UNIVERSITY OF CALIFORNIA,
IRVINE

Numerical Methods for Reaction Diffusion Systems in High Spatial Dimensions

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematics

by

Dongyong Wang

Dissertation Committee:
Professor Qing Nie, Chair
Professor Frederic Y.-M. Wan
Professor Long Chen

2014

To my parents,
Haiou He and Jianzhong Wang,
Without whom I would not succeed in anything

Table of Contents

List of Figures

Acknowledgments

I would like to thank my advisor, Dr. Qing Nie, for all the help during my time as a graduate student. From him, I have learned how to be a computational scientist, numerical analysist, and an independent researcher. Without his support, I would not continue my graduate study. For these lessons and more, I will always be grateful.

I am grateful to my collaborators, Dr. Xing Dai and Dr. Briana Lee Altman, who always helped me better understand the biological problems and for all their hard work.

I want to express my appreciation to my dissertation committee: Dr. Frederic Wan and Dr. Long Chen. It has always been a pleasure discussing my projects with you, I appreciate all the help. Moreover, they are not only my committee members. Dr. Frederic Wan signed the offer letter and brought me to UC Irvine; Dr. Long Chen taught me a bunch of lessons related to numerical methods.

I want to express my gratitude to all the current and past members of the Nie lab for all their help: Jeremy Ovadia, Jinzhi Lei, Catherine Ta, Seth Figueroa, Tao Peng, Alexander Gord, Huijing Du, Weitao Chen, Tian Hong, Likun Zheng, Qixuan Wang, Bill Holmes, Meng Chen, Su Zhao, Lei Zhang, Jiajun Zhang, Jiang Xie, Hsiao-Mei Lu, Scott Christley, Liming Wang, Anna Cai, Zhenzhen Zheng and Jon Lo. Having lunch them was always wonderful.

I would like to thank the School of Physical Sciences, the Department of Mathematics, the Mathematical, Computational, and Systems Biology graduate gateway program, and the Center for complex Biological Systems for all their help.

Curriculum Vitae

Dongyong Wang

EDUCATION

Doctor of Philosophy in Mathematics **2014**
University of California, Irvine *Irvine, California, USA*

Master of Science in Mathematics **2012**
University of California, Irvine *Irvine, California, USA*

Bachelor of Science in Mathematics **2009**
Tsinghua University *Beijing, China*

RESEARCH EXPERIENCE

Graduate Research Assistant June 2009 – June 2014
University of California, Irvine *Irvine, California*

HONORS AND AWARDS

Center for Complex Biological Systems Opportunity Award 2011,2012

JOURNAL PUBLICATIONS J. Lei, D. Wang, Y. Song, Q. Nie, and Y. Wan, "Robustness of Morphogen Gradients with "Bucket Brigade" Transport through membrane-associated Non-receptor", *Discrete and Continuous Dynamical Systems-B*, 18(3), 2013.

D. Wang, L. Zhang, and Q. Nie, " Array-representation Integration Factor Method for High-dimensional Systems", *Journal of Computational Physics*, 258, 2014.

D. Wang, W. Chen, and Q. Nie, "Semi-implicit Integration Factor Methods on Sparse Grid", *Manuscript in preparation.*

Abstract

Numerical Methods for Reaction Diffusion Systems in High Spatial Dimensions

DISSERTATION

By

Dongyong Wang

Doctor of Philosophy in Mathematics

University of California, Irvine, 2014

Professor Qing Nie, Chair

Reaction diffusion equations are widely used to model biological phenomena and in some situation, the spatial dimension can be much larger. Numerical efficiently solving high-dimensional reaction-diffusion equations is a huge challenge. To solve the high-dimensional equation, the “curse of dimensions” has to be dealt with. Also, an efficiently time integration method is needed to solve the afterwords time dependent problem. The sparse grid technique can deal with the problem, and the semi-implicit integration factor method can handles the second one. The combination of two methods will be an efficient method to solve high-dimensional reaction-diffusion equations.

In Chapter 2 we present one application of reaction-diffusion equations. In Chapter 3 we study the hair follicle cycle. In Chapter 4 we present an numerical method to solve high-dimensional PDE systems on regular grids. The method is numerically tested to be efficient. In Chapter 5, we present how to combine the sparse grid with semi-implicit integration

factor method, finally construct a new method to solve high-dimensional reaction-diffusion equations efficiently.

Chapter 1

Introduction

In biology, reaction-diffusion equations are widely used. For example, the morphogen gradient, which is a key factor in embryo development, is believed to be formed by the diffusion process. Also, it can describe the the population density in certain region, e.g. diffusive logistic equation. In these applications, the spatial dimension of the equation is usually less or equal to three. When the stochastic behavior of a bio-chemical network is concerned, the Fokker-Planck equation may be used to calculate the probability density function of the network, and its spatial dimension can be much larger.

Due to the more interests on stochasticity, how to efficiently solve reaction-diffusion equations, especially for high dimensions, becomes important. There are two major challenges for the problem. The first is the "curse of dimensions" for high dimensional problems. When dimension becomes larger, the number of the unknown of the equation is huge so that store and manipulate these unknowns can be too expensive and unaffordable, if traditional methods are applied. The second challenge is solving the afterwords time dependent problem. It can be ineffective if some improper time integration methods are used.

The first challenge can be overcame by the sparse grid method. The sparse grid method largely decrease the number of knowns for high dimensions, with the trade off that lower the accuracy. The afterword time dependent problem can be solved by a series of semi-implicit integration factor method, which is proved to be efficient, no matter in high dimensions or not. Combination of the two methods can leads to an efficient method to solve high dimensional reaction-diffusion equations.

This thesis consists four topics. In Chapter 2, we study a reaction-diffusion system that model the morphogen gradient. We mainly focus on the robustness of the morphogen gradient. In Chapter 3, we apply a cell lineage model to study the hair follicle system. We present some hypotheses that can be key factors for hair follicle cyclic growth. The robustness of the system is also discussed. In Chapter 4, we construct an efficient numerical method to solve high dimensional PDE systems on regular grids. The unknown of the equation is represented as high-dimensional array, and the differential operators can have compact representation. In Chapter 5, we construct the semi-implicit integration factor method on the sparse grid. Numerical tests show that our new method is more efficient than some existing ones.

Chapter 2

Robustness of Morphogen Gradients With “Bucket Brigade” Transport Through Membrane-Associated Non-Receptors

NOTE: This chapter was done in collaboration with Jinzhi Lei, You Song, Qing Nie and Frederic Y.M. Wan and was published in [2].

2.1 Introduction

At some stages of embryonic development, signaling protein molecules known as Morphogens (ligands) are synthesized at some specific localized site. Some of them may disperse from their production site, bind to cell receptors along the way, and result in different receptor occupancies at different cell locations. The spatial concentration gradient of morphogen-receptor complexes (signaling gradients) induces spatially graded differences in cell signaling. The differential cell signaling in turn gives rise to different gene expressions. This fact leads cells in different locations to go through different cell lineages, finally reach different stable cell fates, which is essential for tissue patterns and organs during development.

In general, it is important for a developing biological organism to form appropriate morphogen gradients that appear at a proper time and proper place and are robust with respect to perturbations in system architecture, environmental changes, or signaling noise. As conflicting biochemical processes and strategies may be required to attain precision and robustness, the delineation of how both characteristics can be achieved in biological development remains a challenge in systems biology [3]. Different mechanisms and processes involved in the formation of different morphogen gradients are either known or have been proposed [4, 5, 6, 7, 8, 9, 10, 11, 12]. One process that remains indispensible is the transport of morphogens away from their localized source.

A number of models have been proposed for morphogen transport; most are based on diffusion of morphogen molecules and their interactions with non-signaling extracellular molecules [13, 6, 14, 15, 9, 11, 16, 17]. It has been argued that diffusion alone may not be a reliable mechanism as the resulting gradients are sensitive to substantial changes in system parameters, leading to a signaling gradient that is no longer biologically useful [3, 18, 19]. Many additional mechanisms such as transcytosis, dynamin-mediated endocytosis, feedback control and regulations by membrane-associated non-receptors have been suggested for achieving robustness [20, 13, 6, 1, 18].

Regulations by non-signaling receptor (or non-receptor for short) such as heparan sulfate proteoglycans (HSPG) in morphogen movement are observed in many experiments [21, 4, 5, 22, 8, 23], and the effects of the presence of non-receptors on the existence and characteristics of the steady-state signaling morphogen gradients have been studied in [24, 25, 1, 18]. From these studies, the desired robustness with respected to substantial perturbations of morphogen synthesis rate is seen to be achievable through two different mechanisms including regulations of non-receptors:

Mechanism 1 Substantial (reversible) binding of *slowly turned over* morphogen molecules with membrane-bound non-receptors with the resulting non-signaling (morphogen) complexes degrading at a sufficiently rapid rate [18].

Mechanism 2 Fast binding of *rapidly turned over* free morphogen molecules with non-receptors so that the non-signaling complexes move downstream through a "bucket brigade" process [1].

Mathematical analysis of the effectiveness of these two robustness mechanisms have been carried out in [1, 18] based on reaction-diffusion models with only one diffusion term in either free or non-signaling bound-morphogens. These one diffusion models however are biologically incomplete because it is possible to have both types of transport. Models with two diffusions have been developed and studied in [25, 26, 27] with the numerical simulations carried out in [25] suggesting that Mechanism 2 continuous to ensure robustness if the free morphogen degradation is large enough.

We study numerically a model of morphogen gradient formation with both diffusion of free morphogens and the movement of non-signaling morphogen complexes through a "bucket brigade" process to provide some evidence for the previous numerical simulation study in [25]. The model here (corresponding to the Dpp gradient in the wing imaginal disc) is unrelated to, and conceptually different from those of [26, 27] where the non-receptors are freely diffusing (Sog) molecules.

2.2 Model equations

We refer the mathematical model in [25, 1], which was based on the formation of morphogen gradient in Drosophila wing imaginal discs regulated by the glypican members of heparan sulfate proteoglycans [5]. The model involves concentration of free ligands [L], receptors [R], ligand-receptor complexes [LR], non-receptors [N], and ligand-non-receptor complexes [LN]. Distributions of various morphogen concentrations in the wing imaginal discare assumed to be sufficiently uniform in two directions (except possibly for boundaryers) to change only along the antero-posterior axis. The anterior and posterior compartments are taken to be sufficiently symmetric so that we can focus on the posterior compartment that spans the range $-d_0 \leq X \leq X_{\max}$ with a narrow region of ligand synthesis at $-d_0 \leq X \leq 0$. Diffusions of both [L] and [LN] are allowed; other main reactions include binding and unbinding of ligand to receptors and non-receptors, degradation of ligands, receptors and the ligand-receptor complexes. The total concentration of non-receptor binding sites is abundant and assumed to be a constant N_0. Therefore, the resulting reaction-diffusion equations is (see [1] for an expanded discussion of the D_{LN} term):

$$\begin{aligned}
\frac{\partial[\mathrm{L}]}{\partial T} &= D_{\mathrm{L}}\frac{\partial^2[\mathrm{L}]}{\partial X^2} - k_{\mathrm{on}}[\mathrm{L}]\,[\mathrm{R}] + k_{\mathrm{off}}[\mathrm{LR}] \qquad (2.1)\\
&\quad - j_{\mathrm{on}}[\mathrm{L}]\,(N_0 - [\mathrm{LN}]) + j_{\mathrm{off}}[\mathrm{LN}] - \delta_{\mathrm{L}}[\mathrm{L}] + V(X)\\
\frac{\partial[\mathrm{R}]}{\partial T} &= \omega_{\mathrm{R}}([\mathrm{LR}]) - k_{\mathrm{on}}[\mathrm{L}]\,[\mathrm{R}] + k_{\mathrm{off}}[\mathrm{LR}] - \delta_{\mathrm{R}}[\mathrm{R}] \qquad (2.2)\\
\frac{\partial[\mathrm{LR}]}{\partial T} &= k_{\mathrm{on}}[\mathrm{L}]\,[\mathrm{R}] - k_{\mathrm{off}}[\mathrm{LR}] - \delta_{\mathrm{LR}}[\mathrm{LR}] \qquad (2.3)\\
\frac{\partial[\mathrm{LN}]}{\partial T} &= D_{\mathrm{LN}}\frac{\partial^2[\mathrm{LN}]}{\partial X^2} + j_{\mathrm{on}}[\mathrm{L}]\,(N_0 - [\mathrm{LN}]) - j_{\mathrm{off}}[\mathrm{LN}], \qquad (2.4)
\end{aligned}$$

where $X \in (-d_0, X_{\max})$. The production rate of ligands is given in terms of the *Heaviside unit step function* $H(z)$:

$$V(X) = v_0 H(-X), \qquad H(z) \equiv \begin{cases} 0 & (z < 0) \\ 1 & (0 < z) \end{cases} \tag{2.5}$$

where v_0 is a constant ligand synthesis rate in the production region $-d_0 \leq X < 0$. The boundary of the region is specified by $X = X_{\max}$. The receptor synthesis rate is assumed to depend only on [LR] and not on X and T explicitly.

Assumed symmetry at the border $X = -d_0$ requires the no flux conditions:

$$\frac{\partial[\mathrm{L}]}{\partial X} = 0, \quad \frac{\partial[\mathrm{LN}]}{\partial X} = 0 \quad \text{at} \quad X = -d_0. \tag{2.6}$$

The edge $X = X_{\max}$ is taken to be a sink for all diffusive elements so

$$[\mathrm{L}] = [\mathrm{LN}] = 0 \quad \text{at} \quad X = X_{\max}. \tag{2.7}$$

With $V(X)$ discontinuous at $X = 0$, we stipulate the continuity of [L], [LN] and the partial derivatives $\partial[\mathrm{L}]/\partial X$, $\partial[\mathrm{LN}]/\partial X$ at $X = 0$, consistent with the Eq. (2.1) requiring $\partial^2[L]/\partial X^2$ to have only a simple jump discontinuity at $X = 0$.

Before the onset of ligand production ([L] $= 0$), there is no ligand concentration of any kind so that

$$[\mathrm{L}] = [\mathrm{LR}] = [\mathrm{LN}] = 0, \quad (-d_0 < X < X_{\max}, \quad T \leq 0). \tag{2.8}$$

The receptors are expected to be in a steady-state prior to the onset of ligand production, so that

$$\frac{\partial[\mathrm{R}]}{\partial T} = 0.$$

This implies

$$T = 0: \quad [\mathrm{R}] = \omega_{\mathrm{R}}(0)/\delta_{\mathrm{R}} \equiv R_0 \tag{2.9}$$

where R_0 is the concentration of unbound receptors at the steady-state prior to the onset of ligand production.

The conditions Eq. (2.1-2.9) define an initial-boundary value problem (IBVP) for morphogen gradient formation. The model in [1] corresponding to the special case with $D_{\mathrm{L}} = 0$. With $D_{\mathrm{L}} \neq 0$ herein, the corresponding mathematical problem becomes more difficult to analyze and requires a new approach.

Similar to [1], we introduce the following non-dimensional quantities:

$$l = \frac{[\mathrm{L}]}{j_{\mathrm{off}} N_0/\delta_{\mathrm{L}}}, \quad r = \frac{[\mathrm{R}]}{R_0}, \quad u = \frac{[\mathrm{LR}]}{R_0 k_{\mathrm{on}} j_{\mathrm{off}} N_0/(\delta_{\mathrm{L}}\delta_{\mathrm{LR}})}, \quad w = \frac{[\mathrm{LN}]}{N_0}, \tag{2.10}$$

$$t = \delta_{\mathrm{L}} T, \quad x = \frac{X + d_0}{X_{\max} + d_0}, \quad d = \frac{d_0}{(X_{\max} + d_0)}, \quad \omega_R = \delta_{\mathrm{R}} R_0 k(u), \tag{2.11}$$

$$\varepsilon = \frac{j_{\mathrm{on}} N_0}{\delta_{\mathrm{L}}}, \quad \lambda^2 = \frac{j_{\mathrm{off}}(X_{\max} + d_0)^2}{D_{\mathrm{LN}}}, \quad \delta_r = \frac{\delta_{\mathrm{R}}}{\delta_{\mathrm{L}}}, \quad \delta_u = \frac{\delta_{\mathrm{LR}}}{\delta_{\mathrm{L}}}, \tag{2.12}$$

$$\alpha = \frac{k_{\mathrm{off}}}{\delta_{\mathrm{LR}}}, \quad \gamma = \frac{k_{\mathrm{on}} R_0}{j_{\mathrm{on}} N_0}, \quad \mu = \frac{k_{\mathrm{on}} j_{\mathrm{off}} N_0}{\delta_{\mathrm{R}} \delta_{\mathrm{L}}}, \quad \eta = \frac{v_0 j_{\mathrm{on}}}{\delta_{\mathrm{L}} j_{\mathrm{off}}}, \tag{2.13}$$

$$\theta_l = \frac{D_{\mathrm{L}}}{(X_{\max} + d_0)^2 \delta_{\mathrm{L}}}, \quad \theta_w = \frac{D_{\mathrm{LN}}}{(X_{\max} + d_0)^2 \delta_{\mathrm{L}}}. \tag{2.14}$$

Here $k(u)$ is the normalized receptor synthesis rate satisfying $k(0) = 1$. The normalized ligand production rate is given by

$$v(x) = (\eta/\varepsilon) H(d - x).$$

Here η is the effective production rate which is important for our discussion below.

Using the above non-dimensional variables, Eq. (2.1- 2.4) become

$$\begin{aligned}
\frac{\partial l}{\partial t} &= \theta_l \frac{\partial^2 l}{\partial x^2} - (l-w) - \varepsilon\left(l(1-w) - \gamma(\alpha u - lr)\right) + v(x) && (2.15)\\
\frac{\partial r}{\partial t} &= \delta_r\left(\mu(\alpha u - lr) + (k(u) - r)\right) && (2.16)\\
\frac{\partial u}{\partial t} &= -\delta_u((\alpha+1)u - lr) && (2.17)\\
\frac{\partial w}{\partial t} &= \theta_w\left(\frac{\partial^2 w}{\partial x^2} - \lambda^2(w - \varepsilon l(1-w))\right), && (2.18)
\end{aligned}$$

with initial conditions

$$l = u = w = 0, \quad r = 1, \quad (t = 0, 0 \leq x \leq 1) \qquad (2.19)$$

and boundary conditions

$$\left.\frac{\partial l}{\partial x}\right|_{x=0} = \left.\frac{\partial w}{\partial x}\right|_{x=0} = l|_{x=1} = w|_{x=1} = 0, \qquad (t \geq 0) \qquad (2.20)$$

2.3 Numerical methods for the steady state

The existence, uniqueness and stability of the biologically acceptable gradients of Eq. (2.15-2.20) have been established in [2]. Here we consider the robustness of the corresponding signaling gradient when ligand production rate is changed. We focus on the dependence of the signaling gradient on the ligand production rate characterized by the dimensionless constant η. Denoting the dimensionless signaling concentration by $\overline{u}(x;\eta)$, we characterize the robustness of the signaling gradient with respect to changes in ligand production rate by a robustness index $R(\eta,\eta')$, defined as the relative change of the signaling gradient when the parameter η is changed to η':

$$R(\eta,\eta') = \frac{1}{\Delta\eta/\eta}\frac{1}{1-d}\int_d^1 \frac{\Delta\bar{u}(x)}{\bar{u}(x;\eta)}dx \tag{2.21}$$

where

$$\Delta\eta = |\eta-\eta'|, \quad \Delta\bar{u}(x) = |\bar{u}(x;\eta)-\bar{u}(x;\eta')|. \tag{2.22}$$

Evidently, the smaller the value of R, the more robust is the signaling gradient. We adopt $R < 0.2$ for acceptable robustness as in [18].

When both ε and θ_l, by reasonable approximation, the robustness of the system can be derived and analyzed, as in [2]. It is shown that at this situation, when η is large, the system has good robustness. For general cases when either ε or θ_l is not small, such analytic methods may not be applied, since the analytic calculation of the robustness R is difficult. Here we perform numerical simulations to study the robustness for such situations.

To study the robustness Eq. (2.21), we first need to compute the signaling gradient $\overline{u}(x;\eta)$ in Eq. (2.15)-2.20). Some methods may be applied to directly solve the steady state [28]. To avoid solving the large non-linear system, we apply the semi-implicit integration factor method (IIF) [29] to solve the time dependent problem, and wait until the system reach the steady state. The second order IIF method for the reaction-diffusion system:

$$u(x,t) = \Delta u + f(u), \tag{2.23}$$

is:

$$u_{n+1} = e^{D\Delta t}\left(u_n + \frac{\Delta t}{2}f(u_n)\right) + \frac{\Delta t}{2}f(u_{n+1}), \tag{2.24}$$

where u_n is the spatial discretization of $u(x,t)$ at time $t = n\Delta t$, and matrix D is the discretized form of the Laplacian Δ, in this case a tridiagonal matrix. After computing the signaling concentration, $R(\eta,\eta')$ is calculated based on a quadrature rule.

For deferent pair of ε (range from 10^{-4} to 10^{-2}) and θ_l (range from 10^{-2} to 1) and different value of η, we compute the robustness index R and find the η value, as a function of (ε, θ_l) so that $R = 0.2$. The result is given in Fig. 2.1(b).

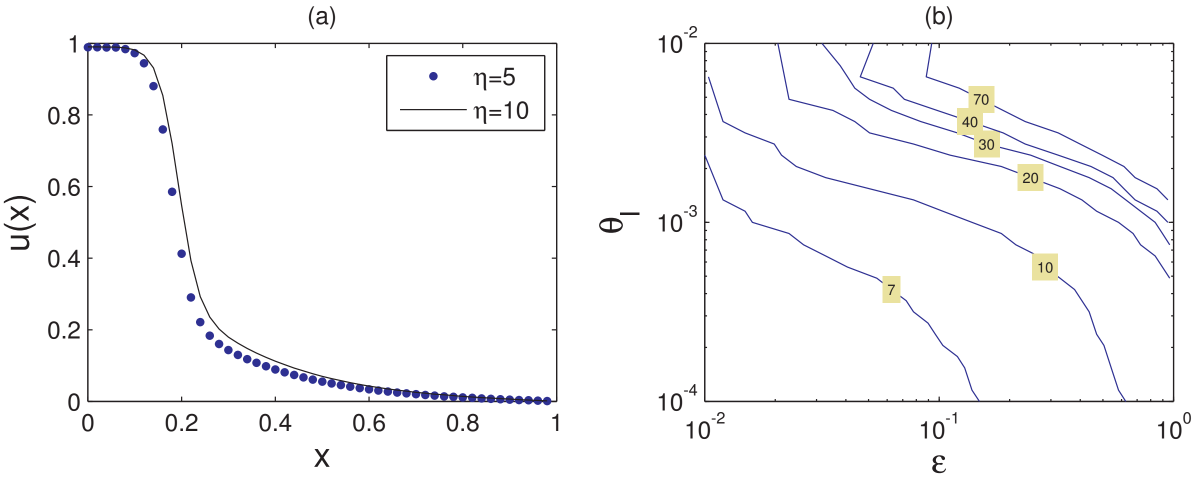

Figure 2.1: Direct numerical simulations for robustness of morphogen gradient. (a) Profiles of $u(x)$ at the steady-state for two morphogen production rates. Parameters used are referred to [1]: $d = 0.06$, $\lambda = 5.0$, $\gamma = 0.8$, $\mu = 0.6$, $\alpha = 0.1$, $\varepsilon = 0.01$, $\theta_l = 4.0 \times 10^{-4}$, and $\eta = 5$ (circle dots) or $\eta = 10$ (solid line). The feedback function is $k(u) = \frac{1}{1+(u/1.2)^2}$. (b) Contour plot of η (values shown on the lines) as a function of (ε, θ_l) so that the robustness $R = 0.2$.

Chapter 3

Cell lineage model on hair cyclic behavior

NOTE: This chapter was done in collaboration with Qixuan Wang, Qing Nie, Xing Dai and Briana Lee and the manuscript is under preparation.

