



135
381
THS

1
2000

This is to certify that the

dissertation entitled

Multigrid Methods for
Solving Reaction-Diffusion Systems

presented by

Hsiu-Chuan Wei

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Mathematics



Major professor

Date August 15, 2000

**LIBRARY
Michigan State
University**

**PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.**

| DATE DUE | DATE DUE | DATE DUE |
|------------|----------|----------|
| 00509 2002 | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Multigrid Methods for Solving Reaction-Diffusion Systems

By

Hsiu-Chuan Wei

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Mathematics

August 15, 2000

ABSTRACT

Multigrid methods for solving reaction-diffusion systems

By

Hsiu-Chuan Wei

A Multigrid method as an iterative method is proposed to solve a reaction-diffusion system. The model we are concerned in this thesis is a system of parabolic partial differential equations for two chemical species. A fully implicit finite difference scheme is used to discretize the differential equations. A V-cycle scheme with one smoothing step per grid is then applied to the Helmholtz equations [54] arising at each time step. The convergence of the V-cycle scheme for solving the linear system at each time step is obtained. The stability and convergence of the fully implicit scheme along with V-cycle as the iterative solver are proved. Numerical results for a reaction-diffusion system are presented.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Dr. ChiChia Chiu, my thesis advisor, for her kind guidance during each phase of this project. Our many and varied discussions are vital to my development. With each obstacle encountered, she offered me countless suggestions that help me overcome each hurdle and move forward to succeed. I also wish to thank Dr. Chang Yi Wang, Dr. Charles MacCluer, Dr. Tien-Yien Li, Dr. Baisheng Yan and Dr. Zhengfang Zhou on my committee for their kind guidance and invaluable suggestions. Without their expertise, I could not have finished this project.

TABLE OF CONTENTS

| | |
|---|-----------|
| LIST OF FIGURES | v |
| LIST OF TABLES | vi |
| INTRODUCTION | 1 |
| 1 Fully implicit discretization | 8 |
| 2 Two-grid algorithm and its convergence | 13 |
| 3 V-cycle and its convergence | 25 |
| 4 Stability and Convergence | 34 |
| 5 Experimental Results | 43 |
| 5.1 The Thomas System | 44 |
| 5.2 The Schnakenberg System | 49 |
| 5.3 Comparison | 52 |
| BIBLIOGRAPHY | 56 |

LIST OF FIGURES

| | | |
|-----|--|----|
| 2.1 | Grids for Ω_k and Ω_{k-1} | 14 |
| 5.1 | Possible patterns estimated by linear stability analysis. | 46 |
| 5.2 | Turing's patterns obtained by numerical simulation. | 47 |
| 5.3 | Numerical simulation of Turing pattern of (5.1) and residual plot with $\gamma = 8$ | 47 |
| 5.4 | Numerical simulation of Turing pattern of (5.1) and residual plot with $\gamma = 48$ | 48 |
| 5.5 | Numerical simulation of Turing pattern of (5.1) and residual plot on a cone surface. | 48 |
| 5.6 | Numerical simulation of Turing pattern of (5.3) and residual plot of with $\gamma = 15$ | 50 |
| 5.7 | Numerical simulation of Turing pattern of (5.3) and residual plot with $\gamma = 30$ | 51 |
| 5.8 | Numerical simulation of Turing pattern of (5.3) and residual plot with $\gamma = 50$ | 51 |
| 5.9 | Numerical simulation of Turing pattern on a square. | 53 |

LIST OF TABLES

| | | |
|-----|--|----|
| 5.1 | Comparison between single grid and multigrid with $h = 0.05$ and $\Delta t = 0.01$. | 53 |
| 5.2 | Comparison between single grid and multigrid with $h = 0.025$ and $\Delta t = 0.0025$. | 53 |
| 5.3 | Comparison between single grid and multigrid with $h = 0.025$ and $\Delta t = 0.005$. | 53 |
| 5.4 | Comparison between single grid and multigrid with $h = 0.0125$ and $\Delta t = 0.0025$. | 54 |

Introduction

Reaction-diffusion systems are systems of parabolic partial differential equations. They have been used successfully to model chemical and biological processes which involve pattern formation in morphogenesis. Since all models for spatial pattern generation are necessarily nonlinear, analytical solutions are generally not available. Numerical solutions of such equations are thus important for computer simulation of patterns and data analysis. In this thesis, general two-dimensional reaction diffusion systems are considered. A multigrid method is proposed for solving the system. Analysis and applications will be discussed.

Fascinating spatial patterns, such as animal coat patterns and butterfly wing patterns, are exhibited on living creatures. The development of pattern and form in embryology is known as morphogenesis. It begins from a more or less homogeneous egg. The final complexity of pattern and form are generated during its development. How the patterns are established is still unknown. Although the process of the development is genetically determined, the genes themselves cannot generate the pattern since the genetic material in most cells of an organism can be assumed the same [39]. Much research has tried to determine the mechanisms that can explain the generation of patterns developed from less structured tissues. Therefore any mechanism assumed to explain this development must be capable of generating spatial patterns from almost homogeneous initial conditions. Several mechanisms have been proposed as possible pattern formation mechanisms in morphogenetic situations. Among these,

the most powerful mechanism for forming coat patterns is a reaction diffusion mechanism proposed by Turing [3]. A. M. Turing first suggested in 1952 that some patterns that occur in chemistry result from interaction between chemical reaction and diffusion. Since then a substantial amount of research has been done on this subject. See [41] for a survey of related developments.

According to Turing's chemical prepattern approach, the first step of development in the morphogenetic process is the creation of a morphogen concentration spatial pattern. The underlying prepattern is believed to be laid down in the very early stages of the embryogenesis. Consider the case of the zebra. The gestation is about 360 days with the prepattern laid down about 3-5 weeks. After the prepattern is once established, the cells differentiate accordingly to produce melanin to reflect the spatial pattern of morphogen concentration. For example, the pattern of the giraffe could be generated when the morphogen concentration level is greater than some threshold level.

When applying Turing's theory of morphogenesis, an important step is to identify the chemical elements for the morphogens. For example, calcium has been identified as a morphogen for hair initiation in *Acetabularia*. In this thesis, we will assume that the pattern is generated when one of the morphogen concentrations is above some threshold level. Now let us consider the following two-dimensional reaction-diffusion system:

$$\begin{cases} u_t = D_u \Delta u + f(u, v) \\ v_t = D_v \Delta v + g(u, v), \end{cases} \quad (0.1)$$

$$\left. \frac{\partial u}{\partial n} \right|_{\partial \Omega} = \left. \frac{\partial v}{\partial n} \right|_{\partial \Omega} = 0,$$

where $u(x, 0)$ and $v(x, 0)$ are given.

Here we take the rectangular domain : $\Omega = [0, L_1] \times [0, L_2] \subset R^2$, $\partial \Omega$ is the boundary of Ω , and u and v are considered as two morphogen concentrations; $f(u, v)$ and

$g(u, v)$ are reaction kinetics which describe the interrelation between morphogens. The diffusion terms, $D_u \Delta u$ and $D_v \Delta v$, reflect that each molecule can move around randomly with diffusivities $D_u > 0$ and $D_v > 0$.

Reaction-diffusion mechanisms suggest that spatial patterns are evolved from diffusion-driven instability. Turing also demonstrated that, under certain conditions, two interactive morphogens could form a stable inhomogeneous concentration pattern. He suggested that without diffusion, u and v tend to a uniform steady state – no pattern will form. In the presence of diffusion, the system is unstable to small disturbances but the instability will be bounded by nonlinear reaction terms. Thus the inhomogeneous steady state is obtained and spatial patterns are generated [52]. All that is required for the creation of pattern is some sort of nonlinear activator-inhibitor mechanism [26]. The analysis of whether or not the systems are able to generate spatial patterns and how the pattern and mode is selected can be found in [41].

To formulate the mathematical model, we need initial and boundary conditions. Since the formation of the underlying prepatter could arise from instability to small perturbations in the featureless tissue, a natural way to simulate this situation in numerical computations is by introducing a small random perturbation about the uniform steady state. In the system given above, we assume initial conditions are given. As for the choice of boundary conditions, we take zero flux boundary conditions because it implies no external forces.

Under certain conditions, the reaction-diffusion system described above has a unique solution. The proof of the existence and uniqueness of the solution can be found in [11]. The system is useful for modeling patterns in chemistry and biology. A few examples can be found in [13]. In these examples, the concentration of nutrient and the concentration of buffer satisfy a reaction-diffusion system as given in (0.1).

