.24E-02</td>
<td>1.02</td>
</tr>
<tr>
<td>704</td>
<td>10.58</td>
<td>8.45E-04</td>
<td>1.96</td>
<td>1.01E-03</td>
<td>1.95</td>
<td>3.38E-03</td>
<td>1.88</td>
</tr>
<tr>
<td>2816</td>
<td>303.36</td>
<td>2.12E-04</td>
<td>1.99</td>
<td>2.55E-04</td>
<td>1.99</td>
<td>8.72E-04</td>
<td>1.96</td>
</tr>
<tr>
<td>11264</td>
<td>55415</td>
<td>5.31E-05</td>
<td>2.00</td>
<td>6.37E-05</td>
<td>2.00</td>
<td>2.21E-04</td>
<td>1.99</td>
</tr>
<tr>
<td>DG-Strang symmetrical OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.02</td>
<td>5.91E-02</td>
<td>-</td>
<td>7.03E-02</td>
<td>-</td>
<td>1.34E-01</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>0.17</td>
<td>2.25E-02</td>
<td>1.40</td>
<td>2.78E-02</td>
<td>1.34</td>
<td>1.22E-02</td>
<td>1.36</td>
</tr>
<tr>
<td>704</td>
<td>2.19</td>
<td>6.94E-03</td>
<td>1.69</td>
<td>9.30E-03</td>
<td>1.58</td>
<td>4.92E-03</td>
<td>1.31</td>
</tr>
<tr>
<td>2816</td>
<td>57.95</td>
<td>1.84E-03</td>
<td>1.91</td>
<td>2.52E-03</td>
<td>1.88</td>
<td>1.39E-03</td>
<td>1.82</td>
</tr>
<tr>
<td>11264</td>
<td>2441</td>
<td>4.68E-04</td>
<td>1.98</td>
<td>6.43E-04</td>
<td>1.97</td>
<td>3.59E-04</td>
<td>1.96</td>
</tr>
<tr>
<td>DG-trapezoidal OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.02</td>
<td>9.40E-02</td>
<td>-</td>
<td>1.14E-02</td>
<td>-</td>
<td>2.51E-02</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>0.16</td>
<td>2.72E-02</td>
<td>1.79</td>
<td>3.25E-03</td>
<td>1.81</td>
<td>9.13E-03</td>
<td>1.46</td>
</tr>
<tr>
<td>704</td>
<td>1.78</td>
<td>6.92E-04</td>
<td>1.98</td>
<td>8.27E-04</td>
<td>1.97</td>
<td>2.45E-03</td>
<td>1.90</td>
</tr>
<tr>
<td>2816</td>
<td>35.45</td>
<td>1.74E-04</td>
<td>1.99</td>
<td>2.08E-04</td>
<td>1.99</td>
<td>6.31E-04</td>
<td>1.96</td>
</tr>
<tr>
<td>11264</td>
<td>1484</td>
<td>4.35E-05</td>
<td>2.00</td>
<td>5.20E-05</td>
<td>2.00</td>
<td>1.69E-04</td>
<td>1.90</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$. 

56
Figure 2.15. Error as a function of $h_{\text{min}}$ for three different approaches for Example 2.9, 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.
Figure 2.16. Error as a function of CPU time for three different approaches for Example 2.9, 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.
Example 2.10 Consider the two-dimensional nonlinear problem the same as equation 2.64

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

(2.69)

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, as in Example 2.9 (Figure 2.20). The simulation is carried up to \( T = 1.0 \) at which the \( L^1, L^2 \) and \( L^\infty \) errors are measured. The time step size is \( \Delta t = 0.1h_{\min} \). We use the same three different approaches as in Example 2.9, and list the CPU time, errors, and order of accuracy for the DG w/o OS, the DG-Strang symmetrical OS, and the DG-trapezoidal OS in Table 2.10. The second order accuracy is achieved for all of these three approaches in Figure 2.17. In Figure 2.18, we 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 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.
**TABLE 2.10**

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 2.10

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>( L^1 ) error</th>
<th>order</th>
<th>( L^2 ) error</th>
<th>order</th>
<th>( L^\infty ) error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG method w/o OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.4</td>
<td>1.42E-02</td>
<td></td>
<td>1.74E-02</td>
<td></td>
<td>3.83E-02</td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>6.31</td>
<td>3.74E-03</td>
<td>1.92</td>
<td>4.56E-03</td>
<td>1.93</td>
<td>1.04E-02</td>
<td>1.89</td>
</tr>
<tr>
<td>704</td>
<td>153.26</td>
<td>9.50E-04</td>
<td>1.98</td>
<td>1.16E-03</td>
<td>1.98</td>
<td>2.73E-03</td>
<td>1.92</td>
</tr>
<tr>
<td>2816</td>
<td>4957</td>
<td>2.39E-04</td>
<td>1.99</td>
<td>2.91E-04</td>
<td>1.99</td>
<td>7.41E-04</td>
<td>1.88</td>
</tr>
<tr>
<td>11264</td>
<td>753876</td>
<td>5.98E-05</td>
<td>2.00</td>
<td>7.30E-05</td>
<td>2.00</td>
<td>2.12E-04</td>
<td>1.81</td>
</tr>
<tr>
<td>DG-Strang symmetrical OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.23</td>
<td>1.36E-02</td>
<td></td>
<td>1.67E-02</td>
<td></td>
<td>3.84E-02</td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>1.85</td>
<td>3.80E-03</td>
<td>1.84</td>
<td>4.66E-03</td>
<td>1.84</td>
<td>1.29E-02</td>
<td>1.58</td>
</tr>
<tr>
<td>704</td>
<td>21.26</td>
<td>1.01E-03</td>
<td>1.91</td>
<td>1.24E-03</td>
<td>1.91</td>
<td>3.74E-03</td>
<td>1.78</td>
</tr>
<tr>
<td>2816</td>
<td>540.04</td>
<td>2.38E-04</td>
<td>2.09</td>
<td>2.94E-04</td>
<td>2.08</td>
<td>8.18E-04</td>
<td>2.19</td>
</tr>
<tr>
<td>11264</td>
<td>20081</td>
<td>6.31E-05</td>
<td>1.92</td>
<td>7.74E-05</td>
<td>1.92</td>
<td>2.34E-04</td>
<td>1.81</td>
</tr>
<tr>
<td>DG-trapezoidal OS scheme</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.2</td>
<td>1.20E-02</td>
<td></td>
<td>1.46E-02</td>
<td></td>
<td>3.35E-02</td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>1.7</td>
<td>3.04E-03</td>
<td>1.98</td>
<td>3.71E-03</td>
<td>1.98</td>
<td>8.93E-03</td>
<td>1.91</td>
</tr>
<tr>
<td>704</td>
<td>16.92</td>
<td>7.62E-04</td>
<td>1.99</td>
<td>9.34E-04</td>
<td>1.99</td>
<td>2.34E-03</td>
<td>1.93</td>
</tr>
<tr>
<td>2816</td>
<td>319.39</td>
<td>1.91E-04</td>
<td>2.00</td>
<td>2.33E-04</td>
<td>2.00</td>
<td>6.46E-04</td>
<td>1.86</td>
</tr>
<tr>
<td>11264</td>
<td>10844</td>
<td>4.77E-05</td>
<td>2.00</td>
<td>5.84E-05</td>
<td>2.00</td>
<td>1.88E-04</td>
<td>1.78</td>
</tr>
</tbody>
</table>

Final time \( T = 1.0 \).
Figure 2.17. Error as a function of $h_{\text{min}}$ for three different approaches for Example 2.10, 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.
Figure 2.18. Error as a function of CPU time for three different approaches for Example 2.10, 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.
Example 2.11 Consider the two-dimensional linear problem the same as equation

\[ u_t = u_{xx} + u_{yy} - u + 2\pi^2 e^{-t} \cos(\pi x) \cos(\pi y), \quad (x, y) \in (0, 1) \times (0, 1); \]
\[ u(x, y, 0) = \cos(\pi x) \cos(\pi y), \]

(2.70)

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 as that in Example 2.9 (Figure 2.20), to perform the convergence study. The refinement of the meshes is done in a uniform way, by cutting each triangular into four smaller similar ones. The simulation is carried up to \( T = 8.0 \) at which the \( L^1 \), \( L^2 \) and \( L^\infty \) errors are measured. The time step size is increased successively as \( \Delta t = h_{\min}, 2h_{\min}, 4h_{\min}, 8h_{\min}, \text{ and } 16h_{\min} \). CPU time, errors, and order of accuracy are reported in Table 2.11 for different ratios of \( \Delta t \) and \( h_{\min} \). Similarly to the one-dimensional problem in Example 2.3, the two-dimensional problem in Example 2.6 and 2.8, 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 2.11 and in Figure 2.19.
TABLE 2.11: CPU TIME, ERROR, AND ORDER OF ACCURACY OF THE DG-TRAPEZOIDAL OS FOR EXAMPLE 2.11

<table>
<thead>
<tr>
<th># of cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t = h_{\min}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.12</td>
<td>9.55E-06</td>
<td>-</td>
<td>1.13E-05</td>
<td>-</td>
<td>2.81E-05</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>1.04</td>
<td>2.44E-06</td>
<td>1.97</td>
<td>2.91E-06</td>
<td>1.95</td>
<td>7.78E-06</td>
<td>1.85</td>
</tr>
<tr>
<td>704</td>
<td>11.73</td>
<td>6.14E-07</td>
<td>1.99</td>
<td>7.37E-07</td>
<td>1.98</td>
<td>2.06E-06</td>
<td>1.91</td>
</tr>
<tr>
<td>2816</td>
<td>240</td>
<td>1.53E-07</td>
<td>2.00</td>
<td>1.84E-07</td>
<td>2.00</td>
<td>5.29E-07</td>
<td>1.96</td>
</tr>
<tr>
<td>11264</td>
<td>8676</td>
<td>3.52E-08</td>
<td>2.12</td>
<td>4.31E-08</td>
<td>2.10</td>
<td>1.58E-07</td>
<td>1.74</td>
</tr>
<tr>
<td>$\Delta t = 2h_{\min}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.06</td>
<td>9.10E-06</td>
<td>-</td>
<td>1.08E-05</td>
<td>-</td>
<td>3.09E-05</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>0.55</td>
<td>2.29E-06</td>
<td>1.99</td>
<td>2.76E-06</td>
<td>1.97</td>
<td>8.55E-06</td>
<td>1.85</td>
</tr>
<tr>
<td>704</td>
<td>6.23</td>
<td>5.77E-07</td>
<td>1.99</td>
<td>6.95E-07</td>
<td>1.99</td>
<td>2.22E-06</td>
<td>1.94</td>
</tr>
<tr>
<td>2816</td>
<td>123</td>
<td>1.44E-07</td>
<td>2.00</td>
<td>1.74E-07</td>
<td>2.00</td>
<td>5.66E-07</td>
<td>1.97</td>
</tr>
<tr>
<td>11264</td>
<td>4633</td>
<td>3.54E-08</td>
<td>2.02</td>
<td>4.28E-08</td>
<td>2.02</td>
<td>1.64E-07</td>
<td>1.79</td>
</tr>
<tr>
<td>$\Delta t = 4h_{\min}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.03</td>
<td>6.23E-06</td>
<td>-</td>
<td>7.87E-06</td>
<td>-</td>
<td>3.21E-05</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>0.3</td>
<td>1.56E-06</td>
<td>2.00</td>
<td>2.01E-06</td>
<td>1.97</td>
<td>8.94E-06</td>
<td>1.84</td>
</tr>
<tr>
<td>704</td>
<td>3.41</td>
<td>3.88E-07</td>
<td>2.01</td>
<td>5.03E-07</td>
<td>2.00</td>
<td>2.53E-06</td>
<td>1.82</td>
</tr>
<tr>
<td>2816</td>
<td>76</td>
<td>9.65E-08</td>
<td>2.01</td>
<td>1.25E-07</td>
<td>2.01</td>
<td>7.31E-07</td>
<td>1.79</td>
</tr>
<tr>
<td>11264</td>
<td>2784</td>
<td>2.39E-08</td>
<td>2.03</td>
<td>3.09E-08</td>
<td>2.02</td>
<td>2.08E-07</td>
<td>1.81</td>
</tr>
</tbody>
</table>

continued...
<table>
<thead>
<tr>
<th># of cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta t = 8h_{min}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.02</td>
<td>1.59E-05</td>
<td>-</td>
<td>1.96E-05</td>
<td>-</td>
<td>7.81E-05</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>0.17</td>
<td>3.54E-06</td>
<td>2.16</td>
<td>4.34E-06</td>
<td>2.20</td>
<td>1.92E-05</td>
<td>2.02</td>
</tr>
<tr>
<td>704</td>
<td>2.01</td>
<td>8.71E-07</td>
<td>2.02</td>
<td>1.06E-06</td>
<td>2.03</td>
<td>5.18E-06</td>
<td>1.89</td>
</tr>
<tr>
<td>2816</td>
<td>43.69</td>
<td>2.17E-07</td>
<td>2.00</td>
<td>2.65E-07</td>
<td>2.01</td>
<td>1.40E-06</td>
<td>1.89</td>
</tr>
<tr>
<td>11264</td>
<td>1679</td>
<td>5.48E-08</td>
<td>1.99</td>
<td>6.68E-08</td>
<td>1.99</td>
<td>3.77E-07</td>
<td>1.89</td>
</tr>
<tr>
<td>$\Delta t = 16h_{min}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.02</td>
<td>1.13E-04</td>
<td>-</td>
<td>1.39E-04</td>
<td>-</td>
<td>2.74E-04</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>0.1</td>
<td>2.30E-05</td>
<td>2.30</td>
<td>2.85E-05</td>
<td>2.29</td>
<td>6.88E-05</td>
<td>1.99</td>
</tr>
<tr>
<td>704</td>
<td>1.24</td>
<td>4.86E-06</td>
<td>2.24</td>
<td>5.99E-06</td>
<td>2.25</td>
<td>1.54E-05</td>
<td>2.16</td>
</tr>
<tr>
<td>2816</td>
<td>28.16</td>
<td>1.19E-06</td>
<td>2.03</td>
<td>1.46E-06</td>
<td>2.03</td>
<td>3.91E-06</td>
<td>1.98</td>
</tr>
<tr>
<td>11264</td>
<td>1114</td>
<td>2.96E-07</td>
<td>2.00</td>
<td>3.65E-07</td>
<td>2.01</td>
<td>1.00E-06</td>
<td>1.96</td>
</tr>
</tbody>
</table>

