A Stable Algorithm for Non-Negative Invariant Numerical Solution of Reaction-Diffusion Systems on Complicated Domains

Insoon Yang

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-77
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-77.html

May 10, 2012

Acknowledgement

First of all, I owe my deepest gratitude to my advisor, Professor Claire Tomlin, who has extended her valuable support and encouragement throughout this work. I thank her for inspiring me during our discussions to take various perspectives on my work.
I am also thankful to Professors Phillip Colella and L. Craig Evans. I will never forget Prof. Colella's advice that a good algorithm for numerical solution of a partial differential equation (PDE) should be based on a solid understanding of its analytic solution. Prof. Evans provided wonderful PDE courses, in which I learned modern PDE theory. I also appreciate his willingness to answer my questions on reaction-diffusion systems.

A Stable Algorithm for Non-Negative Invariant Numerical Solution of Reaction-Diffusion Systems on Complicated Domains

by

Insoon Yang

B.S. (Seoul National University) 2009

A thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Claire J. Tomlin, Chair
Professor Phillip Colella

Spring 2012

The thesis of Insoon Yang is approved.

Chair	Date
	Date

University of California, Berkeley

Spring 2012

A Stable Algorithm for Non-Negative Invariant Numerical Solution of Reaction-Diffusion Systems on Complicated Domains

by

Insoon Yang

Abstract

A Stable Algorithm for Non-Negative Invariant Numerical Solution of Reaction-Diffusion Systems on Complicated Domains

by

Insoon Yang

Master of Science in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Claire J. Tomlin, Chair

We present a Cartesian grid finite difference numerical method for solving a system of reaction-diffusion initial boundary value problems with Neumann type boundary conditions. The method utilizes adaptive time-stepping, which guarantees stability and non-negativity of the solutions. The latter property is critical for models in biology where solutions represent physical measurements such as concentration. The level set representation of the boundary enables us to handle domains with complicated geometry with ease. We provide numerical validation of our method on synthetic and biological examples. Empirical tests demonstrate second order convergence rate in the L^1- and L^2-norms, as well as in the L^∞-norm for many cases.

Professor Claire J. Tomlin
Thesis Committee Chair

Contents

List of Figures

List of Tables

Acknowledgements

First of all, I owe my deepest gratitude to my advisor, Professor Claire Tomlin, who has extended her valuable support and encouragement throughout this work. I thank her for inspiring me during our discussions to take various perspectives on my work.

I am also thankful to Professors Phillip Colella and L. Craig Evans. I will never forget Prof. Colella's advice that a good algorithm for numerical solution of a partial differential equation (PDE) should be based on a solid understanding of its analytic solution. Prof. Evans provided wonderful PDE courses, in which I learned modern PDE theory. I also appreciate his willingness to answer my questions on reaction-diffusion systems.

Finally, I would like to thank Professors Ronald Fedkiw, Frédéric Gibou, Stanley Osher and Richard Tsai for feedback on previous versions of the thesis, and Professor Jay Groves and his lab for providing image data of Ephrin-A1 intracellular distribution.

Chapter 1

Introduction

Consider the following reaction-diffusion system for a vector valued function $u = (u_1, u_2, \ldots, u_M)$ with Neumann boundary conditions:

$$\frac{\partial u_m}{\partial t} = D_m \Delta u_m + f_m(x, y, t, u) \qquad \text{in } \Omega \times (0, T) \tag{1.1a}$$

$$\frac{\partial u_m}{\partial \nu} = 0 \qquad \text{on } \partial\Omega \times (0, T) \tag{1.1b}$$

$$u_m(x, 0) = u_m^0(x) \qquad \text{on } \Omega \times \{t = 0\}, \tag{1.1c}$$

for $m = 1, \cdots, M$. The domain Ω is an open, bounded and connected subset of $\mathbb{R}^2$, whose boundary is $\partial\Omega$. The vector ν denotes the outward unit normal vector to the domain and D_m denotes the diffusion rate for each $m = 1, \cdots, M$. Similar to u, we write $u^0 := (u_1^0, \cdots, u_M^0)$, $f := (f_1, \cdots, f_M)$. Here we assume that f is continuously differentiable. The system (1.1) has been widely used as a fundamental model for describing biological pattern formation [17], [42], spatio-temporal biological signaling [1], [15], population dynamics [5], [26] and chemical reactions [16], [18]. Although we shall mainly focus on the Neumann boundary conditions (1.1b), the method presented in this thesis is also applicable to general Robin boundary conditions.

While the structure of reaction-diffusion system (1.1) is quite similar to the heat equation, their behaviors are generally very different due to the nonlinear forcing term f. For example, blowing up in L^∞ in a finite time occurs for some f, i.e, the solution u of the reac-

tion diffusion system (1.1) when f are not in $L^\infty(\Omega \times [0,T])$ [32]. In addition, the solution to reaction-diffusion systems often represents physical, chemical, and biological components such as the concentration level of chemicals and morphogens, which are non-negative by definition. Thus, analytical conditions for non-negative invariance of the solution is crucial in modeling these applications. To guarantee the global existence and the non-negative invariance of the classical solution we apply the result of invariant sets of the reaction-diffusion systems [20]–[22] to $[0,+\infty)^M$: assume for all $m = 1, \cdots, M$, and for all $(x,t) \in \Omega \times (0,T)$,

(A1) $f_m(x,t,u_1,\cdots,u_{m-1},0,u_{m+1},\cdots,u_M) \geq 0$ for each $u \in [0,+\infty)^M$, and

(A2) there exist constants c and d, such that $f_m(x,t,u) \leq cu_m + d$ for each $u \in [0,+\infty)^M$.

Condition (A1) is called the *quasi-positivity* of f, which ensures that $[0,+\infty)^M$ is an invariant set of the reaction-diffusion system [20]. Condition (A2) uniformly bounds f_m by an affine function of u_m to prevent the solution from blowing-up in finite time. Note that the conditions are more strict than the conditions given in [31] for global existence of a classical solution to reaction-diffusion systems.

In this thesis, we develop a novel algorithm for solving (1.1) that numerically realizes the constraints (A1) and (A2). We employ a level set representation of $\partial\Omega$; Neumann boundary conditions are enforced at the boundaries using the normal direction extracted from the level set representation. While level set representation of the domain boundary has been used prior to our work, the extraction of the outward normal directions using a third order accurate gradient of the level set function is novel. The main advantages of our approach are:

1. The scheme is provably stable and guarantees the solution to be non-negative invariant.

2. The level set representation allows us to handle arbitrarily complicated domain geometry and the Neumann boundary condition with ease.

3. The method is provably second order in the L^1-norm.

The importance of stability is clear. Furthermore, the non-negative invariance is necessary in modeling a physical phenomena, such as chemical concentration, in a meaningful way. The ease of handling arbitrarily complicated domains is useful, for example, when modeling the temporal evolution of the distribution of chemical species in a biological cell. Also, the robustness of the level set method to moving interfaces or sets may complement our approach to problems of solving (1.1) in moving domains. While we can prove a second order convergence in the L^1-norm, empirical tests have demonstrated a second order convergence in many cases in L^2-norm and the L^∞-norm.

1.1 Related work

Numerical methods for solving reaction-diffusion systems on irregular domains have been developed in the context of the heat equation and the reaction-diffusion equation. We list several related methods below:

- The finite element method using unstructured meshes has been widely used for solving partial differential equations in a domain with complex geometry [14]. However, the mesh generation over an irregular domain is often a complicated process and computationally expensive, which makes the implementation of the finite element method not as simple as those of the finite difference and finite volume methods utilizing Cartesian grids.

- The *immersed interface method* [19] was originally developed for solving the elliptic equations with coefficients and forcing terms that are discontinuous across the interface on Cartesian grids. It uses a local coordinate transformation, which results in a system of linear equations to approximate the solution value at a point near the boundary. Although the original version of the algorithm requires appropriate jump conditions on the solution and its normal derivative, which are not generally available in practice, it has been modified in [6] to solve the heat equations with the Neumann boundary condition without requiring the jump conditions. However, the six-point

stencil selecting method in [6] near the interface is not trivial, and furthermore, it is not clear that the modified method has a consistent convergence rate.

- The *ghost fluid method* [4] is applicable to solving parabolic equations with second-order and fourth-order accuracy [7], [8]. Its advantage is its simplicity of handling irregular domains by construction of a ghost solution on each side of the interface, which is easily generalized in the three dimensional case. However, it requires jump conditions at the boundary to utilize the information from the both sides of the interface.

- The conceptual idea suggested by Morton and Myers using interpolation along the normal is similar to the way we handle the domain with complex geometry, but it lacks of the discussion of stability and the detailed methods for computing normals [25].

- The Cartesian grid *embedded boundary methods* (or the cut-cell methods) are finite volume methods for approximating fluxes between volumes using interpolation; it naturally resolves the problem of mesh generation of irregular domains [13], [23], [40]. This method is also extended to the three-dimensional and surface diffusion problems [35], [36]. An important variant of the cut-cell type method is [29], which yields a symmetric discretization when a domain is divided by an interface. These methods are implicit, which suggest good stability properties; however, no proof has been shown for stability.

- The *moving boundary node method* is another finite volume method that projects the grid points near the boundary onto the boundary. This method is easily applicable to the three dimensional case [43]. However, as with the previous work, no proof has been shown for stability.

1.2 Outline of the thesis

In Section 2, we propose a novel spatial discretization method for the Laplacian or diffusion operator on a domain with complex geometry. Then in Section 3 we discuss

conditions on time steps for guaranteeing the non-negative invariance of the solution and the stability. We also prove that the adaptive time stepping is uniformly-bounded from below, followed by proposing a complete algorithm in Section 4. Numerical tests for the convergence rate of the method are examined and verified with examples in Sections 4.1 and 5, respectively.

Chapter 2

Spatial discretization

2.1 Notation and setup

Without loss of generality we can assume that Ω is contained in a rectangular region $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ since Ω is bounded. Let us discretize the box into a uniform Cartesian grid of size $N \times N$ so that $x_i = x_{\min} + (i-1) \cdot h$ and $y_i = y_{\min} + (i-1) \cdot h$ for $i = 1, \cdots, N$. Here $h = (N-1)/(x_{\max} - x_{\min}) = (N-1)/(y_{\max} - y_{\min})$ is the grid spacing in the x and y directions. We denote by $u^n_{m,i,j}$ the numerical approximation of $u_m(x_i, y_j, t_n)$ for each $m = 1, \cdots, M$.

Let $\phi : \mathbb{R}^2 \to \mathbb{R}$ be a signed-distance (level set) function, whose zero sub-level set corresponds to Ω:

$$\phi(x,y) = \begin{cases} \operatorname{dist}\big((x,y), \Omega\big) & \text{if } (x,y) \in \Omega^c, \\ -\operatorname{dist}\big((x,y), \Omega^c\big) & \text{if } (x,y) \in \Omega, \end{cases} \tag{2.1}$$

where $\operatorname{dist}(\cdot,\cdot)$ is the geodesic distance between two sets in $\mathbb{R}^2$. In this thesis, we assume an exact ϕ is given for the pre-defined domain Ω. In practice, for arbitrarily shaped Ω, highly accurate numerical techniques for constructing signed-distance functions are available [2], [41].