Since the reaction term, f and g , in (0.1) is nonlinear, one can not hope in general

to obtain analytical solutions. Numerical solutions with preliminary linear analyses are required in studying the patterns generated by a given reaction-diffusion system. Attempts have been made to demonstrate the patterns formed by such mechanisms and compare them with the mammalian coat patterns [3] [41]. Since the spatial patterns are generated by diffusion-driven instability, the system has to be unstable. Because of this nature, highly stable numerical methods are necessary for computer simulations of these patterns to ensure that the patterns obtained by computer simulations are formed by the instability of the original system, not by numerical instability.

One approach is an ADI (alternating direction implicit) type of scheme [13] [14]. Other numerical methods that have been used to solve reaction-diffusion systems are finite element techniques, monotone iterative methods and explicit finite difference schemes [21] [27] [40] [46] [56]. Finite difference schemes have been used due to their simplicity. Explicit finite difference schemes are especially easy to implement but the conditional stability forces small step size in time in numerical experiments. Consequently, the explicit schemes require long runtime. Another intuitive approach is using a fully implicit scheme which is unconditionally stable. The discrete problem arising from a fully implicit finite difference scheme is a large sparse system of linear equations. Existing methods for solving linear systems can be grouped into two branches: direct methods and iterative methods. Although direct methods produce exact solutions in a finite number of steps (regardless of roundoff errors), they can not be applied here because the complexity needed to invert the matrix is of cubic order, and because the inverse of a sparse matrix may not be sparse. The computation time and computer memory for a solution obtained by direct methods are much too great. We therefore consider using iterative methods.

Our concern about the numerical scheme now turns to its convergence rate in practice. In order to obtain the inhomogeneous solutions, the computations have to be carried on until equilibrium is reached. Even though the unconditional stability

of fully implicit finite difference schemes enable us to take a much greater step size in t , it still requires thousands of steps in t to reach the steady state. Thus in these numerical computations, we need to solve a pair of large linear systems at each time step for at least thousands of time steps. Therefore, the running time mainly depends on the efficiency of the iterative method used for solving the linear system at each time step. Unfortunately, the classic iterative methods, namely Jacobi, weighted Jacobi and Gauss-Seidel methods, have unsatisfactory convergence rates. They have been known to be able to remove the high frequency modes in an error efficiently, but are unable to damp the low frequency modes. Finding a fast solver for a large linear system becomes an important issue. In this thesis, a multigrid method is proposed for solving this problem. Consider that the smooth components which are efficiently approximated on coarse grids and the oscillatory components are fast to converge on fine grids. Multigrid techniques could eliminate all frequency components.

Multigrid methods as iterative methods have been known to be a fast solver for linear systems arising from the discretization of partial differential boundary-value problems [4] [35]. Numerous works about numerical experiments and theoretical understanding of the convergence properties of these methods have appeared in the past three decades [2] [4]-[8] [20] [29]-[38]. Reported numerical experiments suggest that these methods are very efficient for a wide range of practical problems [10] [54].

There are many convergence proofs to multigrid algorithms. One approach to these methods is local Fourier analysis [22] [28] [54]. It is not generally rigorous. Local Fourier analysis gives realistic quantitative results on the convergence behavior yet assumes an unbounded domain in space. Thus it can be regarded as an analysis only for problems with periodic boundary conditions. With Neumann boundary condition in our model problem (0.1), we can view the multigrid as a single operator and study the norm of the operator. Some of the proofs have been given in [16]-[18] yet these proofs require a sufficiently large number of relaxation sweeps. In practice, multigrid

is used only with one or a few relaxation sweeps. In our numerical computations (Chapter 5), only one presmoothing and one postsmoothing are needed in almost every example.

Other proofs are based on the approach for variational formulation. Intensive research into the convergence of multigrid methods for variational problems can be seen in [4] [7]-[9] [29]-[38]. These proofs usually require a ‘regularity and approximation’ assumption. Convergence rates have been guaranteed with any amount of smoothing for solving linear systems $Ax = b$, with A symmetric positive definite. The linear system arising from discretization of (0.1) is unfortunately nonsymmetric. Then redefining an inner product and its corresponding norm by the midpoint rule [13] is proposed to formulate a variational-like problem and construct the proof of the V-cycle algorithm. Variational framework is a natural formulation when finite element discretization is used. In this thesis, since finite difference discretization is used and a new inner product is defined, more attention needs to be paid to the constructions of the transformations between grids. For the choice of the smoother, a red-black Gauss-Seidel method is applied because it has been used extensively as a smoother in multigrid methods [54] [60]. The convergence of this smoother under the norm used in this thesis is discussed in Chapter 3. The rigorous proof of the convergence of a V-cycle algorithm is also given with any amount of smoothing.

This thesis focuses on the construction of the multigrid algorithm that can be applied to our model problem and the proof of the convergence of the algorithm followed by the demonstration of the efficiency of the algorithm. In the numerical computations, two reaction systems have been used to demonstrate the V-cycle algorithm. They are able to generate spots and stripes on rectangular domains. We should note that even the analysis has been discussed on a general rectangular domain due to its simplicity. We believe the V-cycle algorithm constructed in this thesis is also valid for domains in other shapes. Thus the pattern on the surface of a cone has been

simulated in numerical experiments. Finally, we compare the running time of the V-cycle algorithm with that of the computations carried out only on the fine grid to demonstrate the high efficiency of the former.

The outline of this thesis is as follows. In Chapter 1, the finite difference scheme is constructed. In Chapter 2, the two-grid algorithm is developed and its convergence is proved. In Chapter 3, we extend the two-grid algorithm to a V-cycle algorithm and its convergence is obtained. In Chapter 4, we discuss the stability and convergence of the fully implicit finite difference scheme along with the V-cycle algorithm described in Chapter 1 and 3. Finally in Chapter 5, the experimental results for the numerical solutions to two reaction-diffusion systems are presented.

CHAPTER 1

Fully implicit discretization

In this section we begin our study of the fully implicit finite difference discretization of the reaction-diffusion system (0.1). For simplicity, consider a uniform rectangular mesh on a rectangular domain $\Omega = [0, L_1] \times [0, L_2]$ with mesh size $h = L_1/N_1 = L_2/N_2$. Let $(x_i, y_j) \in \Omega$ be a grid point then $x_{i+1} = x_i + h$ and $y_{j+1} = y_j + h$. The increment in t , $t^{n+1} - t^n$, will be denoted by Δt , and we adopt the standard notation $u_{ij}^n \approx u(x_i, y_j, t^n)$.

The basic idea of the finite difference discretization is to approximate the derivatives in a differential equation by the difference quotients. For example,

$$\frac{\partial u}{\partial t}(x_i, y_j, t^n) \simeq \frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t}$$

can be seen from the derivative formula

$$\frac{\partial u}{\partial t}(x, y, t) = \lim_{\varepsilon \rightarrow 0} \frac{u(x, y, t + \varepsilon) - u(x, y, t)}{\varepsilon}.$$

Similarly, the second order derivative $\frac{\partial^2 u}{\partial x^2}(x_i, y_j, t^n)$ can be approximated as follows:

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j, t^n) \simeq u_{i+1j}^n - 2u_{ij}^n + u_{i-1j}^n.$$

Here we propose to use a fully implicit scheme. We approximate (0.1) by the

backward-time central-space scheme [53]. We have

$$\begin{cases} \frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = D_u \left(\frac{\Delta_{xh} u_{ij}^{n+1}}{h^2} + \frac{\Delta_{yh} u_{ij}^{n+1}}{h^2} \right) + f(u_{ij}^n, v_{ij}^n) \\ \frac{v_{ij}^{n+1} - v_{ij}^n}{\Delta t} = D_v \left(\frac{\Delta_{xh} v_{ij}^{n+1}}{h^2} + \frac{\Delta_{yh} v_{ij}^{n+1}}{h^2} \right) + g(u_{ij}^n, v_{ij}^n), \end{cases} \quad (1.1)$$

where Δ_{xh} and Δ_{yh} are the centered second order difference operators such that

$$(\Delta_{xh} + \Delta_{yh})u_{ij}^n = (u_{i+1j}^n - 2u_{ij}^n + u_{i-1j}^n) + (u_{ij+1}^n - 2u_{ij}^n + u_{ij-1}^n).$$