Final time $T = 8.0$. Time step size is increased successively.
Figure 2.19. Error as a function of $h_{\text{min}}$ for the DG-trapezoidal OS for Example 2.11.
Figure 2.20. The coarsest triangular mesh of 44 cells
CHAPTER 3

FIXED GRIDS AND 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 are distributed over a tissue domain or field, and when bound to cell receptors, assign different cell fates at different concentrations \[99\] \[108\]. 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 \[42\] \[55\]. In this chapter, we apply the DG methods with trapezoidal operator splitting \[2.57\] \[2.59\] to two reaction-diffusion systems in modeling 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.

3.1 The Schnakenberg Model

The Schnakenberg system \[92\] 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* \[39\]). It is also a classical example to test numerical methods for
reaction-diffusion equations (e.g. [48, 62, 63, 91]). The Schnakenberg system has the form

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

(3.1)

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

(3.2)

where \( C_a \) and \( C_i \) denote the concentration of activator and inhibitor respectively, \( D_1 \) and \( D_2 \) are diffusion coefficients, \( \kappa \), \( a \) and \( b \) are rate constants of the biochemical reactions. Following the setup in [48], we take the initial conditions as

\[
C_a(x, y, 0) = a + b + 10^{-3} \exp^{-100((x-x_0)^2+(y-y_0)^2)},
\]

(3.3)

\[
C_i(x, y, 0) = \frac{b}{(a+b)^2},
\]

(3.4)

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 3.1 - 3.4 on the unit square domain \( \Omega = (0, 1)^2 \). To study the performance and convergence of the DG methods with trapezoidal operator splitting 2.57 - 2.59, we list in Table 3.1 the CPU time, error, and order of accuracy for simulations of the Schnakenberg model 3.1 - 3.2. In this case, the spatial resolution is fixed with 1024 elements, 553 nodes and 1576 sides, as shown in Figure 3.1. The error at \( \Delta t \) 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.

\[
E_{\Delta t} = ||C_{a,\Delta t} - C_{a,2\Delta t}||.
\]

(3.5)

The DG methods with trapezoidal operator splitting 2.57 - 2.59 clearly shows a
second order of accuracy in time as expected.

We compute \( 3.1 - 3.4 \) on the same rectangular domain and the time evolutions of the concentration of activator \( C_a \) is shown in Figure 3.2. We can observe that the initial perturbation in \( 3.3 - 3.4 \) is amplified and spread, leading to formations of spot-like patterns.

Next we vary the geometrical shape and size of the domain, but keep all of parameters in the model \( 3.1 - 3.4 \) unchanged. The time evolution plots of concentrations of activator \( C_a \) are for circular domains \( \Omega = \{ (x, y) | x^2 + y^2 < 1 \} \) (Figure 3.3) and \( \Omega = \{ (x, y) | (x-0.5)^2 + (y-0.5)^2 < 0.5^2 \} \) (Figure 3.4), for elliptic domains \( \Omega = \{ (x, y) | (\frac{x}{0.5})^2 + y^2 < 1 \} \) (Figure 3.5), \( \Omega = \{ (x, y) | (\frac{x-0.5}{0.25})^2 + (\frac{y-0.5}{0.5})^2 < 1 \} \) (Figure 3.6), and \( \Omega = \{ (x, y) | (\frac{x}{0.1})^2 + (\frac{y}{0.5})^2 < 1 \} \) (Figure 3.7), and narrow rectangular domains \( \Omega = [0, 0.05] \times [0, 1] \) (Figure 3.8) and \( \Omega = [0, 0.1] \times [0, 1] \) (Figure 3.9). It is interesting to notice that spot-like patterns are formed on the circular and elliptic domains in Figures 3.4 - 3.7, but stripe-like patterns are formed on the narrow rectangular domains in Figure 3.8 and 3.9. These simulations of the Schnakenberg system on different domains give an example about the sensitivity of patterns in reaction-diffusion systems with respect to the domain size and geometrical shape.
TABLE 3.1

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR DG-TRAPEZOIDAL OS APPLIED TO THE SCHNAKENBERG MODEL

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>5e-4</td>
<td>3400</td>
<td>6.23E-03</td>
<td>-</td>
<td>1.21E-02</td>
<td>-</td>
<td>9.31E-02</td>
<td>-</td>
</tr>
<tr>
<td>2.5e-4</td>
<td>5846</td>
<td>1.69E-03</td>
<td>1.88</td>
<td>3.36E-03</td>
<td>1.85</td>
<td>2.58E-02</td>
<td>1.85</td>
</tr>
<tr>
<td>1.25e-4</td>
<td>10464</td>
<td>4.33E-04</td>
<td>1.97</td>
<td>8.63E-04</td>
<td>1.96</td>
<td>6.63E-03</td>
<td>1.96</td>
</tr>
<tr>
<td>6.25e-5</td>
<td>19018</td>
<td>1.09E-04</td>
<td>1.99</td>
<td>2.16E-04</td>
<td>1.99</td>
<td>1.67E-03</td>
<td>1.99</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$.

Figure 3.1. Triangular mesh for Schnakenberg model of 1024 cells
Figure 3.2. Numerical solution of the Schnakenberg model on $(0, 1)^2$. Contour plots of time evolution of the concentration of the activator $Ca$. 

(a) Mesh

(b) $t = 0.5$

(c) $t = 1.0$

(d) $t = 2.0$
Figure 3.3. Numerical solution of the Schnakenberg model on circular domain \( \{(x, y) | x^2 + y^2 < 1\} \). Contour plots of time evolution of the concentration of the activator \( Ca \).
Figure 3.4. Numerical solution of the Schnakenberg model on circular domain \( \{(x, y) \mid (x - 0.5)^2 + (y - 0.5)^2 < 0.5^2 \} \). Contour plots of time evolution of the concentration of the activator \( Ca \).
Figure 3.5. Numerical solution of the Schnakenberg model on elliptic domain \( \{(x, y)| 0.5^2 + y^2 < 1\} \). Contour plots of time evolution of the concentration of the activator \( Ca \).

Figure 3.6. Numerical solution of the Schnakenberg model on elliptic domain \( \{(x, y)| (\frac{x}{0.5} - 0.25)^2 + (\frac{y}{0.5} - 0.5)^2 < 1\} \). Contour plots of time evolution of the concentration of the activator \( Ca \).
Figure 3.7. Numerical solution of the Schnakenberg model on elliptic domain \( \{(x, y) | (\frac{x-0.5}{0.1})^2 + (\frac{y-0.5}{0.5})^2 < 1 \} \). Contour plots of time evolution of the concentration of the activator \( Ca \).
Figure 3.8. Numerical solution of the Schnakenberg model on $(0, 0.05) \times (0, 1)$. Contour plots of time evolution of the concentration of the activator $Ca$. 
Figure 3.9. Numerical solution of the Schnakenberg model on 
$(0, 0.1) \times (0, 1)$. Contour plots of time evolution of the concentration of
the activator $Ca$. 

(a) Mesh  (b) $t = 0.5$  (c) $t = 1.0$  (d) $t = 2.0$
3.2 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 [81, 103]. 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 [79, 82]. It thus lends itself to being modeled by systems like [1.1] which readily generate spot- and stripe-like patterns.

The most detailed model for vertebrate limb development presented thus far is that of [45], 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 mathematically and computationally. Recently in [4], 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

\[
\frac{\partial C_a}{\partial t} = D_a \nabla^2 C_a + U(C_a) - k_a C_a C_i; \tag{3.6}
\]

\[
\frac{\partial C_i}{\partial t} = D_i \nabla^2 C_i + V(C_a) - k_a C_a C_i, \tag{3.7}
\]

where \( C_a \) denotes the concentration of the activator TGF-\( \beta \), \( C_i \) 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. The system are subject to no-flux boundary conditions and zero initial concentrations for \( C_a \) and \( C_i \). The functions \( U \) and \( V \) are given by

\[
U(C_a) = [J_a^1 \alpha(C_a) + J_a(C_a) \beta(C_a)] R_{eq}, \tag{3.8}
\]

\[
V(C_a) = J_i(C_a) \beta(C_a) R_{eq}, \tag{3.9}
\]

where \( J_a(C_a) = J_{a\text{max}}(C_a/s)^n/[1 + (C_a/s)^n] \), \( J_i(C_a) = J_{i\text{max}}(C_a/\delta)^q/[1 + (C_a/\delta)^q] \), and \( \beta(C_a) = \beta_1 C_a/[(\beta_2 + C_a)] \). Following [4], the parameter values in the system are taken as \( D_a = 1, D_i = 100.3, J_{a\text{max}} = 6.0\lambda, J_{i\text{max}} = 8.0\lambda, s = 4.0, k_a = \lambda, J_a^1 \alpha(C_a) = 0.05\lambda, \beta_1 = 0.693473, \beta_2 = 2.66294, R_{eq} = 2.0, n = q = 2 \), where the value of \( \lambda \) 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 [45] have non-standard geometries, it is important to study the effects of domain geometry on the pattern generated by the model 3.6 - 3.7. We use the DG-trapezoidal OS methods described in Section 2.1.2 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 activator-inhibitor dynamics, initially comprises the entire limb bud, but is increasingly confined to narrower bands of tissue at the limb buds tip [79]. 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 Figure 3.10a. This domain has height 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 Figure 3.10d. The parameter \( \lambda = 500 \) in this case, and a one-stripe pattern is observed. Next we shrink the domain width in the horizontal direction to be 0.1 and 0.07 respectively, shown in Figure 3.10b and 3.10c. Two-stripe and three-stripe patterns can be obtained on these shrunken domains if we choose \( \lambda = 3900 \), shown in Figure 3.10e and 3.10f. 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 [79, 81].

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 Figure 3.11a, 3.11b and 3.11c. Again we obtain one, two, and three-stripe patterns on successive shrinking domains with \( \lambda = 500, 1900, 3900 \) respectively, shown in 3.11d, 3.11e and 3.11f. 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 $\lambda$, and only vary the shape of the domain, we observe the transition of three-stripe patterns to two-stripe patterns, shown in Figure 3.12, and the transition of stripe patterns to spot patterns, shown in Figure 3.13. In the numerical experiments, Figure 3.12, the value of $\lambda$ is fixed to be 3900, and in Figure 3.13, the value of $\lambda$ is fixed to be 6900. Configurations of nodular skeletal elements arranged like the activator spots in Figure 3.13, are also found in portions of vertebrate limbs, where they form the wrist and ankle bones that intervene between tiers of rod-like elements.
Figure 3.10. Numerical solution of the model 3.6 - 3.7 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).
Figure 3.11. Numerical solution of the model 3.6 - 3.7 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).
Figure 3.12. Numerical solution of the model 3.6 - 3.7 on domains with irregular shapes, with $\lambda = 3900$. (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).