2.2 Regular and Irregular grid points

Our first goal is approximating $\Delta u = u_{xx} + u_{yy}$ on the right hand side of (1.1a). For notational convenience we suppress the vector index m and the time index n, in $u^n_{m,i,j}$ and u_m. We begin by categorizing the grid points inside the domain as follows: a grid point is called *regular* if its four neighboring grid points are inside the domain; otherwise, it is called *irregular*. For a regular grid point at the location (i, j) we use the standard five point stencil finite difference scheme for approximating the operator:

$$\Delta u(x_i, x_j, t_n) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}. \tag{2.2}$$

We discretize (1.1) in time using the forward Euler scheme in which case the local truncation error is second-order in space. If (i, j) is irregular, however, the standard finite difference method is not applicable since at least one of $u_{i-1,j}, u_{i+1,j}, u_{i,j-1}$ and $u_{i,j+1}$ is not defined. We further assume that at least one of $(i-1, j)$ and $(i+1, j)$ lies inside the domain, and at least one of $(i, j-1)$ and $(i, j+1)$ lies inside the domain. Without loss of generality, we suppose that $(i-1, j)$ and $(i, j+1)$ are inside the domain, and $(i+1, j)$ and $(i, j-1)$ are outside the domain. We note the case in which only one neighboring point is outside the domain is naturally included. As shown in Figure 2.1, the right and bottom arms of (i, j) have intersections with the boundary. Let us refer to these intersection points as R and B with coordinates $(i + a_{i+1,j}, j)$ and $(i, j - a_{i,j-1})$, respectively, where $a_{i+1,j}, a_{i,j-1} \in [0, 1)$. In addition, we let U_R and U_B denote the analytic solutions at R and B. If we have a third order accurate approximation u_R and u_B of U_R and U_B, respectively, the following scheme with the forward Euler time integration has the first-order local truncation error in space:

$$\begin{aligned} \Delta u(x_i, x_j, t_n) \approx & \left(\frac{u_R - u_{i,j}}{a_{i+1,j}h} - \frac{u_{i,j} - u_{i-1,j}}{h} \right) \frac{2}{(a_{i+1,j} + 1)h} \\ & + \left(\frac{u_{i,j+1} - u_{i,j}}{h} - \frac{u_{i,j} - u_B}{a_{i,j-1}h} \right) \frac{2}{(a_{i,j-1} + 1)h}. \end{aligned} \tag{2.3}$$

Definition 1. Denote $\Delta_h u_{i,j}$ as the discretization (2.2) if (i, j) is a regular point and (2.3) if (i, j) is a irregular point.

Our first main contribution is a method for obtaining third order accurate approxima-

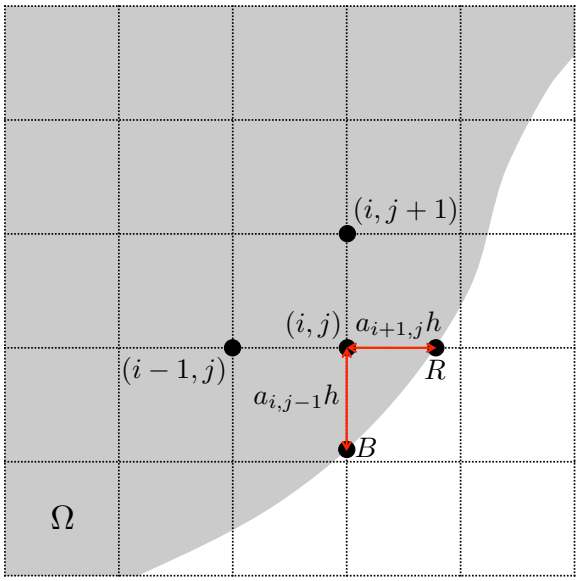

Figure 2.1. (i, j) is an irregular point whose right and bottom arms are intersected by the boundary of the domain.

tions of the solution at the boundary points U_R and U_B with Neumann boundary conditions. We outline the procedure below:

Step 1: **Compute the outward normal directions**, i.e. slope of ν, on the boundary points with third-order accuracy. This is achieved by computing a third order approximation of the gradient of the level set function ϕ.

Step 2: **Extend the normal inward to the domain** and choose two points that intersect the grid lines.

Step 3: **Approximate the solution at these intersecting points** using the second-order interpolation of three neighboring points.

Step 4: **Approximate the solution at the boundary points R and B** by extrapolating the two solution values with the information of the normal derivative at the boundary.

Now we examine each step in detail and justify the third-order accuracy.

Remark 2. We note that problems with Dirichlet boundary conditions can also be handled by our method because the values of u_R and u_B are given as boundary conditions. Therefore

we can handle the problems with Robin boundary conditions, $p\partial u/\partial \nu + qu = b$, without significant additional effort.

2.3 Computing the outward normal directions

Approximation of ϕ_x and ϕ_y at (i, j) with third-order accuracy is achieved by

$$\phi_x(x_i, y_j) \approx \frac{1}{3}\left(2\frac{\phi_{i+1,j} - \phi_{i-1,j}}{h} - \frac{\phi_{i+2,j} - \phi_{i-2,j}}{4h}\right) =: \phi^{num}_{x,i,j}, \tag{2.4}$$

$$\phi_y(x_i, y_j) \approx \frac{1}{3}\left(2\frac{\phi_{i,j+1} - \phi_{i,j-1}}{h} - \frac{\phi_{i,j+2} - \phi_{i,j-2}}{4h}\right) =: \phi^{num}_{y,i,j}. \tag{2.5}$$

Furthermore, we show that the discretizations (2.4) and (2.5) are sufficient to approximate the slopes ϕ_x/ϕ_y or ϕ_y/ϕ_x to third order accuracy.

Proposition 2.3.1. *Suppose that $\phi^{num}_{x,i,j}$ and $\phi^{num}_{y,i,j}$ are defined as in* (2.4) *and* (2.5)*. Then we have the following estimates:*

$$\frac{\phi_x(x_i, y_j)}{\phi_y(x_i, y_j)} - \frac{\phi^{num}_{x,i,j}}{\phi^{num}_{y,i,j}} = O(h^3), \quad \text{if } \phi_y \neq 0, \tag{2.6}$$

$$\frac{\phi_y(x_i, y_j)}{\phi_x(x_i, y_j)} - \frac{\phi^{num}_{yi,j}}{\phi^{num}_{x,i,j}} = O(h^3), \quad \text{if } \phi_x \neq 0. \tag{2.7}$$

Proof. We shall only prove (2.6); (2.7) follows via a similar argument. First, suppress i, j, x_i, y_j in the expression for notational convenience, and note that

$$\begin{aligned}\frac{\phi^{num}_x}{\phi^{num}_y} &= \frac{\phi_x + O(h^3)}{\phi_y + O(h^3)} \\ &= \frac{\phi_x}{\phi_y} + \frac{(\phi_y - \phi_x)}{(\phi_y + O(h^3))\phi_y} O(h^3).\end{aligned}$$

Note that $(\phi_y - \phi_x)/((\phi_y + O(h^3))\phi_y)$ is bounded by a constant as $h \to 0$ since $\phi_y \neq 0$. □

Interpolating this approximation of slopes at grid points near the boundary, we now approximate the slopes at the points on the boundary with third-order accuracy. Note that, points on the boundary with very large absolute value of curvature would have abrupt changes in the numerical values of the slopes. Therefore, the interpolation should be

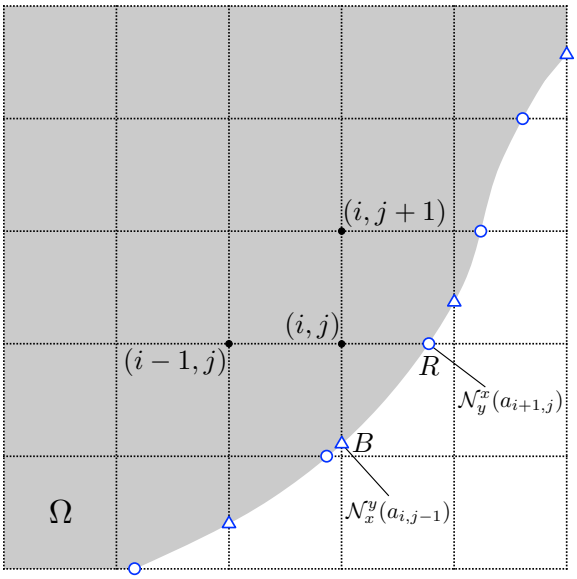

Figure 2.2. The rule for choosing $\mathcal{N}_y^x$ or $\mathcal{N}_x^y$ for approximating the slope of normal vector: if the boundary intersects a horizontal grid line we approximate ϕ_x/ϕ_y by $\mathcal{N}_y^x$ (Circles); if the boundary intersects a vertical grid line we approximate ϕ_y/ϕ_x by $\mathcal{N}_x^y$ (Triangles).

designed to robustly treat the possibility of a numerical discontinuity. *Essentially non-oscillatory* (ENO) interpolation [38], [39] is well-suited for interpolating such functions. In particular, to approximate the slope with third-order accuracy, it suffices to use the second-order ENO interpolation.

Recall R, the point $(i + a_{i+1,j}, j)$ between (i, j) and $(i + 1, j)$ where the horizontal grid line $y = y_j$ intersects the boundary of Ω. Let $\phi_{x,R}$ be the exact value of ϕ_x at R, and similarly for $\phi_{y,R}$. We approximate the slope of the gradient $\phi_{x,R}/\phi_{y,R}$ by a quadratic ENO polynomial

$$\begin{aligned}\frac{\phi_{x,R}}{\phi_{y,R}} &\approx \frac{\phi_{x,i,j}}{\phi_{y,i,j}} + \left(\frac{\phi_{x,i+1,j}}{\phi_{y,i+1,j}} - \frac{\phi_{x,i,j}}{\phi_{y,i,j}}\right) a_{i+1,j} + H_x a_{i+1,j}(a_{i+1,j} - 1)h^2 \\ &=: \mathcal{N}_y^x(a_{i+1,j}),\end{aligned} \tag{2.8}$$

which is third order accurate. Here, H_x is chosen as the coefficient corresponding to less oscillatory polynomial interpolation, i.e., let

$$H_x = \begin{cases} H_x^+ & \text{if } |H_x^+| < |H_x^-|, \\ H_x^- & \text{otherwise,} \end{cases}$$

where

$$H_x^+ = \frac{1}{2h^2}\left(\frac{\phi_{x,i+2,j}}{\phi_{y,i+2,j}} - 2\frac{\phi_{x,i+1,j}}{\phi_{y,i+1,j}} + \frac{\phi_{x,i,j}}{\phi_{y,i,j}}\right),$$
$$H_x^- = \frac{1}{2h^2}\left(\frac{\phi_{x,i+1,j}}{\phi_{y,i+1,j}} - 2\frac{\phi_{x,i,j}}{\phi_{y,i,j}} + \frac{\phi_{x,i-1,j}}{\phi_{y,i-1,j}}\right).$$

Similarly, for a boundary point $B = (i, j - a_{i,j-1})$ on the vertical grid line $x = x_i$ between (i, j) and $(i, j-1)$, we approximate the slope as

$$\begin{aligned}\frac{\phi_{y,B}}{\phi_{x,B}} &\approx \frac{\phi_{y,i,j}}{\phi_{x,i,j}} + \left(\frac{\phi_{y,i,j-1}}{\phi_{x,i,j-1}} - \frac{\phi_{y,i,j}}{\phi_{x,i,j}}\right) a_{i,j-1} + H_y a_{i,j-1}(a_{i,j-1} - 1)h^2 \\ &=: \mathcal{N}_x^y(a_{i,j-1})\end{aligned} \tag{2.9}$$

Since all intermediate procedures are third-order accurate in space, we can conclude that the approximations (2.8) and (2.9) are third-order accurate in space, i.e.,

$$\begin{aligned}\frac{\phi_{x,R}}{\phi_{y,R}} - \mathcal{N}_y^x(a_{i+1,j}) &= O(h^3), \\ \frac{\phi_{y,B}}{\phi_{x,B}} - \mathcal{N}_x^y(a_{i,j-1}) &= O(h^3).\end{aligned} \tag{2.10}$$

2.4 Extending the normal line into Ω

Next, we extend the normal line inward from the boundary and find two intersecting points with the grid lines: we refer to these as *extension points*. As shown in Figures 2.3, for example, P and Q are two extension points with respect to the boundary point R.