3.1 Introduction

Hair follicle is an approximately cylindrical mini-organ from which hairs are formed. It is unique to mammals in that it is the only organ being able to undergo cyclic transformations through the mammalian entire lifetime. Hair follicle cycle consists of three main phases: anagen, an active regeneration phase during which massive epithelial cell proliferation happens; catagen, a phase of rapid, apoptosis-driven organ involution; and telogen, a relative quiescence phase during which the follicle awaits entering the following anagen [30, 31, 32, 33].

The stem cells are mainly localized in the bulge region of a hair follicle, which marks the lower end of the permanent portion of the hair follicle and is the source of cells for HF renewal. Stem cells are slow cycling. Upon the induction of anagen, HF stem cells located in the bulge region actively proliferate in response to messages from the dermal papilla (DP), and the daughter transient amplifying cells migrate to the secondary hair germ (sHG),which is considered an extension of the bulge present only during telogen [34, 35]. During anagen the bulge stem cells produce excessive diversity of daughter cell lineages in the downward direction, resulting in the descending outer root sheath (ORS) and provides the penetration of the hair follicle into deep dermis. During mature anagen, the ORS-driven downward growth of the HF is complete, mitotic activity is evident in the bulge region [36, 37], yet we still observe active downward migration of bulge-derived cells along the ORS [38]. During catagen, matrix keratinocytes are considered to play an important role in the HF involution. These cells originate from the bulge cells and are transient amplifying cells, thus have only a limited proliferative potential. During catagen, the matrix cells cease proliferation and undergo extensive apoptosis, which drives the upward movement of DP. At the end of catagen, matrix cells are almost completely eliminated, hair germ cells emerge, and DP returns to its initial position. Through telogen, the bulge, hair germ and DP stay in proximity.

Up to date, the fundamental mechanism underlying the follicle clock remains unclear. Recently, genetic studies show that several molecular pathways becoming activated in different hair follicle phases. Macro-environmental bone morphogenetic protein (BMP) ligands and Wnt antagonists act collectively to keep hair follicle stem cells in a quiescent state [39]. Another study shows that both Fgf18 and BMP6 slow keratinocyte growth without inducing terminal differentiation, indicating that such signals favor bulge stem cells in a quiescent state [40]. The correlation between hair follicle phases and various molecular pathway activities leads to the hypothesis that genetic interaction among various molecular pathways automatically result in the cyclic hair follicle growth dynamics.

In this chapter we develop a cell-lineage model for hair follicle, in which we consider three main types of cells: stem cells, transient amplifying cells (TA) and terminally differentiated cells (TD). We study various of feedback controls and pin point some reasonable mechanisms that produce periodicity, and explore how different feedbacks affect the dynamics of the whole hair follicle system. First, we find that while many feedbacks fail a cyclic growth dynamics, certain feedbacks are crucial in regulating a cyclic hair follicle growth dynamics. Second, we study the robustness of some feedbacks to internal white noise. Moreover, we find that dynamics regulated by different feedbacks has its featured timescales, and a combination of various feedback permits flexibility of timescales of the follicle growth dynamics.

3.2 Three-stage cell lineage model

Consider the cell lineage model in Figure 3.1 which contains three stages: the stem cell (SC), transient amplifying cells (TA) and thermally differentiated cells (TD). Each SC cell has a probability of p_0 to replicate itself, and a probability of $1 - p_0$ to differentiate to two TA cells. The cell division rate of SC cell is denote as v_0. Each TA cell has a probability of p_1 to replicate itself, and a probability of $1 - p_1$ to differentiate to two TD cells. The cell division rate of TA cell is denote as v_1. For TD cell, its degradation rate is denoted as d_0. Using [.] to denote the population of each type of cell, the cell lineage model is described by the following ODE system:

$$\begin{aligned}
\frac{d[SC]}{dt} &= (2p_0 - 1)v_0[SC] \\
\frac{d[TA]}{dt} &= 2(1 - p_0)v_0[SC] + (2p_1 - 1)v_1[TA] \\
\frac{d[TD]}{dt} &= 2(1 - p_1)v_1[TA] - d_0[TD]
\end{aligned} \tag{3.1}$$

We study feedbacks that marked in Figure 3.1 with black or red arrows. Here the arrow-headed line denotes the positive feedback and bar-headed line denotes the negative feedback. For simplicity, the strengthen of the feedback is the function of the corresponding cell population. For example, the positive feedback on p_1 from TD cell is $p_1([TD])$ where $p_1(.)$ is a mono-increasing function.

3.2.1 Negative feedback from TD cell to p_1 leads to periodicity

If only this feedback exists, to ensure that SC cell not extinct or go infinity, p_0 must be 0. Also, we suppose the feedback takes the following Hill function:

$$p_1([TD]) = \frac{p_1}{1 + ([TD]/EC_{50})^n}. \tag{3.2}$$

Thus we can study the following ODE system with two unknowns instead of Eq. (3.1):

$$\begin{aligned} \frac{d[TA]}{dt} &= c + \left(\frac{2p_1}{1+[TD]^n} - 1\right)[TA] \\ \frac{d[TD]}{dt} &= 2\left(1 - \frac{p_1}{1+[TD]^n}\right)[TA] - d[TD]. \end{aligned} \tag{3.3}$$

For above, the following dimensionless process has been applied:

$$\begin{gathered} t \to v_1 t, \quad c \to \tfrac{v_0}{v_1}\tfrac{[SC]}{EC_{50}}, \quad d \to v_1 d, \\ [TA] \to \tfrac{[TA]}{EC_{50}}, \quad [TD] \to \tfrac{[TD]}{EC_{50}}. \end{gathered} \tag{3.4}$$

By parameter search, we can find regions that generate periodicity. For example, when $n = 3$, the regions that a stable periodic solution exists is shown in Figure 3.2, for different values of p_1. We can see that, larger proliferation probability makes the system easier to get periodic solution. Mathematically, the stable periodic solution exists since the system is bounded, and contains one unstable node. From Poincare-Bendixson theorem, the system has a stable limit cycle.

We also learn the parameters that affect the length of the period. By increasing the value of p_1, we found that the period becomes longer and longer, and approaches to the real hair follicle cycle length. The results are shown in Figure 3.3.

In a similar way, we state that the negative feedback from TA cell to the proliferation probability of SC cell can also generate periodicity.

3.2.2 Positive feedback from TD cell to v_1 leads to periodicity

Consider the following ODE system with two unknowns:

$$\frac{dx_1}{dt} = X_1(x_1, x_2), \quad \frac{dx_2}{dt} = X_2(x_1, x_2). \tag{3.5}$$

Suppose the system has one critical point (x_1^*, x_2^*), then near the critical point, the linearized ODE system is:

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = A\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \tag{3.6}$$

where

$$A = \left.\begin{pmatrix} \frac{\partial X_1}{\partial x_1} & \frac{\partial X_1}{\partial x_2} \\ \frac{\partial X_2}{\partial x_1} & \frac{\partial X_2}{\partial x_2} \end{pmatrix}\right|_{(x_1^*, x_2^*)} \tag{3.7}$$

Then if the following conditions are satisfied, Eq. (3.5) has a stable periodic solution:

(i) The system is bounded, which means either x_1 or x_2 cannot go infinity.

(ii) The following inequality holds:

$$\det(A) > 0, \quad \text{trace}(A) > 0. \tag{3.8}$$

This is part of the Poincare-Bendixson theorem and we use this theorem to find the conditions for a periodic solution. Assume the feedback is of the following form:

$$v_1([TD]) = v_1\left(s + \frac{([TD]/EC_{50})^n}{1+([TD]/EC_{50})^n}\right) \tag{3.9}$$

Then after dimensionless process, we study the following system:

$$\begin{aligned} \frac{d[TA]}{dt} &= c - \left(s + \frac{[TD]^n}{1+[TD]^n}\right)[TA] \\ \frac{d[TD]}{dt} &= a\left(s + \frac{[TD]^n}{1+[TD]^n}\right)[TA] - d[TD], \end{aligned} \tag{3.10}$$

where $a = 2(1-p_1)/(1-2p_1) \in [2,\infty)$, since for this case, to prevent the excessive number of TA cell, $p_1 < 1/2$. The only critical point for the system is:

$$[TA]^* = \frac{c\,(1+(ac/d)^n)}{(ac/d)^n + s + (ac/d)^n s}, \quad [TD] = \frac{ac}{d}. \tag{3.11}$$

Then to fulfill the condition (ii), we have the following sufficient condition that leads to the existence of a stable periodic solution:

$$1 + d + s + \frac{dns}{s + (ac/d)^n(1+s)} < \frac{1+dn}{1+(ac/d)^n}, \quad p_1 < 1/2. \tag{3.12}$$

3.2.3 Feedbacks that cannot generate periodic patterns

By check the trace and the determinant of the linearized matrix around the critical point, we state that the following feedbacks alone cannot generate stable periodic patterns:

(i) Positive feedback from TD cell to p_1.

(ii) negative feedback from TD cell to v_1.

(iii) negative/positive feedback from TD cell to v_0.

(iv) negative/positive feedback from TA cell to v_1.

We'll study the negative feedback from TD cell to p_0 in Section 3.4, and this feedback gives the periodic patterns.

3.3 Robustness of the periodic patterns

In this section we study the robustness of the periodic patterns with respect to internal noise. In the ODE description of the cell lineage model, Eq. (3.1), we dismiss the stochastic behavior of the system. Actually, when a cell divide, it divide with some probability, while ODE system only model its statistical behavior. Let's say that at certain time, $p_0 = 0.7$, in ODE system, exact 70 percent of the dividing SC cells will proliferate and 30 percent of them will differentiate.

However, in reality, it is entire possible that 71 percent, or even more dividing SC cells will proliferate, since the SC cells divide with some probabilities. When the cell number is large enough, the stochastic behavior can be neglect. But when the cell number is relative small, the stochastic behavior may affect the periodic patterns: alter the period, change the amplitude, or even destroy the periodic patterns. In previous section we mainly studied two feedbacks that generate periodicity: negative feedback on proliferation probability, or positive feedback on cell cycle rate. Two schemes will react to noise differently.

The Gillespie algorithm [41] is used to study the stochastic behavior of the cell lineage in this chapter. The core idea is that once a cell divide, we generate a random number to indicate whether it proliferates or differentiates. We run the simulation long enough so that the system goes multiple cycles, and plot the phase portrait in Figure 3.4. We can see that the negative feedback on proliferation probability is more robust to noise (Figure 3.4(a)), and the noise destroys the periodic pattern comes from the positive feedback on cell cycle rate (Figure 3.4(b)).

This can be explained by the following: the stochasticity of the cell lineage mainly comes from the proliferation probability. Once the cell is more determined to do one thing than another (p_1 is close to 0 or 1), the system is less stochastic. For the negative feedback control, when the TD cell population is high, TA cells are more likely to differentiate, and vice versa. While for the positive feedback control on cell cycle rate, the proliferation probability is constant, less than 0.5, and cannot be too small. Thus the system is not robust to the noise.

3.4 Mechanisms to prolong the length of the period

For humans and mouses, the hair follicle cycle length can be very long: from months to years. Within normal parameter ranges, the cell lineages in Section 3.2 cannot have such long period. In this section we give some mechanisms that efficiently prolong the period of the cyclic patterns.

3.4.1 Combination of feedbacks: An excitable system

In Section 3.2, we study the cell lineage model with only one feedback. We have shown that the negative feedback on proliferation probability, or positive feedback on cell cycle rate could generate periodicity. In this section we study the cell lineage model with two feedbacks. We show that the new system can be an excitable system.

The cell lineage model we study here is shown in Figure 3.5. Besides the negative feedback on proliferation probability, which leads to periodic behavior, another positive feedback from TA cell to its proliferation probability is added. We assume the feedback takes the Hill-function type, so the ODE system for the cell lineage is (Since $p_0 = 0.5$, the population of stem cell remains constant):

$$\begin{aligned} \frac{d[TA]}{dT} &= (1-2p_0)v_0[SC] + \left(\frac{2b([TA]/K_1)^n}{r_1 + ([TA]/K_1)^n + r_2([TD]/K_2)^m} - 1\right) v_1[TA] \\ \frac{d[TD]}{dT} &= 2\left(1 - \frac{b([TA]/K_1)^n}{r_1 + ([TA]/K_1)^n + r_2([TD]/K_2)^m}\right) v_1[TA] - d^*[TD] \end{aligned} \tag{3.13}$$

With the following dimensionless process:

$$\begin{aligned} &x = [TA]/K_1, \quad Y = [TD]/K_2, \quad T = tv_1, \quad d = d^*/v_1, \\ &K = K_1/K_2, \quad c = (1-2p_0)v_0[SC]/(v_1K_1), \end{aligned} \tag{3.14}$$

The ODE system becomes:

$$\begin{aligned} \frac{dX}{dt} &= c + \left(\frac{2bX^n}{r + X^n + Y^m} - 1\right) X, \\ \frac{dY}{dt} &= 2K\left(1 - \frac{bX^n}{r + X^n + Y^m}\right) X - dY. \end{aligned} \tag{3.15}$$

With the parameter: $n = 1, m = 2, b = 1, r = 0.05, d = 0.8$ and $K = 0.1$, when parameter c increases, the phase portrait is given in Figure 3.6(a) ($c = 0.0026$) and Figure 3.6(b) ($c = 0.01$). When c is small, the system has three critical point: one stable node, one unstable node and a saddle point. When c increases, the stable node and the saddle point will approach to each other, finally reach each other and disappear. Then the system only left one unstable node, thus a periodic solution with long period exists, as in Figure 3.6(c).

3.4.2 Multi-stage cell lineage model

In this section we present a mechanism to prolong the period by adding multiple middle stages to the cell lineages, as in Figure 3.7. Numerical simulation shows that when the more the middle stage presents, the longer the period is, as in Figure 3.8.

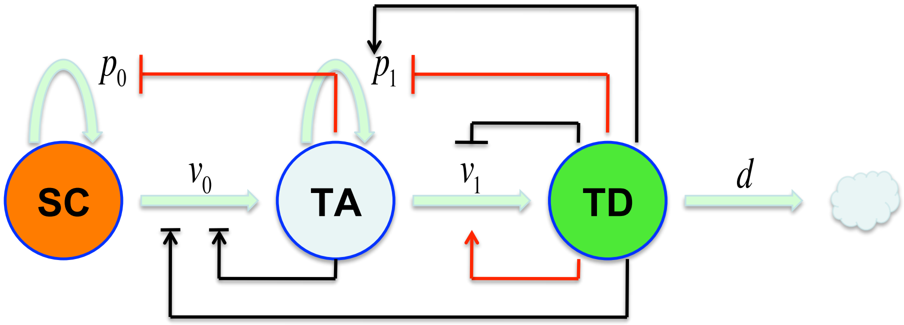

Figure 3.1: A three-stage cell lineage model with interested feedbacks

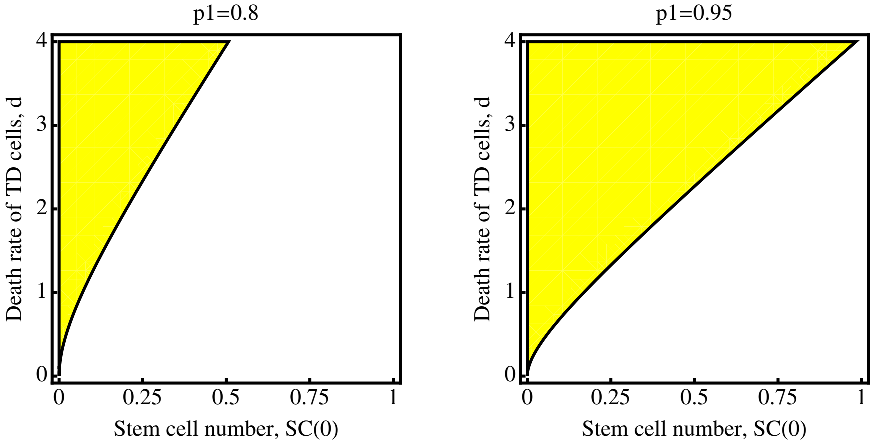

Figure 3.2: Parametric region that Eq. (3.3) has a stable periodic solution (marked as yellow) for $p_1 = 0.8$ and $p_1 = 0.95$.

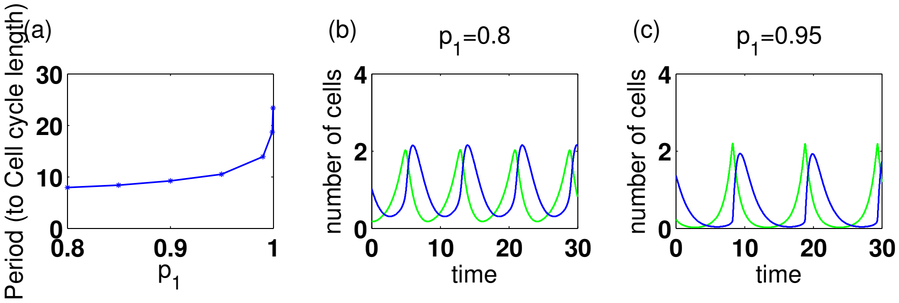

Figure 3.3: The relationship between p_1 and the period of the cycle. (a) As p_1 increases, the period becomes longer. (b) For relative small p_1, the period is short. (c) For larger p_1, the period is long.

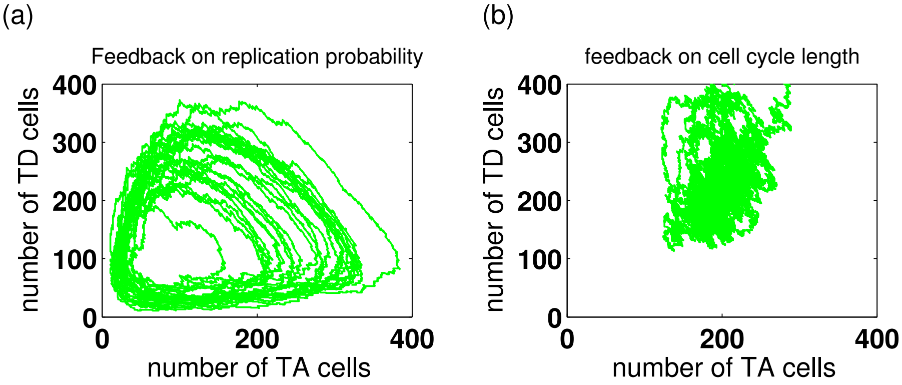

Figure 3.4: The phase portrait of the cell lineage model when the internal noise is considered. (a) The phase portrait for the negative feedback on proliferation probability. The periodic pattern still remains. (b) The phase portrait for the positive feedback on cell cycle rate. The periodic pattern is somehow ruined.

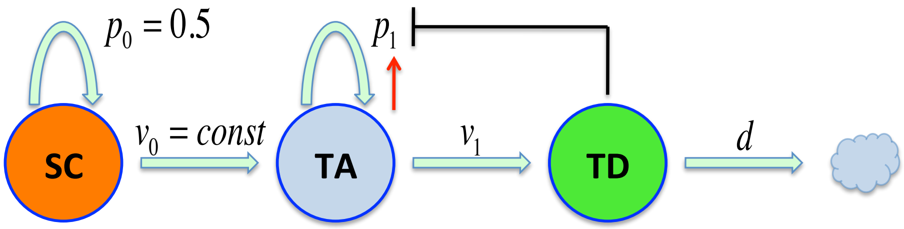

Figure 3.5: Three stage cell lineage model with two feedbacks: (i) A negative feedback from TD cell to TA cell's proliferation probability; (ii) A positive feedback from TA cell to its proliferation probability.

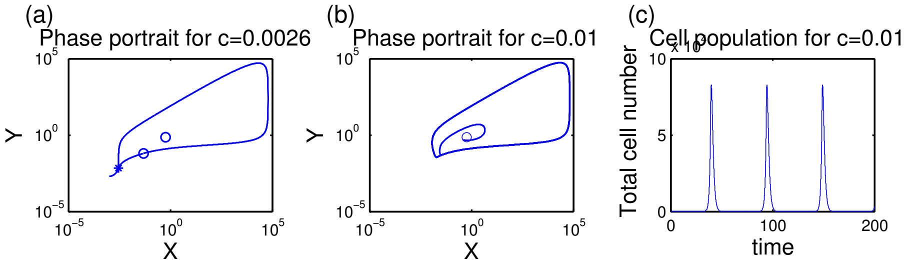

Figure 3.6: (i) Phase portrait for $c = 0.0026$. Three critical points exist: one stable node, one saddle point and one unstable node; (ii) Phase portrait for $c = 0.01$. One unstable node and a stable limit cycle exist. (iii) Cell population vs. time for $c = 0.01$.

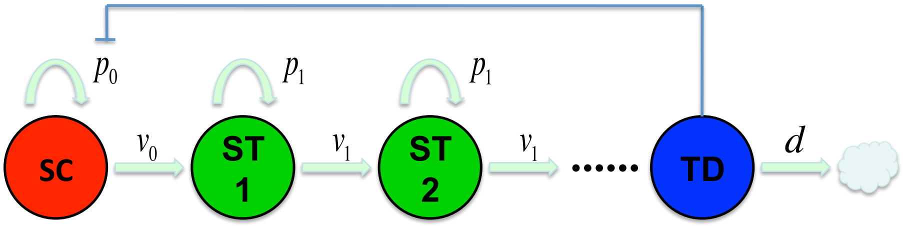

Figure 3.7: Cell lineages with multiple middle stages.

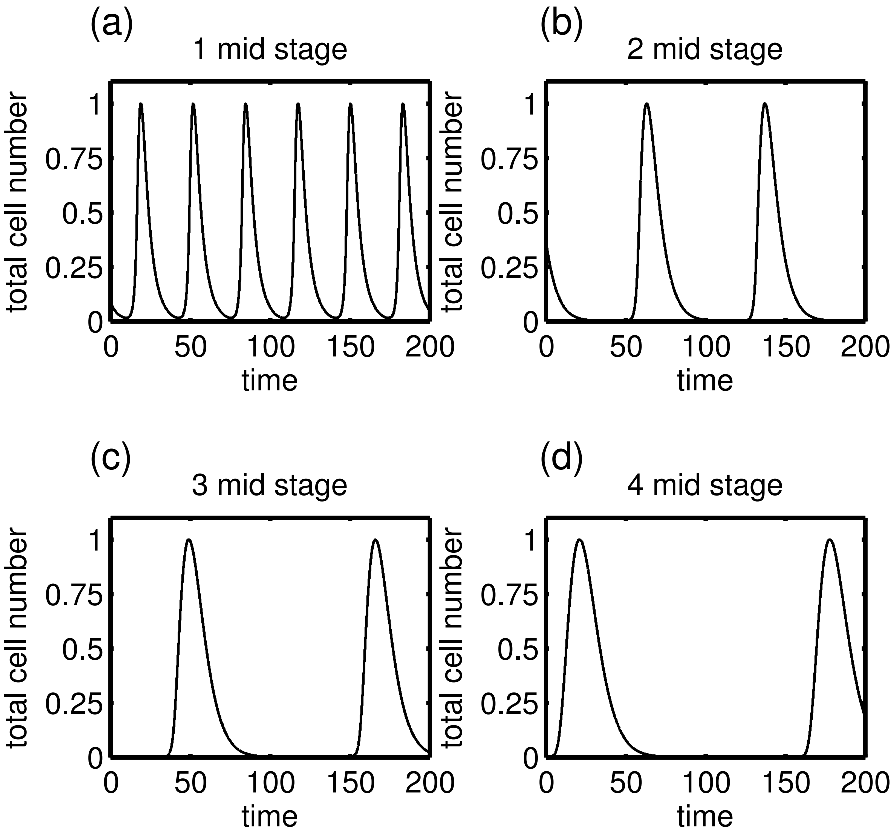

Figure 3.8: Cell population for different number of multi stages.

Chapter 4

Array-representation Integration Factor Method for High-dimensional Systems

NOTE: This chapter was done in collaboration with Lei Zhang and Qing Nie and was published in [42].