The zero flux boundary condition is approximated by

$$\begin{cases} u_{iN_2+1}^n = u_{iN_2-1}^n, & u_{i(-1)}^n = u_{i1}^n, & \text{for all } i, n, \\ u_{N_1+1j}^n = u_{N_1-1j}^n, & u_{(-1)j}^n = u_{1j}^n, & \text{for all } j, n. \end{cases}$$

The boundary treatment for v can be defined in the same way. Let $u^n = [u_{ij}^n]$ and $v^n = [v_{ij}^n]$ be the vectors obtained by the usual ordering. Then (1.1) can be written in the following matrix forms:

$$(I + \tau \tilde{A})u^{n+1} = u^n + \Delta t f^n \quad (1.2)$$

$$(I + \tau' \tilde{A}')v^{n+1} = v^n + \Delta t g^n, \quad (1.3)$$

where $\tau = D_u \Delta t / h^2$, $\tau' = D_v \Delta t / h^2$, \tilde{A} is the standard matrix resulting from the discretization operator $-\Delta_{xh} - \Delta_{yh}$, $f^n = [f(u_{ij}^n, v_{ij}^n)]$, and $g^n = [g(u_{ij}^n, v_{ij}^n)]$. Let $A = I + \tau \tilde{A}$ and $A' = I + \tau' \tilde{A}'$. Then (1.2) and (1.3) become

$$Au^{n+1} = u^n + \Delta t f^n \quad (1.4)$$

$$A'v^{n+1} = v^n + \Delta t g^n. \quad (1.5)$$

where

$$r_{ij}^{n+1}, s_{ij}^{n+1} = \begin{cases} O(\Delta t + h^2) & \text{if } ij \text{ is an interior index,} \\ O(\Delta t + h) & \text{if } ij \text{ is a boundary index.} \end{cases}$$

Proof: Use the Taylor series expansion

$$\tilde{u}_{ij}^n = \tilde{u}_{ij}^{n+1} - \Delta t (\tilde{u}_t)_{ij}^{n+1} + O(\Delta t^2)$$

to obtain

$$\frac{\tilde{u}_{ij}^{n+1} - \tilde{u}_{ij}^n}{\Delta t} = (\tilde{u}_t)_{ij}^{n+1} + O(\Delta t) \quad \text{for all } ij .$$

Consider the Taylor series expansions

$$\tilde{u}_{i+1j}^{n+1} = \tilde{u}_{ij}^{n+1} + h(\tilde{u}_x)_{ij}^{n+1} + \frac{h^2}{2}(\tilde{u}_{xx})_{ij}^{n+1} + \frac{h^3}{6}(\tilde{u}_{xxx})_{ij}^{n+1} + O(h^4)$$

and

$$\tilde{u}_{i-1j}^{n+1} = \tilde{u}_{ij}^{n+1} - h(\tilde{u}_x)_{ij}^{n+1} + \frac{h^2}{2}(\tilde{u}_{xx})_{ij}^{n+1} - \frac{h^3}{6}(\tilde{u}_{xxx})_{ij}^{n+1} + O(h^4).$$

Add the equations above and use the approximation for the zero flux boundary condition to obtain

$$\Delta_{xh} \tilde{u}_{ij}^{n+1} = \begin{cases} (\tilde{u}_{xx})_{ij}^{n+1} + O(h^2) & \text{if } i \text{ is an interior index,} \\ (\tilde{u}_{xx})_{ij}^{n+1} n + O(h) & \text{if } i \text{ is a boundary index.} \end{cases}$$

Similar expansions also hold for $\Delta_{yh} \tilde{u}_{ij}^{n+1}$ as well as the other concentration function \tilde{v} . Also,

$$f(\tilde{u}_{ij}^{n+1}, \tilde{v}_{ij}^{n+1}) = f(\tilde{u}_{ij}^n, \tilde{v}_{ij}^n) + f_u(\zeta_{ij}^{n+1}, \eta_{ij}^{n+1})(\zeta_{ij}^{n+1} - \tilde{u}_{ij}^n) + f_v(\zeta_{ij}^{n+1}, \eta_{ij}^{n+1})(\eta_{ij}^{n+1} - \tilde{v}_{ij}^n),$$

where $(\zeta_{ij}^{n+1}, \eta_{ij}^{n+1})$ is between $(\tilde{u}_{ij}^{n+1}, \tilde{v}_{ij}^{n+1})$ and $(\tilde{u}_{ij}^n, \tilde{v}_{ij}^n)$. Since $\tilde{u}_{ij}^{n+1} - \tilde{u}_{ij}^n = O(\Delta t)$, we have $\zeta_{ij}^{n+1} - \tilde{u}_{ij}^n = O(\Delta t)$ and $\eta_{ij}^{n+1} - \tilde{v}_{ij}^n = O(\Delta t)$. Thus $f(\tilde{u}_{ij}^{n+1}, \tilde{v}_{ij}^{n+1}) = f(\tilde{u}_{ij}^n, \tilde{v}_{ij}^n) + O(\Delta t)$. We have

$$\begin{aligned} \frac{\tilde{u}_{ij}^{n+1} - \tilde{u}_{ij}^n}{\Delta t} &= D_u \left(\frac{\Delta_{xh} \tilde{u}_{ij}^{n+1}}{h^2} + \frac{\Delta_{yh} \tilde{u}_{ij}^{n+1}}{h^2} \right) + f(\tilde{u}_{ij}^n, \tilde{v}_{ij}^n) \\ &+ \begin{cases} O(\Delta t + h^2) & \text{if } ij \text{ is an interior index,} \\ O(\Delta t + h) & \text{if } ij \text{ is a boundary index.} \end{cases} \end{aligned}$$

This proves the consistency of this discretization. □

Normally (1.4) and (1.5) are large systems of linear equations and the matrices A and A' are banded and sparse. Then direct methods are often impractical because the cost for inverting these two matrices is too expensive. Therefore we propose to use an iterative method to solve (1.4) and (1.5). However the classical iterative methods seem to have unsatisfactory convergence properties. Typically the high frequency components of the error damp out quickly while there is a very slow decay of the error on low frequency components. Then a multigrid technique becomes a natural way to accelerate the convergence.

CHAPTER 2

Two-grid algorithm and its convergence

In this chapter, we describe the fundamental multigrid idea. To this end, we construct a two-grid algorithm which is the basis of the multigrid method. There are many iterative methods that are directly relevant to the multigrid method. Within the two-grid method we will use the Gauss-Jacobi iteration as the smoother.

Let $\{\Omega_k\}_{k=0}^{k=J}$ be a nested sequence of vector spaces :

$$\Omega_0 \subset \Omega_1 \subset \dots \subset \Omega_J.$$

The vector space Ω_k corresponds to the spatial discretization size, $h_k = h_{k-1}/2$. Let $m_{1k} = L_1/h_k$ and $m_{2k} = L_2/h_k$. Then Ω_k is a mesh containing $(m_{1k} + 1) \times (m_{2k} + 1)$ points.

To define a multigrid algorithm, we need to define the linear projectors transferring vectors between vector spaces. There are various choices of such projectors [54]. Here

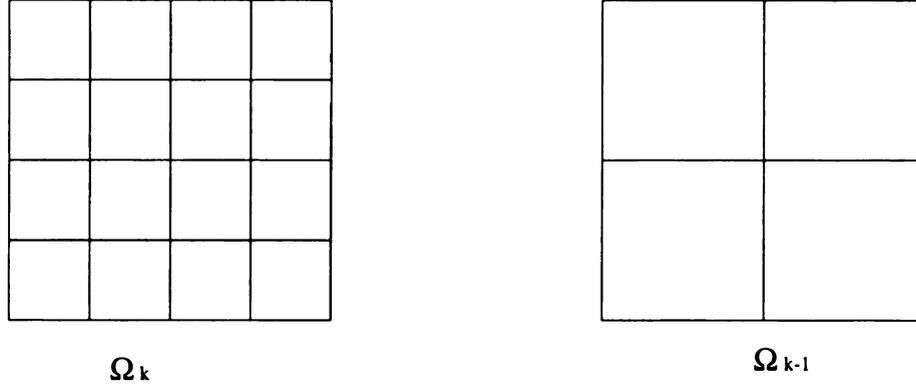


Figure 2.1: Grids for Ω_k and Ω_{k-1} .

we use the full weighting restriction $P_k^{k-1} : \Omega_k \rightarrow \Omega_{k-1}$ whose stencil is