85
Figure 3.13. Numerical solution of the model 3.6 - 3.7 on domains with irregular shapes, with $\lambda = 6900$. (a), (b), (c), (d): domains with different shapes and their meshes; (e), (f), (g), (h): Contour plots of the concentration of the activator $C_a$ at time $T = 1.0$ (close to the steady-state).
CHAPTER 4
NUMERICAL METHODS ON MOVING GRIDS

The system for limb development belongs to the class of reaction-diffusion systems of two chemical species. On a fixed domain, they can be written in the general form

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

where \(u \in \mathbb{R}^2\) represent concentrations of molecular species, \(D \in \mathbb{R}^{2 \times 2}\) is the diffusion constant matrix and it is diagonal, \(\nabla^2 u\) is the Laplacian associated with the diffusion of the molecules \(u\), and \(F(u)\) describes the biochemical reactions.

To model vertebrate limb development, we consider system on a moving domain. Let \(\Omega(t) = \{(x(t), y(t))\}\) be an open, bounded, and time-dependent domain on which the reaction-diffusion system is defined, where \((x(t), y(t))\) is any point in the domain. We triangulate \(\Omega(t)\) by \(\Omega_h(t)\) which consists of time-dependent nonoverlapping triangles \(\{\triangle_m(t)\}_{m=1}^{N}\). Let \(h_{\text{min}}(t) = \min_{1 \leq m \leq N} \rho_m(t)\), where \(\rho_m(t)\) is the diameter of the inscribed circle of the triangle \(\triangle_m(t)\). \(\{(x_i(t), y_i(t))\}_{i=1}^{M}\) denote the grid points of \(\Omega_h(t)\). All spatial variables are functions of temporal variable.
4.1 Reaction-diffusion system on a moving domain

On a moving domain, by the Reynolds transport theorem [54], system 4.1 can be extended to

\[
\frac{Du}{Dt} + u \nabla \cdot \vec{a} = D \nabla^2 u + F(u),
\]

where \( \frac{Du}{Dt} \) is the material derivative of chemical species \( u \), i.e.,

\[
\frac{Du}{Dt} = \frac{\partial u}{\partial t} + \vec{a} \cdot \nabla u,
\]

and

\[
\vec{a}(x(t),y(t),t) = \left( \frac{dx(t)}{dt}, \frac{dy(t)}{dt} \right)^T
\]

is the velocity of the spatial point \((x(t),y(t))\) in the moving domain.

4.2 The DG spatial discretization on a moving domain

Define the time-dependent finite element space

\[
V^k_h(t) = \{ v : v|_{\Delta_m(t)} \in P^k(\Delta_m(t)), m = 1, \cdots, N \},
\]

where \( P^k(\Delta_m(t)) \) denotes the set of all polynomials of degree at most \( k \) on \( \Delta_m(t) \).

As in Section 2.1.2 we formally apply the DG formulation [16] to discretize the reaction-diffusion equations 4.2 in the spatial dimensions, but keep the time variable continuous. The difference from Section 2.1.2 is that now the finite element space is time-dependent since we are solving the problem on a moving domain. We characterize the semi-discrete scheme as:
find \( u \in V^k_h(t) \), such that

\[
\int_{\Delta_m(t)} \frac{D}{Dt} u dx + \int_{\Delta_m(t)} u \nabla \cdot \tilde{a} v dx - D \int_{\Delta_m(t)} u \nabla^2 v dx \\
+ D \int_{\partial \Delta_m(t)} \tilde{u} \nabla v \cdot \tilde{n}_{\partial \Delta_m(t)} dS - D \int_{\partial \Delta_m(t)} \tilde{v} \nabla \tilde{u} \cdot \tilde{n}_{\partial \Delta_m(t)} dS \\
= \int_{\Delta_m(t)} F(u) v dx
\]

(4.6)

holds true for any \( v \in V^k_h(t) \) and \( m = 1, \ldots, N \). The numerical fluxes on the element edges \( \partial \Delta_m(t) \) are chosen as

\[
\hat{u} = \frac{u^{in} + u^{ext}}{2}, \tag{4.7}
\]

\[
\tilde{\nabla} u = \frac{(\nabla u)^{in} + (\nabla u)^{ext}}{2} + \beta [u], \tag{4.8}
\]

where the jump term

\[
[u] = (u^{ext} - u^{in}) \big|_{\partial \Delta_m(t)} \cdot \tilde{n}_{\partial \Delta_m(t)},
\]

(4.9)

\( u^{in} \) and \( u^{ext} \) are the limits of \( u \) at \( x \in \partial \Delta_m(t) \) taken from the interior and the exterior of \( \Delta_m(t) \) respectively, \( \tilde{n}_{\partial \Delta_m(t)} \) is the outward unit normal to the element \( \Delta_m(t) \) at \( x \in \partial \Delta_m(t) \), and \( \beta \) is a positive quantity that is of the order \( O(h_{\min}^{-1}(t)) \).

Following [16], we take \( \beta = 10/h_{\min}(t) \). The choice of numerical fluxes (4.7 - 4.9) is crucial for the stability and convergence of the DG scheme (4.6).

Following Section 2.1.2, we use the Strang type second-order symmetrical operator splitting schemes [47, 97] to avoid solving the completely coupled nonlinear system from the fully implicit temporal discretization and overcome the computational challenge from the stiffness of reaction-diffusion equations (4.2) and the DG
spatial discretization operator. We consider the $P^1$ case in this paper such that the order of accuracy in the spatial direction corresponds to the splitting error order in the temporal direction, and they are both 2.

Next we describe the detailed numerical formulae for the scalar case of 4.6. The corresponding system can be solved component by component using similar formulae. For each element $\triangle_m(t)$, denote its three neighboring elements by $i_m$, $j_m$, and $k_m$. To simplify notations in the following presentation, we will still omit the subscript $m$ and just use $i$, $j$, $k$ to represent the neighboring cells of $\triangle_m(t)$. Since limb development is accompanied by moderate growth, the apical zone of the limb bud does not deform rapidly. Hence in this computational model the mesh movement is controlled such that there is no degenerate element formed during movement of the domain. Therefore the neighboring elements of each element do not merge and the indexes $i, j, k$ for neighboring elements are time-independent.

The linear polynomial on $\triangle_m(t)$ is represented by

$$u(x, y, t) = a_m(t) + b_m(t)\xi_m(x(t), y(t), t) + c_m(t)\eta_m(x(t), y(t), t), \quad (4.10)$$

where $\xi_m$ and $\eta_m$ are time-dependent local basis functions on $\triangle_m(t)$

$$\xi_m(x(t), y(t), t) = \frac{x(t) - x_m(t)}{h_m(t)}, \quad (4.11)$$

$$\eta_m(x(t), y(t), t) = \frac{y(t) - y_m(t)}{h_m(t)}, \quad (4.12)$$

and $(x_m(t), y_m(t))$ is the barycenter of the element $\triangle_m(t)$, $h_m(t) = \sqrt{||\Delta_m(t)||}$ with $||\Delta_m(t)||$ denoting the area of $\triangle_m(t)$. The movement of $\triangle_m(t)$ and the whole mesh $\Omega_h(t)$ are pre-determined by the development of the LALI zone of limb bud.

By taking $v = 1, \xi_m, \eta_m$ on $\triangle_m(t)$ and $v = 0$ elsewhere, the DG formulation
4.6 is converted from the integral form to the following system, for \( m = 1, \cdots, N \):

\[
\begin{align*}
\text{for } m = 1, \cdots, N: \\
p_{11}(t)a'_m(t) + p_{12}(t)b'_m(t) + p_{13}(t)c'_m(t) + \\
q_{11}(t)a_m(t) + q_{12}(t)b_m(t) + q_{13}(t)c_m(t) + \\
k_{11}(t)a_m(t) + k_{12}(t)b_m(t) + k_{13}(t)c_m(t) = \\
D\{w_{a11}(t)a(t) + w_{b11}(t)b(t) + w_{c11}(t)c(t) + \\
\sum_{l=i,j,k} [w_{al1}(t)a_l(t) + w_{bl1}(t)b_l(t) + w_{cl1}(t)c_l(t)]\} + \\
\frac{(p_{11}(t)/3)}{l=i,j,k} \sum F(u(x_{m,l}(t), y_{m,l}(t))),
\end{align*}
\]

(4.13)

\[
\begin{align*}
\text{for } m = 1, \cdots, N: \\
p_{21}(t)a'_m(t) + p_{22}(t)b'_m(t) + p_{23}(t)c'_m(t) + \\
q_{21}(t)a_m(t) + q_{22}(t)b_m(t) + q_{23}(t)c_m(t) + \\
k_{21}(t)a_m(t) + k_{22}(t)b_m(t) + k_{23}(t)c_m(t) = \\
D\{w_{a21}(t)a(t) + w_{b21}(t)b(t) + w_{c21}(t)c(t) + \\
\sum_{l=i,j,k} [w_{al2}(t)a_l(t) + w_{bl2}(t)b_l(t) + w_{cl2}(t)c_l(t)]\} + \\
\frac{(p_{11}(t)/3)}{l=i,j,k} \sum F(u(x_{m,l}(t), y_{m,l}(t))),
\end{align*}
\]

(4.14)
\[
p_{31}(t)a_m(t) + p_{32}(t)b_m(t) + p_{33}(t)c_m(t) + q_{31}(t)a_m(t) + q_{32}(t)b_m(t) + q_{33}(t)c_m(t) +
k_{31}(t)a_m(t) + k_{32}(t)b_m(t) + k_{33}(t)c_m(t) =
\]
\[
D\{w_{am3}(t)a_m(t) + w_{bm3}(t)b_m(t) + w_{cm3}(t)c_m(t) +
\sum_{l=i,j,k} [w_{al3}(t)a_l(t) + w_{bl3}(t)b_l(t) + w_{cl3}(t)c_l(t)]\} +
\]
\[
(p_{11}(t)/3) \sum_{l=i,j,k} F(u(x_{m,l}(t), y_{m,l}(t))) \eta_m(x_{m,l}(t), y_{m,l}(t), t),
\] (4.15)

where the coefficients \{\{w_{alr}\}_r=1\}^3, \{w_{blr}\}_r=1, \{w_{clr}\}_r=1, \{p_{rs}\}_r,s=1, \{q_{rs}\}_r,s=1, \{k_{rs}\}_r,s=1 depend on the local geometry of the mesh (i.e., triangle \(\Delta_m(t)\) and its neighboring cells \(i, j, k\) and \(\vec{n}_{\partial\Delta_m(t)}\)), the value \(\beta\), and the local basis functions \(1, \{\xi_l(x, y, t), \eta_l(x, y, t)\}_{l=m,i,j,k}\). \(\{x_{m,l}(t), y_{m,l}(t)\}_{l=i,j,k}\) are the mid-points of the three edges \(\{e_l\}_{l=i,j,k}\) of \(\Delta_m(t)\) which serve as Gaussian quadrature points for the integral involving the nonlinear reaction terms in equation 4.6.