We employ the following two-stage rule for selecting extension points:

1. Depending on the inward normal direction from R on $\partial\Omega$, determine the *inner box* and *outer box* corresponding to R; see Figure 2.3 for an example in the case of a boundary point on a horizontal grid line.

2. Extend normal line from R inward, and let P and Q be the intersections of this line with the edges of the outer and inner boxes, respectively.

The case of a boundary point on a vertical grid line, B, is shown in Figure 2.4. This rule obeys the following considerations for the extension points: first, they should lie on

grid lines; second, we should space them as equally as possible. The significance of the former is that the solution can be approximated with a one dimensional interpolation using nearby grid values; this will be presented in Section 2.5. The latter prevents P and Q from being too close to each other, which would deteriorate the accuracy in the extrapolation in approximating the value at R. Note how the rule guarantees that the distance between P and Q is at least one grid spacing. Also, the positions of the extensions are approximated with the same order of accuracy as $\mathcal{N}_y^x$ to the true normal as per (2.10).

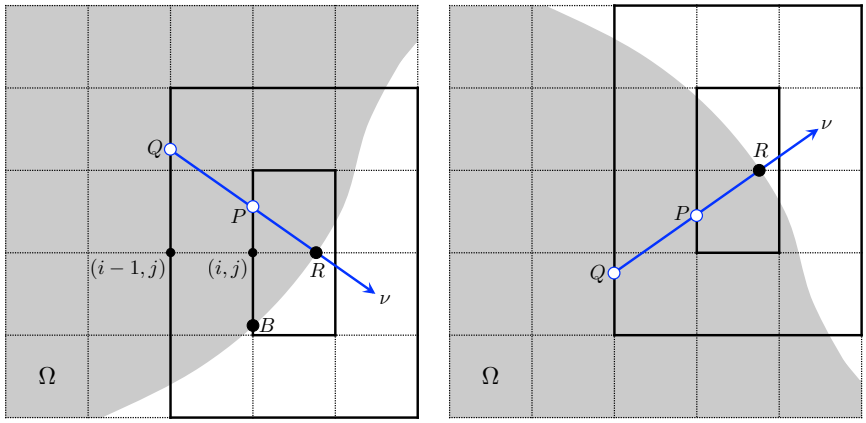

Figure 2.3. Normal lines extended inwardly (blue) and their intersections (extension points) with inner and outer boxes of the boundary point of interest, R.

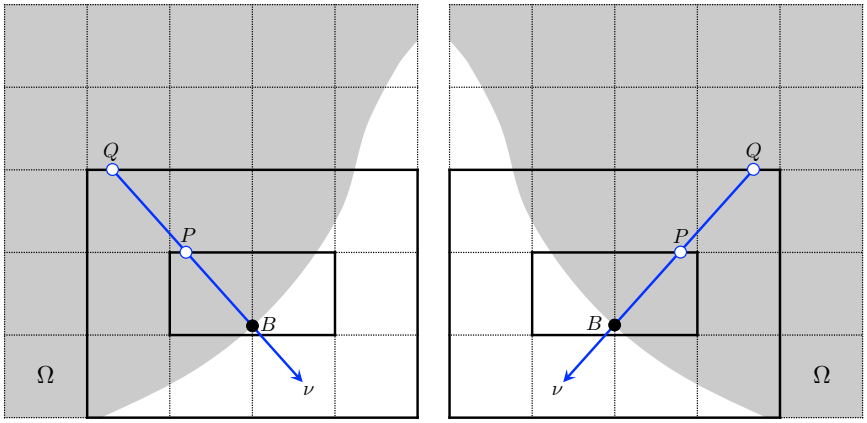

Figure 2.4. Normal lines extended inwardly (blue) and their intersections (extension points) with inner and outer boxes of the boundary point of interest, B.

2.5 Approximating u on the extension points

To explain Step 3 in detail, let us consider the case of Figure 2.5. Our aim is to compute the approximated solution u_P and u_Q at two extension points P and Q of R, respectively. To approximate u_P, we first choose the adjacent grid points $u_{i,j}$ and $u_{i,j+1}$. If $(i, j-1)$ (resp. $(i, j+2)$) is outside the domain, we pick $(i, j+2)$ (resp. $(i, j-1)$) for the third point that is used for the interpolation. If both are inside the domain, we choose the one closer to P. If the three points are $(i, j), (i, j+1), (i, j+2)$, for example, we use the standard second-order interpolation to obtain u_P:

$$u_P = u_{i,j} + (u_{i,j+1} - u_{i,j})\theta + \frac{1}{2}(u_{i,j+2} - 2u_{i,j+1} + u_{i,j})\theta(\theta - 1),$$

where $\theta := \text{dist}(P, (x_i, y_j))/h \in [0, 1]$. The solution at the point Q can be approximated with the same interpolation method. Since the interpolation we proposed is second-order, the approximated solutions are third-order accurate in space.

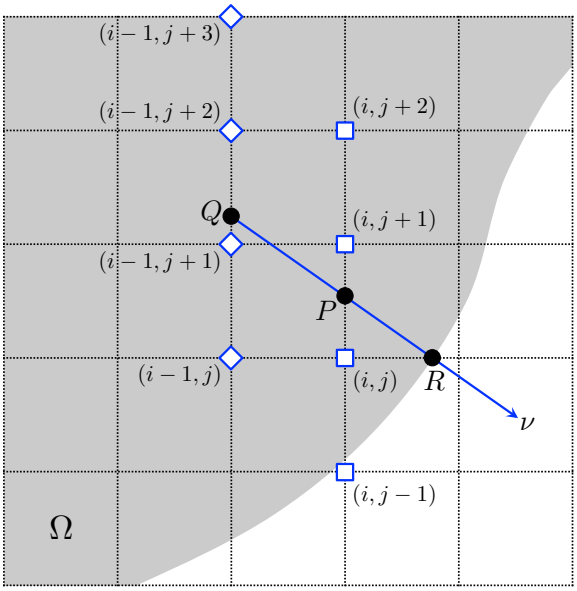

Figure 2.5. Approximation of the solution at the boundary point R interpolating the extension points P and Q. Note that, for approximating u_P, we consider the four square points and eventually use (i, j), $(i, j+1)$ and $(i, j+2)$ for the approximation because $(i, j-1)$ is outside of the domain. To approximate u_Q, we first detect the four diamond points and then utilize $(i-1, j)$, $(i-1, j+1)$ and $(i-1, j+2)$ for the approximation since the distance between $(i-1, j)$ and Q is shorter than the distance between $(i-1, j+3)$ and Q.

2.6 Approximating u on the boundary

Step 4 is the core part of the algorithm. It allows us to construct a solution on the boundary with third-order accuracy despite the Neumann boundary condition. Let us choose the boundary point R as shown in Figure 3. Suppose the solutions at the extensions P and Q are approximated as u_P and u_Q, respectively, as per Step 3. Let U_P and U_Q denote analytic solutions at P and Q. If we also let αh be the distance between P and R, and βh be the distance between Q and R, the following Taylor expansions are obtained:

$$U_P = U_R - (\alpha h)\frac{\partial}{\partial \nu}U_R + \frac{1}{2}(\alpha h)^2 D_\nu^2 U_R + O(h^3), \tag{2.11}$$

$$U_Q = U_R - (\beta h)\frac{\partial}{\partial \nu}U_R + \frac{1}{2}(\beta h)^2 D_\nu^2 U_R + O(h^3). \tag{2.12}$$

Here $D_\nu^2 u = \nu^T(\mathcal{H}u)\,\nu$, where $\mathcal{H}u$ is the Hessian of u. Due to the Neumann boundary condition given in our problem, we have $\partial U_R/\partial \nu = 0$. Multiplying α^2 in (2.11) and β^2 in (2.12), and subtracting one from another, we get

$$U_R = \frac{\alpha^2 U_Q - \beta^2 U_P}{\alpha^2 - \beta^2} + O(h^3).$$

Since $U_P = u_P + O(h^3)$ and $U_Q = u_Q + O(h^3)$, the formula

$$u_R = \frac{\alpha^2 u_Q - \beta^2 u_P}{\alpha^2 - \beta^2}$$

is a third order accurate approximation of U_R.

To retain a non-negative solution on the boundary, we use a thresholding for the approximated solution as follows:

$$u_R = \max\left(0, \frac{\alpha^2 u_Q - \beta^2 u_P}{\alpha^2 - \beta^2}\right). \tag{2.13}$$

Under the conditions (A1), (A2) and the continuity of the solution, the activation of the above thresholding implies that u_P and u_Q are close to zero. Thus, the effect of thresholding is not significant for the accuracy of the scheme; indeed, the numerical tests in a later section demonstrate that the thresholding (2.13) does not affect the convergence rate of our method.

Remark 3. The algorithm can, in theory, be generalized to three dimensional domains. The only significant difficulty is extending the normal inward and choosing the two extension points, i.e. the three dimensional analogue to Section 2.4. This step is achieved by defining the inner and outer cubes of the boundary point of interest, and choosing the extension points as the intersections between the normal line and the surfaces of the inner and outer cubes.

Chapter 3

Non-negative invariance and stability

3.1 Temporal discretization

We discretize the time derivative by the forward Euler method

$$\frac{\partial u(t_n)}{\partial t} = \frac{u^{n+1} - u^n}{k},$$

where k is the time step of the discretization. While this explicit method suffers from a restrictive stability criterion, $k = O(h^2)$, not seen in implicit methods [3], [30], as we shall see, it is better suited for designing schemes with the non-negative invariance property.

As a shorthand, we write $f^n_{m,i,j}$ in place of $f_m(x_i, y_j, t_n, u^n_{m,i,j})$. Then, the update formula for solving (1.1a) is:

$$u^{n+1}_{m,i,j} = u^n_{m,i,j} + k(D_m \Delta_h u^n_{m,i,j} + f^n_{m,i,j}), \tag{3.1}$$

for $m = 1, 2, \ldots, M$.

3.2 Non-negative invariance

If f is quasi-positive, as we assumed in (A1), the solution to the reaction-diffusion system (1) with non-negative initial conditions remains non-negative for all positive time. When the reaction-diffusion system is discretized, however, the quasi-positivity is not enough to guarantee the non-negativity of the numerical solution. Thus, we introduce a notion of ϵ-*thresholding*, and prove that the numerical solution starting from non-negative initial values, with an appropriately constructed scheme, is non-negative.

Definition 4. Denote $u^{\epsilon,n}_{m,i,j}$ as the *ϵ-thresholded solution* of $u^n_{m,i,j}$ defined as

$$u^{\epsilon,n}_{m,i,j} = \begin{cases} 0 & \text{if } 0 \le u^n_{m,i,j} \le \epsilon/2, \\ \epsilon & \text{if } \epsilon/2 < u^n_{m,i,j} \le \epsilon, \\ u^n_{m,i,j} & \text{otherwise.} \end{cases} \tag{3.2}$$

The following proposition gives a condition on the time step k to guarantee the non-negativity of the solution.

Proposition 3.2.1. *Suppose $u^{\epsilon,n}_{m,i,j} \ge 0$ for all m and (i,j). Then each $u^{\epsilon,n+1}_{m,i,j}$ computed by the formulas* (3.1) *and* (3.2) *is non-negative provided*

$$k \le \frac{u^{\epsilon,n}_{m,i,j}}{|D_m \Delta_h u^{\epsilon,n}_{m,i,j} + f^n_{m,i,j}|} \quad \text{whenever } D_m \Delta_h u^{\epsilon,n}_{m,i,j} + f^n_{m,i,j} < 0, \tag{3.3}$$

for all m and (i,j).