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{for an interior point,}$$

$$\frac{1}{16} \begin{bmatrix} 2 & 4 & 2 \\ 2 & 4 & 2 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{for a boundary point with index } j = 0,$$

and

$$\frac{1}{16} \begin{bmatrix} 0 & 4 & 4 \\ 0 & 4 & 4 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{for the corner point with index } i, j = 0.$$

This restriction takes fine grid vectors and produces the coarse grid vectors according to the rule $P_k^{k-1}u = v$ for $u \in \Omega_k$ and $v \in \Omega_{k-1}$, where

$$\begin{aligned} v_{ij} &= \frac{1}{16} [u_{2i+1,2j+1} + u_{2i+1,2j-1} + u_{2i-1,2j+1} + u_{2i-1,2j-1} + 2(u_{2i,2j+1} + u_{2i,2j-1} \\ &\quad + u_{2i+1,2j} + u_{2i-1,2j}) + 4u_{2i,2j}], \quad 1 \leq i \leq (m_{1k-1} - 1) \text{ and } 1 \leq j \leq (m_{2k-1} - 1), \\ v_{0j} &= \frac{1}{16} [2u_{1,2j+1} + 2u_{1,2j-1} + 2u_{0,2j+1} + 2u_{0,2j-1} + 4u_{1,2j} + 4u_{0,2j}], \end{aligned}$$

stencil notation is given by

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{for an interior point,}$$

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{for a boundary point with index } j = 0,$$

and

$$\frac{1}{4} \begin{bmatrix} 0 & 2 & 1 \\ 0 & 4 & 2 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{for the corner point with index } i, j = 0.$$

If $u \in \Omega_k$ and $v \in \Omega_{k-1}$, then $Q_{k-1}^k v = u$ with the components of u given by

$$\begin{aligned} u_{2i,2j} &= v_{ij}, & 0 \leq i \leq m_{1k-1} \text{ and } 0 \leq j \leq m_{2k-1}, \\ u_{2i+1,2j} &= \frac{1}{2}(v_{ij} + v_{i+1,j}), & 0 \leq i \leq m_{1k-1} - 1 \text{ and } 0 \leq j \leq m_{2k-1}, \\ u_{2i,2j+1} &= \frac{1}{2}(v_{ij} + v_{i,j+1}), & 0 \leq i \leq m_{1k-1} \text{ and } 0 \leq j \leq m_{2k-1} - 1, \end{aligned}$$

and

$$\begin{aligned} u_{2i+1,2j+1} &= \frac{1}{4}(v_{ij} + v_{i+1,j} + v_{i,j+1} + v_{i+1,j+1}), & 0 \leq i \leq m_{1k-1} - 1 \\ & & \text{and } 0 \leq j \leq m_{2k-1} - 1. \end{aligned}$$

the grid Ω_{k-1} and with components

$$\begin{aligned}\alpha_{k-1} &= (36\alpha_k + 96\beta_k + 64\gamma_k)/64, \\ \beta_{k-1} &= (6\alpha_k + 32\beta_k + 32\gamma_k)/64, \\ \gamma_{k-1} &= (\alpha_k + 8\beta_k + 16\gamma_k)/64.\end{aligned}$$

Recall that the fine grid operator $A_J = A$ is five-point. The transformation leads to a nine-point stencil for A_{J-1} . Fortunately, the coarser grids continue to have the nine-point stencils as shown above. This increases the computation cost somewhat. However the choices of the operators defined previously lead to nice properties by which the convergence of the multigrid process is guaranteed.

Now \tilde{A} in (1.4) is symmetric and nonnegative definite with respect to the discrete L^2 inner product $(\cdot, \cdot)_J$ on $L^2(\Omega_J)$ determined by the midpoint rule [12]

$$(u, v)_J = h_J^2 \gamma_i \gamma_j \sum_{k,l=0}^{k,l=N} u_{kl} v_{kl},$$

where $\gamma_i = 1/2$ if $i = 0$ or $i = N$, and $\gamma_i = 1$ otherwise, and $h_J = h$. The corresponding norm is $\|u\|_J = \sqrt{(u, u)_J}$.

We also define the inner product on each Ω_k in the same way. The matrix $A = I + \tau\tilde{A}$ is SPD (symmetric positive definite) with respect to $(\cdot, \cdot)_J$. Furthermore we have $(P_k^{k-1}u, v)_{k-1} = (u, Q_{k-1}^k v)_k$ for every $u \in \Omega_k$ and $v \in \Omega_{k-1}$. It will be shown in the following lemma that each A_k is also SPD with respect to $(\cdot, \cdot)_k$.

Lemma 2.1 *If A_k is SPD with respect to $(\cdot, \cdot)_k$, then A_{k-1} is SPD with respect to $(\cdot, \cdot)_{k-1}$. Therefore $(A_{k-1})^{-1}$ exists.*

Proof: Let $x \in \Omega_{k-1}$ and $x \neq 0$. Then

$$\begin{aligned}(A_{k-1}x, x)_{k-1} &= (P_k^{k-1}A_k Q_{k-1}^k x, x)_{k-1} \\ &= (A_k Q_{k-1}^k x, Q_{k-1}^k x)_k && \text{(positive)} \\ &= (Q_{k-1}^k x, A_k Q_{k-1}^k x)_k\end{aligned}$$

$$\begin{aligned}
&= (x, P_k^{k-1} A_k Q_{k-1}^k x)_{k-1} \\
&= (x, A_{k-1} x)_{k-1}. \quad (\text{symmetric})
\end{aligned}$$

Now A_{k-1} is SPD with respect to $(\cdot, \cdot)_{k-1}$. Consider the linear system $A_{k-1}x = 0$ with $x \in \Omega_{k-1}$. Note $(A_{k-1}x, x)_{k-1} = (0, x)_{k-1} = 0$. But $(A_{k-1}x, x)_{k-1} > 0$ unless $x = 0$. But $x = 0$ is the only solution to $A_{k-1}x = 0$. Thus $(A_{k-1})^{-1}$ exists. \square

Since A_J is SPD, from Lemma 2.1, A_k is SPD for $1 \leq k \leq J$ and $(A_k)^{-1}$ exists.

Now we can define an inner product by $A(u, v) = (A_k u, v)_k$, with corresponding norm $\|u\| = \sqrt{A(u, u)}$ for $u, v \in \Omega_k$.

We recall the following well known properties of real symmetric matrices. The proof can be obtained by following the same procedure as for the standard 2-norm [1].

Lemma 2.2 (a) Let B be a $(m_{1k}+1)(m_{2k}+1) \times (m_{1k}+1)(m_{2k}+1)$ matrix, symmetric with respect to $(\cdot, \cdot)_k$. Then $\|B\|_k = \rho(B)$, where $\rho(B)$ is the spectral radius of B .

(b) Let C be a $(m_{1k}+1)(m_{2k}+1) \times (m_{1k}+1)(m_{2k}+1)$ matrix, symmetric with respect to $A(\cdot, \cdot)$. Then $\|C\| = \rho(C)$.

For the purpose of analysis, we define an auxiliary operator $S_k^{k-1} : \Omega_k \longrightarrow \Omega_{k-1}$ by $S_k^{k-1} = (A_{k-1})^{-1} P_k^{k-1} A_k$. So, we have

$$A_{k-1} S_k^{k-1} = P_k^{k-1} A_k.$$

Next, we shall construct a two-grid scheme for the linear system in (1.4). Let D_J be the diagonal matrix which consists of the diagonal elements of $A = A_J$, $f_J^n = u^n + \Delta t f^n$, and x^{n+1} be the exact solution of (1.4). So (1.4) becomes

$$A_J x^{n+1} = f_J^n. \quad (2.1)$$

The Gauss-Jacobi method can be expressed as follows. Let $A_J = B_J + D_J$,

$$D_J u^{(i+1)} = -B_J u^{(i)} + f_J^n,$$

$$\begin{aligned}
D_J u^{(i+1)} &= (D_J - A_J)u^{(i)} + f_J^n, \\
u^{(i+1)} &= (D_J)^{-1}(D_J - A_J)u^{(i)} + (D_J)^{-1}f_J^n, \\
u^{(i+1)} &= u^{(i)} - (D_J)^{-1}A_J u^{(i)} + (D_J)^{-1}f_J^n, \\
u^{(i+1)} &= u^{(i)} + (D_J)^{-1}(f_J^n - A_J u^{(i)}).
\end{aligned}$$

The two-grid algorithm for solving (2.1) is given below.