Rewrite equations 4.13 - 4.15 to the matrix-vector form

\[
P_m(t)\vec{V}_m'(t) + Q_m(t)\vec{V}_m(t) + K_m(t)\vec{V}_m(t) = D \sum_{l=m,i,j,k} W_l(t)\vec{V}_l(t) + \tilde{F}_m(\vec{V}_m(t)).
\] (4.16)

Denote \(\frac{dx}{dt}\) as \(x'\), then we have

\[
x_m(t) = \frac{x_i(t) + x_j(t) + x_k(t)}{3}, \quad x'_m = \frac{x'_i + x'_j + x'_k}{3},
\] (4.17)

\[
y_m(t) = \frac{y_i(t) + y_j(t) + y_k(t)}{3}, \quad y'_m = \frac{y'_i + y'_j + y'_k}{3}.
\] (4.18)

92
Let

$$\text{ss} = \det \begin{pmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ 1 & 1 & 1 \end{pmatrix},$$

(4.19)

We have

$$|\Delta_m(t)| = |\text{ss}|/2,$$

(4.20)

$$h_m = \sqrt{\frac{1}{2} |\text{ss}|},$$

(4.21)

$$h'_m = \frac{1}{2\sqrt{2}} \left( \frac{|\text{ss}|'}{h_m} \right) = \begin{cases} \frac{\text{ss}'}{4h_m} & \text{if } \text{ss} >= 0 \\ -\frac{\text{ss}'}{4h_m} & \text{if } \text{ss} < 0 \end{cases}$$

(4.22)

and

$$\beta = 10 \cdot \min_{m=1,\ldots,N} \frac{2|\text{ss}|}{|e_i| + |e_j| + |e_k|},$$

(4.23)

$$\xi'_m = \frac{(x' - x_m)h_m - (x - x_m)h'_m}{h_m^2},$$

(4.24)

$$\eta'_m = \frac{(y' - y_m)h_m - (y - y_m)h'_m}{h_m^2},$$

(4.25)

$$u'_m = a'_m + b'_m \xi_m + c'_m \eta_m + b_m \xi'_m + c_m \eta'_m.$$  

(4.26)

Then we have

$$\vec{V}_m(t) = \begin{pmatrix} a_m(t) \\ b_m(t) \\ c_m(t) \end{pmatrix}, \quad \vec{V}_l(t) = \begin{pmatrix} a_l(t) \\ b_l(t) \\ c_l(t) \end{pmatrix},$$

(4.27)
\[ P_m(t) = \begin{pmatrix} p_{11}(t) & p_{12}(t) & p_{13}(t) \\ p_{21}(t) & p_{22}(t) & p_{23}(t) \\ p_{31}(t) & p_{32}(t) & p_{33}(t) \end{pmatrix} \]

\[
\begin{pmatrix}
\int_{\Delta_m(t)} \xi_m(t) \, dx & \int_{\Delta_m(t)} \xi_m(t) \eta_m(t) \, dx \\
\int_{\Delta_m(t)} \eta_m(t) \, dx & \int_{\Delta_m(t)} \eta_m(t) \eta_m(t) \, dx
\end{pmatrix}
\]

\[ Q_m(t) = \begin{pmatrix} q_{11}(t) & q_{12}(t) & q_{13}(t) \\ q_{21}(t) & q_{22}(t) & q_{23}(t) \\ q_{31}(t) & q_{32}(t) & q_{33}(t) \end{pmatrix} \]

\[
\begin{pmatrix}
0 & \int_{\Delta_m(t)} \xi_m'(t) \, dx & \int_{\Delta_m(t)} \eta_m'(t) \, dx \\
0 & \int_{\Delta_m(t)} \xi_m(t) \xi_m'(t) \, dx & \int_{\Delta_m(t)} \xi_m(t) \eta_m'(t) \, dx \\
0 & \int_{\Delta_m(t)} \eta_m(t) \xi_m'(t) \, dx & \int_{\Delta_m(t)} \eta_m(t) \eta_m'(t) \, dx
\end{pmatrix}
\]

\[ K_m(t) = \begin{pmatrix} k_{11}(t) & k_{12}(t) & k_{13}(t) \\ k_{21}(t) & k_{22}(t) & k_{23}(t) \\ k_{31}(t) & k_{32}(t) & k_{33}(t) \end{pmatrix} \]

\[
\begin{pmatrix}
\int_{\Delta_m(t)} \nabla \cdot \vec{a} \, dx & \int_{\Delta_m(t)} \xi_m(t) \nabla \cdot \vec{a} \, dx & \int_{\Delta_m(t)} \eta_m(t) \nabla \cdot \vec{a} \, dx \\
\int_{\Delta_m(t)} \xi_m(t) \nabla \cdot \vec{a} \, dx & \int_{\Delta_m(t)} \xi_m(t)^2 \nabla \cdot \vec{a} \, dx & \int_{\Delta_m(t)} \xi_m(t) \eta_m(t) \nabla \cdot \vec{a} \, dx \\
\int_{\Delta_m(t)} \eta_m(t) \nabla \cdot \vec{a} \, dx & \int_{\Delta_m(t)} \eta_m(t) \eta_m(t) \nabla \cdot \vec{a} \, dx & \int_{\Delta_m(t)} \eta_m(t)^2 \nabla \cdot \vec{a} \, dx
\end{pmatrix}
\]
\[
W_m(t) = \begin{pmatrix}
    w_{am1}(t) & w_{bm1}(t) & w_{cm1}(t) \\
    w_{am2}(t) & w_{bm2}(t) & w_{cm2}(t) \\
    w_{am3}(t) & w_{bm3}(t) & w_{cm3}(t)
\end{pmatrix}
\]

\[
= \sum_{l=i,j,k} \begin{pmatrix}
    0 & \frac{r_{11}(t)n_{i,x}(t)}{2h_m(t)} & \frac{r_{11}(t)n_{i,y}(t)}{2h_m(t)} \\
    -\frac{r_{11}(t)n_{i,x}(t)}{2h_m(t)} & 0 & \frac{r_{21m}(t)n_{i,y}(t)-r_{31m}(t)n_{i,x}(t)}{2h_m(t)} \\
    -\frac{r_{11}(t)n_{i,y}(t)}{2h_m(t)} & \frac{r_{31m}(t)n_{i,x}(t)-r_{21m}(t)n_{i,y}(t)}{2h_m(t)} & 0
\end{pmatrix}
\]

\[
- \beta \sum_{l=i,j,k} \begin{pmatrix}
    r_{11}(t) & r_{21m}(t) & r_{31m}(t) \\
    r_{21m}(t) & s_{1mml}(t) & s_{2mml}(t) \\
    r_{31m}(t) & s_{2mml}(t) & s_{3mml}(t)
\end{pmatrix} ,
\]

(4.31)

\[
W_i(t) = \begin{pmatrix}
    w_{ai1}(t) & w_{bi1}(t) & w_{ci1}(t) \\
    w_{ai2}(t) & w_{bi2}(t) & w_{ci2}(t) \\
    w_{ai3}(t) & w_{bi3}(t) & w_{ci3}(t)
\end{pmatrix}
\]

\[
= \begin{pmatrix}
    0 & \frac{r_{11}(t)n_{i,x}(t)}{2h_l(t)} & \frac{r_{11}(t)n_{i,y}(t)}{2h_l(t)} \\
    -\frac{r_{11}(t)n_{i,x}(t)}{2h_m(t)} & \frac{r_{21m}(t)n_{i,x}(t)}{2h_m(t)} & \frac{r_{21m}(t)n_{i,y}(t)-r_{31m}(t)n_{i,x}(t)}{2h_m(t)} \\
    -\frac{r_{11}(t)n_{i,y}(t)}{2h_m(t)} & \frac{r_{31m}(t)n_{i,x}(t)-r_{21m}(t)n_{i,y}(t)}{2h_m(t)} & \frac{r_{31m}(t)n_{i,y}(t)}{2h_m(t)}
\end{pmatrix}
\]

\[
+ \beta \begin{pmatrix}
    r_{11}(t) & r_{21}(t) & r_{31}(t) \\
    r_{21m}(t) & s_{1m}(t) & s_{2m}(t) \\
    r_{31m}(t) & s_{2m}(t) & s_{3m}(t)
\end{pmatrix} ,
\]

(4.32)
\[ F_m(\mathbf{V}_m(t)) = p_{11}(t)/3 \left( \begin{array}{c} \sum_{l=i,j,k} F(u(x_{m,l}(t), y_{m,l}(t))) \\ \sum_{l=i,j,k} F(u(x_{m,l}(t), y_{m,l}(t))) \xi_m(x_{m,l}(t), y_{m,l}(t), t) \\ \sum_{l=i,j,k} F(u(x_{m,l}(t), y_{m,l}(t))) \eta_m(x_{m,l}(t), y_{m,l}(t), t) \end{array} \right) \] (4.33)

and

\[ r_{1l}(t) = \int_{e_l(t)} \xi_l(t) dS, \quad r_{2l}(t) = \int_{e_l(t)} \xi_l(t) dS, \quad r_{3l}(t) = \int_{e_l(t)} \eta_l(t) dS, \]
\[ r_{2lm}(t) = \int_{e_l(t)} \xi_m(t) dS, \quad r_{3lm}(t) = \int_{e_l(t)} \eta_m(t) dS, \]
\[ s_{1lm}(t) = \int_{e_l(t)} \xi_m(t) \xi_l(t) dS, \quad s_{2lm}(t) = \int_{e_l(t)} \xi_m(t) \eta_l(t) dS, \]
\[ s_{3lm}(t) = \int_{e_l(t)} \xi_l(t) \eta_m(t) dS, \quad s_{4lm}(t) = \int_{e_l(t)} \eta_m(t) \eta_l(t) dS. \]
\[ s_{1mml}(t) = \int_{e_l(t)} \xi_m(t) \eta_m(t) dS, \quad s_{2mml}(t) = \int_{e_l(t)} \eta_m(t) \eta_l(t) dS. \] (4.34)

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

\[ \mathbf{V}_m'(t) = P_m(t)^{-1} \left[ (D W_m(t) - Q_m(t) - K_m(t)) \mathbf{V}_m(t) + D \sum_{l=i,j,k} W_l(t) \mathbf{V}_l(t) \right] \]
\[ + P_m(t)^{-1} \mathbf{F}_m(\mathbf{V}_m(t)). \] (4.35)

4.3 Temporal discretization

The ODE (4.35) has a linear term resulting from the diffusion and domain movement and a nonlinear term coming from the reaction term in (4.2). Both of these terms can cause stiffness in the reaction-diffusion system and present
challenges for temporal discretization schemes. Hence we need to use fully implicit schemes to solve \( \text{4.35} \). In order to avoid solving a large coupled nonlinear system of equations at every time step, we adopted the trapezoidal operator splitting (OS) scheme \([47]\), which belongs to the class of Strang type second-order symmetrical operator splitting schemes \([97]\), to split the linear terms from the nonlinear terms of \( \text{4.35} \). The large nonlinear problem is decoupled, and hence we can solve the linear part and the nonlinear part individually by implicit temporal schemes. The resulting nonlinear problems are local for each element and they can be solved efficiently by an iterative method such as Newton’s method.

We denote the numerical solution of the ODE system \( \text{4.35} \) at \( t = t^n \) by \( \vec{V}_m^n \).

To evolve the system from step \( t^n \) to \( t^{n+1} \), we apply the trapezoidal OS scheme for \( \text{4.35} \):

**Step 1** – apply forward Euler method for the linear term at \( [t^n, t^{n+\frac{1}{2}}] \),

\[
\vec{v}_{0,m} = \vec{V}_m^n, \quad m = 1, \cdots, N, \\
\vec{v}_{1,m} = \vec{v}_{0,m} + \frac{1}{2} \Delta t P_m(t^n)^{-1} \left[ [D W_m(t^n) - Q_m(t^n) - K_m(t^n)] \vec{v}_{0,m} + D \sum_{l=i,j,k} W_l(t^n) \vec{v}_{0,l} \right].
\]  
(4.36)

**Step 2** – apply Crank-Nicholson method for the nonlinear term at \( [t^n, t^{n+1}] \), with \( \vec{v}_{1,m} \) as input data:

\[
\vec{v}_{2,m} = \vec{v}_{1,m} + \frac{1}{2} \Delta t [P_m(t^n)^{-1} \bar{F}_m(\vec{v}_{1,m}) + P_m(t^{n+1})^{-1} \bar{F}_m(\vec{v}_{2,m})].
\]  
(4.37)

The local nonlinear system \( \text{4.37} \) on the element \( m \) is solved by Newton iterations, with the initial guess \( \vec{v}_{1,m} \), for \( m = 1, \cdots, N \).

**Step 3** – apply backward Euler method for the linear term at \( [t^{n+\frac{1}{2}}, t^{n+1}] \), with
\( \vec{v}_{2,m} \) as input data,

\[
\vec{v}_{3,m} = \vec{v}_{2,m} + \frac{1}{2} \Delta t P_m(t^{n+1})^{-1}.
\]

\[
\left[ [DW_m(t^{n+1}) - Q_m(t^{n+1}) - K_m(t^{n+1})]\vec{v}_{3,m} + D \sum_{l=i,j,k} W_l(t^{n+1})\vec{v}_{3,l} \right],
\]

\( \vec{v}_{m+1}^{n+1} = \vec{v}_{3,m}, \quad m = 1, \ldots, N. \) \hspace{1cm} (4.38)

The sparse linear system 4.38 is solved by the solver “lin_sol_gen_coordinate” of the IMSL package.
4.4 Numerical Tests

4.4.1 One dimension

**Example 4.1** Consider the one-dimensional linear problem

\[
\begin{align*}
    u_t &= u_{xx} - u + \pi^2 e^{-t} \cos(\pi x) - e^{-t} \pi \sin(\pi x) x' + \sigma_x e^{-t} \cos(\pi x), \quad 0 < x < 1; \\
    x' &= \sigma_x x, \\
    u(x, 0) &= \cos(\pi x),
\end{align*}
\]

(4.39)

with the no flux boundary conditions. The initial domain is \((0, 1)\). For any \(x\), \(x' = \sigma_x x\). Thus, at \(T = 1\), the domain changes to \((0, e^{\sigma x})\). In this problem, we take \(\sigma_x = 0.5\). The exact solution is \(u(x, t) = e^{-t} \cos(\pi x)\) where \(x\) is a function of the time variable \(t\). The simulation is carried up to \(T = 1.0\). We perform the numerical convergence analysis on successively refined meshes. The \(L^1\), \(L^2\) and \(L^\infty\) errors, order of accuracy and CPU times are measured and listed in Table 4.1. The time step size is taken to be \(\Delta t = \Delta x\) where \(\Delta x = 1/N\) and \(N\) is the number of spatial elements. The second order accuracy in Table 4.1 has the expected values.
TABLE 4.1

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 4.1

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.01</td>
<td>5.84E-03</td>
<td>-</td>
<td>5.70E-03</td>
<td>-</td>
<td>7.58E-03</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.02</td>
<td>1.55E-03</td>
<td>1.91</td>
<td>1.50E-03</td>
<td>1.93</td>
<td>1.95E-03</td>
<td>1.96</td>
</tr>
<tr>
<td>40</td>
<td>0.04</td>
<td>3.95E-04</td>
<td>1.97</td>
<td>3.81E-04</td>
<td>1.98</td>
<td>4.90E-04</td>
<td>1.99</td>
</tr>
<tr>
<td>80</td>
<td>0.19</td>
<td>9.95E-05</td>
<td>1.99</td>
<td>9.59E-05</td>
<td>1.99</td>
<td>1.23E-04</td>
<td>2.00</td>
</tr>
<tr>
<td>160</td>
<td>0.85</td>
<td>2.50E-05</td>
<td>1.99</td>
<td>2.40E-05</td>
<td>2.00</td>
<td>3.06E-05</td>
<td>2.00</td>
</tr>
<tr>
<td>320</td>
<td>3.72</td>
<td>6.25E-06</td>
<td>2.00</td>
<td>6.01E-06</td>
<td>2.00</td>
<td>7.66E-06</td>
<td>2.00</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$. 