Proof. The update formula (3.1) gives

$$u^{n+1}_{m,i,j} = u^{\epsilon,n}_{m,i,j} + k(D_m \Delta_h u^{\epsilon,n}_{m,i,j} + f^n_{m,i,j}) \ge 0.$$

Applying ϵ-thresholding (3.2) on $u^{n+1}_{m,i,j}$, we have $u^{\epsilon,n+1}_{m,i,j} \ge 0$ as claimed. □

Note how the time step k is *adaptive* due to its dependence on $\Delta_h u^{\epsilon,n}_{m,i,j}$ and $u^{\epsilon,n}_{m,i,j}$. In addition, the Euler method allows us to obtain these conditions for each time step since the solution at time $n+1$ is explicitly predictable based on the information given at time n. In the algorithm, at each time step, we choose the largest k that satisfies the criterion in Proposition 3.2.1 at all grid points (i,j).

It can be shown that the local truncation error upon invoking the ϵ-thresholding (3.2) is

$$\tau_{i,j} = \begin{cases} O(h^2) + O(k) + O(\epsilon/h^2) & \text{if } (i,j) \text{ is regular,} \\ O(h) + O(k) + O(\epsilon/h^2) & \text{if } (i,j) \text{ is irregular.} \end{cases}$$

We choose $\epsilon = O(h^4)$ thereby maintaining the second order in space rate of convergence.

At first glance, it may appear that the time step k may rapidly decrease for increasing n, such that the algorithm gets "stuck" at some time prior to the final time T. We shall later prove that this never happens, in Proposition 4.0.2, when we present the complete algorithm.

3.3 Stability

Next, we derive a condition which guarantees the stability of the scheme, in the sense that there exists a constant K independent of k and n such that

$$\|u_m^{\epsilon,n}\|_\infty := \max_{i,j} |u_{m,i,j}^{\epsilon,n}| \leq K$$

for all $0 \leq nk \leq T$ and $m = 1, 2, \ldots, M$. Let us assume that the solution is non-negative, which is guaranteed by the conditions given in Proposition 3.1, so that the constraint (A2) on reaction term f holds. Since we want to find a stability condition that yields the second-order convergence rate for the algorithm, we assume that $k = Sh^2$ for some constant S. The notion that $k = O(h^2)$ is consistent with the standard stability criteria for time explicit methods for parabolic PDEs.

The following proposition suggests an estimate of this constant S.

Proposition 3.3.1. *The scheme* (3.1) *is stable if*

$$k \leq \frac{h^2}{4 \max_m\{D_m\}} \min_{i,j} a_{i,j}. \tag{3.4}$$

Proof. Throughout this proof we drop ϵ from $u_m^{\epsilon,n}$ for notational convenience. 1. Assume

first that (i,j) is a regular point. We first bound $u^{n+1}_{m,i,j}$ by an affine function of $\|u^n_m\|_\infty$:

$$\begin{aligned} u^{n+1}_{m,i,j} &= (1-4\nu_m)u^n_{m,i,j} + \nu_m(u_{m,i-1,j} + u_{m,i+1,j} + u_{m,i,j-1} + u_{m,i,j+1}) + kf^n_{m,i,j} + O(\epsilon) \\ &\le (1-4\nu_m)\|u^n_m\|_\infty + 4\nu_m\|u^n_m\|_\infty + k(c\|u^n_m\|_\infty + d)^+ + O(\epsilon) \\ &\le (1+kc)\|u^n_m\|_\infty + kd^+ + O(\epsilon) \end{aligned}$$

where $\nu_m = kD_m/h^2$, $g^+ = \max\{0, g\}$. The first inequality holds since $\min_{i,j} a_{i,j} \le 1$ and thus $\nu_m \le 1/4$ from (3.4). Thus $\|u^{n+1}_m\|_\infty \le (1+kc)\|u^n_m\|_\infty + kd^+ + O(\epsilon) \le (1+kc)\|u^n_m\|_\infty + k\Gamma$ for some constant Γ since $\epsilon = O(k^2)$. Note that this holds for all n,

$$\begin{aligned} \|u^n_m\|_\infty &\le (1+kc)^n\|u^0_m\|_\infty + \sum_{l=0}^{n-1}(1+kc)^l k\Gamma \\ &= (1+kc)^n\|u^0_m\|_\infty + \frac{(1+kc)^n - 1}{kc}k\Gamma \\ &\le \exp(ckn)\|u^0_m\|_\infty + \frac{\exp(ckn)-1}{c}\Gamma. \end{aligned}$$

Let $K = \exp(cT)\|u^0_m\|_\infty + (\exp(cT)-1)\Gamma/c$. Then for all k and n such that $0 \le nk \le T$, we have $\|u^n_m\|_\infty \le K$ as desired.

2. Next suppose (i,j) is irregular. Without loss of generality we consider the case of Figure 2.1. Note that $U_P = U_R + O(h^2)$ by applying the Neumann boundary condition to (2.11). Thus we have $u_P = u_R + O(h^2)$ since u_P and u_R are third-order accurate approximations of u_P and u_R, respectively. That is, there exists a constant C_0 such that

$$u_{m,R} = u_{m,P} + C_0 h^2.$$

Also note that $u_{m,P} \le \|u_m\|_\infty$, since $u_{m,P}$ is obtained by a standard second-order interpolation of the numerical solutions. Then, under the assumption (3.4),

$$\begin{aligned} u^{n+1}_{m,i,j} &= \left(1 - \nu_m\left(\frac{2}{a_{i+1,j}} + \frac{2}{a_{i,j-1}}\right)\right)u^n_{m,i,j} + \frac{2\nu_m}{(a_{i+1,j}+1)}\left(\frac{u^n_{m,R}}{a_{i+1,j}} + u^n_{m,i-1,j}\right) \\ &+ \frac{2\nu_m}{(a_{i,j-1}+1)}\left(u^n_{m,i,j+1} + \frac{u^n_{m,B}}{a_{i,j-1}}\right) + kf^n_{m,i,j} + O(\epsilon) \\ &\le \left(1 - \nu_m\left(\frac{2}{a_{i+1,j}} + \frac{2}{a_{i,j-1}}\right)\right)\|u^n_m\|_\infty + \frac{2\nu_m}{a_{i+1,j}}\|u^n_m\|_\infty + \frac{2\nu_m}{a_{i,j-1}}\|u^n_m\|_\infty \\ &+ C_1h^2 + k(c\|u^n_m\|_\infty + d)^+ + O(\epsilon) \\ &\le (1+kc)\|u^n_m\|_\infty + C_2 k, \end{aligned}$$

In the last inequality, the terms $C_1 h^2$ and $O(\epsilon)$ were absorbed into $C_2 k$ via the assumption that $k = Sh^2$ and $\epsilon = Wh^4$. Using the same argument as in the case of regular points, we conclude the stability of the method. □

Remark 5. When $\min_{i,j} a_{i,j}$ is very small, the time step restriction (3.4) makes the scheme computationally expensive. In practice, however, we have found that convergence is attained even with larger time step $k = C^* h^2/(4 \max_m D_m)$ for some constant $C^* > \min_{i,j} a_{i,j}$. The constant C^* should be chosen less than 1 due to the stability condition for regular points. In our numerical example, presented in later sections, an appropriate choice of C^* depends on the geometry of the domain. We used C^* ranging from 10^{-2} to 10^{-1}.

Chapter 4

Complete algorithm and convergence rate analysis

The discretization of the reaction-diffusion system (1), and the way to choose time steps for maintaining the solution to be non-negative invariant and the method to be stable can be summarized in Algorithm 1. The iteration from line 9 to line 14 is a numerical implementation of Proposition 3.1. Note that k^n, the n-th time step, is chosen to be the largest number satisfying both the non-negativity (3.3) and stability (3.4) conditions; See line 16. In the following proposition, we show that the adaptive time steps will not become prohibitively small so as to make the algorithm stuck at some premature time.

Proposition 4.0.2. *Given a final time $0 < T < \infty$, Algorithm 1 converges in finitely many steps.*

Proof. It suffices to prove that the right hand sides of (3.4) and (3.3) are bounded from below by a positive constant, when they are invoked. The former bound is constant so it is clearly bounded from below.

For the latter, assume for the moment that $u^{\epsilon,n}_{m,i,j}$ is positive. Then, by the ϵ-thresholding, we have $u^{\epsilon,n}_{m,i,j} \geq \epsilon$. Furthermore, $|D_m \Delta_h u^{\epsilon,n}_{m,i,j} + f^n_{m,i,j}|$ is bounded from

above since u does not blow up in T. Therefore, the right hand side of (3.3) is bounded from below, as desired.

Finally, if $u^{\epsilon,n}_{m,i,j} = 0$, we argue that $\Delta_h u^{\epsilon,n}_{m,i,j} + f^n_{m,i,j} \geq 0$; This implies that the condition (3.3) will not be invoked. To see this, first note that $\Delta_h u^{\epsilon,n}_{m,i,j} \geq 0$ since all the neighboring grid points have non-negative values. Furthermore, the quasi-positivity of f implies that $f^n_{m,i,j} \geq 0$. Thus we have $\Delta_h u^{\epsilon,n}_{m,i,j} + f^n_{m,i,j} \geq 0$, as desired. □

Algorithm 1: Pseudocode for solving (1.1) on an irregular domain.

Initialization:

Choose $\epsilon > 0$, the thresholding parameter of u, see (3.2);

ϕ signed distance function to Ω, see (2.1);

u^0 (non-negative) initial condition;

foreach $n = 1, 2, 3, \ldots$ **do**

  Compute u^n's on the points where grid lines intersect with the boundary using (2.13);

  Compute $\Delta_h u^n$ using (2.2) and (2.3);

  foreach *Grid node* (i, j) **do**

    Set $k_{m,i,j} = \infty$;

    Compute thresholded solution $u^{\epsilon,n}_{m,i,j}$ using (3.2);

    Set $u^n_{m,i,j} \leftarrow u^{\epsilon,n}_{m,i,j}$;

    if $D_m \Delta_h u^n_{m,i,j} + f^n_{m,i,j} < 0$ **then**

      $k_{m,i,j} \leftarrow u^n_{m,i,j} / |D_m \Delta_h u^n_{m,i,j} + f^n_{m,i,j}|$;

    end

  end

  $k^n \leftarrow \min\{\min_{m,i,j} k_{m,i,j},\ (h^2 \min_{i,j} a_{ij})/(4 \max_m D_m)\}$;

  $u^{n+1}_{m,i,j} \leftarrow u^n_{m,i,j} + k^n (D_m \Delta_h u^n_{m,i,j} + f^n_{m,i,j})$;

end

To avoid having a regular point lie on the zero level set of ϕ, in practice, we perturb ϕ as follows [24]: for all (i,j)

$$\phi_{i,j}^{\kappa} = \begin{cases} \kappa & \text{if } 0 \le \phi_{i,j} \le \kappa \\ -\kappa & \text{if } -\kappa \le \phi_{i,j} < 0 \\ \phi_{i,j} & \text{otherwise.} \end{cases}$$

This perturbation guarantees that each a_{ij} is strictly greater than 0, which results in the algorithm for computing the discrete Laplace operator $\Delta_h u_{i,j}$ in (2.3) being well-defined.