Algorithm 2.1

$$\begin{aligned}
\text{Step1.} \quad u^{(0)} &= u^n + (D_J)^{-1}(f_J^n - A_J u^n). \\
\text{Step2.} \quad u^{(1)} &= u^{(0)} + Q_{J-1}^J q, \\
&\quad A_{J-1} q = P_{J-1}^{J-1}(f_J^n - A_J u^{(0)}). \\
\text{Step3.} \quad u^{n+1} &= u^{(1)} + (D_J)^{-1}(f_J^n - A_J u^{(1)}).
\end{aligned}$$

Where Step1 is a pre-relaxation sweep using the Gauss-Jacobi iteration on the fine grid. Step2 is coarse grid correction which solves the coarse grid problem exactly. Step3 is a post-relaxation sweep on the fine grid.

To establish the convergence of the two-grid algorithm, we need the following lemma.

Lemma 2.3 (a) $S_{k+1}^k Q_k^{k+1} = I_k$.

(b) $Q_{k-1}^k S_k^{k-1}$ is symmetric with respect to $A(\cdot, \cdot)$, i.e. for $u, v \in \Omega_k$, $A(Q_{k-1}^k S_k^{k-1} u, v) = A(u, Q_{k-1}^k S_k^{k-1} v)$.

(c) $Q_{k-1}^k S_k^{k-1}$ is idempotent, i.e. $(Q_{k-1}^k S_k^{k-1})^2 = Q_{k-1}^k S_k^{k-1}$, so its eigenvalues are 0 and 1.

Proof: (a) The result follows directly from the definitions of A_k and S_{k+1}^k .

$$S_{k+1}^k Q_k^{k+1} = (A_k)^{-1} P_{k+1}^k A_{k+1} Q_k^{k+1} = (A_k)^{-1} (A_k) = I_k.$$

(b) Let $u, v \in \Omega_k$, we have

$$\begin{aligned}
A(Q_{k-1}^k S_k^{k-1} v, u) &= (A_k Q_{k-1}^k S_k^{k-1} v, u)_k \\
&= (S_k^{k-1} v, P_k^{k-1} A_k u)_{k-1} \\
&= (S_k^{k-1} v, A_{k-1} S_k^{k-1} u)_{k-1} \\
&= (A_{k-1} S_k^{k-1} v, S_k^{k-1} u)_{k-1} \\
&= (P_k^{k-1} A_k v, S_k^{k-1} u)_{k-1} \\
&= (A_k v, Q_{k-1}^k S_k^{k-1} u)_k \\
&= A(v, Q_{k-1}^k S_k^{k-1} u).
\end{aligned}$$

(c) It follows from part (a) that $(Q_{k-1}^k S_k^{k-1})^2 = Q_{k-1}^k S_k^{k-1} Q_{k-1}^k S_k^{k-1} = (Q_{k-1}^k I_{k-1} S_k^{k-1}) = (Q_{k-1}^k S_k^{k-1})$. An idempotent matrix is similar to a diagonal matrix of the form $diag(1, \dots, 1, 0, \dots, 0)$ [59]. \square

Let x^{n+1} be the exact solution of the linear system (2.1) and u^{n+1} be the numerical solution obtained by the two-grid algorithm described above. Let the initial error be $e_0 = x^{n+1} - u^n$ and the error after the two-grid scheme be $e_1 = x^{n+1} - u^{n+1}$.

Theorem 2.1 *The error $e_1 = K e_0$, and $|||e_1||| \leq |||K||| \cdot |||e_0|||$ where K is the error operator with $|||K||| < 1$. The two-grid scheme is convergent.*

Proof: Since

$$\begin{aligned}
e_1 &= x^{n+1} - u^{n+1} \\
&= x^{n+1} - u^{(1)} - (D_J)^{-1} (f_J^n - A_J u^{(1)}) \\
&= x^{n+1} - u^{(1)} - (D_J)^{-1} A_J (x^{n+1} - u^{(1)}) \\
&= (I - (D_J)^{-1} A_J) (x^{n+1} - u^{(1)}) \\
&= (I - (D_J)^{-1} A_J) (x^{n+1} - u^{(0)} - Q_{J-1}^J (A_{J-1})^{-1} P_J^{J-1} A_J (x^{n+1} - u^{(0)})) \\
&= (I - (D_J)^{-1} A_J) (I - Q_{J-1}^J (A_{J-1})^{-1} P_J^{J-1} A_J) (x^{n+1} - u^{(0)})
\end{aligned}$$

$$\begin{aligned}
&= (I - (D_J)^{-1}A_J)(I - Q_{J-1}^J(A_{J-1})^{-1}A_{J-1}S_J^{J-1})(x^{n+1} - u^{(0)}) \\
&= (I - (D_J)^{-1}A_J)(I - Q_{J-1}^J S_J^{J-1})(x^{n+1} - u^{(0)}) \\
&= (I - (D_J)^{-1}A_J)(I - Q_{J-1}^J S_J^{J-1})(I - D_J^{-1}A_J)(x^{n+1} - u^n) \\
&= (I - (D_J)^{-1}A_J)(I - Q_{J-1}^J S_J^{J-1})(I - (D_J)^{-1}A_J)e_0,
\end{aligned}$$

We have $K = (I - (D_J)^{-1}A_J)(I - Q_{J-1}^J S_J^{J-1})(I - (D_J)^{-1}A_J)$. Next we shall verify that $|||K||| < 1$.

Since A is strictly diagonally dominant, $\rho(I - (D_J)^{-1}A_J) < 1$. Also from Lemma 2.3(c), $\rho(I - Q_{J-1}^J S_J^{J-1}) = 1$. Since D_J is symmetric with respect to $(\cdot, \cdot)_J$, $I - (D_J)^{-1}A_J$ is symmetric with respect to $A(\cdot, \cdot)$. From Lemma 2.3(b), $I - Q_{J-1}^J S_J^{J-1}$ is also symmetric with respect to $A(\cdot, \cdot)$, so we have $|||I - D_J^{-1}A_J||| = \rho(I - D_J^{-1}A_J) < 1$ and $|||I - Q_{J-1}^J S_J^{J-1}||| = \rho(I - Q_{J-1}^J S_J^{J-1}) = 1$.

Thus $|||K||| \leq |||I - (D_J)^{-1}A_J||| \cdot |||I - Q_{J-1}^J S_J^{J-1}||| \cdot |||I - (D_J)^{-1}A_J||| < 1$. This completes the convergence of two-grid algorithm. \square

To see that the coarse grid correction Step2 in the two-grid algorithm reduces the norm of the error, assume that after Step1, the initial error e_0 becomes $e^{(1)}$. Let $x = Q_{J-1}^J S_J^{J-1} e^{(1)}$ and $y = (I - Q_{J-1}^J S_J^{J-1})e^{(1)}$, so $e^{(1)} = x + y$. From Lemma 2.3, parts (b) and (c), we have that $A(x, y) = 0$ and $Q_{J-1}^J S_J^{J-1}$ is an A -orthogonal projector. We have $|||e^{(1)}|||^2 = |||x|||^2 + |||y|||^2$. After Step2, the norm of error $|||e^{(1)}|||$ is altered to $|||(I - Q_{J-1}^J S_J^{J-1})e^{(1)}||| = |||y|||$, i.e. the coarse grid correction eliminates component errors in $Range(Q_{J-1}^J S_J^{J-1})$.

Two-grid methods, however, usually are not used in practice. Note that (1.4) and (1.5) are large systems. Even if we reduce the number of grid points by 1/2 in each direction in coarse grid correction, the resulting system in the next coarser grid is still large. Consequently, in the coarse grid correction step, we still need to invert a matrix of large size. To solve this problem, as we have pointed out, two-grid scheme

serves only as the basis of the multigrid methods. Thus instead of solving the grid $J - 1$ problem exactly as in Step2 in the two-grid algorithm, we can modify Step2 as a recursive procedure which suggests a V-cycle algorithm. In the next chapter, we will further discuss the V-cycle algorithm which is used to solve (1.4) and (1.5) for the examples given in Chapter 5.