100
Example 4.2 Consider the one-dimensional nonlinear problem

\[
\begin{aligned}
  u_t &= u_{xx} - u^2 + e^{-2t} \cos^2(\pi x) + (\pi^2 - 1)e^{-t} \cos(\pi x) - e^{-t} \pi \sin(\pi x)x' \\
  &\quad + \sigma_x e^{-t} \cos(\pi x), \quad 0 < x < 1; \\
  x' &= \sigma x, \\
  u(x, 0) &= \cos(\pi x),
\end{aligned}
\]  

(4.40)

with the no flux boundary conditions. The initial domain is (0,1). For any \( x \), \( x' = \sigma x \). Thus, at \( T = 1 \), the domain changes to \((0, e^{\sigma x})\). In this problem, we take \( \sigma_x = 0.5 \). The exact solution is \( u(x, t) = e^{-t} \cos(\pi x) \) where \( x \) is a function of the time variable \( t \). The simulation is carried up to \( T = 1.0 \). We perform the numerical convergence analysis on successively refined meshes. The \( L^1 \), \( L^2 \) and \( L^\infty \) errors, order of accuracy and CPU times are measured and listed in Table 4.2.

The time step size is taken to be \( \Delta t = \Delta x \) where \( \Delta x = 1/N \) and \( N \) is the number of spatial elements. The second order accuracy in Table 4.2 has the expected values.
TABLE 4.2

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR
EXAMPLE 4.2

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.01</td>
<td>1.42E-02</td>
<td>-</td>
<td>1.41E-02</td>
<td>-</td>
<td>2.07E-02</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>0.02</td>
<td>4.79E-04</td>
<td>4.89</td>
<td>4.28E-04</td>
<td>5.04</td>
<td>5.53E-04</td>
<td>5.23</td>
</tr>
<tr>
<td>40</td>
<td>0.07</td>
<td>2.28E-04</td>
<td>1.07</td>
<td>2.16E-04</td>
<td>0.99</td>
<td>3.22E-04</td>
<td>0.78</td>
</tr>
<tr>
<td>80</td>
<td>0.25</td>
<td>7.21E-05</td>
<td>1.66</td>
<td>6.81E-05</td>
<td>1.66</td>
<td>9.44E-05</td>
<td>1.77</td>
</tr>
<tr>
<td>160</td>
<td>1.06</td>
<td>1.91E-05</td>
<td>1.92</td>
<td>1.79E-05</td>
<td>1.93</td>
<td>2.41E-05</td>
<td>1.97</td>
</tr>
<tr>
<td>320</td>
<td>4.67</td>
<td>4.85E-06</td>
<td>1.98</td>
<td>4.54E-06</td>
<td>1.98</td>
<td>6.01E-06</td>
<td>2.00</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$. 
4.4.2 Two dimensions

Example 4.3 Consider the two-dimensional linear problem

\[
\begin{aligned}
\frac{D u}{D t} + u \nabla \cdot \vec{a} &= \nabla^2 u - u + 2\pi^2 e^{-t} \cos(\pi x) \cos(\pi y) + (\sigma_x + \sigma_y) e^{-t} \cos(\pi x) \cos(\pi y) \\
- e^{-t} \pi \sin(\pi x) \cos(\pi y) x' - e^{-t} \pi \cos(\pi x) \sin(\pi y) y', & \quad (x, y) \in (0, 1) \times (0, 1); \\
\vec{a} &= (x', y')^T = (\sigma_x x, \sigma_y y)^T, \\
u(x, y, 0) &= \cos(\pi x) \cos(\pi y),
\end{aligned}
\]

(4.41)

with no flux boundary conditions. The initial domain is \((0, 1) \times (0, 1)\). For any \((x, y), x' = \sigma_x x\) and \(y' = \sigma_y y\). Thus, at \(T = 1\), the domain changes to \((0, e^{\sigma_x}) \times (0, e^{\sigma_y})\). In this problem, we take \(\sigma_x = \sigma_y = 0.5\). The exact solution is \(u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y)\) where \(x\) and \(y\) are both functions of the time variable \(t\). The simulation is carried up to \(T = 1.0\). We perform the numerical convergence analysis on successively refined meshes. The coarsest mesh is shown in Figure 2.20. The refinement of the meshes is done in a uniform way, namely by cutting each triangle into four smaller similar ones. The \(L^1\), \(L^2\) and \(L^\infty\) errors, order of accuracy and CPU times are measured and listed in Table 4.3. The time step size is taken to be \(\Delta t = h_{\text{min}}(0)\). The second order accuracy in Table 4.3 has the expected values.
TABLE 4.3

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 4.3

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>0.08</td>
<td>5.05E-02</td>
<td>-</td>
<td>4.09E-02</td>
<td>-</td>
<td>8.25E-02</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>0.39</td>
<td>1.39E-02</td>
<td>1.86</td>
<td>1.13E-02</td>
<td>1.86</td>
<td>2.21E-02</td>
<td>1.90</td>
</tr>
<tr>
<td>704</td>
<td>7.81</td>
<td>3.62E-03</td>
<td>1.95</td>
<td>2.91E-03</td>
<td>1.95</td>
<td>5.80E-03</td>
<td>1.93</td>
</tr>
<tr>
<td>2816</td>
<td>450.79</td>
<td>9.19E-04</td>
<td>1.98</td>
<td>7.36E-04</td>
<td>1.98</td>
<td>1.50E-03</td>
<td>1.95</td>
</tr>
<tr>
<td>11264</td>
<td>33796</td>
<td>2.32E-04</td>
<td>1.99</td>
<td>1.85E-04</td>
<td>1.99</td>
<td>3.82E-04</td>
<td>1.97</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$.

**Example 4.4** Consider the two-dimensional nonlinear problem

\[
\begin{align*}
\frac{Du}{Dt} + u \nabla \cdot \vec{a} &= \nabla^2 u - u^2 + e^{-2t} \cos(\pi x)^2 \cos(\pi y)^2 + (2\pi^2 - 1)e^{-t} \cos(\pi x) \cos(\pi y) \\
&\quad - e^{-t} \pi \sin(\pi x) \cos(\pi y) x' - e^{-t} \pi \cos(\pi x) \sin(\pi y) y' + (\sigma_x + \sigma_y) e^{-t} \cos(\pi x) \cos(\pi y), \\
\vec{a} &= (x', y')^T = (\sigma_x x, \sigma_y y)^T, \\
u(x, y, 0) &= \cos(\pi x) \cos(\pi y),
\end{align*}
\]

(4.42)

with no flux boundary conditions. The initial domain is $(0, 1) \times (0, 1)$. For any $(x, y)$, $x' = \sigma_x x$ and $y' = \sigma_y y$. Thus, at $T = 1$, the domain changes to $(0, e^{\sigma_x}) \times (0, e^{\sigma_y})$. In this problem, we take $\sigma_x = \sigma_y = 0.5$. The exact solution is $u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y)$ where $x$ and $y$ are both functions of the time variable $t$. The
TABLE 4.4

CPU TIME, ERROR, AND ORDER OF ACCURACY FOR EXAMPLE 4.4

<table>
<thead>
<tr>
<th># of Cells</th>
<th>CPU(s)</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^2$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>0.47</td>
<td>5.13E-02</td>
<td>-</td>
<td>4.15E-02</td>
<td>-</td>
<td>7.99E-02</td>
<td>-</td>
</tr>
<tr>
<td>176</td>
<td>4.4</td>
<td>1.39E-02</td>
<td>1.88</td>
<td>1.13E-02</td>
<td>1.87</td>
<td>2.26E-02</td>
<td>1.82</td>
</tr>
<tr>
<td>704</td>
<td>68.45</td>
<td>3.57E-03</td>
<td>1.96</td>
<td>2.91E-03</td>
<td>1.96</td>
<td>6.02E-03</td>
<td>1.90</td>
</tr>
<tr>
<td>2816</td>
<td>1957</td>
<td>9.04E-04</td>
<td>1.98</td>
<td>7.36E-04</td>
<td>1.98</td>
<td>1.55E-03</td>
<td>1.96</td>
</tr>
<tr>
<td>11264</td>
<td>208170</td>
<td>2.28E-04</td>
<td>1.99</td>
<td>1.85E-04</td>
<td>1.99</td>
<td>3.92E-04</td>
<td>1.99</td>
</tr>
</tbody>
</table>

Final time $T = 1.0$.

Simulation is carried up to $T = 1.0$. We perform the numerical convergence analysis on successively refined meshes. The coarsest mesh is shown in Figure 2.20. The refinement of the meshes is done in a uniform way, namely by cutting each triangle into four smaller similar ones. The $L^1$, $L^2$ and $L^\infty$ errors, order of accuracy and CPU times are measured and listed in Table 4.4. The time step size is taken to be $\Delta t = 0.1 h_{\text{min}}(0)$. The second order accuracy in Table 4.4 has the expected values.
5.1 Cellular and molecular dynamics of limb formation

The limbs of vertebrate animals emerge from the embryonic flank as buds of the mesenchymal cells covered by an epithelial layer, the ectoderm, flattening into paddle shapes as they grow. The most thoroughly studied aspect of limb development is the formation of the skeleton, an array of jointed bone or cartilage elements having a typical pattern that has sustained only moderate alterations over the course of evolution [29, 81]. The mechanism of limb skeletal pattern formation is incompletely understood. There is broad agreement, however, concerning the cellular and molecular-genetic interactions underlying the differentiation of the cartilage tissue that forms the primordia of the bony skeleton of tetrapod limbs and the endoskeleton of fish fins during embryogenesis [79, 116].

A population of multipotent skeletogenic progenitor cells [77, 78, 84], is maintained in a differentiated state [53] in the distal-most 0.3 mm of the avian and mammalian limb bud by the action of morphogens of the fibroblast growth factor family (FGFs) [66]. These are secreted by a narrow A-P-oriented ectodermal thickening, the apical ectodermal ridge (AER), along the limb bud tip [50]. While the mesenchymal cells can give rise to cartilage, connective tissue, and bone, and can undergo cell death, the limb musculature is formed by a separate population
of cells [84]. The non-myogenic limb mesenchyme (termed precartilage in what follows), forms the skeletal pattern by being caused to choose among its potential fates of cartilage, soft connective tissue fibroblasts and apoptotic cells, in a spatiotemporally regulated fashion.

In most amniote species, emergence of the cartilage cell of the limb skeleton occurs first in proximal regions of the limb bud and then in increasingly more distal regions [50] (Figures 5.3C, 5.4). This process is dependent on an intact AER, the removal of which leads to truncations of the skeleton with progressively more distal elements forming the longer the AER remains in place [50] (Figure 5.12a). The growth of the skeletal structures (and portions thereof), that are spared after AER removal is unaffected, with these components attaining nearly their normal size and shapes, except for terminal deletions, as development proceeds [50]. The limbs of certain amphibians seem to represent partial exceptions to the proximal-to-distal rule, with some distal cartilage elements appearing before proximal ones [52].