4.1 Summary

High order spatial derivatives and stiff reactions often introduce severe temporal stability constraints on the time step in numerical methods. Implicit integration factor (IIF) method, which treats diffusion exactly and reaction implicitly, provides excellent stability properties with good efficiency by decoupling the treatment of reactions and diffusions. One major challenge for IIF is the storage and calculation of the potential dense exponential matrices of the sparse discretization matrices resulted from the linear differential operators. Motivated

by a compact representation for IIF (cIIF) for Laplacian operators in two and three dimensions, we introduce an array-representation technique for efficient handling of exponential matrices from a general linear differential operator that may include cross-derivatives and non-constant diffusion coefficients. In this approach, exponentials are only needed for matrices of small size that depend only on the order of derivatives and number of discretization points, independent of the size of spatial dimensions. This method is particularly advantageous for high dimensional systems, and it can be easily incorporated with IIF to preserve the excellent stability of IIF. Implementation and direct simulations of the array-representation compact IIF (AcIIF) on systems, such as Fokker-Planck equations in three and four dimensions and chemical master equations, in addition to reaction-diffusion equations, show efficiency, accuracy, and robustness of the new method. Such array-presentation based on methods may have broad applications for simulating other complex systems involving high-dimensional data.

4.2 Introduction

In the previous chapter, we discussed the noise effects on the hair follicle system. There, we used a Gillespie algorithm [43] to simulate multiple samples of the cell lineage model, and got some statistical behaviors. Alternative approach is to make some assumptions and approximations on the copy number of each cell, then derive the following Fokker-Planck equation (aka Kolmogorov backward equations)[44]:

$$\partial_t u(\mathbf{x},t) = \sum_{i,j=1}^{n} \partial_{x_i}(D_{ij}(u,\mathbf{x})\partial_{x_j} u(\mathbf{x},t)) + F(u,\mathbf{x}), \tag{4.1}$$

The unknown functions $u(x,t)$ is the probability density function and once we solve for it, we get all the statistical information about the system. In general, for above equation, n is the spatial dimension and $\mathbf{x} = \{x_1, ..., x_n\}$. A non-linear term $F(u,\mathbf{x})$ is often interpreted as a reaction term. Coefficients D_{ij} can be either constants or functions of u and $\mathbf{x}$. In morphogen formation model, the function $u(\mathbf{x}, t)$ usually represents concentrations of physical or biological species with reactions among them, for which n is usually 2 or 3. In Fokker-Planck equation, n can be much larger.

Spatial discretization for partial differential equations of higher spatial dimensions (even for $n = 3$) often requires large, sometimes prohibitive, data storage and management as well as expensive CPU time at a fixed time point. In addition, temporal discretization, which strongly depends on the stiffness of reactions and treatment of the high order derivatives (e.g. the diffusion term), may lead to severe stability conditions that require very small time steps, resulting in excessive computational cost.

Integration factor (IF) or exponential time differencing (ETD) methods are effective approaches to deal with temporal stability constraints associated with high order derivatives [45, 46, 47]. By treating linear operators of the highest order derivative exactly, IF or ETD methods are able to achieve excellent temporal stability [45, 48, 49]. To deal with additional stability constraints from stiff reactions, a class of semi-implicit integration factor (IIF) methods [50] were developed for implicit treatment of the stiff reactions. In the IIF approach, the diffusion term is solved exactly like the IF method while the nonlinear equations resulted from the implicit treatment of reactions is decoupled from the diffusion term to avoid solving large nonlinear systems involving both diffusions and reactions, such as in a standard implicit method for reaction-diffusion equations. IIF methods have a great stability property with its second order scheme being linearly unconditionally stable.

In IF or ETD type of methods, the dominant computational cost arises from the storage and calculation of exponentials of matrices resulting from discretization of the linear differential operators in the PDEs. To deal with this difficulty, compact representation of the discretization matrices was introduced in the context of IIF method [51]. In compact implicit integration factor method (cIIF), the discretized solutions are represented in a matrix form rather than a vector while the discretized diffusion operator are represented in matrices of much smaller size than the standard matrices for IIF while preserving the stability property of the IIF. For two or three dimensions, cIIF is significantly more efficient in both storage and CPU cost. In addition, cIIF method is robust in its implementation and integration with other spatial and temporal algorithms. It can handle general curvilinear coordinates as well as combine with adaptive mesh refinements in a straightforward fashion [52]. One can also apply cIIF to stiff reactions and diffusions while using other specialized hyperbolic solvers (e.g WENO methods [53, 54]) for convection terms to solve reaction-diffusion-convection equations efficiently [55]).

One alternative approach for IF (or ETD) methods to avoid storage of the exponentials of large matrices is to use Krylov subspace method to compute the multiplication between the vector and the exponentials of matrices without explicitly forming the matrices [56, 57]. The advantage of applying Krylov subspace method is that it can handle complicated diffusion operators, e.g. diffusion coefficients are spatial functions or elliptic operators contains cross derivatives, while cIIF in previous studies [51] can only handle systems of constant diffusion coefficients and Laplacian operators restricted to two and three dimensions. In contrast to cIIF, in which exponentials of matrices are pre-calculated only once and stored for repeated usages at each time step of the temporal updating, the Krylov subspace method needs to be carried out at each time step, leading to a significant increase in CPU time.

In this chapter, we introduce an array representation for the linear differential operators that may contain non-constant diffusion coefficients as well as cross-derivatives in two, three or higher dimensions. This array-representation approach is based on the idea of compact Implicit Integration Factor (cIIF), that is, when discretizing the terms with partial derivatives, regard the unknown solution as a vector with index connected to corresponding variables, while keeping other indexes fixed with unrelated variables. This new approach yields several discretization matrices of a small size that depend only on the number of derivatives in the continuous operators and the number of spatial discretization points in the direction of each derivative, in contrast to IF (or ETD) that requires exponentials of matrices whose size depend on the number of dimensions. In particular, the array representation can be incorporated into IIF to maintain the nice stability property of IIF as well as the implicit local treatment of the reactions decoupled from the diffusions. Like IIF, the second order array-representation (compact) implicit integration method (AcIIF) is A-stable. An operator splitting technique is incorporated into AcIIF for certain differential operators, resulting in non-commutable operations between discretization matrices. The AcIIF method is an extension of cIIF method that is able to deal with cross derivatives and non-constant diffusion coefficients in addition to other applications.

To study the accuracy and efficiency of AcIIF, we implement AcIIF methods and compare it with several other existing methods for both two and three dimensional reaction-diffusion equations. In addition, we apply AcIIF to solve Fokker-Planck equations in three and four dimensions. To demonstrate other applications of the array representation, we also use this approach to directly solve chemical master equations. In CMEs, the structure of the rate matrix for a reaction containing k species of molecules is very similar to the discretization matrix for a k-th order partial differential equation with cross derivatives. The overall direct simulations show the excellent properties of AcIIF and its distinct advantage in high spatial dimensions.

4.3 Array-representation (compact) Implicit Integration Factor Method (AcIIF)

4.3.1 Array representation for reaction-diffusion systems in three dimension without cross derivatives

To illustrate the array-representation approach, we first consider three-dimensional reaction-diffusion equations without cross-derivatives and with constant diffusion coefficients and periodic boundary conditions:

$$u_t(x,y,z,t) = D\Delta u(x,y,z,t) + f(u(x,y,z,t)), \tag{4.2}$$

where $(x,y,z) \in \Omega = \{0 < x,y,z < l\}$. Let N_x, N_y, N_z be the number of spatial grid points in each spatial direction and h_x, h_y, h_z be the grid size, respectively. Denote U_{k_1,k_2,k_3} as the approximated solution of u at the grid point (k_1h_x, k_2h_y, k_3h_z). The approximation of $D\partial^2/\partial x^2$ using the second order central difference discretization can be written in terms of multi-dimensional arrays, $U = (U_{k_1,k_2,k_3})$, through a linear mapping $\mathcal{L}_x$,

$$(\mathcal{L}_x U)_{k_1,k_2,k_3} := \frac{D}{h_x^2}\left(U_{k_1+1,k_2,k_3} - 2U_{k_1,k_2,k_3} + U_{k_1-1,k_2,k_3}\right) \tag{4.3}$$

where $1 \le k_1 \le N_x, 1 \le k_2 \le N_y$, and $1 \le k_3 \le N_z$. Similarly, using $\mathcal{L}_y$ and $\mathcal{L}_z$ to represent the approximations $D\partial^2/\partial y^2$ and $D\partial^2/\partial z^2$, respectively, Eq. (4.2) is approximated by

$$\frac{dU}{dt} = \mathcal{L}_x U + \mathcal{L}_y U + \mathcal{L}_z U + f(U). \tag{4.4}$$

Multiplying the integration factor, $e^{(\mathcal{L}_x+\mathcal{L}_y+\mathcal{L}_z)t}$, to both sides and integrating from t_n to t_{n+1}, two adjacent discretized temporal points, we derive a class of semi-implicit integration factor methods (IIF) after approximating the integral [50]. For example, the second order IIF takes the form

$$U^{n+1} - \frac{\Delta t}{2} f(U^{n+1}) = e^{(\mathcal{L}_x+\mathcal{L}_y+\mathcal{L}_z)\Delta t}(U^n + \frac{\Delta t}{2} f(U^n)) \tag{4.5}$$

where $U^n \approx U$ at time point t_n.

In a typical representation of the linear differential operator, the matrix $(\mathcal{L}_x + \mathcal{L}_y + \mathcal{L}_z)\Delta t$ has a size of $N_xN_yN_z \times N_xN_yN_z$. Although the matrix itself is sparse, its exponential is usually not, leading to prohibitive storage and computing cost for any fine spatial meshes. Next, we decompose this matrix into small matrices based on an array representation.

If one defines a vector by fixing the last two indices, k_2, k_3, of the the three-dimensional array U,

$$U(:, k_2, k_3) = (U_{1,k_2,k_3}, U_{2,k_2,k_3}, ..., U_{N_x,k_2,k_3})^T, \tag{4.6}$$

Then the three dimensional array U, can be treated as the collection of all such one dimensional vector on a two-dimensional array, with all k_2, k_3 going through from 1 to N_y and from 1 to N_z, respectively. We present this collection using symbol $\bigotimes$, with the super index indicates that this collection is along x_i axis, then we have:

$$U = \bigotimes_{\substack{1 \le k_2 \le N_y \\ 1 \le k_3 \le N_z}} U(:, k_2, k_3). \tag{4.7}$$

Next, we define a $N_x \times N_x$ matrix $M_x = D/h_x^2 W_{N_x \times N_x}$, where

$$W_{N \times N} = \begin{pmatrix} -2 & 1 & 0 & ... & 1 \\ 1 & -2 & 1 & ... & 0 \\ 0 & 1 & -2 & 1 & ... \\ ... & ... & ... & ... & ... \\ 1 & 0 & ... & 1 & -2 \end{pmatrix}_{N \times N} . \tag{4.8}$$

Then, $M_x U(:, k_2, k_3)$ represents the vector and matrix multiplication for any fixed pair of k_2, k_3. Using this approach, the linear mapping $\mathcal{L}_x$ in the array representation becomes,

$$\mathcal{L}_x U = \bigotimes_{\substack{1 \le k_2 \le N_y \\ 1 \le k_3 \le N_z}} M_x U(:, k_2, k_3). \tag{4.9}$$

Consequently, the exponential of $\mathcal{L}_x$ in the array representation takes the following form,

$$e^{\mathcal{L}_x} U = \bigotimes_{\substack{1 \le k_2 \le N_y \\ 1 \le k_3 \le N_z}} e^{M_x} U(:, k_2, k_3), \tag{4.10}$$

as induced from the relation,

$$(\mathcal{L}_x)^m U = \bigotimes_{\substack{1 \le k_2 \le N_y \\ 1 \le k_3 \le N_z}} (M_x)^m U(:, k_2, k_3), \forall m \in \mathbb{N}^+. \tag{4.11}$$

Applying the definition of linear mapping exponential yields Eq. (4.10).

Clearly, $\mathcal{L}_y$ and $\mathcal{L}_z$ have similar array representations,

$$\mathcal{L}_y U = \bigotimes_{\substack{1 \le k_1 \le N_x \\ 1 \le k_3 \le N_z}} M_y U(k_1, :, k_3), \quad \mathcal{L}_z U = \bigotimes_{\substack{1 \le k_1 \le N_x \\ 1 \le k_2 \le N_y}} M_z U(k_1, k_2, :), \tag{4.12}$$

where $M_y = D/h_y^2 W_{N_y \times N_y}$ and $M_z = D/h_z^2 W_{N_z \times N_z}$.

Using the array representations, one can easily show that the three linear mappings $\mathcal{L}_x, \mathcal{L}_y$ and $\mathcal{L}_z$ commute with each other, i.e.,

$$\mathcal{L}_a \mathcal{L}_b U = \mathcal{L}_b \mathcal{L}_a U, \qquad \text{for } a, b \in \{x, y, z\}. \tag{4.13}$$

This commuting property results in

$$e^{\mathcal{L}_x + \mathcal{L}_y + \mathcal{L}_z} U = e^{\mathcal{L}_x} e^{\mathcal{L}_y} e^{\mathcal{L}_z} U. \tag{4.14}$$

Direct application of Eq. (4.14) to Eq. (4.5) results in the following second order array-representation Implicit Integration Factor (AcIIF) method:

Algorithm 1 Second order AcIIF (AcIIF2)

$$U^{n+1} - \frac{\Delta t}{2} f(U^{n+1}) =$$

$$\bigotimes_{\substack{1 \le k_2 \le N_y \\ 1 \le k_3 \le N_z}} e^{M_x \Delta t} \left(\bigotimes_{\substack{1 \le k_1 \le N_x \\ 1 \le k_3 \le N_z}} e^{M_y \Delta t} \left(\bigotimes_{\substack{1 \le k_1 \le N_x \\ 1 \le k_2 \le N_y}} e^{M_z \Delta t} V(k_1, k_2, :) \right) (k_1, :, k_3) \right) (:, k_2, k_3), \tag{4.15}$$

where $V = U^n + \Delta t/2f(U^n)$.

Previously, a compact IIF (cIIF) was derived in a different fashion in two spatial dimensional systems by treating unknowns as a matrix, then the action of $\mathcal{L}_x$ is like a left product to the matrix and $\mathcal{L}_y$ is as its right product. And in three spatial dimensional cases, in addition to the left and right multiplications, a middle multiplication represents $\mathcal{L}_z$[51]. One major advantage of both cIIF and AcIIF methods that one only needs to compute the exponentials of M_x, M_y and M_z, which are much smaller matrices (only about $N \times N$), in comparison to standard IF or ETD methods [58, 59], for which the exponential $e^{(\mathcal{L}_x+\mathcal{L}_y+\mathcal{L}_z)\Delta t}$ of dimension of $N_xN_yN_z \times N_xN_yN_z$ are needed. Clearly, cIIF and AcIIF have significant savings in both CPU cost and storage, in particular, for equations in three or higher dimensions.

For the systems without cross-derivatives (e.g. Eq. (4.15)), the second order AcIIF (4.2) is equivalent to the second order cIIF method [51]. As it will be shown next, the advantage of AcIIF lies in its potential applications to reaction-diffusion systems with cross derivatives and non-constant diffusion coefficients for which cIIF is unable to achieve.

4.3.2 AcIIF method for three-dimensional reaction-diffusion systems with cross derivatives

Consider the reaction-diffusion equations with second order cross derivatives:

$$\begin{aligned} u_t = &\left(a_1\frac{\partial^2}{\partial x^2} + 2b_1\frac{\partial^2}{\partial x \partial y} + c_1\frac{\partial^2}{\partial y^2}\right)u + \left(a_2\frac{\partial^2}{\partial x^2} + 2b_2\frac{\partial^2}{\partial x \partial z} + c_2\frac{\partial^2}{\partial z^2}\right)u \\ &+ \left(a_3\frac{\partial^2}{\partial y^2} + 2b_3\frac{\partial^2}{\partial y \partial z} + c_3\frac{\partial^2}{\partial z^2}\right)u + f(u), \end{aligned} \tag{4.16}$$

in a cube, $\{(x, y, z) : 0 < x, y, z < l\}$, with periodic boundary conditions, satisfying the conditions $a_ic_i > b_i^2$, for $i = 1, 2, 3$. Applying a standard second order central difference approximation to $a_1\frac{\partial^2}{\partial x^2} + 2b_1\frac{\partial^2}{\partial x\partial y} + c_1\frac{\partial^2}{\partial y^2}$, one obtains its approximation, denoted by $\mathcal{L}_{xy}$, as the following

$$\begin{aligned}
&(\mathcal{L}_{xy}U)_{k_1,k_2,k_3} = \frac{a_1}{h_x^2}\left(U_{k_1+1,k_2,k_3} - 2U_{k_1,k_2,k_3} + U_{k_1-1,k_2,k_3}\right)\\
&\frac{2b_1}{4h_xh_y}\left(U_{k_1+1,k_2+1,k_3} + U_{k_1-1,k_2-1,k_3} - U_{k_1+1,k_2-1,k_3} - U_{k_1-1,k_2+1,k_3}\right)\\
&+\frac{c_1}{h_y^2}\left(U_{k_1,k_2+1,k_3} - 2U_{k_1,k_2,k_3} + U_{k_1,k_2-1,k_3}\right).
\end{aligned} \tag{4.17}$$

Using similar definitions for $\mathcal{L}_{yz}$ and $\mathcal{L}_{xz}$, the spatial approximation of Eq. (4.16) becomes

$$\frac{dU}{dt} = (\mathcal{L}_{xz} + \mathcal{L}_{xy} + \mathcal{L}_{yz})U + f(U). \tag{4.18}$$

To derive the array representation of the operator $\mathcal{L}_{xy}$, we first fix k_3 in $U(:,:,k_3)$ that represents a $N_xN_y \times N_xN_y$ matrix. Collect all these two dimensional matrices along a vector leads to:

$$U = \bigotimes_{1\le k_3\le N_z} U(:,:,k_3). \tag{4.19}$$

Define a linear mapping, $\mathcal{A}_{xy}$, from a matrix space consisting of all $N_x \times N_y$ matrices to itself as follows:

$$\begin{aligned}
(\mathcal{A}_{xy}M)_{i,j} &= \frac{2b_1}{4h_xh_y}\left(M_{i+1,j+1} + M_{i-1,j-1} - M_{i-1,j+1} - M_{i+1,j-1}\right)\\
&+\frac{a_1}{h_x^2}\left(M_{i+1,j} - 2M_{i,j} + M_{i-1,j}\right) + \frac{c_1}{h_y^2}\left(M_{i,j+1} - 2M_{i,j} + M_{i,j-1}\right).
\end{aligned} \tag{4.20}$$

Then, the array representation of $\mathcal{L}_{xy}$, in terms of $\mathcal{A}_{xy}$, and its exponential become

$$\mathcal{L}_{xy}U = \bigotimes_{1\le k_3\le N_z} \mathcal{A}_{xy}U(:,:,k_3), \quad e^{\mathcal{L}_{xy}}U = \bigotimes_{1\le k_3\le N_z} e^{\mathcal{A}_{xy}}U(:,:,k_3). \tag{4.21}$$

Similarly, the array representation for $\mathcal{L}_{yz}$ and $\mathcal{L}_{xz}$ may be written in terms of $\mathcal{A}_{yz}$ and $\mathcal{A}_{xz}$, respectively.

As long as $\mathcal{L}_{xy}$, $\mathcal{L}_{yz}$ and $\mathcal{L}_{xz}$ commute with each other, applying Eq. (4.5) using the array representation to Eq. (4.18) leads to the following algorithm:

Algorithm 2 AcIIF2 for reaction-diffusion systems with cross derivatives

$$\begin{aligned} &U^{n+1} - \frac{\Delta t}{2} f(U^{n+1}) = \\ &\bigotimes_{1\leq k_1\leq N_x} e^{\mathcal{A}_{yz}\Delta t} \left(\bigotimes_{1\leq k_2\leq N_y} e^{\mathcal{A}_{xz}\Delta t} \left(\bigotimes_{1\leq k_3\leq N_z} e^{\mathcal{A}_{xy}\Delta t} V(:,:,k_3) \right)(:,k_2,:) \right)(k_1,:,:), \end{aligned} \tag{4.22}$$

where $V = U^n + \Delta t/2 f(U^n)$.

In this algorithm, the exponential of $\mathcal{A}_{xy}$ is a $N_xN_y \times N_xN_y$ matrix, in comparison to $\mathcal{L}_{xy}$ that is a $N_xN_yN_z \times N_xN_yN_z$ matrix. Thus applying array representation leads to significant saving.

Cross derivatives may affect the commutable property of the discretized operators, resulting in the questions: under what conditions, $\mathcal{L}_{xy}$, $\mathcal{L}_{yz}$ and $\mathcal{L}_{xz}$ can commute with each other and what to do with the algorithms if the commuting property doesn't hold. In Section 3, we will give a sufficient condition for the commuting property. Alternatively, we next introduce a splitting technique to deal with the cases without such commuting property.

4.3.3 AcIIF method for reaction-diffusion systems with non-constant diffusion coefficients

When the diffusion coefficients in Eq. (4.16) are functions in space, minor modification is needed for the three discretization operators $\mathcal{L}_{xy}$, $\mathcal{L}_{yz}$, $\mathcal{L}_{xz}$ in array representations. Because each of the three operators depends on the other spatial dimension, we introduce a super index for $\mathcal{A}_{xy}$ to represent the operator at difference value of $z = k_3h_z$ for the array representation of $\mathcal{L}_{xy}$:

$$\mathcal{L}_{xy}U = \bigotimes_{1\leq k_3 \leq N_z} \mathcal{A}_{xy}^{k_3} U(:,:,k_3) \tag{4.23}$$

Similarly, the array representation of $\mathcal{L}_{yz}$ and $\mathcal{L}_{xz}$ can be written in terms of $\mathcal{A}_{yz}^{k_1}$ and $\mathcal{A}_{xz}^{k_2}$.

As a result, the three operators can no longer commute with each other. Notice that

$$e^{(\mathcal{L}_{xy}+\mathcal{L}_{yz}+\mathcal{L}_{xz})\Delta t} = e^{\mathcal{L}_{xy}\Delta t} e^{\mathcal{L}_{xz}\Delta t} e^{\mathcal{L}_{yz}\Delta t} + O(\Delta t^2), \tag{4.24}$$

the algorithm for Eq. (4.22) for this general case becomes only first order in time.