CHAPTER 3

V-cycle and its convergence

In this chapter, we shall construct a V-cycle algorithm to solve (2.1) and prove its convergence. Let $(R_k)^{-1}$ be the smoothing matrix for solving the linear system $A_k u_k = g_k$:

$$\begin{aligned}(R_k)^{-1} u_k^{(i)} &= ((R_k)^{-1} - A_k) u_k^{(i-1)} + g_k, \\(R_k)^{-1} u_k^{(i)} &= (R_k)^{-1} u_k^{(i-1)} + g_k - A_k u_k^{(i-1)}, \\u_k^{(i)} &= u_k^{(i-1)} + R_k (g_k - A_k u_k^{(i-1)}).\end{aligned}$$

The V-cycle algorithm is given below.

Algorithm 3.1 *If $k = 1$, solve $A_1 u_1 = g_1$ exactly. We may assume that $B_1 = (A_1)^{-1}$.*

For $k > 1$, solve $A_k u_k = g_k$ by the algorithm described below with $g_J = f_J$ and B_k is defined in terms of B_{k-1} .

$$\begin{aligned}\text{Step1.} \quad u_k^{(1)} &= u_k^{(0)} + R_k (g_k - A_k u_k^{(0)}), \\ &\text{with } u_k^{(0)} = 0 \text{ if } k < J \text{ and } u_k^{(0)} = u^n \text{ if } k = J.\end{aligned}$$

$$\begin{aligned}\text{Step2.} \quad u_k^{(2)} &= u_k^{(1)} + Q_{k-1}^k B_{k-1} g_{k-1}, \\ &\text{where } g_{k-1} = P_k^{k-1} (g_k - A_k u_k^{(1)}).\end{aligned}$$

$$\text{Step3.} \quad B_k g_k = u_k^{(2)} + R_k(g_k - A_k u^{(2)}).$$

For the smoother R_k , it is sufficient to choose a convergent iterative method. Here we will use the red-black Gauss-Seidel method because of its good smoothing rate for elliptic problems [54] [60]. Let $((R_k)^{-1})^t$ be the adjoint of $(R_k)^{-1}$ with respect to $(\cdot, \cdot)_k$. It can be verified that $((R_k)^{-1})^t = A_k + D_k - (R_k)^{-1}$, where D_k is the diagonal matrix which consists of the diagonal elements of A_k . In the following lemma, we will show that $(R_k)^{-1} + ((R_k)^{-1})^t - A_k = D_k$ is positive definite. Thus red-black Gauss-Seidel method is convergent with respect to $||| \cdot |||$.

Lemma 3.1 (a) *The diagonal elements of A_k are positive for $1 \leq k \leq J$, so D_k is positive definite.*

$$(b) \quad |||I_k - R_k A_k||| < 1 .$$

Proof: (a) Since A_k is SPD, $(A_k e_i, e_i)_k$ is positive and so is the i th diagonal element of A_k for $0 \leq i \leq m_k$, where e_i is the vector with '1' on the i th position and '0' at all other entries. Thus D_k is also positive definite.

(b) For $u, v \in \Omega_k$, we need first to show that $A(R_k A_k u, v) = A(u, (R_k)^t A_k v)$ and $((R_k)^t)^{-1} = ((R_k)^{-1})^t$. Now

$$\begin{aligned} A(R_k A_k u, v) &= (R_k A_k u, A_k v)_k \\ &= (A_k u, (R_k)^t A_k v)_k \\ &= A(u, (R_k)^t A_k v). \end{aligned}$$

But

$$\begin{aligned} (u, v)_k &= (R_k (R_k)^{-1} u, v)_k \\ &= ((R_k)^{-1} u, (R_k)^t v)_k \\ &= (u, ((R_k)^{-1})^t (R_k)^t v)_k. \end{aligned}$$

Since $((R_k)^{-1})^t (R_k)^t = I_k$, we have $((R_k)^{-1})^t = ((R_k)^t)^{-1}$.

Next we will prove the inequality in part (b).

$$\begin{aligned}
0 &\leq A((I_k - R_k A_k)u, (I_k - R_k A_k)u) \\
&= A((I_k - (R_k)^t A_k)(I_k - R_k A_k)u, u) \\
&= A([I_k - (R_k)^t A_k - R_k A_k + (R_k)^t A_k R_k A_k]u, u) \\
&= A(u, u) - A([(R_k)^t A_k + R_k A_k - (R_k)^t A_k R_k A_k]u, u) \\
&= A(u, u) - A([(R_k)^t (R_k)^{-1} R_k A_k + (R_k)^t ((R_k)^t)^{-1} R_k A_k - (R_k)^t A_k R_k A_k]u, u) \\
&= A(u, u) - A((R_k)^t [(R_k)^{-1} + ((R_k)^t)^{-1} - A_k] R_k A_k u, u) \\
&= A(u, u) - ((R_k)^t [(R_k)^{-1} + ((R_k)^t)^{-1} - A_k] R_k A_k u, A_k u)_k \\
&= A(u, u) - ([(R_k)^{-1} + ((R_k)^t)^{-1} - A_k] R_k A_k u, R_k A_k u)_k \\
&= A(u, u) - (D_k R_k A_k u, R_k A_k u)_k \\
&< A(u, u).
\end{aligned}$$

Since D_k is positive definite, $(D_k R_k A_k u, R_k A_k u)_k > 0$ if u is not zero. So $\|I_k - R_k A_k\| < 1$ and the red-black Gauss-Seidel smoother is convergent for each k . \square

For analysis purposes we will define the following operators:

$$S_J^k = S_{k+1}^k S_{k+2}^{k+1} \cdots S_J^{J-1},$$

$$P_J^k = P_{k+1}^k P_{k+2}^{k+1} \cdots P_J^{J-1},$$

and

$$Q_k^J = Q_{J-1}^J Q_{J-2}^{J-1} \cdots Q_k^{k+1}.$$

Then we have the following lemma.

Lemma 3.2 (a) $(P_J^k v, w)_k = (v, Q_k^J w)_J$ if $v \in \Omega_J$ and $w \in \Omega_k$.

(b) $A_k = P_J^k A_J Q_k^J$.

$$(c) A_k S_J^k = P_J^k A_J.$$

$$(d) S_J^k Q_k^J = I_k.$$

Proof: To prove part (a),

$$\begin{aligned} (P_J^k v, w)_k &= (P_{k+1}^k P_{k+2}^{k+1} \cdots P_J^{J-1} v, w)_k \\ &= (P_{k+2}^{k+1} \cdots P_J^{J-1} v, Q_k^{k+1} w)_{k+1} \\ &= (v, Q_{J-1}^J Q_{J-2}^{J-1} \cdots Q_k^{k+1} w)_J \\ &= (v, Q_k^J w)_J. \end{aligned}$$

For part (b),

$$\begin{aligned} P_J^k A_J Q_k^J &= P_{k+1}^k P_{k+2}^{k+1} \cdots P_J^{J-1} A_J Q_{J-1}^J Q_{J-2}^{J-1} \cdots Q_k^{k+1} \\ &= P_{k+1}^k \cdots P_{J-1}^{J-2} A_{J-1} Q_{J-2}^{J-1} \cdots Q_k^{k+1} \\ &= P_{k+1}^k A_{k+1} Q_k^{k+1} \\ &= A_k. \end{aligned}$$

For part (c),

$$\begin{aligned} A_k S_J^k &= A_k S_{k+1}^k S_{k+2}^{k+1} \cdots S_J^{J-1} \\ &= A_k A_k^{-1} P_{k+1}^k A_{k+1} S_{k+2}^{k+1} \cdots S_J^{J-1} \\ &= P_{k+1}^k A_{k+1} S_{k+2}^{k+1} \cdots S_J^{J-1} \\ &= P_{k+1}^k P_{k+2}^{k+1} A_{k+1} \cdots S_J^{J-1} \\ &= P_{k+1}^k P_{k+2}^{k+1} \cdots P_J^{J-1} A_J \\ &= P_J^k A_J. \end{aligned}$$

For part (d),

$$\begin{aligned} S_J^k Q_k^J &= S_{k+1}^k \cdots S_{J-1}^{J-2} S_J^{J-1} Q_{J-1}^J Q_{J-2}^{J-1} \cdots Q_k^{k+1} \\ &= S_{k+1}^k \cdots S_{J-1}^{J-2} I_{J-1} Q_{J-2}^{J-1} \cdots Q_k^{k+1} \end{aligned}$$

$$\begin{aligned}
&= S_{k+1}^k \cdots S_{J-1}^{J-2} Q_{J-2}^{J-1} \cdots Q_k^{k+1} \\
&= S_{k+1}^k \cdots I_{J-2} \cdots Q_k^{k+1} \\
&= S_{k+1}^k Q_k^{k+1} \\
&= I_k
\end{aligned}$$