Cartilage differentiation, or chondrogenesis, is preceded by condensation of the precartilage mesenchyme, in which cell density increases and the cells enter into broad, transient contact with one another [102]. The precartilage cells are embedded in a dilute extracellular matrix (ECM) and condensation is accompanied by and dependent on local accumulation of the ECM molecule fibronectin [35], with markers of prospective condensation appearing earlier than ECM and morphological changes [72]. Molecules secreted by the dorsal and ventral ectoderm, including FGFs and Wnt, inhibit chondrogenesis [72, 100, 116], thereby confining the developing one-bar proximal cartilage primordium (stylopod, i.e., humerus, femur) to a central planar sector of the paddle-shaped limb bud. As development proceeds, the skeleton remains confined to this plane but expands laterally in more distal
regions as the stylopod gives way to the two-bar (zeugopod, i.e., radius and ulna, tibia and fibula) and multiple-bar (autopod, i.e., digits) primordia of the mid and terminal regions of the limb (Figure 5.3C). This brings the developing skeletal elements increasingly closer to the anterior and posterior edges of the limb bud, reflecting attenuation of the peripheral inhibitory effect. Attenuation of inhibition can also be seen in the proximity of the more distal elements to the dorsal and ventral surfaces as the limb bud tapers towards its tip, and to the apical boundary, as the potency of the AER wanes [51].

The developing limb bud contains all the components of a skeletogenic "local autoactivation-lateral inhibition (LALI) mechanism [70, 80, 82], and this Turing-type biological framework is schematized by Meinhardt and Gierer [67]. LALI system is self-organizing: that is, with appropriate choices of reaction (i.e., activator-inhibitor) and diffusion parameters, in domains of appropriate size, shape and boundary conditions, a spatially uniform distribution of the diffusible morphogens becomes dynamically unstable, giving way to nonuniform distributions, typically periodic arrangements of stripes and spots in two dimensions, and their three-dimensional analogs (bars, nodules) [2, 34].

Specifically, precartilage condensation and subsequent chondrogenesis are promoted by morphogens of the TGF-β superfamily [56, 68] that form an activator subnetwork (Figure 5.3B). In brief, one or more TGF-βs and activins set off a train of events whereby BMPs (one or more of BMP2, 4 and 7), acting via the receptor BMPR-Ia, induces regions of high BMP signaling activity, marked by phosphorylated Smads [73]. Signaling by the activator subnetwork induces extracellular matrix, matricellular and cell adhesion molecules, promoting mesenchymal condensation and subsequent chondrogenesis [85, 118] (Figure 5.3B).
The spatial profile of BMP action is dictated not by its receptor distribution, which is uniform, by a prepattern of the diffusible morphogen itself, which in the autopod takes the form of narrow crescents at the tips of the forming digits [73]. This pattern results form the spatiotemporal dynamics of the activator network morphogens interacting with inhibitory factors. The latter include the extracellular antagonistic BMP binding partners Noggin and Ventroptin (CHL2), the BMP receptor antagonist BAMBI and the antagonistic intracellular co-receptor Smad6, the expression of which are all induced by the activator subnetwork [17, 73]. Consistent with this, the prospective digits and interdigits are uncommitted as to their fates late into autopod formation [73], and this plasticity likely pertains in the fields of precartilage cells that form the stylopod and zeugopod at earlier stages. In particular, exogenous TGF-β can cause noncondensing limb bud pre-cartilage mesenchyme to condense and form ectopic cartilage in vitro [56] and in vivo [17].

The FGF pathway is involved in restricting the expansion of precartilage condensations as they form [72], and the Notch pathway plays a similar role [36]. Together with the inhibitory factors described above, these perinodular chondrogenesis restricting components constitute what we refer to as the inhibitor subnetwork (I) (Figure 5.3B).

The activator (A) and inhibitor (I) subnetworks together constitute a core skeletal patterning network (Figure 5.3B). Subnetwork A functions in a paracrine and (via its TGF-β component [68]), a positively autoregulatory fashion. It induces the production of key cell-cell transmissible components of subnetwork I, which in turn limit As radius of action (Figure 5.3B). These components and properties constitute the basic elements of a LALI framework [67, 104].
These and possibly additional activator-inhibitor interactions cause the limb bud mesenchyme to behave as a self-organizing system even outside its normal biological context. Dissociated and reaggregated limb mesoderm cells, packed into an ectodermal hull, form limb-like arrays of rod-like and nodular skeletal elements [90]. In high-density culture, moreover, these cells form spots and stripes of cartilage with similar quantitative characteristics to the in vivo patterns [18]. When the physical properties of the culture microenvironment [69], or putative LALI components are manipulated in vitro and vivo, the pattern behaves in ways predictable from LALI models [56, 68, 72, 73].

5.2 Computation methods

Simulations of simple LALI systems under geometric and growth constraints partially approximating those of developing limbs show that pattern formation of normal and mutant (e.g., polydactyly, human Apert syndrome, mouse *Doublefoot*) limbs is consistent with such mechanisms [15, 70]. These simulations, however, have variously employed highly schematic and ad hoc molecular and cellular representations, and stationary approximations of inherently temporal dynamics [70, 82].

The model of Hentschel and coworkers [45] contains eight coupled partial differential equations that model interactions among all of the following: reaction-diffusion dynamics of a network based on a TGF-β related activator and a FGF/FGF receptor 2-induced inhibitor, the diffusion of FGF from fixed and moving sources, TGF-β induced production of fibronectin, and the short-range movement of cells and their differentiation from uncondensed mesenchyme to condensed mesenchyme and then cartilage. While it would be desirable to follow the time evolution of this
system in an initially unpatterned 3D domain that mimics the growing limb bud, the complexity of this prospective simulation exceeds the capacity of available methods.

We have simplified this problem using mathematical arguments and biologically motivated assumptions [4], extracting an activator-inhibitor subnetwork from the elaborate system described in [15]:

\[
\frac{\partial C_a}{\partial t} = D_a \nabla^2 C_a + U(C_a) - k_a C_a C_i; \quad (5.1)
\]
\[
\frac{\partial C_i}{\partial t} = D_i \nabla^2 C_i + V(C_a) - k_a C_a C_i, \quad (5.2)
\]

We apply Section 4.1 to simulate the time-dependent behavior of such reaction-diffusion networks in regions of arbitrary shape. Here we describe pattern development in a growing limb-bud-like domain with no imposed structure or pattern other than a simple P-D gradient representing the effect of the AER [53] (Figure 5.3C). Our analysis demonstrates that major features of normal, experimentally manipulated, genetically aberrant and evolutionary transitional limb development emerge from the inherent self-organizing properties of the core skeletal patterning mechanism. The roles of factors important in establishing asymmetries in the the limb axes and differences among the skeletal elements, such as the morphogen Sonic hedgehog and the Hox gene families [79], have natural interpretations in this computational framework as regulators of limb bud shape and interaction parameters of the activator-inhibitor network.
5.3 Computation results

5.3.1 Normal development

To simulate the developing limb we allowed system 5.1-5.2 to operate in a geometric setting that represents the section of the growing limb bud equidistant from the dorsal and ventral surfaces (Figure 5.3C). The morphogens are constrained not to leak from the borders of the domain (no-flux boundary conditions) and the model limb is programmed to grow uniformly in the P-D direction.

There are three color scales in the simulation pictures (Figure 5.5): gray scale, representing formed skeletal elements of the frozen zone; white-violet scale, representing the active zone; and blue-green-red scale, representing the activator morphogen concentration in the apical zone. The LALI zone, in which the reaction-diffusion system is solved, consists of the active zone and apical zone.
At the very beginning the LALI zone comprises the entire limb bud. The computational mesh at $t = 0$ is shown in (Figure 5.1). Since the FGF gradient gets more and more shallow, the LALI zone shrinks in the simulation in the P-D direction ($x-$direction), and the LALI zone growing rate is modeled as

$$x'(t) = \sigma_x(x - vt) + v$$

with $\sigma_x = -0.2896$, and $v = 1$. It remains the same in the A-P direction ($y-$direction) in the simulation, i.e.,

$$y'(t) = 0.$$

FGF is at its highest concentration value at the AER, but this value decreases nonlinearly with time, described by curve A in Figure 5.2. It satisfies the following formula

$$C_{\text{tip}}(t) = 0.4 + 0.6e^{\sigma_x t}.$$  

The FGF concentration decreases with distance from the distal tip of the limb bud. When the value falls below a fixed value 0.4 arbitrary units, a portion of the LALI zone becomes the active zone (Figure 5.3C) at the boundary represented by curve B in Figure 5.2. Thus, the FGF concentration at the distal end of the active zone is always 0.4.

The FGF concentration declines further as it approaches the common proximal ends of the LALI and active zones, where the FGF concentration vanishes (curve C in Figure 5.2). The FGF concentration at this level is always 0.

The FGF concentration profiles across the entire LALI zone at time $T = 0$ and $T = 3.0$ are described by curves D and E in Figure 5.2. These concentration
profiles satisfy the following formula

\[ C_{FGF}(t, x) = \frac{4}{0.09} (e^{\sigma_s t} - 1)e^{-2\sigma_s t}x^2 - \frac{4}{3}(8e^{\sigma_s t} - 3)e^{-\sigma_s t}x + 0.4 + 0.6e^{\sigma_s t} \]  \quad (5.6)

where \( t \) is time and \( x \) is the distance from AER. When \( t \) is fixed, \( C_{FGF} \) depends non-linearly on \( x \). Thus, neither curve D nor E in Figure 5.2 are straight lines. Both the distance from level A to level B and from level A to level C in Figure 5.2 decrease with time, which means that the LALI zone is shrinking during limb development.

The frozen zone grows at the speed \( v = 1 \). At every 0.05 unit of time, we copy the concentration values on the computation grids of the left boundary of LALI zone to the new grid points in the frozen zone.

The width (\( x \)-direction) of the starting point of LALI (the very tip) and the
starting point of active zone is

\[ W_{Apical} = 0.09 \left( \frac{2}{3} \right)^{t/\tau} \]  

(5.7)

and the width of the active zone is

\[ W_{Active} = \frac{2}{3} W_{Apical}. \]  

(5.8)

The values of the key parameters

\[ \lambda = 1500, \quad \delta = 4.7, \quad t \in [0, 1.4]; \]

\[ \lambda = 5000, \quad \delta = 4.9, \quad t \in [1.4, 2.4]; \]

\[ \lambda = 16500, \quad \delta = 4.9, \quad t \in [2.4, 3]. \]  

(5.9)

The time step size is \( \Delta t = 2 \times 10^{-5} \) in the simulations. Parameter values are in Table 5.1. Take \( \lambda = 17000 \) after \( t = 2.4 \) in Table 5.1, we have the simulations in Figure 5.6, with 4 digits in the last stage instead of 3 in Figure 5.5.

Varying key parameters in system 5.1, 5.2 and geometric properties of the simulation template (Figure 5.3C) allowed us to explore their effects on the generated pattern and the robustness of the entire process. The reaction kinetic parameters \( \lambda \) and \( \delta \) and the A-P and P-D lengths of the LALI zone (Figure 5.8 - 5.11) are key determinants of the character of the patterns formed. This can be illustrated by the effect of changes in these dimensionless parameters in the context of the standard developmental sequence (Figure 5.5). During the first phase of simulated development (time \( t \) from 0 to 1.4) as the LALI zone shrinks in the P-D direction to 2/3 its original width, one stripe will form when \( \lambda \) and \( \delta \) takes smaller
### TABLE 5.1

PARAMETER VALUES

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>$t$</th>
<th>$\lambda$</th>
<th>$\delta$</th>
<th>$C_{tip}$</th>
<th>$W_{LALI}$</th>
<th>$W_{Active}$</th>
<th>$W_{Apical}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 \times 10^{-5}$</td>
<td>[0,1.4]</td>
<td>1500</td>
<td>4.7</td>
<td>0.8</td>
<td>0.1</td>
<td>0.04</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>[1.4,2.4]</td>
<td>5000</td>
<td>4.9</td>
<td>0.7</td>
<td>0.075</td>
<td>0.03</td>
<td>0.045</td>
</tr>
<tr>
<td></td>
<td>[2.4,3.0]</td>
<td>16500</td>
<td>4.9</td>
<td>0.65</td>
<td>0.063</td>
<td>0.025</td>
<td>0.038</td>
</tr>
</tbody>
</table>

Parameter values for Figure 5.5

Parameter values for Figure 5.6

| $2 \times 10^{-5}$ | [0,1.4] | 1500 | 4.7 | 0.8 | 0.1 | 0.04 | 0.06 |
| | [1.4,2.4] | 5000 | 4.9 | 0.7 | 0.075 | 0.03 | 0.045 |
| | [2.4,3.0] | 17000 | 4.9 | 0.65 | 0.063 | 0.025 | 0.038 |
and individual changes (Figure 5.7). These results suggest a causal basis for the robustness of normal development to mild environmental variation and developmental noise, and certain types of mutations (any of which could influence kinetic parameter values), but sensitive to teratogenic effects of more severe versions of each of these conditions.