The order of accuracy can be improved by using a Strang splitting scheme to approximate $e^{(\mathcal{L}_{xy}+\mathcal{L}_{yz}+\mathcal{L}_{xz})\Delta t}$. In the Strang splitting method, two linear operators $\mathcal{L}_1$ and $\mathcal{L}_2$ defined on the same linear space have the following property [60]:

$$e^{(\mathcal{L}_1+\mathcal{L}_2)\Delta t} = e^{\mathcal{L}_1\Delta t/2} e^{\mathcal{L}_2\Delta t} e^{\mathcal{L}_1\Delta t/2} + R(\Delta t), \tag{4.25}$$

where

$$R(\Delta t) = (\mathcal{L}_2\mathcal{L}_1^2 d + \mathcal{L}_1^2\mathcal{L}_2 + 4\mathcal{L}_2\mathcal{L}_1\mathcal{L}_2 - 2\mathcal{L}_1\mathcal{L}_2\mathcal{L}_1 - 2\mathcal{L}_1\mathcal{L}_2^2 - 2\mathcal{L}_2^2\mathcal{L}_1)\Delta t^3/24. \tag{4.26}$$

For multiple linear operators, $\mathcal{L}_1, \mathcal{L}_2, ...\mathcal{L}_m$, the Strang splitting method can be extended to the following by induction:

$$e^{\sum_{i=1}^m \mathcal{L}_i \Delta t} = \prod_{i=1}^{m} e^{\mathcal{L}_i \Delta t/2} \prod_{i=m}^{1} e^{\mathcal{L}_i \Delta t/2} + O(\Delta t^3). \tag{4.27}$$

Now, applying Strang splitting to Eq. (4.5) leads to a second order method in both time and space:

$$U^{n+1} - \frac{\Delta t}{2} F(U^{n+1}) = e^{\frac{\Delta t}{2}\mathcal{L}_{xy}} e^{\frac{\Delta t}{2}\mathcal{L}_{xz}} e^{\Delta t \mathcal{L}_{yz}} e^{\frac{\Delta t}{2}\mathcal{L}_{xz}} e^{\frac{\Delta t}{2}\mathcal{L}_{xy}} \left(U^n + \frac{\Delta t}{2} F(U^n) \right). \tag{4.28}$$

Consequently, the array representation for solving Eq. (4.16) with non-constant diffusion coefficients leads to

Algorithm 3 AcIIF2 for system (4.16)

$$V = U^n + \frac{\Delta t}{2} f(U^n),$$

$$V^* =$$

$$\bigotimes_{1\leq k_1 \leq N_x} e^{\mathcal{A}_{yz}^{k_1}\Delta t} \left(\bigotimes_{1\leq k_2 \leq N_y} e^{\mathcal{A}_{xz}^{k_2}\Delta t/2} \left(\bigotimes_{1\leq k_3 \leq N_z} e^{\mathcal{A}_{xy}^{k_3}\Delta t/2} V(:,:,k_3) \right) (:,k_2,:) \right) (k_1,;,:), \tag{4.29}$$

$$U^{n+1} - f(U^{n+1}) = \bigotimes_{1\leq k_3 \leq N_z} e^{\mathcal{A}_{xy}^{k_3}\Delta t/2} \left(\bigotimes_{1\leq k_2 \leq N_y} e^{\mathcal{A}_{xz}^{k_2}\Delta t/2} V^*(:,k_2,:) \right) (:,:,k_3).$$

Operator splitting leads to twice as many exponential-matrix and vector multiplication compared to the non-splitting case in Algorithm 2. Therefore, it is important to use appropriate order of splitting if a subset of operators can commute with each other to improve computational efficiency. For instance, for three operators $\mathcal{L}_i, i = 1,2,3$ where $\mathcal{L}_1$ and $\mathcal{L}_2$ can commute, however, $\mathcal{L}_3$ cannot, one may have two different kinds of splittings:

$$e^{(\mathcal{L}_1+\mathcal{L}_2+\mathcal{L}_3)\Delta t} = e^{\mathcal{L}_3\Delta t/2} e^{\mathcal{L}_2\Delta t} e^{\mathcal{L}_1\Delta t} e^{\mathcal{L}_3\Delta t/2} + O(\Delta^3 t) \tag{4.30}$$

and

$$e^{(\mathcal{L}_1+\mathcal{L}_2+\mathcal{L}_3)\Delta t} = e^{\mathcal{L}_1\Delta t/2} e^{\mathcal{L}_2\Delta t/2} e^{\mathcal{L}_3\Delta t} e^{\mathcal{L}_2\Delta t/2} e^{\mathcal{L}_1\Delta t/2} + O(\Delta^3 t). \tag{4.31}$$

Clearly, the splitting in Eq. (4.30) computes one fewer exponential matrix and vector multiplication than the splitting in Eq. (4.31).

4.3.4 AcIIF method for high dimensional reaction-diffusion systems

We next extend AcIIF to the reaction-diffusion equation in d spatial dimensions with $d \geq 3$:

$$u_t = \sum_{1 \leq i < j \leq d} L_{x_i x_j} u + f(u) \quad 0 < x_i < 1 \tag{4.32}$$

where

$$L_{x_i x_j} := a_{ij} \frac{\partial^2}{\partial x_i^2} + 2b_{ij} \frac{\partial^2}{\partial x_i \partial x_j} + c_{ij} \frac{\partial^2}{\partial x_i^2}, \quad 1 \leq i < j \leq d, \tag{4.33}$$

and we assume that diffusion coefficients, a_{ij}, b_{ij} and c_{ij} are spatial functions that satisfy the elliptical conditions:

$$a_{ij} > 0, c_{ij} > 0, a_{ij} c_{ij} > b_{ij}^2. \tag{4.34}$$

We also assume that the boundary conditions for the system are periodic.

Similar to the three dimensional case, in each direction x_i, there are N_{x_i} grid points with the grid size of h_{x_i}. We use a $N_{x_1} \times N_{x_2} \times ... N_{x_d}$ d-dimensional array $U = (U_{k_1, k_2, ..., k_d}), 1 \leq k_i \leq N_{x_i}, i = 1, 2, ..., d$ to represent the solution, and $\mathcal{L}_{x_i x_j}$ to represent the discretized operator of $L_{x_i x_j}$, $1 \leq i < j \leq d$. Next, we denote $U|_{x_i, x_j}^{(k_r), r \neq i,j}$ as the matrix derived from U by fixing the dimensional index $k_r, r \neq i, j$. Thus, the array representation of $\mathcal{L}_{x_i x_j}$ becomes

$$\mathcal{L}_{x_i x_j} U = \bigotimes_{\substack{1 \leq k_r \leq N_{x_r} \\ r \neq i, j}} \mathcal{A}_{x_i x_j}^{(k_r), r \neq i,j} \, U|_{x_i, x_j}^{(k_r), r \neq i,j} \tag{4.35}$$

where $\mathcal{A}_{x_i x_j}^{(k_r), r \neq i,j}$ are linear mappings from the matrix space with all $N_{x_i} \times N_{x_j}$ matrices to itself and is similarly defined in the three-dimensional case.

If $\mathcal{L}_{x_i x_j}$ commute with each other, we are able to directly apply array representation to the IIF2 method to obtain a second order AcIIF method for solving Eq. (4.32):

$$U^{n+1} - \frac{\Delta t}{2}F(U^{n+1}) = \bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq 1,2}} e^{\mathcal{A}_{x_1x_2}^{(k_r),r\neq 1,2}\Delta t} \bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq 1,3}} e^{\mathcal{A}_{x_1x_3}^{(k_r),r\neq 1,3}\Delta t} \dots \bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq d-1,d}} e^{\mathcal{A}_{x_{d-1}x_d}^{(k_r),r\neq d-1,d}\Delta t} V. \tag{4.36}$$

If $\mathcal{L}_{x_ix_j}$ are not commutable, we apply Strang splitting and array representation to obtain the second order AcIIF method:

$$\begin{aligned} U^{n+1} - \frac{\Delta t}{2}F(U^{n+1}) = & \bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq 1,2}} e^{\mathcal{A}_{x_1x_2}^{(k_r),r\neq 1,2}\Delta t/2} \dots \bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq d-1,d}} e^{\mathcal{A}_{x_{d-1}x_d}^{(k_r),r\neq d-1,d}\Delta t/2} \\ & \bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq d-1,d}} e^{\mathcal{A}_{x_{d-1}x_d}^{(k_r),r\neq d-1,d}\Delta t/2} \dots \bigotimes_{\substack{1\le k_r\le N_{x_r}\\ r\neq 1,2}} e^{\mathcal{A}_{x_1x_2}^{(k_r),r\neq 1,2}\Delta t/2} V, \end{aligned} \tag{4.37}$$

where $V = U^n + \Delta t/2F(U^n)$ and $1 \le k_r \le N_{x_r}, r = 1, 2, ..., d$.

4.3.5 A sufficient condition for operator commuting

As evident in Strang splitting, proper choice of the order of operators Eq. (4.29) will decrease the computational cost, which can be improved if commuting operators can be found. Now, we give a sufficient condition for commutable operators.

Proposition 1 *All linear operators $\mathcal{L}_{x_ix_j}, 1 \le i < j \le d$ commute with each other if the system in Eq. (4.32) satisfies:*

(1) the diffusion coefficients are constant,

(2) the boundary conditions are periodic along each direction.

PROOF With a given basis, the linear operator $\mathcal{L}_{x_i x_j}$ for the central difference discretization are $N_1 N_2 ... N_d \times N_1 N_2 ... N_d$ matrices in the following form,

$$\begin{pmatrix} m_1 & m_2 & ... & m_{N-1} & m_N \\ m_N & m_1 & m_2 & ... & m_{N-1} \\ m_{N-1} & m_N & m_1 & ... & m_{N-2} \\ ... & ... & ... & ... & ... \\ m_2 & m_3 & ... & m_N & m_1 \end{pmatrix} \tag{4.38}$$

where $N = N_1 N_2 ... N_d$ and $m_i, i = 1, 2, ..., N$ are real. Let two matrices $A = (a_i)_{N\times N}$ and $B = (b_i)_{N\times N}$ both take the form of Eq. (4.38). One can show directly that

$$(AB)_{ij} = \sum_{k=1}^{N} a_{k-i+2} b_{j-k+1} = \sum_{s=1}^{N} b_{s-i+2} a_{j-s+1} = (BA)_{ij}, \quad \forall i, j, \tag{4.39}$$

because $a_{i\pm N} = a_i$, $b_{j\pm N} = b_j$ for $\forall i, j$ where $s = j + i - k - 1$. Q.E.D

This shows that Strang splitting is unnecessary for constant diffusion coefficients in high spatial dimension and Algorithm 2 is applicable for such reaction-diffusion equations.

4.4 Stability analysis, higher-order methods, and computational costs

Next, we study the linear stability of second order AcIIF methods, derive a third method, and discuss the computational costs of the methods.

4.4.1 Stability analysis

Based on linear stability analyses in [50] and [51], we claim that the second order AcIIF methods, Eq. (4.36) and Eq. (4.37), are asymptotically stable for the case of $F(U) = dU$ and $L_{x_ix_j}u = -cu$, where $d < 0$ and $c > 0$ correspond to stable reactions and elliptic operators. For such a linear case, one has

$$u^{n+1} = e^{-c\Delta t}\left(u^n + \frac{d\Delta t}{2}u^n\right) + \frac{d\Delta t}{2}u^{n+1}. \tag{4.40}$$

Assuming $u^n = e^{in\theta}$, we obtain

$$e^{i\theta} = e^{-c\Delta t}\left(1 + \frac{1}{2}\lambda\right) + \frac{1}{2}\lambda e^{i\theta}, \tag{4.41}$$

where $\lambda = d\Delta t$ has a real part λ_r and imaginary part λ_i, leading to

$$\begin{aligned} \lambda_r &= \frac{2(1-e^{-2c\Delta t})}{(1-e^{-c\Delta t})^2 + 2(1+\cos\theta)e^{-c\Delta t}}, \\ \lambda_i &= \frac{4(\sin\theta)ce^{-c\Delta t}}{(1-e^{-c\Delta t})^2 + 2(1+\cos\theta)e^{-c\Delta t}}, \end{aligned} \tag{4.42}$$

since $c > 0$ and $\lambda_r > 0$ for $0 \leq \theta \leq 2\pi$. Then, the second order AcIIF is A-stable since the stability region includes the complex plane for all λ with $\lambda_r < 0$.

If we apply AcIIF methods Eq. (4.36) and Eq. (4.37) to Fokker-Planck equations or chemical master equations, where in each the operator $\mathcal{L}_{x_ix_j}$ defines a Markov process,

$$\frac{dU}{dt} = \mathcal{L}_{x_ix_j}U, \tag{4.43}$$

then we claim that AcIIF methods Eq. (4.36) and Eq. (4.37) are still A-stable.

In order to show the A-stability, we prove that, under certain norm, $\left\|e^{\mathcal{L}_{x_ix_j}\Delta t}\right\| \leq 1$ holds for any $\Delta t > 0$ for such $\mathcal{L}_{x_ix_j}$. Then, each $\mathcal{L}_{x_ix_j}$ can be treated as elliptic operators and the remaining proof goes through Eq. (4.40) to Eq. (4.42). Since Eq. (4.43) defines a Markov process, the total probability of all states, $\sum U(\Delta t) = \sum e^{\mathcal{L}_{x_i x_j}\Delta t} U(0)$, maintains a value of 1 for any time step, when $U(0)$ is a proper probability distribution, i.e. $U(0) > 0$ for each compartment and $\sum U(0) = 1$. Using the maximum norm $||.||_1$ and defining the corresponding linear operator norm, we first prove that for any U with $U > 0$,

$$||e^{\mathcal{L}_{x_i x_j}\Delta t} U||_1 = \sum \left| e^{\mathcal{L}_{x_i x_j}\Delta t} \left(\frac{U}{\sum U} \right) \right| \sum U = ||U||_1. \tag{4.44}$$

Then, for $U = U^+ - U^-$ where U^+ is with all positive compartments of U,

$$\begin{aligned} \left\| e^{\mathcal{L}_{x_i x_j}\Delta t} \right\| &= \max_{U \neq 0} \frac{||e^{\mathcal{L}_{x_i x_j}\Delta t} U||_1}{||U||_1} = \max_{U \neq 0} \frac{\left\| e^{\mathcal{L}_{x_i x_j}\Delta t} U^+ - e^{\mathcal{L}_{x_i x_j}\Delta t} U^- \right\|}{\sum |U|} \\ &\leq \max_{U \neq 0} \left(\frac{||e^{\mathcal{L}_{x_i x_j}\Delta t} U^+||_1}{\sum |U|} + \frac{||e^{\mathcal{L}_{x_i x_j}\Delta t} U^-||_1}{\sum |U|} \right) = 1. \end{aligned} \tag{4.45}$$

Next, we can replace each $\mathcal{L}_{x_i x_j}$ in the Fokker-Planck equation or the chemical master equation by a negative scalar and the proof of A-stability for Eq. (4.36) and Eq. (4.37) for these two cases is done.

4.4.2 High order AcIIF method

If discretization operators are commutable, higher order (in time) AcIIF methods can be derived from the IIF method in a similar manner. For example, the third order IIF scheme [50] has the form:

$$U^{n+1} = e^{\mathcal{L}\Delta t} U^n + \Delta t \left(\frac{5}{12} F(U^{n+1}) + \frac{2}{3} e^{\mathcal{L}\Delta t} F(U^n) - \frac{1}{12} e^{2\mathcal{L}\Delta t} F(U^{n-1}) \right) \tag{4.46}$$

where $\mathcal{L} = \sum_{i,j} \mathcal{L}_{x_i x_j}$. If all $\mathcal{L}_{x_i x_j}, \forall i, j$ commute with each other, we then obtain

$$e^{\mathcal{L}} U = \prod_{1 \leq i < j \leq d} e^{\mathcal{L}_{x_i x_j}} U. \tag{4.47}$$

If all discretized operators $\mathcal{L}_{x_i x_j}$ commute with each other, applying the array representation leads to the third order AcIIF method:

Algorithm 4 (Third order AcIIF method)

$$\begin{aligned} U^{n+1} - \frac{5\Delta t}{12} F(U^{n+1}) = & \bigotimes_{(k_r), r \neq 1,2} e^{\mathcal{A}_{x_1 x_2} \Delta t} \dots \bigotimes_{(k_r), r \neq d-1,d} e^{\mathcal{A}_{x_{d-1} x_d} \Delta t} V_1 \\ & - \bigotimes_{(k_r), r \neq 1,2} e^{2\mathcal{A}_{x_1, x_2} \Delta t} \dots \bigotimes_{(k_r), r \neq d-1,d} e^{2\mathcal{A}_{x_{d-1} x_d} \Delta t} V_2 \end{aligned} \tag{4.48}$$

$$V_1 = U^n + \frac{2\Delta t}{3} F(U^n), \quad V_2 = \frac{\Delta t}{12} F(U^{n-1}) \tag{4.49}$$

where $1 \leq k_r \leq N_r, r = 1, 2, ..., d$.

In the case that some $\mathcal{L}_{x_i x_j}$ are not commutable, the splitting techniques may be applied to this subset of operators to achieve high order accuracy. However, since the formulation becomes much more tedious and complicated, we omit them here.

Remarks on higher order derivatives: The array representation can also be extended for equations with operators that contain high order and cross derivatives, such as those in the following form

$$\frac{\partial^m}{\partial x_{i_1} \partial x_{i_2} ... \partial x_{i_m}}. \tag{4.50}$$

Similarly, the second order central difference approximation in the multi-dimensional array U representation results in the discretization linear operator $\mathcal{L}_{x_{i_1}, x_{i_2}, ..., x_{i_m}}$ of the following compact form,

$$\mathcal{L}_{x_{i_1},x_{i_2},...,x_{i_m}} U = \bigotimes_{1 \leq k_r \leq N_r, r \neq i_1, i_2, ..., i_m} \mathcal{A}_{x_{i_1} x_{i_2} ... x_{i_m}} \; U|^{(k_r), r \neq i_1, i_2, ..., i_m}_{x_{i_1}, x_{i_2}, ..., x_{i_m}} \tag{4.51}$$

where $U|^{(k_r),r\neq i_1,i_2,...,i_m}_{x_{i_1},x_{i_2},...,x_{i_m}}$ is a m-dimensional array by fixing the index $k_r, r \neq i_1, i_2, ..., i_m$ of U and $\mathcal{A}_{x_{i_1} x_{i_2} ... x_{i_m}}$, resulting from the central difference approximation, represents a linear mapping from m-dimensional array linear space to itself.

4.4.3 Computational cost

For stiff reaction-diffusion equations, the size of the time step usually dictates the overall cost of the temporal updating method. For an A-stable method such as AcIIF and IIF, the cost mainly results from the formation of the exponential-matrix and the corresponding vector-matrix multiplication during each time step. In array representation, small matrices of size in $N_i \times N_i$ for the Laplacian operator or $N_i N_j \times N_i N_j$ when the second order cross derivatives are presented in contrast to IIF in which the exponential of a $N_1 N_2 ... N_d \times N_1 N_2 ... N_d$ matrix is required. The advantage of AcIIF becomes more prominent for three or higher dimensional systems.

For a d-spatial dimensional case ($d \geq 3$) with second order cross derivatives, the computational cost for manipulating the exponential matrices in IIF is $O((N_1 N_2 ... N_d)^2)$, or $O(N^{2d})$ for $N_i = N, i = 1, 2, ..., d$, while the corresponding cost for AcIIF is

$$\sum_{1 \leq i < j \leq d} O((N_i N_j)^2) \frac{N_1 N_2 ... N_d}{N_i N_j}. \tag{4.52}$$

For the case of non-constant coefficients in diffusion, it is $O(d^2 N^{d+2})$ when $N_i = N, i = 1, 2, ..., d$. For example, a six-dimensional system requires calculating an exponential of matrix with an approximated size of $10^8 \times 10^8$ when $N = 20$, in contrast to the AcIIF method that only needs exponentials of matrices of a size of 400×400.

Because the exponential-matrices are small in AcIIF, one may pre-calculate the exponential matrices once and store them during the calculations. An alternative approach, which is particularly useful for matrices with sizes exceeding the memory size, is to compute the exponential-matrix vector multiplication without explicit formation of the matrices through, for example, the Krylov subspace method [56, 61, 62]. In the direct simulations shown in the next section, we implement Padé approximation, which has a computational cost of $O(N^2)$ (both in storage and time) to compute a matrix exponential of $N \times N$ matrix [63], for reaction-diffusion equations with or without cross-derivatives in three dimensions, and we use Krylov subspace method for the Fokker-Planck equations in three or four dimensions and chemical master equations.

4.4.4 Array representation for Chemical Master Equations

Chemical master equations (CME) is a system of first-order ordinary differential equations for stochastic description of the time evolution of a network of biochemical reactions [64]. The solution of the system yields the probability density vector at discrete states of the biochemical network in time. The system is typically stiff and it can have many components and states, presenting difficulties for numerical methods and simulations [65, 66]. As seen below, the array representation provides a convenient approach to decompose the large rate matrix into smaller matrices for efficient usage of integration factor methods that can best deal with stiffness in the system.

If a given chemical reaction system consists of of d molecular species, namely $X_1, X_2, ..., X_d$, with maximal copy numbers of $N_1, N_2, ..., N_d$, respectively, then the system has $N_{tot} = N_1 N_2 ... N_d$ possible states, for which a vector $x = (x_1, x_2, ..., x_d)$ denotes each state. The

$R-th$ reaction takes the form

$$R_r:\ w_1^r X_1 + w_2^r X_2 + ... + w_d^r X_d \to v_1^r X_1 + v_2^r X_2 + ... + v_d^r X_d, \tag{4.53}$$

where v_i^r and w_i^r, for every i and r, are non-negative integers and the system contains R number of reactions. The reaction rate at state x is $a_r(x) = a_r(x_1, x_2, ..., x_d)$ and the vector

$$\alpha^r = (v_1^r - w_1^r, v_2^r - w_2^r, ..., v_d^r - w_d^r) \tag{4.54}$$

denotes the change of copy number of the molecular species after the $r-th$ reaction occurs once.

In array representation, one can use an $N_1 \times N_2 ... \times N_d$ d-dimensional array U to denote the probability density function, and each component of U, U_x, as the probability density at state x, which can be written as

$$U_x(t) = \mathrm{Prob}[X_1 = x_1, X_2 = x_2, ..., X_d = x_d, \text{at time } t]. \tag{4.55}$$

Define the linear mapping $\mathcal{L}_r$,

$$(\mathcal{L}_r U)_x = a_r(x-\alpha^r)U_{x-\alpha^r}(t) - a_r(x)U_x. \tag{4.56}$$

The CME for the probability density functions becomes

$$\dot{U}(t) = \sum_{r=1}^{R} \mathcal{L}_r U(t). \tag{4.57}$$

To introduce the array representation for $\mathcal{L}_r$, we let $i_1^r, i_2^r, ..., i_{m_r}^r$ denote the indices of non-zero entries in α^r. Using the same notation as in Eq. (4.51), $U|^{(x_j),j\neq i_1^r, i_2^r, ..., i_{m_r}^r}_{X_{i_1^r}, X_{i_2^r}, ..., X_{i_{m_r}^r}}$ denotes a

m_r-dimensional array by fixing indexes $x_j, j \neq i_1^r, i_2^r, ..., i_{m_r}^r$. Then, one obtains

$$U = \bigotimes_{\substack{1 \leq x_j \leq N_j \\ j \neq i_1^r, ..., i_{m_r}^r}} U|_{X_{i_1^r}, X_{i_2^r}, ..., X_{i_{m_r}^r}}^{(x_j), j \neq i_1^r, i_2^r, ..., i_{m_r}^r}. \tag{4.58}$$

Define the linear mapping $\mathcal{A}_{X_{i_1^r}, X_{i_2^r}, ..., X_{i_{m_r}^r}}^{(x_j),, j \neq i_1^r, i_2^r, ..., i_{m_r}^r}$ on m_r-dimensional array V as

$$\begin{aligned} &(\mathcal{A}_{X_{i_1^r}, X_{i_2^r}, ..., X_{i_{m_r}^r}}^{(x_j),, j \neq i_1^r, i_2^r, ..., i_{m_r}^r} V)_{(x_i), i \in i_1^r, i_2^r, ..., i_{m_r}^r} \\ &= a_r(x - \alpha^r) V_{(x_i - \alpha_i^r), i \in i_1^r, i_2^r, ..., i_{m_r}^r} - a_r(x) V_{(x_i), i \in i_1^r, i_2^r, ..., i_{m_r}^r}. \end{aligned} \tag{4.59}$$

Then the array representation of $\mathcal{L}_r$ becomes

$$\mathcal{L}_r U = \bigotimes_{\substack{1 \leq x_j \leq N_j \\ j \neq i_1^r, i_2^r, ..., i_{m_r}^r}} \mathcal{A}_{X_{i_1}, X_{i_2}, ..., X_{i_{m_r}}}^{(x_j), j \neq i_1^r, i_2^r, ..., i_{m_r}^r} \; U|_{X_{i_1}, ..., X_{i_{m_r}}}^{(x_j), j \neq i_1^r, i_2^r, ..., i_{m_r}^r}. \tag{4.60}$$

For typical systems, each $\mathcal{L}_r, r = 1, 2, ..., R$ cannot commute with one another, thus the Strang splitting method is applied to approximate the solution, resulting in a second order integration factor method (AcIIF2) for CME Eq. (4.57):

$$U^{n+1} = \prod_{r=1}^{R} e^{\mathcal{L}_r \Delta t/2} \prod_{r=R}^{1} e^{\mathcal{L}_r \Delta t/2} U^n, \tag{4.61}$$

where U^n denotes the probability density functions at time $t_n = n\Delta t$.