□

Next we want to find the relation between the errors before and after the V-cycle. Let x_k be the solution of $A_k x_k = g_k$ and u_k be the numerical solution obtained at Step3 in the V-cycle algorithm. Then

$$\begin{aligned}
&(I_k - B_k A_k) x_k \\
&= x_k - B_k g_k \\
&= x_k - u_k \\
&= x_k - u_k^{(2)} - R_k A_k (x_k - u_k^{(2)}) \\
&= (I_k - R_k A_k) (x_k - u_k^{(2)}) \\
&= (I_k - R_k A_k) (I_k - Q_{k-1}^k B_{k-1} P_k^{k-1} A_k) (x_k - u_k^{(1)}) \\
&= (I_k - R_k A_k) (I_k - Q_{k-1}^k B_{k-1} A_{k-1} S_k^{k-1}) (x_k - u_k^{(1)}) \\
&= (I_k - R_k A_k) (I_k - Q_{k-1}^k B_{k-1} A_{k-1} S_k^{k-1}) (I_k - R_k A_k) (x_k - u_k^{(0)}). \quad (3.1)
\end{aligned}$$

For $k < J$, $u_k^{(0)} = 0$,

$$(I_k - B_k A_k) x_k = (I_k - R_k A_k) (I_k - Q_{k-1}^k B_{k-1} A_{k-1} S_k^{k-1}) (I_k - R_k A_k) x_k.$$

Thus

$$I_k - B_k A_k = (I_k - R_k A_k) (I_k - Q_{k-1}^k B_{k-1} A_{k-1} S_k^{k-1}) (I_k - R_k A_k). \quad (3.2)$$

For $k = J$, let $x^{n+1} = x_J$. Then $A_J x^{n+1} = f_J^n = u^n + \Delta t f^n$. Let $u^{n+1} = u_J$, where u_J is obtained by the V-cycle algorithm. Then

$$x^{n+1} - u^{n+1}$$

$$\begin{aligned}
&= x_J - u_J \\
&= (I_J - R_J A_J)(I_J - Q_{J-1}^J B_{J-1} A_{J-1} S_J^{J-1})(I_J - R_J A_J)(x^{n+1} - u^n). \quad (3.3)
\end{aligned}$$

Next, we will show that $x^{n+1} - u^{n+1}$ can be written as a product of operators. We have

$$\begin{aligned}
&I_J - Q_k^J B_k A_k S_J^k \\
&= I_J - Q_k^J S_J^k + Q_k^J S_J^k - Q_k^J B_k A_k S_J^k \\
&= I_J - Q_k^J S_J^k + Q_k^J (I_k - B_k A_k) S_J^k \\
&= I_J - Q_k^J S_J^k \\
&\quad + Q_k^J (I_k - R_k A_k) (I_k - Q_{k-1}^k B_{k-1} A_{k-1} S_k^{k-1}) (I_k - R_k A_k) S_J^k. \quad (3.4)
\end{aligned}$$

Here we used (3.2) to rewrite $(I_k - B_k A_k)$.

Next, using $S_J^k Q_k^J = I_k$, we have

$$\begin{aligned}
Q_k^J (I_k - R_k A_k) &= Q_k^J - Q_k^J R_k A_k \\
&= Q_k^J - Q_k^J R_k A_k S_J^k Q_k^J \\
&= (I_J - Q_k^J R_k A_k S_J^k) Q_k^J.
\end{aligned}$$

Similarly

$$\begin{aligned}
(I_k - R_k A_k) S_J^k &= S_J^k - R_k A_k S_J^k \\
&= S_J^k - S_J^k Q_k^J R_k A_k S_J^k \\
&= S_J^k (I_J - Q_k^J R_k A_k S_J^k).
\end{aligned}$$

Rewrite the last term of (3.4) to obtain

$$\begin{aligned}
&Q_k^J (I_k - R_k A_k) (I_k - Q_{k-1}^k B_{k-1} A_{k-1} S_k^{k-1}) (I_k - R_k A_k) S_J^k \\
&= (I_J - Q_k^J R_k A_k S_J^k) Q_k^J (I_k - Q_{k-1}^k B_{k-1} A_{k-1} S_k^{k-1}) S_J^k (I_J - Q_k^J R_k A_k S_J^k) \\
&= (I_J - Q_k^J R_k A_k S_J^k) (Q_k^J S_J^k - Q_{k-1}^J B_{k-1} A_{k-1} S_J^{k-1}) (I_J - Q_k^J R_k A_k S_J^k).
\end{aligned}$$

Thus we have

$$\begin{aligned}
& I_J - Q_k^J B_k A_k S_J^k \\
= & (I_J - Q_k^J S_J^k) \\
& + (I_J - Q_k^J R_k A_k S_J^k)(Q_k^J S_J^k - Q_{k-1}^J B_{k-1} A_{k-1} S_J^{k-1})(I_J - Q_k^J R_k A_k S_J^k). \quad (3.5)
\end{aligned}$$

Again, since $S_J^k Q_k^J = I_k$,

$$(I_J - Q_k^J S_J^k) Q_k^J R_k A_k S_J^k = 0$$

and

$$Q_k^J R_k A_k S_J^k (I_J - Q_k^J S_J^k) = 0,$$

with

$$(I_J - Q_k^J S_J^k)(I_J - Q_k^J R_k A_k S_J^k) = (I_J - Q_k^J S_J^k)$$

and

$$(I_J - Q_k^J R_k A_k S_J^k)(I_J - Q_k^J S_J^k) = (I_J - Q_k^J S_J^k).$$

Thus

$$(I_J - Q_k^J R_k A_k S_J^k)(I_J - Q_k^J S_J^k)(I_J - Q_k^J R_k A_k S_J^k) = (I_J - Q_k^J S_J^k). \quad (3.6)$$

Then from (3.4), (3.5), and (3.6),

$$I_J - Q_k^J B_k A_k S_J^k = (I_J - Q_k^J R_k A_k S_J^k)(I_J - Q_{k-1}^J B_{k-1} A_{k-1} S_J^{k-1})(I_J - Q_k^J R_k A_k S_J^k).$$

Let

$$\begin{cases} T_J = I_J - R_J A_J, \\ T_k = (I_J - Q_k^J R_k A_k S_J^k) & \text{for } 2 \leq k \leq J-1, \\ T_1 = I_J - Q_1^J S_1^J. \end{cases}$$

Then

$$I_J - Q_k^J B_k A_k S_J^k = T_k (I_J - Q_{k-1}^J B_{k-1} A_{k-1} S_J^{k-1}) T_k, \quad 2 \leq k \leq J. \quad (3.7)$$

Now we can write $x^{n+1} - u^{n+1}$ as a product of operators. From (3.3) and (3.7),

$$\begin{aligned}
x^{n+1} - u^{n+1} &= (I_J - R_J A_J)(I_J - Q_{J-1}^J B_{J-1} A_{J-1} S_J^{J-1})(I_J - R_J A_J)(x^{n+1} - u^n) \\
&= T_J(I_J - Q_{J-1}^J R_{J-1} A_{J-1} S_J^{J-1})T_J(x^{n+1} - u^n) \\
&= T_J T_{J-1}(I_J - Q_{J-2}^J R_{J-2} A_{J-2} S_J^{J-2})T_{J-1}T_J(x^{n+1} - u^n) \\
&= T_J T_{J-1} \cdots T_2 T_1 T_2 \cdots T_{J-1} T_J(x^{n+1} - u^n) \\
&= K(x^{n+1} - u^n),
\end{aligned}$$

where $K = T_J T_{J-1} \cdots T_2 T_1 T_2 \cdots T_{J-1} T_J$.

Since $T_J = I_J - R_J A_J$ and since we have shown that $\|T_J\| < 1$ in Lemma 3.1, the convergence will be obtained if we can show $\|T_k\| = \|I_J - Q_k^J R_k A_k S_J^k\| \leq 1$.

Lemma 3.3 (a) For $1 \leq k \leq J$, $Q_k^J S_J^k$ is symmetric with respect to $A(\cdot, \cdot)$.

(b) The eigenvalues of $Q_k^J S_J^k$ are 0's and 1's, for $1 \leq k \leq J$.