Simulations using the standard parameter set (Table 5.1), but with the AER removed part-way during development, shows the same result as when this manipulation is performed on developing limbs [50]. In vivo, removal of the AER is equivalent to eliminating the distal source of the FGF gradient. After a time lag during which the suppressive gradient decays, the entire LALI zone up to the limb bud tip becomes susceptible to being organized by the activator-inhibitor system, leaving the limb distally truncated and no additional distal structures form (Figure 5.12).

5.3.2 Anteroposteriorly expanded limb buds

In contrast to the chicken forelimb, many limb buds have an apical zone that expands along the A-P axis during skeletogenic patterning (Figure 5.13). This occurs in the avian hindlimb, in reptilian and mammalian limbs, and to an even greater extent in pathological limbs, such as those of chickens which have received ectopic grafts of the zone of polarizing activity (ZPA) [51], embryos which bear the talpid2 mutation [80], or mice in which the Sonic hedgehog (Shh) and Gli3 genes have been knocked down [58]. In the dogfish, while there is no autopod (this being considered an innovation distinguishing fish from tetrapods), a limb bud that is expanded relative to those of birds and mammals produces an array of parallel elements [117], which while not true digits, assume a similar configuration.
Simulations in which the limb bud or its tip is allowed to expand show an increased number of parallel digits in the autopod (Figure 5.14, 5.11).

Let us recall the formula for \( x' \) as equation 5.3

\[
x'(t) = \sigma_x(x - vt) + v
\]

If we denote \([x_{\text{upper}}(t), y_{\text{upper}}(t)]\) and \([x_{\text{lower}}(t), y_{\text{lower}}(t)]\) are the trajectories of top and bottom points of the left boundary of LALI zone. Consider any point \((x_{\text{left}}(t), y_{\text{left}}(t))\) on the left boundary line of LALI zone, we have

\[
y_{\text{left}}(t) = \frac{y_{\text{upper}}(t) - y_{\text{lower}}(t)}{y_{\text{upper}}(0) - y_{\text{lower}}(0)} + y_{\text{lower}}(t),
\]

(5.10)

and

\[
y'_{\text{left}}(t) = \frac{y'_{\text{upper}}(t) - y'_{\text{lower}}(t)}{y_{\text{upper}}(0) - y_{\text{lower}}(0)} + y'_{\text{lower}}(t).
\]

(5.11)

Then for each node \((x(t), y(t))\) in the LALI zone, we simply set

\[
y'(t) = y'_{\text{left}}(t), \quad \text{if } y(t) = y_{\text{left}}(t).
\]

(5.12)

Thus, the vertical growth rate of the LALI zone depends on the location of each node.

For the simulation of Shh-Gli3 knockout (Figure 5.14a), the shrinking of the LALI zone in the \(x\)-direction keeps the same form \(T = 0\) to the final time 3 as the normal development (Table 5.1, Figure 5.5), i.e.,

\[
x'(t) = \sigma_x(x - vt) + v, \quad x = -0.2896, \quad v = 1, \quad t \in [0, 3].
\]

(5.13)
While for the $y$-direction, we have

$$y'(t) = 0, \quad t \in [0, 2.4];$$

$$y'(t) = [y(0) - y_{\text{lower}}(0)] \frac{y'_{\text{upper}}(t) - y'_{\text{lower}}(t)}{y_{\text{upper}}(0) - y_{\text{lower}}(0)} + y'_{\text{lower}}(t), \quad t \in [2.4, 3]; \quad (5.14)$$

where

$$[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, -6.94444(t - 2.4)^3 + 6.25(t - 2.4)^2 + 1], \quad t \in [2.4, 3];$$

$$[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 12.5(T - 2.4)^3 - 7.5(t - 2.4)^2], \quad t \in [2.4, 2.8];$$

$$[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, -0.4], \quad t \in [2.8, 3.0]. \quad (5.15)$$

The values of the key parameters

$$\lambda = 1500, \quad \delta = 4.7, \quad t \in [0, 1.4];$$

$$\lambda = 5000, \quad \delta = 4.9, \quad t \in [1.4, 2.4];$$

$$\lambda = 20000, \quad \delta = 4.9, \quad t \in [2.4, 3.0]. \quad (5.16)$$

For the simulation of ZPA grafting (Figure 5.14b), we have the same parameter values as for the $Shh-Gli3$ knockout. However, there are two differences. The first one is that $y' = 0$ from $t = 2.4$ to 3.0 for the LALI zone nodes whose initial $y$-locations ($t = 0$) are in the range $\frac{1}{3} \leq y(0) \leq \frac{2}{3}$. The other one is the frozen zone grows at the speed $v_1 = 0.5$, other than $v = 1.0$ from $t = 2.4$ to 3.0 for the nodes in the same range, i.e. $\frac{1}{3} \leq y(0) \leq \frac{2}{3}$. Thus, from $t = 2.4$ to 3.0 and for nodes $(x(t), y(t))$ with $\frac{1}{3} \leq y(0) \leq \frac{2}{3}$ in the LALI zone, we have $x(t) = x(0)e^{\sigma x t} + 2.4v + (t - 2.4)v_1$, and $x'(t) = \sigma x [x(t) - v_1 t - 2.4v + 2.4v_1] + v_1$. 
While for other nodes, we have \( x(t) = x(0)e^{\sigma x t} + vt \) and \( x'(t) = \sigma_x [x(t) - vt] + v \) the same as equation 5.3.

For the talpid\(^2\) simulation (Figure 5.14c), we have the same growth speeds both in \( x \) and \( y \) directions as the normal development (Table 5.1, Figure 5.5), i.e.

\[
x'(t) = \sigma_x (x - vt) + v, \quad \sigma_x = -0.2896, \quad v = 1, \quad t \in [0, 1.2];
\]
\[
y'(t) = 0, \quad t \in [0, 1.2]. \quad (5.17)
\]

The differences are the values of the key parameters

\[
\begin{align*}
\lambda &= 1500, \quad \delta = 4.7, \quad &t \in [0, 0.5]; \\
\lambda &= 5000, \quad \delta = 4.9, \quad &t \in [0.5, 0.8]; \\
\lambda &= 20000, \quad \delta = 4.9, \quad &t \in [0.8, 0.9]; \\
\lambda &= 1000, \quad \delta = 4.9, \quad &t \in [0.9, 0.95]; \\
\lambda &= 25000, \quad \delta = 4.9, \quad &t \in [0.95, 1.05]; \\
\lambda &= 500, \quad \delta = 4.9, \quad &t \in [1.05, 1.1]; \\
\lambda &= 25000, \quad \delta = 4.9, \quad &t \in [1.1, 1.2]. \quad (5.18)
\end{align*}
\]

In the dogfish fin simulation (the last picture of Figure 5.13), we have two sets of results, as in Figures 5.14d and 5.14e. The parameter values are

\[
x'(t) = \sigma_x (x - vt) + v, \quad \sigma_x = -0.4055, \quad v = 1, \quad t \in [0, 3];
\]
\[
y'(t) = \left[ y(0) - y_{lower}(0) \right] \frac{y'_{upper}(t) - y'_{lower}(t)}{y_{upper}(0) - y_{lower}(0)} + y'_{lower}(t), \quad t \in [0, 3]; \quad (5.19)
\]

120
where for Figure 5.14d

\[
[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 1.5t^2 + 1], \quad t \in [0, 1];
\]
\[
[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, t^3 - 6t^2 + 12t - 4.5], \quad t \in [1, 2];
\]
\[
[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, -t^3 + 6t^2 - 12t + 11.5], \quad t \in [2, 3];
\]
\[
[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0], \quad t \in [0, 1.5];
\]
\[
[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0.4444t^3 - 2t^2 + 3t - 1.5], \quad t \in [1.5, 3];
\]

(5.20)

and for Figure 5.14e

\[
[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 1.5t^2 + 1], \quad t \in [0, 1];
\]
\[
[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, t^3 - 6t^2 + 12t - 4.5], \quad t \in [1, 2];
\]
\[
[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 3.5], \quad t \in [2, 3];
\]
\[
[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0], \quad t \in [0, 1.5];
\]
\[
[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0.1481t^3 - 0.6667t^2 + t - 0.5], \quad t \in [1.5, 3].
\]

(5.21)

The values of key parameters are the same for Figures 5.14d and 5.14e

\[
\lambda = 3000, \quad \delta = 4.7, \quad t \in [0, 1];
\]
\[
\lambda = 13000, \quad \delta = 4.7, \quad t \in [1, 3].
\]

(5.22)

In all the simulations in Figure 5.14, the ratio of the widths between active zone and apical zone is the same as that in the normal development (Table 5.1, Figure 5.5).
5.3.3 Fossils

Many fossil vertebrate limbs have been characterized, including some that represent transitional forms between fins of fish-like ancestors and limbs of amphibians in which an authentic autopod is present [94] (Figure 5.15). Dinosaur limbs resemble those of modern reptiles and mammals, but ichthyosaurs, swimming dinosaurs, are of interest in that their paddles consist largely of nodular rather than rod-like skeletal elements [74] (Figure 5.15). We performed simulations based on a wide range of hypothetical developmental scenarios (the one leading to the ichthyosaur-like pattern is shown in Figure 5.16a) in which the general rules of the standard progression that generated (Figure 5.5) were maintained, but the contour of the distal region of the limb bud varied during development, as did the parameters $\lambda$ and $\delta$. The simulation end-points shown in (Figure 5.16) indicate that our model exhibits sufficient flexibility to reproduce the general features of a wide variety of fossil limb skeletons.

For the simulation of Brachypterygius (Figure 5.16a), we have the parameter values

\[
\begin{align*}
  x'(t) &= \sigma_x(x - vt) + v, \quad \sigma_x = 0.4621, \quad v = 1, \quad t \in [0, 1.5]; \\
  x'(t) &= \sigma_x(x - vt + 1.5) + v, \quad \sigma_x = 0.4621, \quad v = 2, \quad t \in [1.5, 1.75]; \\
  x'(t) &= \sigma_x(x - vt + 3.25) + v, \quad \sigma_x = 0.4621, \quad v = 3, \quad t \in [1.75, 2.55]; \\
  y'(t) &= [y(0) - y_{lower}(0)] \frac{y'_{upper}(t) - y'_{lower}(t)}{y_{upper}(0) - y_{lower}(0)} + y'_{lower}(t), \quad t \in [0, 2.55]; \quad (5.23)
\end{align*}
\]
where

\[
\begin{align*}
[&x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 1], \quad t \in [0, 0.5]; \\
[&x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 0.1t^3 - 0.05t^2 - 0.025t + 1.0125], \quad t \in [0.5, 1.5]; \\
[&x_{\text{upper}}(t), y_{\text{upper}}(t)] = [2t - 1.5, -10.8(2t - 1.5)^3 + 56.2(2t - 1.5)^2 \\
&- 95.2(2t - 1.5) + 54], \quad t \in [1.5, 1.75]; \\
[&x_{\text{upper}}(t), y_{\text{upper}}(t)] = [3t - 3.25, -0.0556(3t - 3.25)^3 + 0.4167(3t - 3.25)^2 \\
&- (3t - 3.25) + 2.7778], \quad t \in [2.5, 2.55]; \\
[&x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0], \quad t \in [0, 0.5]; \\
[&x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, -0.1t^3 + 0.05t^2 + 0.025t - 0.0125], \quad t \in [0.5, 1.5]; \\
[&x_{\text{lower}}(t), y_{\text{lower}}(t)] = [2t - 1.5, 10.8(2t - 1.5)^3 - 56.2(2t - 1.5)^2 \\
&+ 95.2(2t - 1.5) - 53], \quad t \in [1.5, 1.75]; \\
[&x_{\text{lower}}(t), y_{\text{lower}}(t)] = [3t - 3.25, 0.0556(3t - 3.25)^3 - 0.4167(3t - 3.25)^2 \\
&+ (3t - 3.25) - 1.7778], \quad t \in [2.5, 2.55]; \\
\end{align*}
\]

(5.24)

and the values of key parameters

\[
\begin{align*}
\lambda &= 1500, \quad \delta = 4.7, \quad t \in [0, 1.5]; \\
\lambda &= 20000, \quad \delta = 4.9, \quad t \in [1.5, 1.75]; \\
\lambda &= 5000, \quad \delta = 4.7, \quad t \in [1.75, 2.55]. \\
\end{align*}
\]

(5.25)