The exponential of $\mathcal{L}_r$ can be written in terms of the exponentials of $\mathcal{A}_{X_{i_1}, X_{i_2}, ..., X_{i_{m_r}}}^{(x_j), j \neq i_1^r, i_2^r, ..., i_{m_r}^r}$. If the reaction R_r only affects copy numbers of a few species, implying m_r is small, the calculation of the latter exponential is much more efficient than computing the original one. In other words, the array representation saves storage and CPU time for the system containing many molecular species while each reaction only affects the copy number of a small portion of species.

4.5 Numerical simulations

To explore various applications of the AcIIF methods (Eq. (4.36) and Eq. (4.37)), we apply the second order AcIIF methods to five different systems: three-dimensional reaction-diffusion equations with constant diffusion coefficients or spatially-dependent diffusion constants; three- and four-dimensional Fokker-Planck equations; and chemical master equations arising from a biological application.

4.5.1 Three-dimensional reaction-diffusion equation with constant diffusion coefficients

We first apply AcIIF method Eq. (4.36) to the following reaction-diffusion equation

$$\begin{aligned} u_t = &(0.1u_{xx} - 0.15u_{xy} + 0.1u_{yy}) + (0.1u_{xx} + 0.2u_{xz} + 0.2u_{zz}) \\ &+ (0.2u_{yy} + 0.15u_{yz} + 0.1u_{zz}) + 0.8u, \end{aligned} \tag{4.62}$$

where $x, y, z \in (0, 2\pi)$ with periodic boundary conditions. With the initial condition $u(x, y, z, 0) = \sin(x + y + z)$, the equation has the exact solution $u(x, y, z, t) = e^{-0.2t} \sin(x + y + z)$.

Based on the result from Subsection 4.3.5, for this case, the corresponding linear operators $\mathcal{L}_{xy}, \mathcal{L}_{yz}$ and $\mathcal{L}_{xz}$ can commute with each other. Thus, Eq. (4.36) is a second order scheme in both time and space. We first compare the second order array-representation compact IIF with the standard IIF, both in second order. Because both methods are A-stable, we choose $\Delta t = 1/N = h_x/2\pi$ where $N_x = N_y = N_z = N$ and $\Delta t = 1/N = h_x/2\pi$, and simulation results are evaluated at $t = 1$. As seen in Table 4.1, both methods clearly show second order accuracy with similar sizes of errors as N increases, as one may expect from the analysis of both methods. On the other hand, we observe the CPU time for both methods to achieve the same accuracy is much larger in IIF than in AcIIF, because the exponential matrices in IIF have much larger size than AcIIF. When N becomes 32, IIF fails to compute as the size for matrix exponential becomes exceedingly large, leading to a lack of sufficient memory in a Matlab implementation on typical personal computers (4GB). Even in a cluster where computing a $32^3 \times 32^3$ matrix exponential is possible, the CPU time needed will be about 2 hours, and computation of a $64^3 \times 64^3$ matrix exponential takes more than a day. On the other hand, AcIIF runs normally with good accuracy, showing clear advantages in handling larger grid numbers for convergence of solutions.

4.5.2 Three-dimensional diffusion reaction system with non-constant diffusion coefficients

To test the case with non-commutable differential operators, we consider the following reaction-diffusion equations with non-constant coefficients:

$$\begin{aligned} u_t = &(0.5u_{xx} - 0.5\sin(x+y)u_{xy} + 0.5u_{yy}) + (0.5u_{xx} - 1/3\cos y u_{xz} + 1/3u_{zz}) \\ &+ (1+\cos x)\,(0.5u_{yy} - 0.5u_{yz} + 1/3u_{zz})) + f(x,y,z,u), \end{aligned} \tag{4.63}$$

where $x, y, z \in (0, 2\pi)$ with periodic boundary conditions. With the initial condition $u(x,y,z,0) = \sin(x+y+z)$, the equation has the exact solution $u(x,y,z,t) = e^{-0.2t}\sin(x+y+z)$. Similar to the previous case, we choose $N_x = N_y = N_z = N$, grid size $h_x = h_y = h_z = 2\pi/N$, $\Delta t = 1/N$, and $t = 1$ as the temporal point for evaluating the method.

In this non-commutable case, we need to compute N number of array-representation operators for each of $\mathcal{L}_{xy}, \mathcal{L}_{xz}$ and $\mathcal{L}_{yz}$. For example, to compute $e^{\mathcal{L}_{yz}}$, we need the following calculations

$$\begin{aligned} &\mathcal{A}_{yz}^{k_x} \sim (1+\cos((k_x-1)h_x)((0.5(\cdot)_{yy} - 0.5(\cdot)_{yz} + 1/3(\cdot)_{zz}), \\ &e^{\mathcal{L}_{yz}} = \bigotimes_{k_x} \mathcal{A}_{yz}^{k_x} \, U|_{yz}^{k_x} \end{aligned} \tag{4.64}$$

for $k_x = 1, 2, ..., N$. In the commutable case, we only compute and save three of the exponential matrices of size $N^2 \times N^2$, in contrast to $3N$ of exponential matrices of the same size. As a result, the non-commutable case takes significantly more CPU time than the commutable case as seen in Table 4.1 and Table 4.2. However, compared to the standard IIF method, AcIIF is still significantly much faster.

The order of accuracy for both AcIIF2 and IIF2 remain second order, as seen in Table 4.2. However, the error for the non-commutable case is larger than the commutable case at the same spatial and temporal resolutions, which is likely due to the splitting error in time. Similar to the constant diffusion case, IIF2 fails to run due to the memory problem for relatively larger N.

4.5.3 Three- and four- dimensional Fokker-Planck equations

The Fokker-Planck equation (FPE) describes the time evolution of the probability density function of stochastic systems [44]. The generalized FPE usually takes the following form

$$\frac{\partial p(x,t)}{\partial t} = -\sum_{r=1}^{R} \left\{ \sum_{i}^{N} n_{ri} \frac{\partial}{\partial x_i} \left(q_r(x,t) - \frac{1}{2} \sum_{j=1}^{N} n_{rj} \frac{\partial q_r(x,t)}{\partial x_j} \right) \right\}. \tag{4.65}$$

Here, in the case of bio-chemical reactions, R denotes the total number of chemical reactions involved in the system, N denotes the total number of different species participating the reactions, x_j denotes the copy number of j-th reactant, and n_{ri} denotes the change of copy number of reactant i when the r-th reaction occurs once. $p(x,t)$ represents the probability density of the system at the state $x = (x_1, x_2, ..., x_N)$ ($x \in R^{N+}$) and time t. In addition, we define

$$q_r(x,t) = w_r(x,t)p(x,t), \tag{4.66}$$

where $w_r(x,t)$ is the reaction propensity function for r-th reaction at state x. For example, for the following bio-chemical reactions,

$$X \xrightarrow{k_1[X]} Y, X+Y \xrightarrow{k_2[X][Y]} Z, \emptyset \xrightarrow{k_3} Z, \tag{4.67}$$

we have $n_1 = (-1,1,0), n_2 = (-1,-1,1), n_3 = (0,0,1)$ and

$$w_1(x,y,z,t) = k_1x, w_2(x,y,z,t) = k_2xy, w_3(x,y,z,t) = k_3z. \tag{4.68}$$

In general, FPE is a N-dimensional convection-diffusion equations with non-constant diffusive coefficients and second order cross derivatives. Because the system may be stiff, implicit temporal methods, such as Crank-Nicolson method [67], which requires solving nonlinear systems of large size at each time step, are often needed. While directly apply IIF method, the calculation of the huge matrix exponential is unaffordable in the high dimensional case. AcIIF, which has the good stability like Crank-Nicolson, is a better choice in solving FPE than IIF method as it divides the entire discretization matrix into multiple small pieces by the array-representation technique.

To apply AcIIF to FPE, we first study a three-dimensional case in which there are two metabolites and one enzyme, which is also studied in [67]. The reactions are:

$$\begin{gathered}
\emptyset \xrightarrow{k_1} A \qquad \emptyset \xrightarrow{k_2} B \\
A+B \xrightarrow{k_3} \emptyset \\
A \xrightarrow{k_4} \emptyset \qquad B \xrightarrow{k_5} \emptyset \\
\emptyset \xrightarrow{k_6} E_A \quad E_A \xrightarrow{k_7} \emptyset
\end{gathered} \tag{4.69}$$

The corresponding propensity rates are given as

$$\begin{array}{c} k_1 = \frac{k_A[E_A]}{1+[A]/K_I} \quad k_2 = k_B \quad k_3 = k[A][B] \\ k_4 = \mu[A] \quad k_5 = \mu[B] \quad k_6 = \frac{k_{E_A}}{1+[A]/K_R} \quad k_7 = \mu[E_A] \end{array} \tag{4.70}$$

where $k_A = 0.3s^{-1}, k_B = 2s^{-1}, K_I = 30, k = 0.001s^{-1}, \mu = 0.004s^{-1}, K_R = 30$ and $k_{E_A} = 1s^{-1}$ [67].

The computational domain for this system is chosen to be $\Omega_h = [0, 100] \times [0, 100] \times [0, 45]$, which is large enough such that the probability of $[A] > 100, [B] > 100, [E_A] > 45$ is sufficiently small, implying that the domain covers nearly all the possible states of the chemical reactions. After discretizing the FPE using second order central differences, we represent the density function by a three-dimensional array $U(t)$ to represent the density function. Each component $U_{i_1,i_2,i_3}(t)$ denotes the probability density for system at time t and state $[A] = i_1, [B] = i_2, [E_A] = i_3$. There are seven reactions, thus, in FPE Eq. (4.65), corresponding to $R = 7$. For $r-$th reaction, the corresponding discretized operator, denoted by $\mathcal{L}_r$, becomes

$$\mathcal{L}_r = \sum_{i}^{N} n_{ri} \frac{\partial}{\partial x_i} \left(q_r(x,t) + \frac{1}{2} \sum_{j=1}^{N} n_{rj} \frac{\partial q_r(x,t)}{\partial x_j} \right). \tag{4.71}$$

Because $\mathcal{L}_r$ contains no cross derivatives for $r \neq 3$, we can use the array representation presented in Subsection 4.3.1. On the other hand, $\mathcal{L}_3$ contains a cross derivative $\partial^2/\partial[A]\partial[B]$, we use the array representation presented in Subsection 4.3.3. By direct application of AcIIF (Eq (4.37) based on splitting technique), we obtain an overall second order method. In particular, some of the reactions can be grouped into one matrix to reduce the number of splittings and number of calculations of exponential matrices, such as $\mathcal{R}_1$ and $\mathcal{R}_4$, which both have $\partial^2/\partial[A]^2$ and $\partial/\partial[A]$ in Eq. (4.71).

To study the performance of AcIIF2, we also implement the second order Runge Kutta (RK2) method for a comparison. The error of solution in the maximal norm is based on a simulation result from the finest "spatial" grid ($N_A = N_B = 200, N_{E_A} = 120$) and finest time step ($\Delta t = 5 \times 10^{-3}$). The initial condition for each simulation is a Gaussian distribution centered at point $(30, 40, 20)$ with standard derivation $\sqrt{30}$.

First, we observe in Table 4.3 that a much smaller Δt is required for RK2 to converge compared to AcIIF2 due to the fact that the reactions are stiff, requiring small Δt, for non A-stable methods such as RK2. Interestingly, AcIIF2 can reach the same overall error level as RK2 using a much larger time step for the same-sized "spatial" mesh, indicating that the numerical error for solving this FPE is likely dominated by spatial discretization. Thus, a large time step is sufficient for A-stable methods, such as IIF, while small time steps are still required for RK2 due to its stability constraints. In each time step RK2 is more efficient than AcIIF2; however, AcIIF2, which requires fewer time steps for a given t, still outperforms RK2 significantly in this case. We also plot the numerical results in Figure 4.1, where a grid with $N_A = N_B = 60$ and $N_{E_A} = 30$ and time step $\Delta t = 1s$ are used.

Next, we add another enzyme E_B that synthesizes metabolite B in the same way that E_A synthesizes A in the three-dimensional system Eq. (4.69). This extension leads to a four-dimensional FPE of four molecular species $[A], [B], [E_A]$ and $[E_B]$ [67],

$$\begin{gathered}
\emptyset \xrightarrow{\frac{k_A \cdot e_A}{1+a/K_I}} A \qquad \emptyset \xrightarrow{\frac{k_B \cdot e_B}{1+b/K_I}} B \\
A + B \xrightarrow{k_2 \cdot a \cdot b} \emptyset \\
A \xrightarrow{\mu \cdot a} \emptyset \qquad B \xrightarrow{\mu \cdot b} \emptyset \\
\emptyset \xrightarrow{\frac{k_{E_A}}{1+a/K_R}} E_A \quad E_A \xrightarrow{\mu \cdot e_A} \emptyset \\
\emptyset \xrightarrow{\frac{k_{E_B}}{1+b/K_R}} E_B \quad E_B \xrightarrow{\mu \cdot e_B} \emptyset
\end{gathered} \tag{4.72}$$

where $k_A = k_B = 0.3s^{-1}, k_2 = 0.001s^{-1}, K_I = 60, \mu = 0.002s^{-1}, k_{E_A} = k_{E_B} = 0.02s^{-1}$ and $K_R = 30$.

The computational domain is chosen to be $[0, 80] \times [0, 80] \times [0, 30] \times [0, 30]$ that contains nearly all possible states of the system. We choose zero Dirichlet boundary conditions with the initial condition as a Gaussian distribution centering at $(30, 40, 15, 12)$ with a standard deviation $\sqrt{40}$. There are nine reactions in the system, corresponding to nine array-representation operators. Based on commutability of the operators, we group some of them similar to the three dimensional case to increase the overall computational efficiency. One interesting observation is that that as the fourth dimension grid number N_{E_B} increases, the CPU time for increases only linearly, as seen in Table 4.4. For this set of simulations, we fix the other three grid numbers: $N_A = N_B = N_{E_A} = 10$, and keep doubling N_{E_B} from 4 to 32. IIF method computes the entire matrix exponential, thus its CPU time will increase by a fourth folder. While AcIIF only compute small matrix exponential, its CPU time will linearly depends on N_{E_B}. Finally, we plot the numerical results for $N_A = N_B = 40, N_{E_A} = N_{E_B} = 20$ and time step $\Delta t = 1$, in Figure (4.2).

4.5.4 An application to Chemical Master Equations

The Chemical Master Equation (CME) describes the time evolution of the probability density function. In CME, each reaction on the probability density evolution may be considered as diffusion-like operators with cross derivatives. Thus, AcIIF can be applied to solve such equations. We consider a family of proteins X with different conformational types $X_1, X_2, ..., X_d$. Two conformational types X_i and X_{i+1} can conform to each other through an enzyme E. Suppose that during reactions, no new protein is created; the enzyme is abundant so that one can treat the quantity of the enzyme as a constant; and intermediate products are extremely unstable. As a result, the entire system consists of the following bio-chemical reactions:

$$\begin{aligned} &X_1 \overset{k_1^+}{\rightarrow} X_2 \overset{k_2^+}{\rightarrow} X_3 ... \overset{k_{d-1}^+}{\rightarrow} X_d \\ &X_1 \overset{k_1^-}{\leftarrow} X_2 \overset{k_2^-}{\leftarrow} X_3 ... \overset{k_{d-1}^-}{\leftarrow} X_d \end{aligned} \tag{4.73}$$

for which the total copy number of the protein X is a constant,

$$X_1 + X_2 + ... + X_d = N \tag{4.74}$$

For simplicity, $x = (x_1, x_2, ..., x_d)$ denotes each state where $0 \leq x_i \leq N, i = 1, 2, ..., d$ (although some of the states cannot be reached). In particular, the reaction

$$X_i \overset{k_i^+}{\rightarrow} X_{i+1} \tag{4.75}$$

defines a linear mapping on probability density function in CMEs, with the following array representation,

$$\mathcal{A}_{X_i,X_{i+1}}^{(x_j),j\neq i,i+1} M_{m,n} = -k_1^+ m M_{m,n} + k_1^+(m+1)M_{m+1,n-1} \tag{4.76}$$

where M is a 2-dimensional array. Other reactions can be treated in a similar way.

For a protein family with d conformational types and N total number of copies, the direct calculation of exponential of the linear mapping requires the exponentiation of a $N^d \times N^d$ matrix. However, in the array representation, only $N^2 \times N^2$ matrices' exponentials are required to be calculated. More saving in both storage and CPU time result in using the array representation when the number of species d gets larger.

To demonstrate this through direct simulations, we implement a second order array-representation integration factor method as well as the second order Runge-Kutta (a standard temporal integrator for CMEs) for the case of $N = 30$ and $d = 3$. The initial distribution of the molecules is set to be $P(X_1 = 30) = 1$, that is, initially all molecules take the conformational type X_1. We choose rate coefficients $k_1^+ = k_2^- = 1, k_1^- = 2, k_2^+ = 3$ and we compute the solution up to $t = 3$ using different Δt. The maximal error of the solution is estimated based on an "exact" solution computed using a very small time step by RK2.

First, the second order accuracy of AcIIF2 method is clearly observed in Table 4.5. As expected, RK2 requires a very small time step due to its stability constraint in contrast to AcIIF's stable and good accuracy, even at a time step as large as $\Delta t = 1/4$. As Δt decreases significantly (e.g. $\Delta t \leq 1/128$), RK2 becomes stable and converges as seen in Table 4.5. At the same size of time step, we observe AcIIF2 and RK2 has similar size of errors. Of course, using the same size of time step, RK2 takes less CPU time and storage than AIF2, with both achieving similar accuracy. However, if moderately high accuracy (e.g. 10^{-4} for this particular system) is sufficient, AIF2 shows its advantage. In particular, as the number of species increases or the rate constants become more stiff, a combination of the array representation and the integration factor method becomes even more attractive in achieving both efficiency and accuracy.

4.6 Discussions and Conclusions

Higher order spatial derivatives and reactions of drastically different time scales demand temporal schemes of the generous stability constraint. Implicit integration factor methods, which solve exactly the linear operator of higher order spatial derivatives along with an implicit treatment of the stiff reactions, are effective approaches for such types of different equations. One unique computational challenge associated with such methods is the handling of exponentials of matrices. Here, we have introduced a new array representation for the discretization matrices of the linear differential operators. Because of such representation, computing exponentials of large matrices is reduced to the calculation of exponentials of matrices of significantly smaller sizes. The saving and advantages for array representations in both storage and CPU time escalate as the dimension of the system increases. In addition, this approach can be directly combined with the implicit integration method for an overall efficient method (termed as AcIIF) of excellent temporal stability.

Due to its advantage for high dimensions and stiff reactions, such an approach is particularly appropriate for solving reaction-diffusion equations and other diffusion-like equations, such as Fokker-Planck equations. Our direct implementation and testing of the second order AcIIF, which is linearly absolute stable, has demonstrated its advantages compared to some existing approaches. Interestingly, such array representation can also be applied to chemical master equations, ODE systems of large size that often is stiff. The computational efficiency for such applications become most evident for biochemical networks of a large number of species with each reaction in the system affecting only few species.

Although the array representation has been presented only in the context of compact implicit integration factor methods, the approach can easily be applied to other integration factor or exponential difference methods. Other type of equations of higher order derivatives, (e.g. Cahn-Hilliard equations [68] of fourth order derivatives) in addition to reaction-diffusion equations and Fokker-Planck equations may also be handled using the array representation for better efficiency. To better deal with high spatial dimensions, one can incorporate the sparse grid [69] into the array-representation technique. The flexibility of such representation allows either direct calculation of the exponentials of matrices or using Krylov subspace for computing their exponential matrix-vector multiplications for saving in storages. Overall, the array representation along with integration factor methods provides an efficient approach for solving a wide range of problems arising from biological and physical applications.

	Error in L_∞		Accuracy order		CPU time	
N	AcIIF2	IIF2	AcIIF2	IIF2	AcIIF2	IIF2
8	0.0672	0.0672	-	-	0.05s	0.34s
16	0.0169	0.0169	1.99	1.99	0.15s	30.5s
32	0.0042	-	2.01	-	5.13s	-
64	0.0011	-	1.93	-	246s	-

Table 4.1: A comparison between the second order array-representation compact IIF method (AcIIF2 in Eq. (4.36)) and IIF2 method. The symbol "-" denotes insufficient memory in calculation of the exponential matrix.

	Error in L_∞		Accuracy order		CPU time	
N	AcIIF2	IIF2	AcIIF2	IIF2	AcIIF2	IIF2
8	0.2744	0.2754	-	-	0.29s	1.35s
16	0.0675	0.0678	2.02	2.02	2.2s	155s
32	0.0169	-	2.00	-	133s	-

Table 4.2: A comparison between AcIIF2 method (Eq. (4.37)) and IIF2 for the case of non-constant diffusion coefficients.

		AcIIF2/RK2		
Grids (N_A, N_B, N_{E_A})	Δt	Error in L_∞	CPU time	RK2 unstable when
(25,25,15)	5/0.2	$2.6 \times 10^{-4}/2.6 \times 10^{-4}$	17.6s/71.2s	$\Delta t \geq 0.3$
(40,40,24)	5/0.15	$1.3 \times 10^{-4}/1.4 \times 10^{-4}$	35.3s/182.5s	$\Delta t \geq 0.2$
(50,50,30)	5/0.1	$8.6 \times 10^{-5}/9.1 \times 10^{-5}$	65.5s/470.2s	$\Delta t \geq 0.15$

Table 4.3: A comparison between the second order AcIIF and Runge-Kutta (RK2) for the three-dimensional FPE (4.69) at $t = 30$.

N_{E_B}	4	8	16	32
CPU time(s)	5.6	8.3	14.5	32.6

Table 4.4: The CPU time for different grid numbers of the fourth dimension of the FPE.

	AcIIF2		RK2	
Δt	error	order of acc	error	order of acc
$1/4$	7.95×10^{-4}	-	unstable	
$1/8$	1.96×10^{-4}	2.02	unstable	
$1/16$	4.89×10^{-5}	2.00	unstable	
$1/32$	1.22×10^{-5}	2.00	unstable	
$1/64$	3.06×10^{-6}	2.00	unstable	
$1/128$	7.64×10^{-7}	2.00	3.74×10^{-7}	-
$1/256$	1.91×10^{-7}	2.00	9.30×10^{-7}	2.01

Table 4.5: A comparison between the second order array-representation compact integration factor method (AcIIF2) and Runge-Kutta (RK2) methods for simulating CMEs.

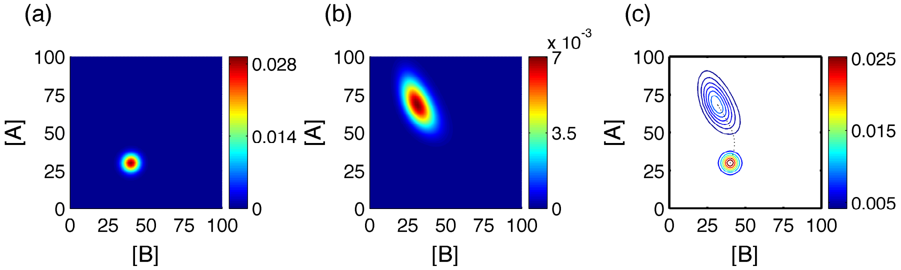

Figure 4.1: Numerical solution of system (4.69) using AcIIF2. Temporal discretization is set by the time step $\Delta t = 1s$, and the simulation is ran up to time $t = 50s$. (a) Shows the initial distribution of molecular species A and B, which are Gaussian distributions centered at $(A, B) = (30, 40)$. (b) The distribution of molecular species A and B at $t = 50s$. (c) The contour plot of initial and final distributions. The dotted black line connects the centers of the solutions of the rate equations of system (4.69).