4.1 Convergence rate analysis

To examine the convergence rate of the method, we recall the local truncation error is $O(h^2)$ if (i,j) is regular, and $O(h)$ if (i,j) is irregular, since the algorithm employs $k = O(h^2)$ and $\epsilon = O(h^4)$. Thus the absolute error between the numerical solution and the analytic solution at (i,j) is

$$E_{i,j} = \begin{cases} O(h^2) & \text{if } (i,j) \text{ is regular,} \\ O(h) & \text{if } (i,j) \text{ is irregular.} \end{cases}$$

Let us consider the L^p-norm of the error E for $1 \le p = \infty$. Since the number of irregular grid points is $O(1/h)$ and the number of regular grid points is $O(1/h^2)$, we have the following estimate of the error:

$$\|E\|_p = \left(\sum_{(i,j) \text{ regular}} E_{i,j}^p h^2 + \sum_{(i,j) \text{ irregular}} E_{i,j}^p h^2 \right)^{1/p} \tag{4.1}$$

$$= \left(O\left(\frac{1}{h^2}\right) O(h^2)^p h^2 + O\left(\frac{1}{h}\right) O(h)^p h^2 \right)^{1/p} \tag{4.2}$$

$$= O(h^2 + h^{1+1/p}). \tag{4.3}$$

Thus the algorithm has the second order convergence rate in the L^1-norm and the first order convergence rate in the L^∞-norm. In the numerical tests to be presented in the next section, the convergence rates in L^1- and L^2-norms are second-order. Even in the L^∞-norm, the method shows second order convergence rate for many cases.

Chapter 5

Numerical results

5.1 Convergence test on a circular domain

We consider a single-species ($M = 1$) example of (1.1) in a circular domain $\Omega = \{(x, y)|x^2 + y^2 < R^2\}$ with

$$f_1(x, y, t, u_1) = -au_1 + \exp(-at)(4b^2 D_1(x^2 + y^2)\cos\theta + 4bD_1 \sin\theta),$$
$$u_1^0(x, y) = \cos\theta,$$

where $\theta = \theta(x, y) = b(x^2 + y^2 - R^2)$ and a, b are constants. Then (1.1) has the analytic solution, $u_1^A(x, t) = \exp(-at)\cos(b(x^2 + y^2 - R^2))$.

In the simulations, we chose the time step $k = D/h^2 \times 10^{-2}$, which is much less restrictive than the stability condition (3.4) but appears to be stable in practice, see Remark 5. We used the parameter values $R = 0.3, D_1 = 0.1, a = 0.1, b = 4\pi$ and final time $T = 1$. A box $[-0.5, 0.5]^2$ containing the circular domain Ω, was discretized by $N \times N$ grids, for $N = 50, 100, 200, 400$. Figures 5.1 shows the simulation result of u_1 for $N = 100$.

The convergence rate, r, can be computed as follows:

$$r = \frac{\log(E_1/E_2)}{\log(h_1/h_2)},$$

where E_1 and E_2 are errors between the numerical and the analytic solutions in norms computed with grid size h_1 and h_2. The errors and convergence rates are presented in

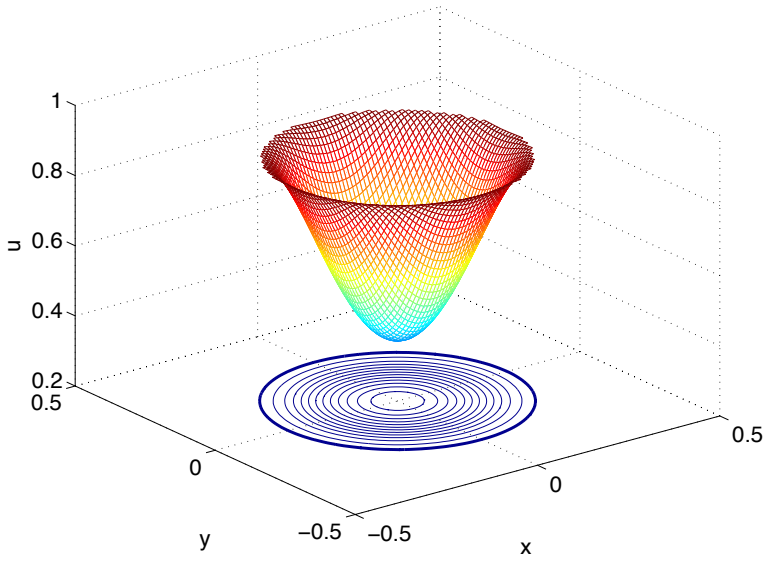

Figure 5.1. The solution u_1 at $T = 1$ on a 100×100 grid.

Table 5.1. Errors and convergence rates of the example simulated in a circular domain for $T = 1$.

Grid size	$\|u - u^A\|_1$	r	$\|u - u^A\|_2$	r	$\|u - u^A\|_\infty$	r
50×50	1.198×10^{-2}	–	2.271×10^{-2}	–	4.733×10^{-2}	–
100×100	2.859×10^{-3}	2.07	5.400×10^{-3}	2.07	1.132×10^{-2}	2.06
200×200	7.089×10^{-4}	2.01	1.337×10^{-3}	2.01	2.880×10^{-3}	1.97
400×400	1.593×10^{-4}	2.15	3.000×10^{-4}	2.16	6.398×10^{-4}	2.17

Table 5.1. As shown in Table 5.1, second-order convergence rates both in L^1-, L^2- and L^∞-norms are achieved.

We also perform the convergence test for a small time step, $T = 0.01$. The convergence rates are summarized in Table 5.2. Note that the convergence rate in the L^1-norm is second order, as predicted by theory.

The error $E = |u_1 - u_1^A|$ is shown in Figure 5.2. Large errors are observed near the boundary as expected. This is due to the fact that the boundary values are not given as the Dirichlet boundary conditions, but approximated using extrapolation of regular points near the boundary with the information of the normal derivative.

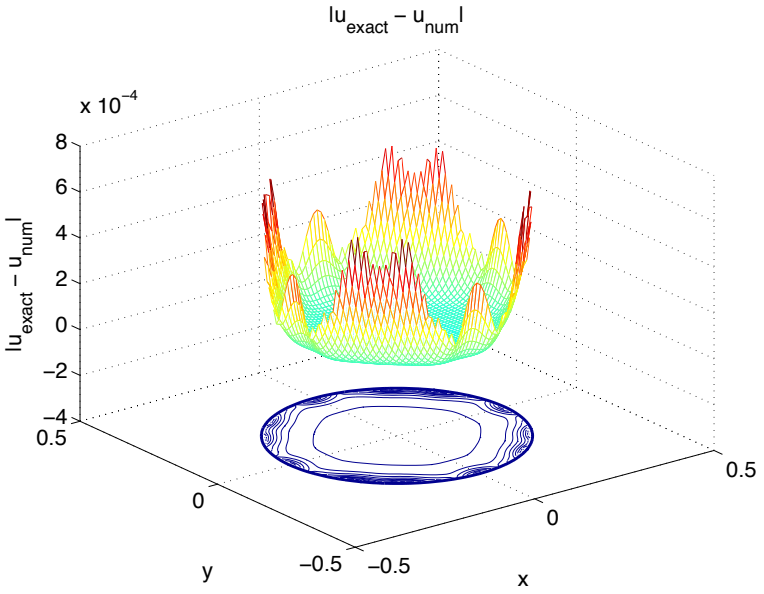

Figure 5.2. The absolute error $|u_1 - u_1^A|$ at $T = 0.01$ on a 100×100 grid.

Table 5.2. Errors and convergence rates of the example simulated in a circular domain for $T = 0.01$.

Grid size	$\|u - u^A\|_1$	r	$\|u - u^A\|_2$	r	$\|u - u^A\|_\infty$	r
50×50	1.330×10^{-4}	–	3.778×10^{-4}	–	2.358×10^{-3}	–
100×100	3.132×10^{-5}	2.09	8.945×10^{-5}	2.08	6.613×10^{-4}	1.83
200×200	7.777×10^{-6}	2.01	2.286×10^{-5}	1.97	2.148×10^{-4}	1.62
400×400	1.757×10^{-6}	2.15	5.026×10^{-6}	2.19	4.357×10^{-5}	2.30

5.2 Convergence test on an annulus

Next, we consider the reaction-diffusion system on an annulus. This type of domain may be useful in modeling biochemical signaling in a cell membrane. Note our method is applicable for domains with arbitrary topology via the level set representation. In this example, $\Omega = \{(x, y) | r_1^2 < x^2 + y^2 < r_2^2\}$, $M = 1$ and

$$f_1(x, y, t, u_1) = -au_1 + \exp(-at)(4b^2 D_1(x^2 + y^2)\cos\theta + 4bD_1 \sin\theta),$$
$$u_1^0(x, y) = 0.2(1 + \cos\theta),$$

where $\theta = \theta(x, y) = b(x^2 + y^2 - r_1^2)$ and $b = \pi/(r_2^2 - r_1^2)$. Then $u_1^A(x, t) = 0.2\exp(-at)(1 + \cos(b(x^2 + y^2 - r_1^2)))$ is the analytic solution to (1.1). Also note that the solution satisfies the Neumann boundary conditions (1.1b) on inner and outer boundaries. The parameter values are chosen as $r_1 = 0.2$, $r_2 = 0.4$, $D_1 = 0.1$ and $a = 0.1$. We solve the problem on a $N \times N$ grid until $T = 0.01$, for $N = 50, 100, 200$, and 400. The result with $N = 100$ is shown in Figure 5.3.

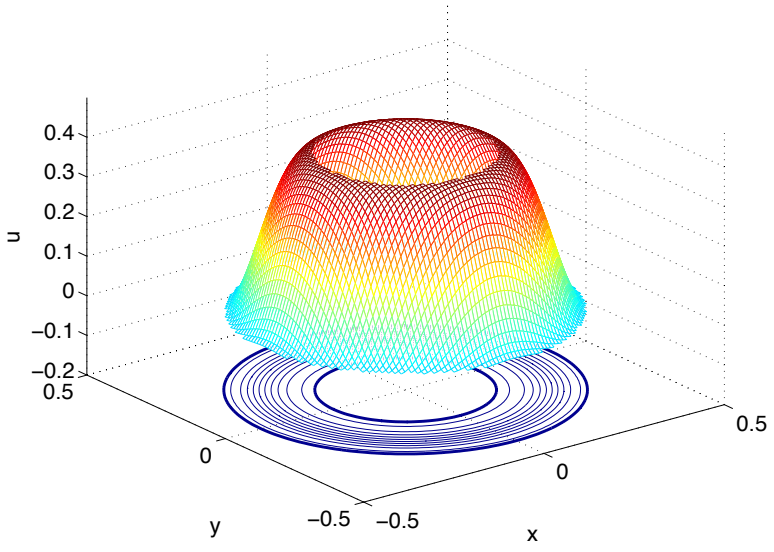

Figure 5.3. The solution u_1 on a 100×100 grid.

As in the previous example, we performed the convergence rate analysis by calculating the absolute errors in L^1-, L^2- and L^∞-norms. The method solves the problem with second order convergence rate in all norms as shown in Table 5.3.

Table 5.3. Errors and convergence rates of the example simulated in a circular domain with a hole.