In equation 5.23, the \(\sigma_x\) is positive and the LALI zone is not shrinking any more.
For the simulation of Sauripterus (Figure 5.16b), we have the parameter values
\[ x'(t) = \sigma_x(x - vt) + v, \quad \sigma_x = -0.2027, \quad v = 1, \quad t \in [0, 3.5]; \]
\[ y'(t) = \left(y(0) - y_{lower}(0)\right) \frac{y'_{upper}(t) - y'_{lower}(t)}{y_{upper}(0) - y_{lower}(0)} + y'_{lower}(t), \quad t \in [0, 3.5]; \quad (5.26) \]

where
\[
\begin{align*}
[x_{upper}(t), y_{upper}(t)] &= [t, 1], \quad t \in [0, 0.5]; \\
[x_{upper}(t), y_{upper}(t)] &= [t, -2.5t^3 + 7.25t^2 - 5.375t + 2.1875], \quad t \in [0.5, 1.5]; \\
[x_{upper}(t), y_{upper}(t)] &= [t, -1.5t^3 + 9.75t^2 - 19.625t + 14.5625], \quad t \in [1.5, 2.5]; \\
[x_{upper}(t), y_{upper}(t)] &= [t, 0.1481t^3 - 1.7778t^2 + 7.1111t - 5.9815], \quad t \in [2.5, 3.5]; \\
[x_{lower}(t), y_{lower}(t)] &= [t, 0], \quad t \in [0, 0.5]; \\
[x_{lower}(t), y_{lower}(t)] &= [t, 0.0233t^3 - 0.1574t^2 + 0.1399t - 0.0335], \quad t \in [0.5, 3.5]; \quad (5.27)
\end{align*}
\]

and the values of key parameters
\[
\begin{align*}
\lambda &= 1500, \quad \delta = 4.7, \quad t \in [0, 0.5]; \\
\lambda &= 1000, \quad \delta = 4.7, \quad t \in [0.5, 1.2]; \\
\lambda &= 1000, \quad \delta = 4.9, \quad t \in [1.2, 1.8]; \\
\lambda &= 3000, \quad \delta = 4.9, \quad t \in [1.8, 2.5]; \\
\lambda &= 10000, \quad \delta = 4.9, \quad t \in [2.5, 3.5]. \quad (5.28)
\end{align*}
\]

For the simulation of Eusthenopteron (Figure 5.16c), we have the parameter
values

\[ x'(t) = \sigma_x(x - vt) + v, \quad \sigma_x = -0.2027, \quad v = 1, \quad t \in [0, 5]; \]
\[ y'(t) = \left[ y(0) - y_{\text{lower}}(0) \right] \frac{y_{\text{upper}}(t) - y_{\text{lower}}(t)}{y_{\text{upper}}(0) - y_{\text{lower}}(0)} + y_{\text{lower}}'(t), \quad t \in [0, 5]; \] (5.29)

where

\[ [x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 1], \quad t \in [0, 1.5]; \]
\[ [x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, -0.2t^3 + 1.5t^2 - 3.15t + 3.025], \quad t \in [1.5, 3.5]; \]
\[ [x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 1.8], \quad t \in [3.5, 5.0]; \]
\[ [x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0], \quad t \in [0, 1.5]; \]
\[ [x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, t^3 - 6t^2 + 11.25t - 6.75], \quad t \in [1.5, 2.5]; \]
\[ [x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, -0.5], \quad t \in [2.5, 3.0]; \]
\[ [x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, -2.6t^3 + 27.3t^2 - 93.6t + 104.8], \quad t \in [3.0, 4.0]; \]
\[ [x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0.8], \quad t \in [4.0, 5.0]; \] (5.30)

and the values of key parameters

\[ \lambda = 1500, \quad \delta = 4.7, \quad t \in [0, 1.5]; \]
\[ \lambda = 5000, \quad \delta = 4.9, \quad t \in [1.5, 2.0]; \]
\[ \lambda = 4000, \quad \delta = 4.9, \quad t \in [2.0, 3.5]; \]
\[ \lambda = 1000, \quad \delta = 4.7, \quad t \in [3.5, 4.0]; \]
\[ \lambda = 5000, \quad \delta = 4.9, \quad t \in [4.0, 4.5]; \]
\[ \lambda = 17000, \quad \delta = 4.9, \quad t \in [4.5, 5.0]. \] (5.31)
For the simulation of Panderichthys (Figure 5.16d and 5.16e), we have the parameter values

\[ x'(t) = \sigma_x(x - vt) + v, \quad \sigma_x = -0.2027, \quad v = 1, \quad t \in [0, 4]; \]

\[ y'(t) = \left[ y(0) - y_{lower}(0) \right] \frac{y'_{upper}(t) - y'_{lower}(t)}{y_{upper}(0) - y_{lower}(0)} + y'_{lower}(t), \quad t \in [0, 4]; \] (5.32)

where

\[ [x_{upper}(t), y_{upper}(t)] = [t, 1], \quad t \in [0, 1.0]; \]

\[ [x_{upper}(t), y_{upper}(t)] = [t, -1.6t^3 + 7.2t^2 - 9.6t + 5], \quad t \in [1.0, 2.0]; \]

\[ [x_{upper}(t), y_{upper}(t)] = [t, 44.4444t^3 - 286.6667t^2 + 613.3333t - 433.7556], \]

\[ t \in [2.0, 2.3]; \]

\[ [x_{upper}(t), y_{upper}(t)] = [t, -0.1628t^3 + 1.5388t^2 - 4.4942t + 5.3777], \]

\[ t \in [2.3, 4.0]; \]

\[ [x_{lower}(t), y_{lower}(t)] = [t, 0], \quad t \in [0, 3.0]; \]

\[ [x_{lower}(t), y_{lower}(t)] = [t, -0.1t^3 + 1.3t^2 - 5.1t + 6.3], \quad t \in [3.0, 4.0]; \] (5.33)

and the values of key parameters

\[ \lambda = 1500, \quad \delta = 4.7, \quad t \in [0, 1]; \]

\[ \lambda = 1000, \quad \delta = 4.7, \quad t \in [1, 2.3]; \]

\[ \lambda = 3000, \quad \delta = 4.9, \quad t \in [2.3, 3]; \]

\[ \lambda = 9000, \quad \delta = 4.9, \quad t \in [3, 3.5]. \] (5.34)
For Figure 5.16d

$$\lambda = 18000, \ \delta = 4.9, \quad t \in [3.5, 4]; \quad (5.35)$$

For Figure 5.16e

$$\lambda = 20000, \ \delta = 4.9, \quad t \in [3.5, 4]. \quad (5.36)$$

For the simulation of Tiktaalik (Figure 5.16f), we have the parameter values

$$x'(t) = \sigma_x(x - v t) + v, \quad \sigma_x = -0.2027, \quad v = 1, \quad t \in [0, 5.5];$$

$$y'(t) = \left[ y(0) - y_{\text{lower}}(0) \right] \frac{y'_{\text{upper}}(t) - y'_{\text{lower}}(t)}{y_{\text{upper}}(0) - y_{\text{lower}}(0)} + y'_{\text{lower}}(t), \quad t \in [0, 5.5]; \quad (5.37)$$

where

$$[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 1], \quad t \in [0, 1.5];$$

$$[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, -t^3 + 6t^2 - 11.25t + 7.75], \quad t \in [1.5, 2.5];$$

$$[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, 1.5], \quad t \in [2.5, 4.5];$$

$$[x_{\text{upper}}(t), y_{\text{upper}}(t)] = [t, -0.5t^2 + 4.5t - 8.625], \quad t \in [4.5, 5.5];$$

$$[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0], \quad t \in [0, 0.5];$$

$$[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0.0741t^3 - 0.4444t^2 + 0.3889t - 0.0926], \quad t \in [0.5, 3.5];$$

$$[x_{\text{lower}}(t), y_{\text{lower}}(t)] = [t, 0.25t^2 - 1.75t + 2.0625], \quad t \in [3.5, 5.5]; \quad (5.38)$$
and the values of key parameters

\[
\begin{align*}
\lambda &= 1500, \quad \delta = 4.7, \quad t \in [0, 0.5]; \\
\lambda &= 1000, \quad \delta = 4.7, \quad t \in [0.5, 2]; \\
\lambda &= 1500, \quad \delta = 4.9, \quad t \in [2, 3.5]; \\
\lambda &= 2700, \quad \delta = 4.9, \quad t \in [3.5, 4.5]; \\
\lambda &= 10000, \quad \delta = 4.9, \quad t \in [4.5, 5]; \\
\lambda &= 3000, \quad \delta = 4.9, \quad t \in [5, 5.5]; \\
\end{align*}
\]

(5.39)

All simulations show that in the parameter range of the standard model both the contraction of the LALI zone in the P-D direction and the level-specific variation of the parameters \( \lambda \) and \( \delta \) are essential to obtaining a normal developmental outcome (Figure 5.9, 5.10). An important result of these simulations, suggested in earlier versions of the model [45, 80], is that contraction of the P-D dimension of the LALI zone alone (due to the attenuation of the suppressive gradient with its source at the AER), tends to cause a proximodistal increase in element number (Figure 5.8, 5.9, 5.10).
(Figure 5.3A) Chicken embryo at 5 days of development with right fore-limb bud accentuated by staining of the embryonic flank with Eosin.

(Figure 5.3B) Schematic representation of core mechanism of chondrogenic pattern formation, comprising activator subnetwork (A, green circle), inhibitory subnetwork (I, red circle), and adhesive, matricellular and ECM molecules induced by A (violet). The molecules in the violet circle promote precartilage condensation, which induces chondrogenesis. Subnetwork A has positively autoregulatory properties; it also induces I, which in turn inhibits A.

(Figure 5.3C) Schematic representation of bare bones limb development model in a 2D template corresponding to a 5-day chicken wing bud. The template is divided into a LALI (lateral autoactivation-lateral inhibition) zone in which the reaction-diffusion process defined by A and I operates. Action of A within the LALI zone is suppressed by a diffusible signal (FGF, red-to-pink gradient), with its source at the AER (apical ectodermal ridge) at the distal end (right). At the low end of the FGF gradient a portion of the LALI zone, termed the active zone, is permissive for the production of the cartilage-promoting molecules shown in (B) (violet). Cartilage (gray) forms in cells that leave the LALI zone due to elongation of the limb bud.
Figure 5.3. Relationship between core cartilage patterning network and bare bones framework for limb development.
Figure 5.4. Developmental progression of chicken forelimb between days 3 and 7 of development (indicated by the corresponding Hamburger-Hamilton stages). Early cartilage, including precartilage condensations, shown in light blue; definitive cartilage shown in darker blue [80].
Figure 5.5. A sequence of snapshots from the simulation of normal limb development(1). Color legend: black/red/violet corresponds to 6.0, white/blue to 0.0, green to the median level 3.0.
Figure 5.6. A sequence of snapshots from the simulation of normal limb development (2). Color legend: black/red/violet corresponds to 6.0, white/blue to 0.0, green to the median level 3.0.
Figure 5.7. Dependence of the number of stripes on the kinetic parameters $\lambda$ and $\delta$ during the first phase of normal development in Figure 5.5.

Figure 5.8. Simulations on a shrinking domain with curved apical zone.
Figure 5.9. LALI zone fixed, other parameter changes as normal development in Figure 5.5. Color legend: black corresponds to 5.6, white to 0.0.

(a) $\lambda = 5000, \delta = 4.9$

(b) $\lambda = 16500, \delta = 4.9$

Figure 5.10. LALI zone narrows as Figure 5.5, but $\lambda$ and $\delta$ remain unchanged. Color legend: black corresponds to 5.6, white to 0.0.
Figure 5.11. Number of parallel stripes in the model limb bud increases with expansion of the A-P (y) direction length of the LALI zone. Color legend: black corresponds to 5.5, white to 0.0.
Figure 5.12. Top: an intact chicken wing bud at an early stage of development and the limb skeleton that it generates. Middle: a wing bud at the same early stage with the AER removed, and the resulting limb skeleton, which attains a normal size but is truncated beginning at the elbow. Bottom: a later stage wing bud whose AER has been removed. The resulting skeleton is truncated from the wrist onward.
Figure 5.13. Effect of limb bud distal expansion
Figure 5.14. Simulations of limb bud distal expansion. Color legend: for (a), (b), (c), black/red/violet corresponds to 6.1, white/blue to 0.0, green to the median level 3.05; for (d), (e), black/red/violet corresponds to 6.5, white/blue to 0.0, green to the median level 3.25.
Figure 5.15. Fossil limb skeletons

Brachypterygius

Sauripterus

Eusthenopteron

Panderichthys

Tihoelk
Figure 5.16.
Figure 5.16. Simulations of fossil limb skeletons. Color legend: black/red/violet corresponds to 6.2, white/blue to 0.0, green to the median level 3.1.