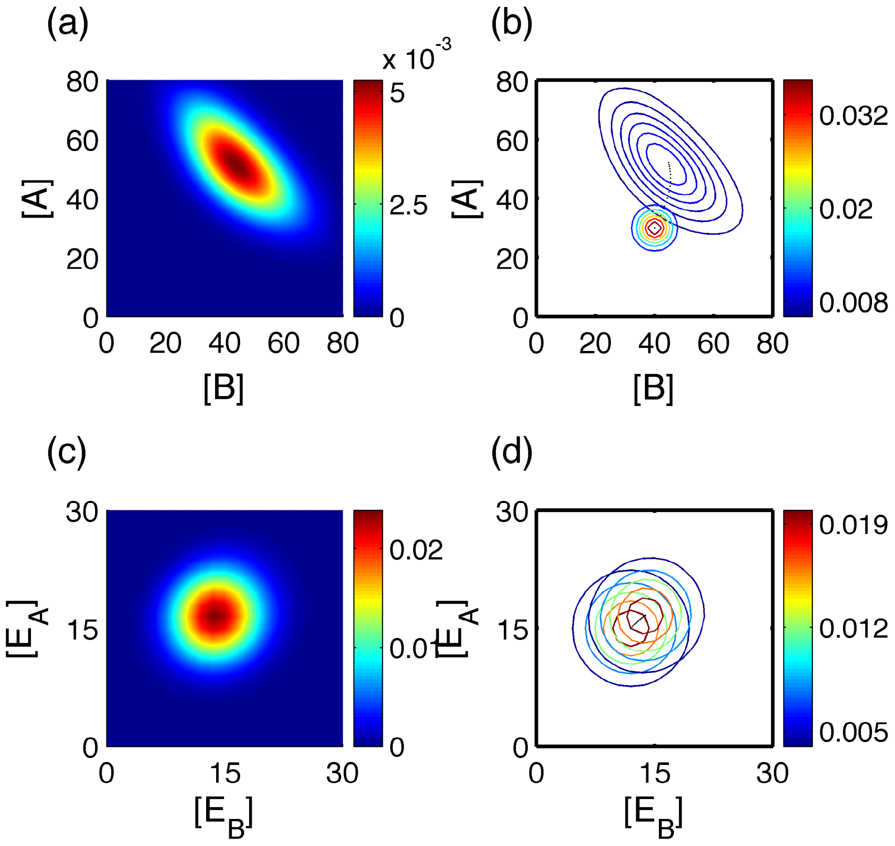

Figure 4.2: Numerical solution of system (4.72) using AcIIF2. Temporal discretization is set by the time step $\Delta t = 1s$, and the simulation is ran up to time $t = 35s$. (a) The distribution of molecular species A and B at $t = 35s$. (b) The contour plot of initial and final distributions of molecular species A and B. The dotted black line connects the centers of the solutions of the rate equations of system (4.72). (c) The distribution of molecular species E_A and E_B at $t = 35s$. (d) The contour plot of initial and final distributions of molecular species E_A and E_B. The dotted black line connects the centers of the solutions of the rate equations of system (4.72).

Chapter 5

Semi-implicit Integration Factor Methods on Sparse Grid

NOTE: This chapter was done in collaboration with Qing Nie and the was submitted in 2014.

5.1 Summary

In order to solve the multi-dimensional partial differential equations, the curse of dimensionality has to be dealt with. Application of sparse grid technique as spatial discretization method can handle this problem, however, how to deal with afterwords time integration problem efficiently is still a challenge. High order spatial derivatives and stiff reactions often introduce extreme temporal stability constraints, forcing people to choose very small time step, or adopt implicit algorithms. Semi-implicit integration factor method (IIF) handles diffusion exactly and reaction implicitly, thus provides good stability properties as well as excellent numerical efficiency for low dimensional cases. Meanwhile, array-representation

compact semi-implicit integration factor method (AcIIF) are developed to significantly decease the time and storage cost of computing the matrix exponentials in IIF method on uniform grids, which is efficient for multi dimensional situations. In this chapter we construct the AcIIF method on several sparse grid spatial discretization techniques (sparse grid finite element scheme, sparse grid finite difference scheme, and a sparse grid combination technique). Our new methods are particularly advantageous for high dimensional systems and are theoretically and numerically proved to be correct and efficient. In addition, a diffusive logistic equation and a Fokker-Planck equation, both driven by real biological problems, are solved by new methods. These examples suggest the broad applications of new methods on high-dimensional systems.

5.2 Introduction

Consider the following generalized reaction-diffusion system:

$$\frac{\partial u(\mathbf{x},t)}{\partial t} = Pu(\mathbf{x},t) + f(u(\mathbf{x},t)), \mathbf{x} = (x_1, ..., x_d) \in (0,1)^d, t \in [0,T], \tag{5.1}$$

where P is an elliptic partial operator respect to space $\mathbf{x}$. This equation has multiple applications. For instance, the formation of the morphogen gradient in the development of embroy is modeled by a reaction-diffusion equation[11], where P is a Laplacian $\Delta_{\mathbf{x}}$. Or the stochastic behavior of a gene network is described by the Fokker-Planck equation [44], where P denotes a second order partial differential operator containing cross derivatives. In finance, the Black-Scholes equation takes the similar form to price the options under several risk factors [70]. While in population genetics this equation is applied to model the site-frequency spectrum [71].

Numerically solving the equation in high-dimensional situation can be a huge challenge. This is mainly because of two reasons. First, people may confront the "curse of dimensions", that is, to achieve an accuracy of the order $O(N_{\mathbf{x}}^{\alpha})$ (for example, $\alpha = 2$ for a second order center difference scheme), $O(N_{\mathbf{x}}^{d})$ number of uniform grids are needed. The storage and manipulation of these grids can be expensive and unaffordable in high-dimensional situations. For example, when solving an equation with fifty spatial dimensions ($d = 50$), even the spatial resolution is set to be small ($N_{\mathbf{x}} = 2$), the number of total unknown, which need to be stored and updated at every time step, is 2^{50}. It would take approximately 10^9 GB of memory to save them if each digit is in double precision.

Second, after space discretization, how to deal with the afterwords time integration problem efficiently can be difficult. High order spatial derivatives (Laplacians, cross derivatives etc.) and stiff reactions may result severe constrains on time step. For explicit methods, e.g. Runge-Kutta method and Euler method, the time step has to be small enough to meet the stability needs, thus increase the total time cost. For implicit methods (Crank-Nicolson method, implicit Euler etc.), although the time step can be larger, a huge non-linear system needs to be solved at each time step.

To handle the first issue, sparse grid is proved to be an efficient technique[72]. Instead of uniform grids, sparse grid can achieve an accuracy of $O(\log N^{d-1}/N^2)$, with $O(N \log N^{d-1})$ number of grids, thus break the "curse of dimensions". The simplest sparse grid approach starts from the 1-dimensional piecewise linear hierarchical basis, which is then extended to the general piecewise d-linear hierarchal basis by a tensor product construction [73, 72]. Then other (polynomial) hierarchical bases are developed[74, 75, 76]

Several sparse grid discretization techniques are developed to solve partial differential equations. Some of them are listed in the following. The first one is the sparse grid finite element method and Galerkin technique [77, 78], which is based on piecewise linear finite elements and weak formula. The second technique of the discretization PDEs is the sparse grid finite differences scheme[79], which is based on the one dimensional second order center difference, and the adaptivity of this method is also developed[79]. Another discretization method is so called sparse grid combination technique [80], which is to solve the PDEs on different grids, then combines all the results. Other discretization scheme based on sparse grid are also developed, e.g. finite volume method [81] and the spectral method [82, 69]. These methods have multiple applications, such as option price model [83] and elliptic problem [69].

For the second issue, dealing with the time integration problem, if on the uniform grid, integration factor (IF) and exponential time differencing (ETD) methods are effective to handle temporal stability constrains raised by high order spatial derivatives [45, 46, 47]. The IF and ETD methods deal with linear operators of the highest order derivative exactly, hence can achieve good temporal stabilities [48, 49, 45]. Besides, a class of semi-implicit integration factor (IIF) methods were developed for implicit treatment of the stiff reactions [50], aiming on relief the stability constraints come from stiff reactions. In the IIF method, the diffusion term is solved exactly while the nonlinear equations resulted from the implicit treatment of reactions is decoupled from the diffusion term, thus avoid solving huge nonlinear systems in each time step, which is unlike and more efficient than other implicit methods. The second order IIF method is proved to be linearly unconditionally stable.

On uniform grid, the dominant computational cost in IIF method is the storage and calculation of exponentials of matrices from discretization of the linear differential operators in the PDEs. To overcome this issue, compact representation [51, 52] and array representation [42] of the discretization matrices was introduced in the context of IIF method. The compact representation works for Laplacian or bi-Laplacian, which does not include cross derivatives, and array representation can be applied general elliptical cases. In both compact implicit integration factor method (cIIF) and array-representation implicit integration factor method (acIIF), the discretized solutions are represented in multi-dimensional array rather than a vector, and the discretized diffusion operator are represented in matrices of much smaller size than the standard matrices used in IIF. Also, both methods preserve the stability property of the IIF method. In addition, cIIF method can handle general curvilinear coordinates as well as combine with adaptive mesh refinements [52]. One can also apply cIIF to stiff reactions and diffusion while using other specialized hyperbolic solvers (e.g. WENO methods [53, 54]) for convection term to solve reaction-diffusion-convection equations efficiently [55].

Since their efficiency on time integration problem, in this chapter we propose to construct the IIF and acIIF method on sparse grid. To deal with a PDE system, we first apply one of the sparse grid discretization technique to discretize the system in space, then deal with the afterword ODE system with the IIF and acIIF methods. Here we introduce the IIF and acIIF methods on three of the sparse grid discretization schemes: (i) finite element scheme; (ii) finite difference scheme; and (iii) sparse grid combination technique. Also, numerical tests are presented to show the correctness, efficiency and broad applications of our new methods.

This chapter is organized as below: in Section 5.3 we present some backgrounds of sparse grid interpolation; in Section 5.4, we construct the IIF method on sparse grid finite element scheme and Galerkin techniques; in Section 5.5, we construct the AcIIF method on the sparse grid finite difference scheme; in Section 5.6, we states that application of AcIIF method on each subproblem required by sparse grid combination technique is computationally efficient, and this approach works well when cross derivatives and non-constant diffusion coefficients are in the system; in Section 5.7, we present some numerical results to demonstrate the correctness, efficiency, and applications of our methods.

5.3 Introduction to sparse grid interpolation

NOTE: This section introduces some backgrounds about sparse grid, and all the materials are refer to [72]. Let $\varphi_{k,j}(x)$ be the following piecewise linear hat function:

$$\varphi_{k,j}(x) = \begin{cases} 1 - 2^{k+1}\left|x - \frac{2j-1}{2^{k+1}}\right|, \ x \in [\frac{(j-1)}{2^k}, \frac{j}{2^k}], \\ 0, \mathit{otherwise} \end{cases} \tag{5.2}$$

And a tensor product leads to the following hierarchical basis:

$$\phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}) = \prod_{i=1}^{d} \varphi_{k_i,j_j}(x_i), \tag{5.3}$$

where $\mathbf{k} = (k_1, k_2, ..., k_d)$, $\mathbf{j} = (j_1, j_2, ..., j_d)$ and $\mathbf{x} = (x_1, x_2, ..., x_d) \in (0,1)^d$. For a smooth function $g(\mathbf{x})$, $\mathbf{x} \in (0,1)^d$ vanishes outside the domain, if g is smooth enough, then for fixed $N_{\mathbf{x}} = 2^K$, we have:

$$g(\mathbf{x}) = \sum_{\substack{k_1 + ... + k_d \leq K, \\ k_i \geq 0}} \sum_{\substack{1 \leq j_i \leq 2^{k_i} \\ i = 1, 2, ..., d}} \varrho_{\mathbf{k},\mathbf{j}} \phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}) + O\left(\frac{\log N_{\mathbf{x}}^{d-1}}{N_{\mathbf{x}}^2}\right). \tag{5.4}$$

Note that to reach the accuracy of $O(\log N_{\mathbf{x}}^{d-1}/N_{\mathbf{x}}^2)$, based on above, there are $O(\log N_{\mathbf{x}}^{d-1} N_{\mathbf{x}}$ basis functions, and the "Curse of dimensions" is beaten. Also note that to get all the coefficients of the above equation, $\varrho_{\mathbf{k},\mathbf{j}}$, the function value at the following point are needed:

$$g_{\mathbf{k},\mathbf{j}} := g\left(\frac{2j_1 - 1}{2^{k_1+1}}, ..., \frac{2j_d - 1}{2^{k_d+1}}\right). \tag{5.5}$$

Also, note that

$$g_{\mathbf{k},\mathbf{j}} \neq \varrho_{\mathbf{k},\mathbf{j}}. \tag{5.6}$$

The value $g_{\mathbf{k},\mathbf{j}}$ is called the **nodal value** of the function, while the value $\varrho_{\mathbf{k},\mathbf{j}}$ is called the **hierarchical value** of the function. Usually a transformation between two value are needed when applying sparse grid. Figure 5.1 shows a two dimensional sparse grid for $N_{\mathbf{x}} = 2^4$.

5.4 Semi-implicit integration factor method (IIF) with sparse grid finite element scheme

In this section we solve the following reaction-diffusion system while the spatial differential operator is the Laplacian.

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) = \Delta_{\mathbf{x}} u(\mathbf{x}, t) + f(u), \ \mathbf{x} = (x_1, ..., x_d) \in (0, 1)^d. \tag{5.7}$$

For the purpose of simplicity, a zero-Dirichlet boundary condition is applied on u.

5.4.1 Weak formula under sparse grid finite element scheme

Fix $N_{\mathbf{x}} = 2^K$, approximate the unknown u by:

$$u(\mathbf{x}, t) \approx \sum_{\substack{k_1 + ... + k_d \leq K \\ k_i \geq 0}} \sum_{\substack{1 \leq j_i \leq 2^{k_i} \\ i = 1, 2, ..., d}} v_{\mathbf{k},\mathbf{j}}(t) \phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}), \tag{5.8}$$

and define the inner product of two functions $\psi_1(\mathbf{x})$ and $\psi_2(\mathbf{x})$:

$$< \psi_1(\mathbf{x}), \psi_2(\mathbf{x}) > := \int_{(0,1)^d} \psi_1(\mathbf{x}) \psi_2(\mathbf{x}) d\mathbf{x}. \tag{5.9}$$

Put Eq. (5.8) into Eq. (5.7), and make the inner product on both sides with $\phi_{\overline{\mathbf{k}},\overline{mathbfj}}(\mathbf{x})$, for all possible $\overline{\mathbf{k}}$ and $\bar{\mathbf{j}}$. This leads to the following weak formula on the sparse grid:

$$\begin{aligned}
&\sum_{\substack{k_1+...+k_d\le K\\ k_i\ge 0}} \sum_{\substack{1\le j_i\le 2^{k_i}\\ i=1,2,...,d}} <\phi_{\mathbf{k},\mathbf{j}},\phi_{\overline{\mathbf{k}},\bar{\mathbf{j}}}> \frac{dv_{\mathbf{k},\mathbf{j}}(t)}{dt}\\
&= \sum_{\substack{k_1+...+k_d\le K\\ k_i\ge 0}} \sum_{\substack{1\le j_i\le 2^{k_i}\\ i=1,2,...,d}} <\Delta_{\mathbf{x}}\phi_{\mathbf{k},\mathbf{j}},\phi_{\overline{\mathbf{k}},\bar{\mathbf{j}}}> v_{\mathbf{k},\mathbf{j}}(t)\\
&+ \sum_{\substack{k_1+...+k_d\le K\\ k_i\ge 0}} \sum_{\substack{1\le j_i\le 2^{k_i}\\ i=1,2,...,d}} <\phi_{\mathbf{k},\mathbf{j}},\phi_{\overline{\mathbf{k}},\bar{\mathbf{j}}}> f_{\mathbf{k},\mathbf{j}}(t)
\end{aligned} \tag{5.10}$$

where $f_{\mathbf{k},\mathbf{j}}(t)$ is the nodal value of function $f(u(\mathbf{x},t))$:

$$f(u(\mathbf{x},t)) \approx \sum_{\substack{k_1+...+k_d\le K\\ k_i\ge 0}} \sum_{\substack{1\le j_i\le 2^{k_i}\\ i=1,2,...,d}} f_{\mathbf{k},\mathbf{j}}(t)\phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}). \tag{5.11}$$

Note that both $<\phi_{\mathbf{k},\mathbf{j}},\phi_{\overline{\mathbf{k}},\bar{\mathbf{j}}}>$ and $<\Delta_{\mathbf{x}}\phi_{\mathbf{k},\mathbf{j}},\phi_{\overline{\mathbf{k}},\bar{\mathbf{j}}}>$ can be exactly calculated since the piece wise linear function are used. By form the following two matrices:

$$M = \left(<\phi_{\mathbf{k},\mathbf{j}},\phi_{\overline{\mathbf{k}},\bar{\mathbf{j}}}>\right), \quad D = \left(<\Delta_{\mathbf{x}}\phi_{\mathbf{k},\mathbf{j}},\phi_{\overline{\mathbf{k}},\bar{\mathbf{j}}}>\right), \tag{5.12}$$

and let vector $V(t) = (v_{\mathbf{k},\mathbf{j}}(t))$ and $F(t) = (f_{\mathbf{k},\mathbf{j}}(t))$ denote the hierarchical value of unknown function and reactions, the following time dependent problem is left to be solved:

$$M\frac{dV}{dt} = D\cdot V + M\cdot F, or \;\; \frac{dV}{dt} = (M^{-1}D)V + F. \tag{5.13}$$

Directly apply the IIF method to the above equation, we construct the following second order (in time) IIF method with sparse grid finite element scheme (**IIF2-SG-FEM**):

$$V(t_{n+1}) = e^{M^{-1}D\Delta t}\left(V(t_n) + \frac{\Delta t}{2}F(t_n)\right) + \frac{\Delta t}{2}F(t_{n+1}). \tag{5.14}$$

Note that since the nodal value is not equal the hierarchical value, the above non-linear system cannot be solved locally. Next we'll introduce an order to solve above equation. Following that order, only localized non-linear system needs to be solved. This is the one major advantage of the IIF method.

5.4.2 Solving non linear system Eq. (5.14)

To solve Eq. (5.14), we first solve the hierarchical value of $\mathbf{k} = (0, ..., 0)$ and $\mathbf{j} = (1, ..., 1)$. Note that at this point, the nodal value equals the hierarchical value so we directly solve the following:

$$u(\frac{1}{2}, ..., \frac{1}{2}, t_{n+1}) = L_{(0,...,0),(1,...,1)} + \frac{\Delta t}{2}f(u(\frac{1}{2}, ..., \frac{1}{2}, t_{n+1})), \tag{5.15}$$

where $L_{\mathbf{k},\mathbf{j}}$ is the computational result of $e^{M^{-1}D\Delta t}\left(V(t_n) + \frac{\Delta t}{2}F(t_n)\right)$ at corresponding location. And

$$v_{(0,...,0),(1,...,1)}(t_{n+1}) = u(\frac{1}{2}, ..., \frac{1}{2}, t_{n+1}). \tag{5.16}$$

The we solve the hierarchical value for $\mathbf{k}$ that $k_1 + k_2 + ... + k_d = 1$. Note that there are two ways to compute the nodal value of f at corresponding point: (i) from the hierarchical value of f (left side of the below equation); and (ii) from the nodal value of u (right side of the below equation).

$$f_{\mathbf{k},\mathbf{j}} + 0.5f_{(0,...,0),(0,...,0)} = f(v_{\mathbf{k},\mathbf{j}} + 0.5v_{(0,...,0),(0,...,0)}),\ k_1 + ... + k_d = 1. \tag{5.17}$$

Since $v_{(0,...,0),(0,...,0)}$ is known, put this into Eq. (aciif-sg:eq:iif-fem):

$$v_{\mathbf{k},\mathbf{j}}(t_{n+1}) = L_{\mathbf{k},\mathbf{j}} + \frac{\Delta t}{2}\left(f(v_{\mathbf{k},\mathbf{j}} + 0.5v_{(0,...,0),(0,...,0)}) - 0.5f_{(0,...,0),(0,...,0)}\right),\ k_1 + ... + k_d = 1. \quad (5.18)$$

Solve above localized non-linear system to get all hierarchical value $v_{\mathbf{k},\mathbf{j}}(t_{n+1})$ such that $k_1 + ... + k_d = 1$.

Typically, after solving all $v_{\mathbf{k},\mathbf{j}}$ (and corresponding $f_{\mathbf{k},\mathbf{j}}$) for which $k_1 + ... + k_d < K$, then for $\overline{\mathbf{k}}$, $\overline{k_1} + ... + \overline{k_d} = K$, and let

$$\mathbf{x}_{\overline{k},\overline{j}} := \left(\frac{2\overline{j_1} - 1}{2^{\overline{k_1}+1}}, ..., \frac{2\overline{j_d} - 1}{2^{\overline{k_d}+1}}\right), \quad (5.19)$$

then followed by two ways to compute the nodal value of f at point $\mathbf{x}_{\overline{k},\overline{j}}$:

$$f_{\overline{\mathbf{k}},\overline{\mathbf{j}}} + \sum_{\substack{k_1 + ... + k_d \leq K - 1 \\ k_i \geq 0}} \sum_{\substack{1 \leq j_i \leq 2^{k_i} \\ i = 1, 2, ..., d}} f_{\mathbf{k},\mathbf{j}}\phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}_{\overline{k},\overline{j}}) = f\left(v_{\overline{\mathbf{k}},\overline{\mathbf{j}}} + \sum_{\substack{k_1 + ... + k_d \leq K - 1 \\ k_i \geq 0}} \sum_{\substack{1 \leq j_i \leq 2^{k_i} \\ i = 1, 2, ..., d}} v_{\mathbf{k},\mathbf{j}}\phi_{\mathbf{k},\mathbf{j}}(\mathbf{x}_{\overline{k},\overline{j}})\right). \quad (5.20)$$

Putting above equation into Eq. (5.14), we derive the local non-linear system for $v_{\overline{\mathbf{k}},\overline{\mathbf{j}}}$.

5.4.3 Error estimation, computational cost and stability

The error of the IIF2-SG-FEM method comes from two parts: (i) the spatial discretization by the sparse grid; and (ii) the second order (in time) IIF method as time integration. So the overall error estimation is:

$$O\left(\frac{\log N_{\mathbf{x}}^{d-1}}{N_{\mathbf{x}}^2}\right) + O(\Delta t^2). \tag{5.21}$$

The majority cost of the IIF2-SG-FEM method is the computation of the exponential of $M^{-1}D\Delta t$. Although it only needs to be computed once and then store for later use, for large dimensions (d) and large spatial resolution ($N_{\mathbf{x}}$), since the size of the matrix is $O(N_{\mathbf{x}} \log N_x^{d-1})$, the cost may still be huge. Also, at each time step, we need to compute Eq. (5.20) in order to get the local non-linear system. This may also takes some time. Later in the chapter, we'll discuss other methods that can reduce the size of the exponential matrix.

Since M is the mass matrix and D comes from the Laplacian operator, $M^{-1}D\Delta t$ only contains non-negative eigenvalues, and the stability of IIF2-SG-FEM can be discussed through a scalar case[50]. It is shown that IIF2-SG-FEM is A-stable if the reaction term is linear.

5.5 Array-representation semi-implicit integration factor method (AcIIF) with the sparse grid finite difference method

In this section we construct the AcIIF with the sparse grid finite difference method to solve Eq. (5.7). For the purpose of simplicity, all other conditions (spatial domain and boundary condition) remain the same as in Section 5.4.