Grid size	$\|u-u^A\|_1$	r	$\|u-u^A\|_2$	r	$\|u-u^A\|_\infty$	r
50×50	3.467×10^{-4}	–	7.647×10^{-4}	–	4.421×10^{-3}	–
100×100	6.770×10^{-5}	2.36	1.421×10^{-4}	2.43	9.167×10^{-4}	2.27
200×200	1.595×10^{-5}	2.09	3.298×10^{-5}	2.11	2.236×10^{-4}	2.04
400×400	3.653×10^{-6}	2.13	7.581×10^{-6}	2.12	5.154×10^{-5}	2.12

5.3 Convergence test of a two-species system

To see how a multi-species system affects the convergence rate of our method, we consider an example of (1.1) in a circular domain $\Omega = \{(x,y)|x^2+y^2 < R^2\}$ with $M = 2$ and

$$\begin{aligned} f_1(x,y,t,u_1,u_2) &= -au_2 + \exp(-at)(4b^2D_1(x^2+y^2)\cos\theta + 4bD_1\sin\theta), \\ f_2(x,y,t,u_1,u_2) &= -au_1 + \exp(-at)(4b^2D_2(x^2+y^2)\cos\theta + 4bD_2\sin\theta), \\ u_1^0(x,y) &= \cos\theta, \\ u_2^0(x,y) &= \cos\theta, \end{aligned}$$

where $\theta = \theta(x,y) = b(x^2+y^2-R^2)$ and a, b are constants. Then (1.1) has the analytic solution, $u_1^A(x,t) = \exp(-at)\cos(b(x^2+y^2-R^2))$ and $u_2^A(x,t) = \exp(-at)\cos(b(x^2+y^2-R^2))$.

In the test, we chose the parameter values $R = 0.3, D_1 = 0.1, D_2 = 0.05, a = 0.1, b = 4\pi$ and final time $T = 0.01$. The test result in Table 5.4 shows the second order convergence rate in L^1 and L^2 as in the single species example.

In order to test whether our method is robust when two diffusion rates are in different orders of magnitude, we simulated the system with $D_1 = 0.1$ and $D_2 = 0.001$. As shown in Table 5.5, the convergence rate in L^1-norm is maintained to be second order despite a drop of the convergence rate of u_2 in L^∞-norm.

Table 5.4. Errors and convergence rates of the example with multiple species simulated in a circular domain, when $D = (0.1, 0.05)$.

Grid size	$\|u_1 - u_1^A\|_1$	r	$\|u_1 - u_1^A\|_2$	r	$\|u_1 - u_1^A\|_\infty$	r
50×50	1.330×10^{-4}	–	3.779×10^{-4}	–	2.358×10^{-3}	–
100×100	3.133×10^{-5}	2.09	8.947×10^{-5}	2.08	6.614×10^{-4}	1.83
200×200	7.779×10^{-6}	2.01	2.287×10^{-5}	1.97	2.148×10^{-4}	1.62
400×400	1.758×10^{-6}	2.15	5.027×10^{-6}	2.19	4.358×10^{-5}	2.30

Grid size	$\|u_2 - u_2^A\|_1$	r	$\|u_2 - u_2^A\|_2$	r	$\|u_2 - u_2^A\|_\infty$	r
50×50	6.775×10^{-5}	–	2.188×10^{-4}	–	1.602×10^{-3}	–
100×100	1.566×10^{-5}	2.11	5.130×10^{-5}	2.09	4.610×10^{-4}	1.80
200×200	3.879×10^{-6}	2.01	1.312×10^{-5}	1.97	1.518×10^{-4}	1.60
400×400	8.777×10^{-7}	2.14	2.873×10^{-6}	2.19	3.045×10^{-5}	2.32

Table 5.5. Errors and convergence rates of the example with multiple species simulated in a circular domain, when $D = (0.1, 0.001)$.

Grid size	$\|u_1 - u_1^A\|_1$	r	$\|u_1 - u_1^A\|_2$	r	$\|u_1 - u_1^A\|_\infty$	r
50×50	1.331×10^{-4}	–	3.780×10^{-4}	–	2.359×10^{-3}	–
100×100	3.133×10^{-5}	2.09	8.949×10^{-5}	2.08	6.615×10^{-4}	1.83
200×200	7.781×10^{-6}	2.01	2.287×10^{-5}	1.97	2.149×10^{-4}	1.62
400×400	1.758×10^{-6}	2.15	5.029×10^{-6}	2.19	4.358×10^{-5}	2.30

Grid size	$\|u_2 - u_2^A\|_1$	r	$\|u_2 - u_2^A\|_2$	r	$\|u_2 - u_2^A\|_\infty$	r
50×50	1.745×10^{-6}	–	9.745×10^{-6}	–	9.683×10^{-5}	–
100×100	3.747×10^{-7}	2.22	2.659×10^{-6}	1.87	3.759×10^{-5}	1.37
200×200	8.475×10^{-8}	2.14	7.183×10^{-7}	1.89	1.697×10^{-5}	1.15
400×400	1.779×10^{-8}	2.25	1.487×10^{-8}	2.27	3.935×10^{-6}	2.11

5.4 A multi-species system in a star-shaped cell

Biochemical reaction between species **A** and **B** that creates species **C** can be described by the following chemical equation:

$$\mathbf{A} + \mathbf{B} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathbf{C}. \tag{5.1}$$

Here k_1 and k_2 represent the forward and backward reaction rates, respectively. The above equation explains many biological processes including receptor-ligand binding [33], protein complex formation [1] and gene expression [11]. Let u_1, u_2 and u_3 denote the concentration levels of $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, respectively. To model (5.1) in a diffusive and bounded environment such as cell and cell membrane, we use the reaction-diffusion system (1) with $M = 3$ and

$$f(u) = (f_1(u), f_2(u), f_3(u)) = (k_2u_3 - k_1u_1u_2,\ k_2u_3 - k_1u_1u_2,\ k_1u_1u_2 - k_2u_3).$$

Suppose that the system is defined in the domain Ω enclosed by a curve $(x(\theta), y(\theta))$ parametrized by angle in radians:

$$x(\theta) = \frac{1}{15}(5 + \sin 5\theta)\cos\theta, \qquad y(\theta) = \frac{1}{15}(5 + \sin 5\theta)\sin\theta.$$

The reaction rates and diffusion rates are $(D_1, D_2, D_3) = (0.1, 0.001, 0.05)$ and $(k_1, k_2) = (10, 0.1)$. The initial conditions are,

$$\begin{aligned} u_1^0 &= 1 + \cos(3.125\pi(x^2 + y^2 - (0.4)^2)), \\ u_2^0 &= 1 - \cos(3.125\pi(x^2 + y^2 - (0.4)^2)), \\ u_3^0 &= 0.1(1 + \cos(3.125\pi(x^2 + y^2 - (0.4)^2))). \end{aligned}$$

The problem is solved on a $N \times N$ grid until $T = 0.01$, for $N = 50, 100, 200$, and 400. In Figures 5.4, 5.5 and 5.6 we show the results for $N = 100$.

Since the analytical solution is not known for this example, we estimate the convergence rate by comparing numerical solutions using different grid sizes:

$$r = \log_2\left(\frac{\|u^h - u^{h/2}\|}{\|u^{2h} - u^h\|}\right),$$

where $u^{h/2}$, u^h and u^{2h} are numerical solutions with grid sizes $h/2$, h and $2h$, respectively. The result are presented in Table 5.6, which shows the second-order convergence of the method in the L^1- and L^2-norms.

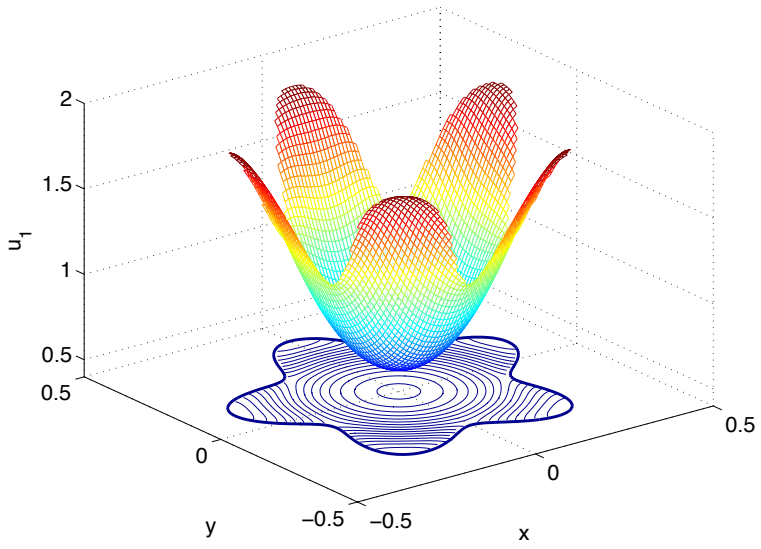

Figure 5.4. The numerical solution of u_1 on a 100×100 grid.

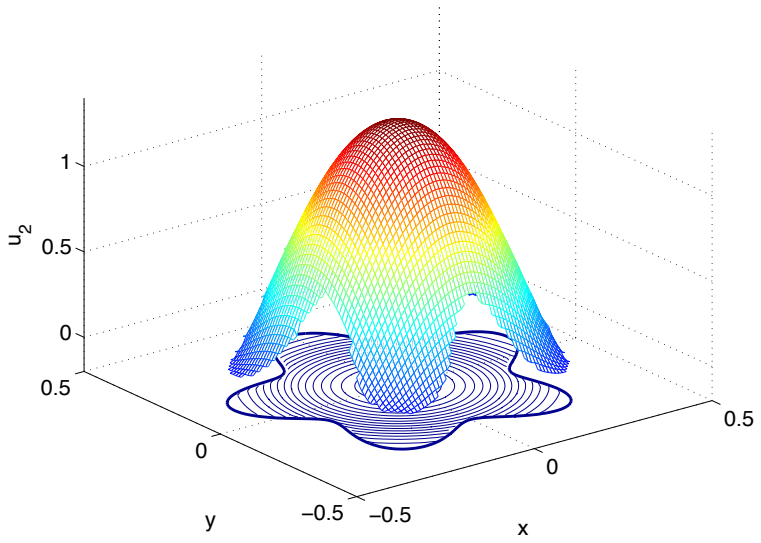

Figure 5.5. The numerical solution of u_2 on a 100×100 grid.

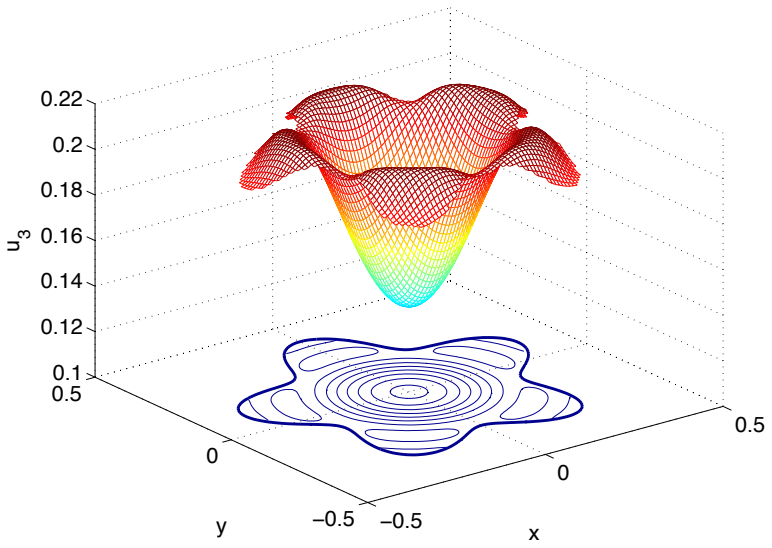

Figure 5.6. The numerical solution of u_3 on a 100×100 grid.

Table 5.6. Errors and convergence rates of the example simulated in a star-shaped domain.

Grid size	$\|u_1^h - u_1^{h/2}\|_1$	r	$\|u_1^h - u_1^{h/2}\|_2$	r	$\|u_1^h - u_1^{h/2}\|_\infty$	r
100×100	8.099×10^{-4}	–	2.488×10^{-3}	–	1.914×10^{-2}	–
200×200	1.764×10^{-4}	2.20	5.563×10^{-4}	2.16	4.360×10^{-3}	2.13
400×400	4.529×10^{-5}	1.96	1.447×10^{-4}	1.94	1.228×10^{-3}	1.83

Grid size	$\|u_2^h - u_2^{h/2}\|_1$	r	$\|u_2^h - u_2^{h/2}\|_2$	r	$\|u_2^h - u_2^{h/2}\|_\infty$	r
100×100	1.318×10^{-3}	–	7.793×10^{-3}	–	8.292×10^{-2}	–
200×200	2.364×10^{-4}	2.48	2.009×10^{-3}	1.96	2.940×10^{-2}	1.50
400×400	4.868×10^{-5}	2.28	4.998×10^{-4}	2.01	1.095×10^{-2}	2.45

Grid size	$\|u_3^h - u_3^{h/2}\|_1$	r	$\|u_3^h - u_3^{h/2}\|_2$	r	$\|u_3^h - u_3^{h/2}\|_\infty$	r
100×100	7.453×10^{-5}	–	3.046×10^{-4}	–	3.380×10^{-3}	–
200×200	1.361×10^{-5}	2.45	5.460×10^{-5}	2.48	6.192×10^{-4}	1.43
400×400	3.160×10^{-6}	2.11	1.371×10^{-5}	1.99	1.882×10^{-4}	1.72

5.5 EphA2/Ephrin-A1 signaling

The temporal evolution of geometric distribution of EphA2 receptors is known to be important because their lateral transport, which is observed in many cell lines, is significantly correlated with the tissue invasion potential. The lateral transport of EphA2 is activated by formulating a complex with Ephrin-A1 ligands. Effects of the spatial distribution of EphA2/Ephrin-A1 complexes to their downstream signaling can be found in [34]. An interesting observation is that the reorganization of actin cytoskeleton may regulate the geometric distribution of EphA2/Ephrin-A1 complexes. In turn, the downstream signaling pathways of EphA2/Ephrin-A1 involving the Rho family of GTPases such as RhoA, Rac1 and Cdc42 control the organization of the actin cytoskeleton [10]. Among several proposed crosstalk schemes of Rho GTPases, the interaction mechanism in Figure 5.7, which we will be using for modeling, is consistent with the bistability of RhoA, Rac1 and Cdc42 [9], [12].

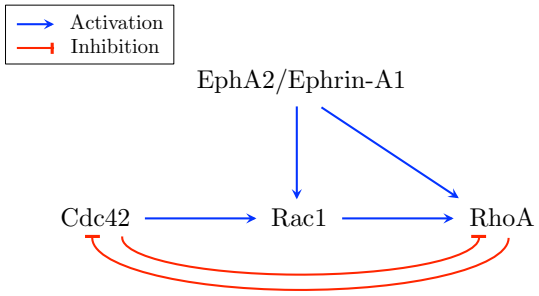

Figure 5.7. An EphA2/Ephrin-A1 downstream signaling mechanism and the crosstalk between RhoA, Rac1 and Cdc42.

The following biochemical equations describe the signaling mechanism in Figure 5.7:

$$\textbf{Ephrin-A1} + \textbf{EphA2} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} p\textbf{EphA2/Ephrin-A1}$$

$$p\textbf{EphA2/Ephrin-A1} + \textbf{RhoA} \underset{k_4}{\overset{k_3}{\rightleftharpoons}} p\textbf{EphA2/Ephrin-A1/RhoA}$$

$$p\textbf{EphA2/Ephrin-A1/RhoA} \underset{k_6}{\overset{k_5}{\rightleftharpoons}} p\textbf{EphA2/Ephrin-A1} + p\textbf{RhoA}$$

$$p\textbf{EphA2/Ephrin-A1} + \textbf{Rac1} \underset{k_8}{\overset{k_7}{\rightleftharpoons}} p\textbf{EphA2/Ephrin-A1/Rac1}$$

$$p\textbf{EphA2/Ephrin-A1/Rac1} \underset{k_{10}}{\overset{k_9}{\rightleftharpoons}} p\textbf{EphA2/Ephrin-A1} + p\textbf{Rac1}$$

$$p\textbf{Cdc42} + \textbf{Rac1} \underset{k_{12}}{\overset{k_{11}}{\rightleftharpoons}} p\textbf{Cdc42/Rac1}$$

$$p\textbf{Cdc42/Rac1} \underset{k_{14}}{\overset{k_{13}}{\rightleftharpoons}} p\textbf{Cdc42} + p\textbf{Rac1}$$

$$p\textbf{Rac1} + \textbf{RhoA} \underset{k_{16}}{\overset{k_{15}}{\rightleftharpoons}} p\textbf{Rac1/RhoA}$$

$$p\textbf{Rac1/RhoA} \underset{k_{18}}{\overset{k_{17}}{\rightleftharpoons}} p\textbf{Rac1} + p\textbf{RhoA}$$

$$p\textbf{Cdc42} + p\textbf{RhoA} \underset{k_{20}}{\overset{k_{19}}{\rightleftharpoons}} p\textbf{Cdc42}/p\textbf{RhoA}$$

$$p\textbf{Cdc42}/p\textbf{RhoA} \underset{k_{22}}{\overset{k_{21}}{\rightleftharpoons}} \textbf{Cdc42} + \textbf{RhoA}$$

Let $u_1 = [\textbf{EphA2}]$, $u_2 = [\textbf{RhoA}]$, $u_3 = [p\textbf{RhoA}]$, $u_4 = [\textbf{Rac1}]$, $u_5 = [p\textbf{Rac1}]$, $u_6 = [\textbf{Cdc42}]$, $u_7 = [p\textbf{Cdc42}]$, $u_8 = [p\textbf{EphA2/Ephrin-A1}]$, $u_9 = [p\textbf{Rac1/RhoA}]$, $u_{10} = [p\textbf{Cdc42/Rac1}]$, $u_{11} = [p\textbf{Cdc42}/p\textbf{RhoA}]$, $u_{12} = [p\textbf{EphA2/Ephrin-A1/RhoA}]$, $u_{13} = [p\textbf{EphA2/Ephrin-A1/Rac1}]$ and $v = [\textbf{Ephrin-A1}]$, where $[\textbf{A}]$ denotes the concentration level of the chemical species **A**. Given the experimental data of $v = [\textbf{Ephrin-A1}]$, the reaction-diffusion system (1.1) with $M = 13$ and the reaction terms:

$$f_1(u) = -k_1 v u_1 + k_2 u_8$$

$$f_2(u) = -k_3 u_8 u_2 + k_4 u_{12} - k_{15} u_5 u_2 + k_{16} u_9 + k_{21} u_{11} - k_{22} u_6 u_2$$

$$f_3(u) = k_5 u_{12} - k_6 u_8 u_3 + k_{17} u_9 - k_{18} u_5 u_3 - k_{19} u_7 u_3 + k_{20} u_{11}$$

$$f_4(u) = -k_7 u_8 u_4 + k_8 u_{13} - k_{11} u_7 u_4 + k_{12} u_{10}$$

$$f_5(u) = k_9 u_{13} - k_{10} u_8 u_5 + k_{13} u_{10} - k_{14} u_7 u_5 - k_{15} u_5 u_2 + k_{16} u_9 + k_{17} u_9 - k_{18} u_5 u_3$$

$$f_6(u) = k_{21} u_{11} - k_{22} u_6 u_2$$

$$f_7(u) = -k_{11} u_7 u_4 + k_{12} u_{10} + k_{13} u_{10} - k_{14} u_7 u_5 - k_{19} u_7 u_3 + k_{20} u_{11}$$

$$
\begin{aligned}
f_8(u) &= k_1 v u_1 - k_2 u_8 - k_3 u_8 u_2 + k_4 u_{12} + k_5 u_{12} - k_6 u_8 u_3 - k_7 u_8 u_4 + k_8 u_{13} \\
&\quad + k_9 u_{13} - k_{10} u_8 u_5 \\
f_9(u) &= k_{15} u_5 u_2 - k_{16} u_9 - k_{17} u_9 + k_{18} u_5 u_3 \\
f_{10}(u) &= k_{11} u_7 u_4 - k_{12} u_{10} - k_{13} u_{10} + k_{14} u_7 u_5 \\
f_{11}(u) &= k_{19} u_7 u_3 - k_{20} u_{11} - k_{21} u_{11} + k_{22} u_6 u_2 \\
f_{12}(u) &= k_3 u_8 u_2 - k_4 u_{12} - k_5 u_{12} + k_6 u_8 u_3 \\
f_{13}(u) &= k_7 u_8 u_4 - k_8 u_{13} - k_9 u_{13} + k_{10} u_8 u_5
\end{aligned}
$$

models the spatio-temporal dynamics of the signaling in Figure 5.7. Here we choose the diffusion rate $D = (D_1, \ldots, D_{13})$ with $D_1, \cdots, D_7$ to be 10^{-4} and $D_8, \cdots, D_{13}$ to be 5×10^{-5}. Reaction rates are given as $k = (k_1, \cdots, k_{22})$, where k_1, k_5, k_9, k_{21} are 1, $k_3, k_7, k_{11}, k_{13}, k_{15}, k_{17}, k_{19}$ are 10^{-1}, and k_{2n}'s are 10^{-5} for $n = 1, \cdots, 11$. We solved the reaction-diffusion system in a circular domain $\Omega = \{(x, y) | x^2 + y^2 < (1/3)^2\}$ until $T = 10$, with initial values $u^0(x) = (u_1^0(x), \cdots, u_{13}^0(x)) = (10, 5, 0.1, 5, 0.1, 5, 0.1, 0.1, 1, 1, 1, 1, 1)$ for all $x \in \Omega$. Note that the concentration level of Ephrin-A1, v, which is given as data, works as an input for the reaction-diffusion system. As the spatial distribution of v evolves over time, the high concentration regions of pRac1, activated by pEphA2/Ephrin-A1, are collocated with those of Ephrin-A1 as shown in Figure 5.8. Due to the clustering behavior of Ephrin-A1, the high concentration regions of pRac1 are centralized despite the diffusion effect. On the other hand the concentration level of Cdc42 near the center of the cell decreases over time since RhoA, activated by pEphA2/Ephrin-A1, inhibits Cdc42. These demonstrate the effects of the spatio-temporal dynamics of Ephrin-A1 to that of its downstream signaling molecules.