5.5.1 An introduction to the sparse grid finite difference method

Let $V(t) = (v_{\mathbf{k},\mathbf{j}}(t))$ be the hierarchical value of unknown u and $U(t) = (u(\mathbf{x}_{\mathbf{k},\mathbf{j}}, t))$ be the nodal value. The nodal-hierarchical transformation can be achieved by a dimensional splitting scheme [79]. Let $\overline{H}_i$ be the (nodal to) hierarchical basis transformation along direction x_i [79], acts on $U(t)$ as below: for all possible

$$\mathbf{k}^i := (k_1, ..., k_{i-1}, \cdot, k_{i+1}, ..., k_d), \quad \mathbf{j}^i := (j_1, ..., j_{i-1}, \cdot, j_{i+1}, ..., j_d), \tag{5.22}$$

do the one dimensional (nodal) to hierarchical transformation on vector $U_{\mathbf{k}^i,\mathbf{j}^i}(t)$, which is a vector composed by elements of $U(t)$ that fix $k_s, j_s, s \neq i$ and only vary k_i and j_i. The (hierarchical to) nodal basis transformation along direction x_i, $\overline{M}_i$, acts on $V(t)$ in a very similar way, instead of a one dimensional (hierarchical to) nodal transformation. The nodal-hierarchical transformation is the composition of $\overline{H}_i$ or $\overline{M}_i$:

$$\overline{H}_i\overline{M}_i = I, \;\; \overline{H}_1\overline{H}_2...\overline{H}_nU(t) = V(t), \;\; \overline{M}_1\overline{M}_2...\overline{M}_nV(t) = U(t). \tag{5.23}$$

Define $\overline{H}_{I/\{i\}} := \overline{H}_1...\overline{H}_{i-1}\overline{H}_{i+1}...\overline{H}_n$ as the (nodal to) hierarchical basis transformation along all direction except direction x_i. Also define the finite difference operator $\overline{D}_i$ acts on any vector $U(t)$ as: for all possible $\mathbf{k}^i$ and $\mathbf{j}^i$, do the regular one dimensional center difference on $U_{\mathbf{k}^i,\mathbf{j}^i}(t)$. The the finite difference scheme on sparse grid for $\partial^2/\partial x_i^2$ is (for fixed $N_{\mathbf{x}} = 2^K$) $\overline{H}_{I/\{i\}}^{-1}\overline{D}_i\overline{H}_{I/\{i\}}U(t)$, with an error estimation:

$$O\left(\frac{\log N_{\mathbf{x}}^{d-1}}{N_{\mathbf{x}}^2}\right) \tag{5.24}$$

Use this scheme in Eq. (5.7) gives the following ODE system:

$$\frac{dU}{dt}(t) = \left(\sum_{1\le i\le d} \overline{H}_{I/\{i\}}^{-1} \overline{D}_i \overline{H}_{I/\{i\}} \right) U(t) + f(U(t)). \tag{5.25}$$

Apply the second order IIF method, we obtain the following second order (in time) IIF with sparse grid finite difference method (**IIF2-SG-FD**):

$$U(t_{n+1}) = e^{\left(\sum_{1\le i\le n} \overline{H}_{I/\{i\}}^{-1} \overline{D}_i \overline{H}_{I/\{i\}}\right)\Delta t} \left(U(t_n) + \frac{\Delta t}{2} f(U(t_n)) \right) + \frac{\Delta t}{2} f(U(t_{n+1})). \tag{5.26}$$

In IIF2-SG-FD scheme, we need to compute and store the exponentials of a $O(N_{\mathbf{x}} \log N_{\mathbf{x}}^{d-1}) \times O(N_{\mathbf{x}} \log N_{\mathbf{x}}^{d-1})$ matrix. When $N_{\mathbf{x}}$ and d are large, its computation is unaffordable.

5.5.2 AcIIF method with the sparse grid finite difference method

To reduce the size of the exponential matrix and related computational cost, we apply array representation on $U(t)$ and $V(t)$, to decompose them to some smaller vectors [42]. Starting from the hierarchical value $V(t)$, for a fixed direction x_i and $\mathbf{k}^i$, $\mathbf{j}^i$, varying k_i and j_i gives a vector of $T(\mathbf{k}^i, \mathbf{j}^i)$ elements, and denote as $V_{\mathbf{k}^i,\mathbf{j}^i}(t)$, where

$$T(\mathbf{k}^i, \mathbf{j}^i) = \sum_{s=0}^{k_i^{\max}} 2^s = 2^{k_i^{\max}+1} - 1, \ k_i^{\max} = K - (k_1 + ... + k_d) + k_i. \tag{5.27}$$

Let $\mathbf{k}^i$ and $\mathbf{j}^i$ go though all possible value, all such smaller vectors joint together will form the original vector $V(t)$, and we use the notation $\bigotimes$ to denote such relation:

$$V(t) = \bigotimes_{\substack{\mathbf{k}^i, \sum_{r\neq i} k_r \le K \\ \mathbf{j}^i, j_r = 1,2,...,2^{k_r}, r\neq i}} V_{\mathbf{k}^i,\mathbf{j}^i}(t). \tag{5.28}$$

Note that it is different than original array representation, where all the subarrays have the same length [42]. Here different sub vectors $V_{\mathbf{k}^i,\mathbf{j}^i}(t)$ have different size $T(\mathbf{k}^i,\mathbf{j}^i)$. But the size is less than $2N_{\mathbf{x}}-1$. With array representation, we can use smaller matrices to represent the (hierarchical to) nodal transformation along direction x_i, $\overline{M}_i$:

$$\overline{M}_i V(t) = \bigotimes_{\substack{\mathbf{k}^i, \sum_{r\neq i} k_r \leq K \\ \mathbf{j}^i, j_r = 1,2,...,2^{k_r}, r\neq i}} M_{\mathbf{k}^i,\mathbf{j}^i} V_{\mathbf{k}^i,\mathbf{j}^i}(t), \tag{5.29}$$

and the center difference scheme $\overline{D}_i$:

$$\overline{D}_i V(t) = \bigotimes_{\substack{\mathbf{k}^i, \sum_{r\neq i} k_r \leq K \\ \mathbf{j}^i, j_r = 1,2,...,2^{k_r}, r\neq i}} D_{\mathbf{k}^i,\mathbf{j}^i} V_{\mathbf{k}^i,\mathbf{j}^i}(t). \tag{5.30}$$

$M_{\mathbf{k}^i,\mathbf{j}^i}$ denotes the one dimensional (hierarchical to) nodal transformation, and $D_{\mathbf{k}^i,\mathbf{j}^i}$ is the following tridiagonal matrix:

$$D_{\mathbf{k}^i,\mathbf{j}^i} = \frac{1}{\mathbf{T(k^i,j^i)^2}} \begin{pmatrix} -2 & 1 & 0 & ... & 0 \\ 1 & -2 & 1 & ... & 0 \\ & ... & ... & ... & \\ 0 & ... & 0 & -2 & 1 \end{pmatrix}, \tag{5.31}$$

and both are $T(\mathbf{k}^i,\mathbf{j}^i)\times T(\mathbf{k}^i,\mathbf{j}^i)$ matrices, which are at most $(2N_{\mathbf{x}}-1)\times(2N_{\mathbf{x}}-1)$. Define $B_{\mathbf{k}^i,\mathbf{j}^i} = M^{-1}_{\mathbf{k}^i,\mathbf{j}^i} D_{\mathbf{k}^i,\mathbf{j}^i} M_{\mathbf{k}^i,\mathbf{j}^i}$, then under array representation, the exponential of $\overline{H}^{-1}_{I/\{i\}}\overline{D}_i\overline{H}_{I/\{i\}}\Delta t$

is calculated as:

$$e^{\overline{M}_i^{-1}\overline{D}_i\overline{M}_i\Delta t}V(t) = \bigotimes_{\substack{\mathbf{k}^i, \sum_{r\neq i} k_r \leq K \\ \mathbf{j}^i, j_r = 1,2,...,2^{k_r}, r\neq i}} e^{B_{\mathbf{k}^i,\mathbf{j}^i}\Delta t}V_{\mathbf{k}^i,\mathbf{j}^i}(t). \tag{5.32}$$

If directly compute, the size of the matrix is $\log N_{\mathbf{x}}^{d-1}$ times greater than computed following the right hand size. Then use array representation in IIF2-SG-FD to replace the process of calculation of the exponential of the matrix, we construct the array-representation semi-implicit integration factor method (**AcIIF2-SG-FD**):

(i) Transform from the nodal value $\big(U(t_n) + \frac{\Delta t}{2}f(U(t_n))\big)$ to the hierarchical value $W^1(t)$,

$$W^1(t_n) = M\left(U(t_n) + \frac{\Delta t}{2}f(U(t_n))\right), \tag{5.33}$$

(ii) Adopt array representation to compute the exponential matrix and vector multiplication:

$$W^2(t_n) = \bigotimes_{\substack{\mathbf{k}^1, \sum_{r\neq 1} k_r \leq K \\ \mathbf{j}^1, j_r = 1,2,...,2^{k_r}, r\neq 1}} e^{B_{\mathbf{k}^1,\mathbf{j}^1}\Delta t}\left(\dots \bigotimes_{\substack{\mathbf{k}^d, \sum_{r\neq d} k_r \leq K \\ \mathbf{j}^d, j_r = 1,2,...,2^{k_r}, r\neq d}} e^{B_{\mathbf{k}^d,\mathbf{j}^d}\Delta t}W^1_{\mathbf{k}^d,\mathbf{j}^d}(t_n)\right)_{\mathbf{k}^1,\mathbf{j}^1}, \tag{5.34}$$

(iii) Update the nodal value by solving the non-linear system, where M^{-1} is the (hierarchical to) nodal value transformation:

$$U(t_{n+1}) = M^{-1}W^2(t_n) + \frac{\Delta t}{2}f(U(t_{n+1})). \tag{5.35}$$

5.5.3 Error estimation, computational cost, stability and high order (in time) method

The error estimation of IIF2-SG-FD and AcIIF2-SG-FD methods is the same as in Eq. (5.21) since the error comes from two parts: (i) spatial discretization by a sparse grid finite difference scheme; and (ii) IIF2/AcIIF2 as time integration.

In the IIF2-SG-FD method, the large matrix exponential still needs to be compute, as the same as in the IIF2-SG-FEM method. While in the AcIIF2-SG-FD method, the computation cost of the matrix exponential is significantly reduced, since we reduce the matrix size by a vector of $\log N_{\mathbf{x}}^{d-1}$. But we introduce the hierarchical-nodal transformation at each time step. These transformation can be achieved in a fast way [69], so its computational cost can be neglect compare to the calculation of matrix exponential.

In both IIF2-SG-FD and AcIIF2-SG-FD method, we all work with matrix with non-positive eigenvalues, so its stability is the same as a scalar case. It is shown that for a scalar case, for the linear reactions, the method is A-stable [50].

To spatial error is constrained by sparse grid, however, we can enhance the time integration error by applying high order (in time) methods. For example, apply the third order IIF method to Eq. (5.25) leads to the following IIF3-SG-FD method:

$$Ut_{n+1} = e^{A\Delta t}U(t_n) + \Delta t\left(\frac{5}{12}f(U(t_n)) + \frac{2}{3}e^{A\Delta t}f(U(t_n)) - \frac{1}{12}e^{2A\Delta t}f(U(t_{n-1}))\right), \quad (5.36)$$

with error:

$$O(\frac{\log N_x^{d-1}}{N_x^2}) + O(\Delta t^3), \quad (5.37)$$

5.6 Cross derivatives and non-constant coefficients: AcIIF with a sparse grid combination technique

In the previous section we mainly focus on the Laplacian differential operator. In this section, we construct an efficient method to solve more general case, Eq. (5.1), where the spatial partial differential operator P may contains cross derivatives and non-constant coefficients. The method based on a sparse grid combination technique and the AcIIF method.

The sparse grid combination technique is another efficient way to deal with high dimensional systems [80]. Instead of sparse grids that handle with irregular grids, the sparse grid combination technique solves the PDE on different uniform grids, then combine each result to get the final solution. One application of the sparse grid combination technique is the option price model [83].

The sparse grid combination technique consists of two major steps:

Step 1: Fix the spatial resolution $N_{\mathbf{x}} = 2^K$, solve Eq. (1) on all the uniform grids below:

$$(2^{k_1} \times 2^{k_2} \times ... \times 2^{k_d}), \quad K \leq k_1 + k_2 + ... + k_d \leq K + d - 1, \quad k_i \geq 1, \tag{5.38}$$

where 2^{k_i} is the grid number along direction x_i. The results on each grid denotes as $w_{k_1,...,k_i}(\mathbf{x}, t)$.

Step 2: Combine all the results as following:

$$w_K(\mathbf{x}, t) = \sum_{m=K}^{K+d-1} (-1)^{d-1-m+K} \begin{pmatrix} d-1 \\ m-K \end{pmatrix} \sum_{k_1+k_2+...+k_n=m} w_{(k_1,...,k_n)}(\mathbf{x}, t). \tag{5.39}$$

If each individual solution $w_{k_1,\ldots,k_i}(\mathbf{x},t)$ satisfies:

$$\left\| u(\mathbf{x},t) - w_{(k_1,k_2,\ldots,k_n)}(\mathbf{x},t) \right\| = \sum_{i,j=1}^{n} O\left(\frac{1}{n_i n_j}\right), \tag{5.40}$$

then the error of $w_K(\mathbf{x},t)$ to the accurate solution $u(\mathbf{x},t)$ is [80]:

$$\| u(\mathbf{x},t) - w_K(\mathbf{x},t) \| = O\left(\frac{\log N_{\mathbf{x}}^{d-1}}{N_{\mathbf{x}}^2}\right). \tag{5.41}$$

In the sparse grid combination technique, each subproblem contains $N_{\mathbf{x}}$ grids, and the overall solution can reach an error estimation like other sparse grid method. Besides, each subproblem is independent to others, and this may lead to a straightforward implementation of parallel scheme. Suppose enough cores are given, then each core is in charge of one subproblem, and then after get all solutions, put them together as Eq. (5.39).

Based on the Cannikin law, under parallel, the subproblem which takes the most CPU time, matters to the overall efficiency. And how to solve each subproblem with less cost needs to be discussed. If take some explicit methods such as RK2, due to the stability constrains, the time step has to be $\Delta t \sim \Delta x^2$. Then for subproblems on grids for which $\exists i$ such that $k_i = K$ and $k_j = 1, j \neq i$, the time step reaches the minimum, that is $\Delta t = 1/2^{2K} = 1/N_{\mathbf{x}}^2$. These problems take more time to be solved. If IIF method is used, the time step can always remain the same, that is $\Delta t = 1/N_{\mathbf{x}}$, since its advantages on stability. However, computation of the exponentials in IIF cost much more CPU time.

We propose to apply the AcIIF method [42] to solve each subproblem. The AcIIF method is as stable as IIF method and moreover, significantly decrease the size of the exponential matrices, thus the total computational cost. With the AcIIF method, the time step for each subproblem is all set to be $\Delta t = 1/N_{\mathbf{x}}$, to fulfill the error condition for each subproblem Eq. (5.40).

The detailed information about the AcIIF method is given in Chapter 4.

5.7 Numerical simulations

We present several numerical tests in this section.

5.7.1 A two dimensional non-linear reaction-diffusion system

We first implement the IIF2-SG-FEM method on the following non-linear reaction-diffusion equation:

$$u_t(x, y, t) = \frac{1}{2\pi^2}\left(u_{xx}(x, y, t) + u_{yy}\right) + u(x, y, t)^2 - e^{-2t}\sin^2\pi x\sin^2\pi y, \tag{5.42}$$

with a zero Dirichlet boundary condition and the initial condition below:

$$u(x, y, 0) = \sin\pi x\sin\pi y. \tag{5.43}$$

The exact solution for the system is:

$$u(x, y, t) = e^{-t}\sin\pi x\sin\pi y. \tag{5.44}$$

Directly solve the weak formula Eq. (5.10) by some implicit methods (e.g. Crank-Nicolson) will have to solve a large non-linear system since the presence of the nonlinear reaction $u(x, y, t)^2$. To make a comparison, we implement both the IIF2-SG-FEM method and the RK2 to solve the weak formula. Numerical results show that in the IIF-SG-FEM, when the time step $\Delta t = 1/N_x$, the method is convergent and the error is consistent with analytical result. While in the RK2 method, if time step $\Delta t = 1/N_x^2$, the method is convergent and the error is very similar to the one of the IIF2-SG-FEM method. And a larger time step $\Delta t = 1/N_x$ leads to instability. The error of the IIF2-SG-FEM method is plotted in Figure 5.2(a), which is consistent with the error estimation Eq. (5.21).

5.7.2 A three dimensional linear reaction-diffusion system

We implement the AcIIF2-SG-FD method on the following system:

$$u_t(x, y, z, t) = \frac{1}{\pi^2} (u_{zz} + u_{yy} + u_{zz}) + (n - 1)u, \tag{5.45}$$

with the initial condition

$$u(x, y, z, 0) = \sin \pi x \sin \pi y \sin \pi z, \tag{5.46}$$

and a zero Dirichlet boundary condition. The exact solution for the system is:

$$u(x, y, z, t) = e^{-t} \sin \pi x \sin \pi y \sin \pi z. \tag{5.47}$$

The L_∞ error with different spatial resolution is plotted in Figure 5.2(b). The time step is step to be $\Delta t = 1/N_{\mathbf{x}}$ and the result is consistent with the analysis.

To compare with other methods, we also implement the IIF2-SG-FD method as well as the RK2 method with sparse grid finite difference. All the results are listed in Table 5.1. It is shown that the AcIIF2-SG-FD method performs much faster than the other two methods. A detailed analysis shows that for the IIF2-SG-FD method, the most consuming part is to calculate the matrix exponential and it ail take about 90 percent of the overall CPU time. And in the AcIIF2-SG-FD method, this part of time consuming is largely decreased. Also, when $N_{\mathbf{x}} = 128$, the size of the exponential matrix is 7423 × 7423 in the IIF2-SG-FD method. With a computer of 4GB memory, such computation is unaffordable and leads to the out of memory error. For the same situation, the AcIIF2-SG-FD method only computes the exponentials of less than 256 × 256 matrices.

In the RK2 method with sparse grid finite difference, due to the stability constrains, the time step has to be $\Delta t = 1/N_{\mathbf{x}}^2$. And this small time step leads to much more integration steps to reach the same final time. While a larger time step is chosen, the method is not convergent.

5.7.3 A diffusive logistic equation

The diffusive logistic equation:

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) = D\Delta_{\mathbf{x}} u + gu - u^2, x \in \Omega, t > 0, \tag{5.48}$$

describes the population density evolution. The unknown function u denote the population density of a specie, with respect to time t and location $\mathbf{x}$. The smooth function g defines the birth rate of the specie, which may be positive or negative. Different methods are develop to solve this type of equation [84]. We implement the AcIIF2-SG-FD method on the following dimensional diffusive logistic equation:

$$\begin{gathered}(x, y, z) \in \Omega = [0,1]^3, \qquad D = \frac{1}{3\pi^2}, \qquad u(x, y, z, 0) = \sin \pi x \sin \pi y \sin \pi z, \\ g(x, y, z, t) = e^{-t} \sin \pi x \sin \pi y \sin \pi z,\end{gathered} \tag{5.49}$$

and a zero Dirichlet boundary condition for $u(x, y, z, t)$. The exact solution for this system is:

$$u(x, y, z, t) = e^{-t} \sin \pi x \sin \pi y \sin \pi z. \tag{5.50}$$

In our implementation, each local non-linear equation is solved by the Newton iteration method, with a tolerance of 10^{-6}. The error is plotted in Figure 5.3(a), which is consistent with the analytic result. Also, the population density corresponding to location x and y is plotted in Figure 5.3(b), here the spatial resolution $N_{\mathbf{x}} = 128$, and $t = 0.5$, $z = 0.5$. It is very similar to the exact result, which is $e^{-0.5} \sin \pi x \sin \pi y$.

To make a comparison, we also implement the RK2 method with sparse grid finite difference. The CPU time for both methods are given in Table 5.2. The time step for RK2 is set to be $1/N_{\mathbf{x}}^2$, due to stability constrains. The results show the numerical efficiency of our method.

5.7.4 A three dimensional reaction-diffusion system with cross derivatives

We implement AcIIF2 method with the sparse grid combination technique to the following PDEs with cross derivatives:

$$u_t = (0.1u_{xx} - 0.15u_{xy} + 0.1u_{yy}) + (0.1u_{xx} + 0.2u_{xz} + 0.2u_{zz}) + (0.2u_{yy} + 0.15u_{yz} + 0.1u_{zz}) + 0.3u \tag{5.51}$$

where $x, y, z \in (0, 2\pi)$, and with a periodic boundary condition on $u(x, y, z, t)$ and a initial condition:

$$u(x, y, z, 0) = \sin(x + y + z). \tag{5.52}$$

The exact solution for the system is

$$u(x, y, z, 0) = e^{-0.2t} \sin(x + y + z). \tag{5.53}$$

The L_∞ error is plotted in Figure 5.3(c), which is consistent with the analytical error estimation. To demonstrate the supreme of apply the AcIIF method on each subproblem, we list the CPU times for some subproblems with different methods in Table 5.3. As it shown, the AcIIF method is always the fastest.

If the diffusive coefficient is not constant, then the pre calculation of matrix exponential may not be efficient in the AcIIF2 method. At this moment, other approximations of matrix exponential may be more suitable. For example, the Krylov subspace method [57, 61]. We also studied this case in Chapter 4.

5.7.5 A four dimensional Fokker-Planck equation

We solve the Fokker-Planck equation with respect to the bio-chemical system (4.72) with the AcIIF2 method and the sparse grid combination technique. All other coefficients are the same as in Subsection 4.5.3.

To solve this FPE, first of all, the domain needs to be determined. We choose the domain to be $[0, 100] \times [0, 100] \times [0, 50] \times [0, 50]$ such that the probability outside of this domain can be neglect. A zero Dirichlet boundary condition is applied for the upper bound. For lower one, the natural boundary is used. We start with a Gaussian distribution centering at $(30, 40, 15, 12)$ with a standard deviation of $\sqrt{40}$. There are nine reactions in the system and for the acIIF method, we group these reactions as the following: the first to fifth reaction denotes are grouped as one array-representation operator; the sixth and seventh are grouped as one; and eighth and ninth are grouped as one.

To use a sparse grid combination techniques, first we set up a spatial resolution $N_{\mathbf{x}} = 2^K$, then this FPE is solved on every subproblems. To meet the accuracy requirement, in the acIIF approach, we choose $\Delta t = 1/\max_i N_{x_i}$. We use multiple cores to run the simulation, then combine all the results. We also implement RK2 to solve each problem and we found that, to maintain the convergence, for certain subproblem (e.g. $N_{x_1} = N_{x_2} = ... = N_{x_{d-1}} = 2$ and $N_d = 2^{K+1-d}$), the time step must be taken very small, and these problems spend much more time than the acIIF approach.

We plot the density distribution in Figure 5.4. This density distribution is consistent with the corresponding system.