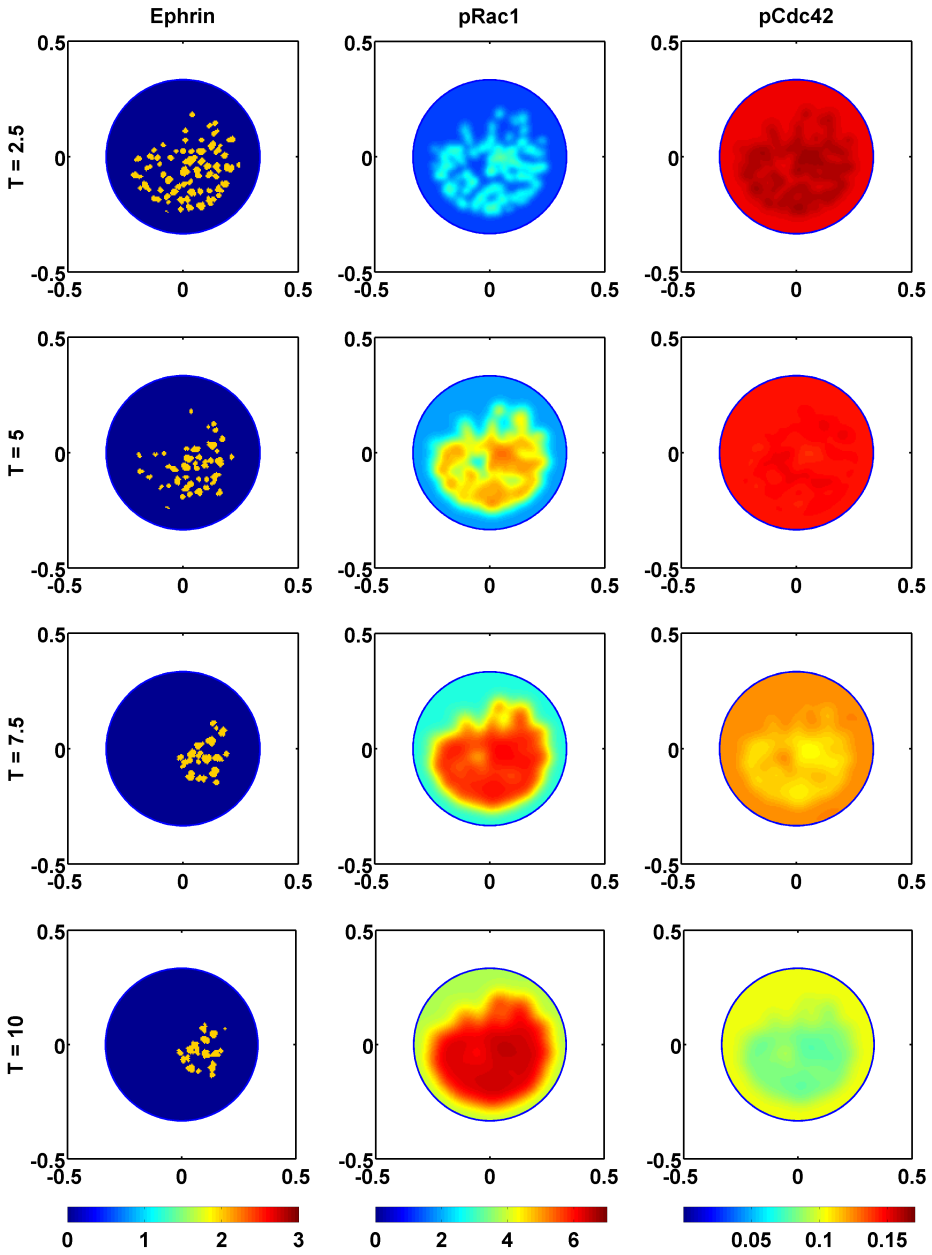

Figure 5.8. The first column shows the experimental data of Ephrin-A1 (courtesy of Jay T. Groves). The second and third columns are numerical solutions of pRac1 and pCdc42, respectively on a 200×200 grid. Rows represent the results at $T = 2.5$, $T = 5$, $T = 7.5$ and $T = 10$ from top to bottom.

Chapter 6

Conclusion

We have proposed a stable second-order accurate method that preserves the non-negativity of the solution with the Neumann boundary conditions on irregular domains. In the spatial discretization, the slopes of normal vectors allow us to extrapolate the solutions inside the domain to approximate the solution on the boundary with third order accuracy. The algorithm employs adaptive time steps, and is well-posed due to the quasi-positivity (A1), to enforce the numerical solution of (1.1) to remain in $[0, +\infty)^M$. In addition, we proved that the method is stable. Numerical examples demonstrates that the method has second-order convergence rate in the L^1-norm, and is robust to domain geometry and to different scales of diffusion rates.

The test involving the EphA2/Ephrin-A1 signaling in a tumor cell, for instance, suggests that our method may serve as a useful simulation tool for testing biological and physical hypotheses in biochemical network in cells.

Our spatial discretization may be applicable to the moving domain problem. In the future, we plan to embed our algorithm within a moving interface using the level set method [27], [28], [37]. The three-dimensional problem and the surface diffusion problem are interesting topics to pursue further, applying our stable and non-negative invariant algorithm.

References

[1] K. Amonlirdviman, N. A. Khare, D. R. P. Tree, W.-S. Chen, J. D. Axelrod, and C. J. Tomlin, "Mathematical modeling of planar cell polarity to understand domineering nonautonomy," *Science*, vol. 307, pp. 423–426, January 2005.

[2] L.-T. Cheng and Y.-H. R. Tsai, "Redistancing by flow of time dependent Eikonal equation," *J. Comput. Phys.*, vol. 227, no. 8, pp. 4002–4017, 2008.

[3] J. Crank and P. Nicolson, "A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type," *P. Camb. Philol. Soc.*, vol. 43, pp. 50–67, 1947.

[4] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, "A non-oscilatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method)," *J. Comput. Phys.*, vol. 152, pp. 457–492, 1999.

[5] R. A. Fisher, "The wave of advance of advantageous genes," *Ann. Eugenic.*, vol. 7, pp. 355–369, 1937.

[6] A. L. Fogelson and J. P. Keener, "Immersed interface methods for Neumann and related problems in two and three dimensions," *SIAM J. Sci. Comput.*, vol. 22, no. 5, pp. 1630–1654, 2000.

[7] F. Gibou and R. P. Fedkiw, "A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the Stefan problem," *J. Comput. Phys.*, vol. 202, pp. 577–601, 2005.

[8] F. Gibou, R. P. Fedkiw, L.-T. Cheng, and M. Kang, "A second-order-accurate symmetric discretization of the Poisson equation on irregular domains," *J. Comput. Phys.*, vol. 176, pp. 205–227, 2002.

[9] E. Giniger, "How do Rho family GTPases direct axon growth and guidance? a proposal relating signaling pathways to growth cone mechanics," *Differentiation*, vol. 70, pp. 385–396, 2002.

[10] A. Hall, "Rho GTPases and the actin cytoskeleton," *Science*, vol. 279, pp. 509–514, 1998.

[11] J. Hasty, J. Pradines, M. Dolnik, and J. J. Collins, "Noise-based switches and amplifiers for gene expression," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 5, pp. 2075–2080, 2000.

[12] A. Jilkine, A. F. M. Marée, and L. Edelsetin-Keshet, "Mathematical model for spatial segregation of the Rho-family GTPases based on inhibitory crosstalk," *B. Math. Biol.*, vol. 69, pp. 1943–1978, 2007.

[13] H. Johansen and P. Colella, "A Cartesian grid embedded boundary method for Poisson's equation on irregular domains," *J. Comput. Phys.*, vol. 147, pp. 60–85, 1998.

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

[15] B. N. Kholodenko, "Cell-signalling dynamics in time and space," *Nat. Rev. Mol. Cell Biol.*, vol. 7, pp. 165–176, 2006.

[16] R. Klajn, M. Fialkowski, I. T. Bensemann, A. Bitner, C. J. Campbell, K. Bishop, S. Smoukov, and B. A. Grzybowski, "Multicolour micropatterning of thin films of dry gels," *Nat. Mater.*, vol. 3, pp. 729–735, 2004.

[17] A. J. Koch and H. Meinhardt, "Biological pattern formation: from basic mechanisms to complex structures," *Rev. Mod. Phys.*, vol. 66, no. 4, pp. 1481–1507, October 1994.

[18] K.-J. Lee, W. D. McCormick, J. E. Pearson, and H. L. Swinney, "Experimental observation of self-replicating spots in a reaction-diffusion system," *Nature*, vol. 369, pp. 215–218, 1994.

[19] R. J. Leveque and Z. Li, "The immersed interface method for Elliptic equation with discontinuous constants and singular sources," *SIAM J. Numer. Anal.*, vol. 31, no. 4, pp. 1019–1044, 1994.

[20] J. H. Lightboune, III and R. H. Martin, Jr., "Relatively continuous nonlinear perturbations of analytic semigroups," *Nonlinear Anal.-Theor.*, vol. 1, no. 3, pp. 277–292, 1977.

[21] R. H. Martin, Jr., "Nonlinear perturbations of linear evolution systems," *J. Math. Soc. Jpn.*, vol. 29, no. 2, pp. 233–252, 1977.

[22] ——, "Abstract functional differential equations and reaction-diffusion systems," *T. Am. Math. Soc.*, vol. 321, no. 1, pp. 1–44, 1990.

[23] P. McCorquodale, P. Colella, and H. Johansen, "A Cartesian grid embedded boundary method for the heat equation on irregular domains," *J. Comput. Phys.*, vol. 173, pp. 620–635, 2001.

[24] C. Min and F. Gibou, "Geometric integration over irregular domains with application to level-set methods," *J. Comput. Phys.*, vol. 226, pp. 1432–1443, 2007.

[25] K. W. Morton and D. F. Mayers, *Numerical Solution of Partial Differential Equations*. Cambridge University Press, New York, 1995.

[26] J. D. Murray, *Mathematical Biology II: Spatial Models and Biomedical Applications*, 3rd ed. Springer, New York, 2003.

[27] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.

[28] S. J. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, 1st ed. Springer, New York, 2002.

[29] J. Papac, F. Gibou, and C. Ratsch, "Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions," *J. Comput. Phys.*, vol. 229, pp. 875–889, 2010.

[30] D. W. Peaceman and H. H. Rachford Jr., "The numerical solution of parabolic and elliptic differential equations," *J. Soc. Ind. Appl. Math.*, vol. 3, no. 1, pp. 28–41, 1955.

[31] M. Pierre, "Global existence in reaction-diffusion systems with control of mass: a survey," *Milan J. Math.*, vol. 78, pp. 417–455, 2010.

[32] M. Pierre and D. Schmitt, "Blowup in reaction-diffusion systems with dissipation of mass," *SIAM Rev.*, vol. 42, no. 1, pp. 93–106, 2000.

[33] S. Y. Qi, J. T. Groves, and A. K. Chakraborty, "Synaptic pattern formation during cellular recognition," *Proc. Natl. Acad. Sci. USA*, vol. 98, no. 12, pp. 6548–6553, 2001.

[34] K. Salaita, P. M. Nair, R. S. Petit, R. M. Neve, D. Das, J. W. Gray, and J. T. Groves, "Restriction of receptor movement alters cellular response: physical force sensing by EphA2," *Science*, vol. 327, pp. 1380–1385, 2010.

[35] P. Schwartz, D. Adalsteinsson, P. Colella, A. P. Arkin, and M. Onsum, "Numerical computation of diffusion on a surface," *Proc. Natl. Acad. Sci. USA*, vol. 102, no. 32, pp. 11 151–11 156, August 2005.

[36] P. Schwartz, M. Barad, P. Colella, and T. Ligocki, "A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions," *J. Comput. Phys.*, vol. 211, pp. 531–550, 2006.

[37] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 2nd ed. Cambridge University Press, New York, 1999.

[38] C.-W. Shu and S. Osher, "Efficient implementation of essentially non-oscillatory shock-capturing schemes," *J. Comput. Phys.*, vol. 77, pp. 439–471, 1988.

[39] ——, "Efficient implementation of essentially non-oscillatory shock-capturing schemes, II," *J. Comput. Phys.*, vol. 83, pp. 32–78, 1989.

[40] W. Strychalski, D. Adalsteinsson, and T. Elston, "A cut-cell method for simulating spatial models of biochemical reaction networks in arbitrary geometries," *Comm. App. Math. Comp. Sci.*, vol. 5, no. 1, pp. 31–53, 2010.

[41] Y.-H. R. Tsai, "Rapid and accurate computation of the distance function using grids," *J. Comput. Phys.*, vol. 178, pp. 175–195, 2002.

[42] A. M. Turing, "The chemical basis of morphogenesis," *Philos. T. Roy. Soc. B*, vol. 237, pp. 37–72, 1952.

[43] C. W. Wolgemuth and M. Zajac, "The moving boundary node method: a level set-based, finite volume algorithm with applications to cell motility," *J. Comput. Phys.*, vol. 229, pp. 7287–7308, 2010.