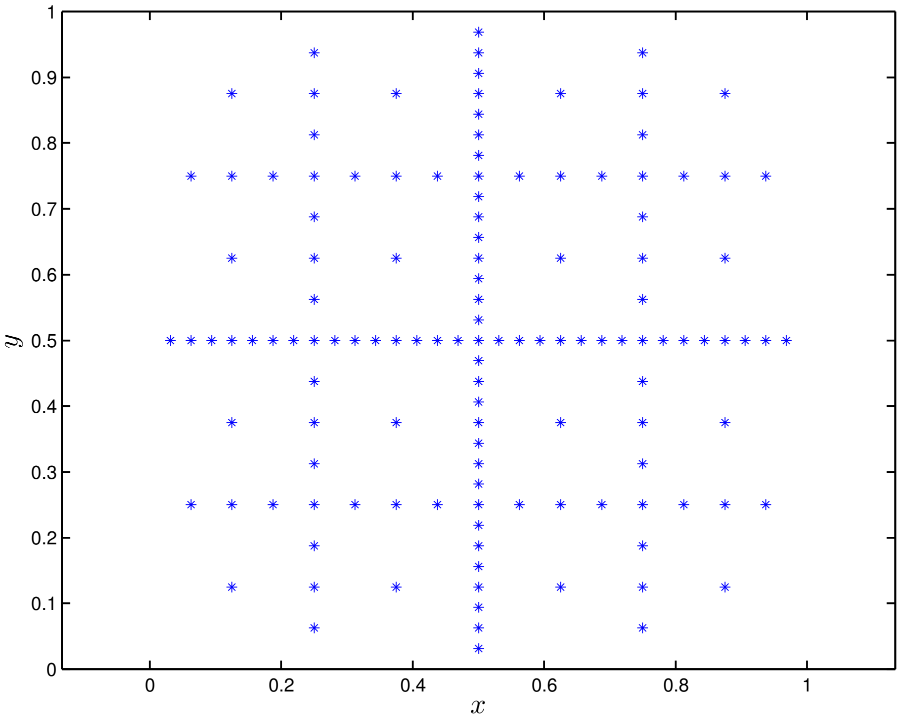

Figure 5.1: Sparse grid for $N_{\mathbf{x}} = 2^4$.

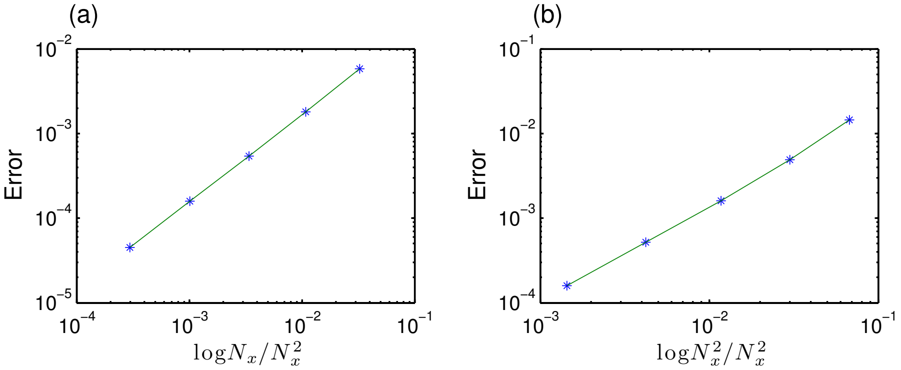

Figure 5.2: The spatial resolution versus L_∞ error for numerical tests. (a) The log-log error plot of the IIF2-SG-FEM on Eq. (5.42). The spatial resolution $N_x = 8, 16, 32, 64, 128$, the time step $\Delta t = 1/N_x$, and simulation ends at $t = 1$. (b) The log-log error plot of the acIIF2-SG-FD on Eq. (5.45). The spatial resolution $N_x = 8, 16, 32, 64, 128$, the time step $\Delta t = 1/N_{\mathbf{x}}$, and simulation ends at $t = 0.5$.

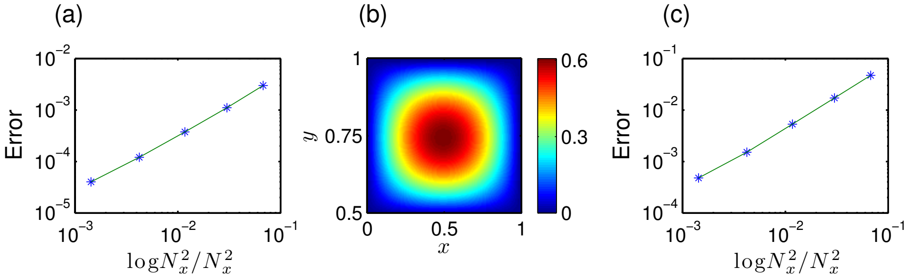

Figure 5.3: (a) The error plot of the AcIIF2-SG-FD method on the diffusive logistic Eq. (5.49). The spatial resolution N_x ranges from 2^3 to 2^7, the time step $\Delta t = 1/N_x$ and simulation ends at $t = 0.5$. (b) The populations density of Eq. (5.49) corresponding to location parameter x and y. The spatial resolution $N_x = 128$, the time $t = 0.5$ and the location parameter $z = 0.5$. (c) The error plot of the AcIIF2 method with the sparse grid combination technique on Eq. (5.51) which contains cross derivatives. The spatial resolution N_x ranges from 2^3 to 2^7, all the time steps for all subproblems are set to be $\Delta t = 1/N_{\mathbf{x}}$, and simulation ends at $t = 0.5$.

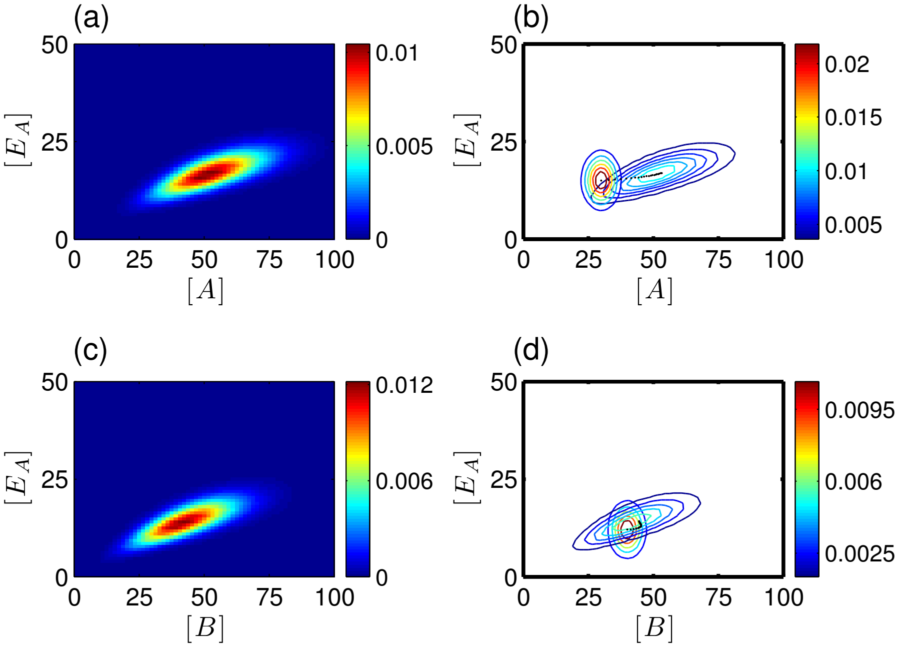

Figure 5.4: Numerical solution of system (4.72) using AcIIF2 with a sparse grid combination technique. The final time step is chosen to be $t = 45s$ and the spatial resolution is $K = 15$. (a) The distribution of metabolite A and related enzyme E_A at $t = 45$. (b) The contour plot of initial distribution and final distribution of A and E_A. The dotted black is the trace of the center by solving the corresponding ODE system. (c) The distribution of metabolite B and related enzyme E_B at $t = 45$. (d) The contour plot of initial distribution and final distribution of B and E_B. The dotted black is the trace of the center by solving the corresponding ODE system.

	acIIF2-SG-FD		IIF2-SG-FD		RK2 on SG-FD	
N_x	L_∞ error	CPU time	L_∞ error	CPU time	L_∞ error	CPU time
8	1.45×10^{-2}	0.07s	1.44×10^{-2}	0.03s	1.1×10^{-2}	0.02s
16	4.9×10^{-3}	0.3s	4.9×10^{-3}	0.4s	4.1×10^{-3}	0.39s
32	1.6×10^{-3}	1.7s	1.6×10^{-3}	8.7s	1.4×10^{-3}	8.5s
64	5.2×10^{-4}	12.8s	5.2×10^{-4}	166s	4.7×10^{-4}	190s
128	1.6×10^{-4}	114s	Out of memory		To long to get results	

Table 5.1: The implementation of acIIF2-SG-FD, IIF2-SG-FD and RK2-SG-FD on Eq. (5.45). In both acIIF2-SG-FD and IIF2-SG-FD methods, the time step is $\Delta t = 1/N_{\mathbf{x}}$. In RK2-SG-FD, the time step is $\Delta t = 1/N_{\mathbf{x}}^2$. All the simulations end at $t = 0.5$.

	acIIF2-SG-FD		RK2 on SG-FD	
N_x	L_∞ error	CPU time	L_∞ error	CPU time
8	3.0×10^{-3}	0.08s	2.9×10^{-3}	0.02s
16	1.1×10^{-3}	0.32s	1.1×10^{-3}	0.4s
32	3.7×10^{-4}	2.0s	3.7×10^{-4}	8.6s
64	1.2×10^{-4}	16s	1.2×10^{-4}	194s
128	4.0×10^{-5}	161s	Too long to get results	

Table 5.2: The numerical results of acIIF2-SG-FD and RK2 on SG-FD for the three spatial dimensional diffusive logistic equation Eq. (5.49). For the acIIF2-SG-FD method, the time step is $\Delta t = 1/N_x$ and for the RK2 on SG-FD, the time step is $\Delta t = 1/N_x^2$ due to the constrains of stability. All the simulation ends at $t = 0.5$.

(d, K)	$(N_{x_1}, ..., N_{x_d})$	CPU time for acIIF on matrix exponentials	CPU time for IIF on matrix exponential
$(3, 11)$	$(2^2, 2^2, 2^7)$	0.3s	3.5s
$(3, 11)$	$(2^3, 2^4, 2^4)$	0.2s	5.4s
$(3, 12)$	$(2^2, 2^2, 2^8)$	4.3s	51s
$(3, 12)$	$(2^4, 2^4, 2^4)$	0.4s	24s
$(4, 11)$	$(2^2, 2^2, 2^2, 2^5)$	0.05s	5.6s
$(4, 12)$	$(2^2, 2^2, 2^2, 2^6)$	0.12s	68s

Table 5.3: The CPU time for the computation of all matrix exponentials in both the acIIF and IIF methods. The time step is set to be $\Delta t = 1/\max_i N_{x_i}$ to meet the error requirement Eq. (5.40) for the sparse grid combination technique.

Bibliography

[1] J.Lei and Y.Song. Mathematical model of the formation of morphogen gradients through membrane-associated non-receptors. *Bull.Math.Biol*, 72:805–829, 2010.

[2] J.Lei, D.Wang, Y.Song, Q.Nie, and F.Y.M Wan. Robustness of morphogen gradients with "bucket brigade" transport through membrane-associated non-receptors. *Disc. Cont. Dyns. Syst. B*, 18(3):721–739, 2013.

[3] A.Lander. Morpheus unbound: Reimagining the morphogen gradient. *Cell*, 128:245–256, 2007.

[4] G.Baeg, E.M.Selva, R.M.Goodman, R.Dasgupta, and N.Perrimon. The wingless morphogen gradient is established by the cooperative action of frizzled and heparin sulfate proteoglycan receptors. *Dev.Biol*, 276:89–100, 2004.

[5] T.Y.Belenkaya, C.Han, D.Yan, R.J.Opoka, M.Khodoun, H.Liu, and X.Lin. Drosophila dpp morphogen movement is independent of dynamin-mediated endocytosis but regulated by the glypican members of heparan sulfate proteoglycans. *Cell*, 119:231–244, 2004.

[6] A.Eldar, D.Rosin, B.Shilo, and N.Barkai. Self-enhanced ligand degradation underlies robustness of morphogen gradients. *Dev.Cell*, 5:635–646, 2003.

[7] E.V.Entchev, A.Schwabedissen, and M.Gonzalez-Gaitan. Gradient formation of the tgf-β homolog dpp. *Cell*, 103:981–991, 2000.

[8] M.Fujise, S.Takeo, K.Kamimura, T.Matsuo, T.Aigaki, S.Izumi, and H.Nakato. Dally regulates dpp morphogen gradient formation in the drosphila wing. *Development*, 130:1515–1522, 2003.

[9] A.Kicheva, P.Pantazis, T.Bollenbach, Y.Kalaidzidis, T.Bittig, F.Jülicher, and M.Gonzàlez-Gaitàn. Kinetics of morphogen gradient formation. *Science*, 315:521–525, 2007.

[10] C.A.Kirkpatrick, B.D.Dimitroff, J.M.Rawson, and S.B.Selleck. Spatial regulation of wingless morphogen distribution and signaling by dally-like protein. *Dev.Cell*, 7:513–523, 2004.

[11] A.Lander, Q.Nie, and F.Y.M.Wan. Do morphogen gradients arise by diffusion? *Dev.Cell*, 2:785–796, 2002.

[12] J.Vincent and L.Dubois. Morphogen transport along epithelia and integrated trafficking problem. *Dev.Cell*, 2:615–623, 2002.

[13] T.Bollenbach, K.Kruse, P.Pantazis, M.Gonzàlez-Gaitàn, and F.Jülicher. Morphogen transport in epithelia. *Phys. Rev. E*, 75, 2007.

[14] M.Kerszberg. Accurate reading of morphogen concentrations by nuclear receptors: a formal model of complex transduction pathways. *J.Theor.Biol*, 183:95–104, 1996.

[15] M.Kerszberg and L.Wolpert. Mechanisms for positional signaling by morphogen transport: a theoretical study. *J.Theor.Biol*, 191:103–114, 1998.

[16] A.Lander, Q.Nie, and F.Y.M.Wan. Spatially distributed morphogen production and morphogen gradient formation. *Math.Biosci*, 2:239–262, 2005.

[17] A.Lander, Q.Nie, and F.Y.M.Wan. Internalization and end flux in morphogen gradient formation. *J. Comp. Appl. Math*, 190:232–251, 2006.

[18] J.Lei, F.Y.M.Wan, A.Lander, and Q.Nie. Robustness of signaling gradient in drosophila wing imaginal disc. *Disc. Cont. Dyns. Syst. B*, 16:835–866, 2011.

[19] L.Wolpert. Position information and patterning revisited. *J. Theor. Biol*, 269:359–365, 2011.

[20] T.Bollenbach, K.Kruse, P.Pantazis, M.Gonzàlez-Gaitàn, and F.Jülicher. Robust formation of morphogen gradients. *Phys. Rev. Lett*, 94, 2005.

[21] T.Akiyama, K.Kamimura, C.Firkus, S.Takeo, O.Shimmi, and H.Nakato. Dally regulates dpp morphogen gradient formation by stabilizing dpp on the cell surface. *Dev.Biol*, 313:408–419, 2008.

[22] D.J.Bornemann, J.E.Duncan, W.Staatz, S.Selleck, and R.Warrior. Abrogation of heparan sulfate synthesis in drosophila disrupts the wingless,hedgehog and decapentaplegic signaling pathways. *Development*, 131:1927–1938, 2004.

[23] L.Hufnagel, J.Kreuger, S.M.Cohen, and B.I.Shraiman. On the role of glypicans in the process of morphogen gradient formation. *Dev.Biol*, 300:512–522, 2006.

[24] A.Lander, Q.Nie, and F.Y.M.Wan. Membrane-associated non-receptors and morphogen gradients. *Bull.Math.Biol*, 69:33–54, 2007.

[25] J.Lei. Mathematical model of the dpp gradient formation in drosophila wing imaginal disc. *Chinese Sci. Bull*, 55:984–991, 2010.

[26] Y.Lou, Q.Nie, and F.Y.M.Wan. Effects of sog on dpp-receptor binding. *SIAM J.Appl.Math*, 65:1748–1771, 2005.

[27] Y.-T.Zhang, A.Lander, and Q.Nie. Computational analysis of bmp gradients in dorsal-ventral patterning of the zebrafish embryo. *J.Theor.Biol*, 248:579–589, 2007.

[28] W.Lo, L.Chen, M.Wang, and Q.Nie. A robust and efficient method for steady state patterns in reaction-diffusion systems. *J.Comp.Phys*, 231:5062–5077, 2012.

[29] Q.Nie, Y.Zhang, and R.Zhao. Efficient semi-implicit schemes for stiff systems. *J. Comp. Phys*, 214:521–537, 2006.

[30] K.S.Stenn and R.Paus. Controls of hair follicle cycling. *Physiological reviews*, 81(1):449, 2001.

[31] R.Paus and K.Foitzik. In search of the "hair cycle clock": a guided tour. *Differentiation*, 72(9):489–511, 2004.

[32] M.R.Schneider, R.Schmidt-Ullrich, and R.Paus. the hair follicle as a dynamic miniorgan. *Current Biology*, 19(3):132–142, 2009.

[33] Y.Al-Nuaimi et al. The cycling hair follicle as an ideal systems biology research model. *Experimental dermatology*, 19(8):707–713, 2010.

[34] T-T.Sun, G.Costsarelis, and R.M.Lavker. Hair follicular stem cells: the bulge-activation hypothesis. *J. of investigative dermatology*, 96:77–78, 1991.

[35] A.A.Panteleyev et al. the role of the hairless ("hr") gene in the regulation of hair follicle catagen transformation. *The American J of pathology*, 155:159–171, 1999.

[36] H.B.Chase. Growth of the hair. *Physiological Reviews*, 34:113, 1954.

[37] E.B.Lane et al. Modeling hair follicle growth dynamics as an excitable medium. *PLoS comput. bio*, 8(12), 2012.

[38] H.Oshima et al. Morphogenesis and renewal of hair follicles from adult mutipotent stem cells. *Cell*, 104(2):233–245, 2001.

[39] M.V.Plikus et al. Self-organizing and stochastic behaviors during the regeneration of hair stem cells. *Science*, 332:586–589, 2011.

[40] V.Greco et al. A two-step mechanism for stem cell activation during hair regeneration. *Cell stem cell*, 4(2):155–169, 2009.

[41] D.T.Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. of Physical Chemistry*, 81(25):2340–2361, 1977.

[42] D.Wang, L.Zhang, and Q.Nie. Array-representation integration factor method for high-dimensional systems. *J. Comp. Phys*, 256:585–600, 2014.

[43] D.T.Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem*, 81(25):2340–2361, 1977.

[44] H.Risken. *The Fokker-Planck equation: Methods of solutions and applications*. Springer, 1996.

[45] A.Kassam, Lloyd, and N.Trefethen. Fourth-order time stepping for stiff PDEs. *SIAM J. Sci. Comp*, 26:1214–1233, 2005.

[46] B.Kleefeld, A.Q.M.Khaliq, and B.A.Wade. An ETD Crank-Nicolson method for reaction-diffusion systems. *Numer. Methods Partial Differential Eq*, 28:1309–1335, 2012.

[47] S.M.Cox and P.C.Matthews. Exponential time differencing for stiff systems. *J. Comp. Phys*, 176(2):430– 455, 2002.

[48] Q.Du and W. Zhu. Stability analysis and application of the exponential time differencing schemes. *J. Comp. Math*, 22(2), 2004.

[49] S.Krogstad. Generalized integrating factor methods for stiff PDEs. *J. Comp. Phys*, 203:72–88, 2005.

[50] Q. Nie, Y.-T.Zhang, and R.Zhao. Efficient semi-implicit schemes for stiff systems. *J. Comp. Phys*, 214(2):521–537, 2006.

[51] Q. Nie, F.Y.M Wan, Y.-T.Zhang, and X. Liu. Compact integration factor methods in high spatial dimensions. *J. Comp. Phys*, 227(10):5238–5255, 2008.

[52] X. Liu and Q. Nie. Compact integration factor methods for complex domains and adaptive mesh refinement. *J. Comp. Phys*, 229:5692–5706, 2010.

[53] X.-D.Liu, S.Osher, and T.Chan. Weighted essentially nonoscillatory schemes. *J. Comp. Phys*, 115:200–212, 1994.

[54] G.-S.Jiang and C.-W.Shu. Efficient implementation of weighted ENO schemes. *J. Comp. Phys*, 126:202–228, 1996.

[55] S. Zhao, J. Ovadia, X. Liu, Y.-T.Zhang, and Q.Nie. Operator splitting implicit integration factor methods for stiff reaction-diffusion-advection systems. *J. Comp. Phys*, 230:5996–6009, 2011.

[56] S. Chen and Y.-T.Zhang. Krylov implicit integration factor methods for spatial discretization on high dimensional unstructured meshes: Application to discontinuous Galerkin method. *J. Comp. Phys*, 230:4336–4352, 2011.

[57] E.Gallopoulos and Y.Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Stat. Comp*, 13(5):1236–1264, 1992.

[58] T.Y. Hou, J.S. Lowengrub, and M.J. Shelley. Removing the stiffness from interfacial flows with surface tension. *J. Comp. Phys*, 114:312–338, 1994.

[59] G. Beylkin, J.M. Keiser, and L. Vozovoi. A new class of tie discretization schemes for the solution of nonlinear PDEs. *J. Com. Phys*, 147:362–387, 1998.

[60] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer Anal*, 5:506–517, 1968.

[61] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal*, 34(5):1911–1925, 1997.

[62] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal*, 29(1):209–228, 1992.

[63] C.Moler and C.V. Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45:3–49, 2003.

[64] D.T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188:404–425, 1992.

[65] T. Jahnke and W. Huisinga. Solving the chemical master equation for monomolecular reaction systems analytically. *J. Math Biol*, 54(1):1–26, 2007.

[66] W. Verena. etc. Solving the chemical master equation using sliding windows. *BMC Systems Biology*, 4(42), 2010.

[67] P. Sjoberg, P.Lotstedt, and J. Elf. Fokker-Planck approximation of the master equation in molecular biology. *Comp. Visual Sci.*, 12:37–50, 2009.

[68] J.W. Cahn and J.E.Hilliard. Free energy of a nonuniform system. I. Interfacial free energy. *J. Chem. Phys*, 28:258–351, 1958.

[69] J. Shen and H.Yu. Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems. *SIAM J. Sci. Comp*, 32:3228–3250, 2010.

[70] Y.Zhu, X.Wu, and I Chern. *Derivative securities and difference methods.* Springer Verlag, New York, 2004.

[71] R.Griffiths. The frequency spectrum of a mutation, and its age, in a general diffusion model. *Theor. Popul. Biol*, 64(2):241–251, 2003.

[72] H.-J.Bungartz and M.Griebel. Sparse grids. *Acta Numer.*, 13:147–269, 2004.

[73] H.-J.Bungartz and M.Griebel. A note on the complexity of solving Poisson's equation for spaces of bound mixed derivatives. *J. Complexity*, 15:167–199, 1999.

[74] G.Deslauries and S.Dubuc. Symmetric iterative interpolation processes. *Constr. Approx*, 5:51–150, 1989.

[75] S.Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.*, 4:240–243, 1963.

[76] H.-J.Bungartz. Concepts for higher order finite elements on sparse grids. In A.V.Ilin and L.R.Scott, editors, *Proceedings of the 3rd Int. Conf. on Spectral and High Order Metehods.* in Houston Journal of Mathematics, 1995.

[77] C.Zenger. Sparse grids. In W.Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations*, volume 31. Notes on Numerical Fluid Mechanics, 1991.

[78] M.Griebel. A parallelizable of vectorizble multi-level algorithm on sparse grids. In W.Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations*, volume 31. Notes on Numerical Fluid Mechanics, 1991.

[79] M.Griebel. Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences. *Computing*, 61:151–179, 1998.

[80] M.Griebel, M.Schneider, and C.Zenger. A combination technique for the solution of sparse grid problems. In P.de Groen and R.Beauwens, editors, *Iterative Methods in Linear Algebra*, pages 263–281, 1992.

[81] P.Hemker. Sparse-grid finite-volume multi grid for 3D problems. *Adv. Comput. Math.*, 4:83–110, 1995.

[82] F.Kupka. *Sparse grid spectral methods for the numerical solution of partial differential equations with periodic boundary conditions*. PhD thesis, Insitut fur Mathematik, Universitat Wien, 1997.

[83] C.C.W. Leentvaar and C.W. Oosterlee. On coordinate transformation and grid stretching for sparse grid pricing of basket options . *Journal of Computational and Applied Mathematics*, 222(1):193 – 209, 2008.

[84] G.A. Afrouzi and S. Khademloo. Numerical solutions of diffusive logistic equation. *Chaos, Solitons and Fractals*, 31(1):112 – 118, 2007.