(c) For $u, v \in \Omega_k$, $A(u, v) = A(Q_k^J u, Q_k^J v)$.

(d) If $\|I_k - R_k A_k\| < 1$, then $\|I_J - Q_k^J R_k A_k S_J^k\| \leq 1$.

Proof: The proofs of (a) and (b) are similar to those of Lemma 2.3, parts (b) and (c).

To prove part (c), let $u, v \in \Omega_k$. Then

$$\begin{aligned}
A(Q_k^J u, Q_k^J v) &= (Q_k^J u, A_J Q_k^J v)_J \\
&= (u, P_J^k A_J Q_k^J v)_k \\
&= (u, A_k v)_k \\
&= A(u, v).
\end{aligned}$$

For part (d), let $u \in \Omega_J$. Then

$$\begin{aligned}
0 &\leq A((I_J - Q_k^J R_k A_k S_J^k)u, (I_J - Q_k^J R_k A_k S_J^k)u) \\
&= A(u, u) - A(u, Q_k^J R_k A_k S_J^k u) - A(Q_k^J R_k A_k S_J^k u, u)
\end{aligned}$$

$$\begin{aligned}
& +A(Q_k^J R_k A_k S_j^k u, Q_k^J R_k A_k S_j^k u) \\
= & A(u, u) - (A_J u, Q_k^J R_k A_k S_j^k u)_J - (Q_k^J R_k A_k S_j^k u, A_J u)_J \\
& +A(R_k A_k S_j^k u, R_k A_k S_j^k u) \\
= & A(u, u) - (P_j^k A_J u, R_k A_k S_j^k u)_k - (R_k A_k S_j^k u, P_j^k A_J u)_k \\
& +A(R_k A_k S_j^k u, R_k A_k S_j^k u) \\
= & A(u, u) - (A_k S_j^k u, R_k A_k S_j^k u)_k - (R_k A_k S_j^k u, A_k S_j^k u)_k \\
& +A(R_k A_k S_j^k u, R_k A_k S_j^k u) \\
= & A(u, u) - A(S_j^k u, R_k A_k S_j^k u) - A(R_k A_k S_j^k u, S_j^k u) \\
& +A(R_k A_k S_j^k u, R_k A_k S_j^k u) \\
= & A(u, u) - A(S_j^k u, S_j^k u) + A(S_j^k u, S_j^k u) \\
& -A(S_j^k u, R_k A_k S_j^k u) - A(R_k A_k S_j^k u, S_j^k u) + A(R_k A_k S_j^k u, R_k A_k S_j^k u) \\
= & A(u, u) - A(S_j^k u, S_j^k u) + A((I_k - R_k A_k) S_j^k u, (I_k - R_k A_k) S_j^k u) \\
\leq & A(u, u) - A(S_j^k u, S_j^k u) + A(S_j^k u, S_j^k u) \\
= & A(u, u).
\end{aligned}$$

Since $0 \leq A((I_J - Q_k^J R_k A_k S_j^k)u, (I_J - Q_k^J R_k A_k S_j^k)u) \leq A(u, u)$,
 $|||(I_J - Q_k^J R_k A_k S_j^k)||| \leq 1$. \square

Theorem 3.1 *Let $e_0 = x^{n+1} - u^n$ be the initial error and $e_1 = x^{n+1} - u^{n+1}$ be the error after the V-cycle algorithm. Then $e_1 = K e_0$ with $|||K||| < 1$. Therefore the V-cycle scheme is convergent.*

Proof: Since $|||I - R_k A_k||| < 1$ for $1 \leq k \leq J$, $|||T_k||| = |||I_J - Q_k^J R_k A_k S_j^k||| \leq 1$ by Lemma 3.3 (d). We also have $|||T_J||| = |||I - R_J A_J||| < 1$ and $|||T_1||| = \rho(T_1) = \rho(I - Q_1^J S_1^J) = 1$. Thus we have $|||K||| < 1$. The V-cycle scheme is convergent. \square

CHAPTER 4

Stability and Convergence

Consider the linear systems (1.4) and (1.5). For each time step n , we solve each of (1.4) and (1.5) with m V-cycles. Let $|||u|||_A = \sqrt{A(u, u)}$ and $|||u|||_{A'} = \sqrt{A'(u, u)}$. For a V-cycle applied to (1.4) and (1.5), we have the convergent rates $|||K|||_A < 1$ and $|||K'|||_{A'} < 1$ respectively. Consider $w = \begin{pmatrix} u \\ v \end{pmatrix}$ on the vector space $\Omega \times \Omega$ with $u \in \Omega$ and $v \in \Omega$. We can define a norm for $w \in \Omega \times \Omega$ by $|||w||| = |||u|||_A + |||v|||_{A'}$. It can be verified that $|||\cdot|||$ satisfies the following properties:

$$\begin{aligned} |||w||| &> 0 && \text{if } w \neq 0, \\ |||w||| &= 0 && \text{if } w = 0, \\ |||cw||| &= |c| \cdot |||w||| && \text{for any complex number } c, \\ |||w_1 + w_2||| &\leq |||w_1||| + |||w_2|||. \end{aligned}$$

To show the stability and convergence of the scheme we need the following lemma.

Lemma 4.1 Let $B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}$, where B_1 and B_2 have the same size as that of A . Then $|||B||| = \max \left\{ |||B_1|||_A, |||B_2|||_{A'} \right\}$.

Proof: Note that

$$|||B||| = \max_{w \neq 0} \frac{|||Bw|||}{|||w|||} = \max_{w \neq 0} \frac{|||B_1u|||_A + |||B_2v|||_{A'}}{|||u|||_A + |||v|||_{A'}}.$$

But

$$\begin{aligned} \frac{|||B_1 u|||_A + |||B_2 v|||_{A'}}{|||u|||_A + |||v|||_{A'}} &\leq \max \left\{ \frac{|||B_1 u|||_A}{|||u|||_A}, \frac{|||B_2 v|||_{A'}}{|||v|||_{A'}} \right\} \\ &\leq \max \left\{ |||B_1|||_A, |||B_2|||_{A'} \right\}. \end{aligned}$$

Therefore

$$|||B||| \leq \max \left\{ |||B_1|||_A, |||B_2|||_{A'} \right\}.$$

□

We want to study the stability of the fully implicit discretization with V-cycle scheme as a solver. The following theorem shows that the scheme is unconditionally stable. Then any instability exhibited in computation should come from the original reaction-diffusion system.

Theorem 4.1 *Let u^n and v^n be the numerical solution of (1.4) and (1.5) with the initial values $u_{ij}^0 = u_0(x_{ij})$ and $v_{ij}^0 = v_0(x_{ij})$. Let \tilde{u}^n and \tilde{v}^n be the numerical solution of (1.4) and (1.5) with the initial values \tilde{u}_{ij}^0 and \tilde{v}_{ij}^0 . Let $w = \begin{pmatrix} u \\ v \end{pmatrix}$ and $\tilde{w} = \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix}$. If the partial derivatives f_u , f_v , g_u , and g_v are continuous and uniformly bounded, then*

$$|||w^n - \tilde{w}^n||| \leq (1 + C\Delta t)^n |||w^0 - \tilde{w}^0|||,$$

where $C > 0$ is a constant independent of h and Δt .

Remark: The general definition of stability for finite difference schemes can be found in [51].

Proof: Let K and K' be the error operator for a V-cycle (see Chapter 3) performed on (1.4) and (1.5) respectively. For each n , let x^n , \tilde{x}^n , y^n , and \tilde{y}^n satisfy

$$Ax^n = u^{n-1} + \Delta t f^{n-1},$$

$$A\tilde{x}^n = \tilde{u}^{n-1} + \Delta t \tilde{f}^{n-1},$$

$$A'y^n = v^{n-1} + \Delta t g^{n-1},$$

$$A'\tilde{y}^n = \tilde{v}^{n-1} + \Delta t \tilde{g}^{n-1},$$

and u^n , \tilde{u}^n , v^n , and \tilde{v}^n be the numerical solutions after m V-cycles are applied to the above linear systems respectively. So

$$x^n - u^n = K^m(x^n - u^{n-1}),$$

$$\tilde{x}^n - \tilde{u}^n = K^m(\tilde{x}^n - \tilde{u}^{n-1}),$$

$$y^n - v^n = (K')^m(y^n - v^{n-1}),$$

$$\tilde{y}^n - \tilde{v}^n = (K')^m(\tilde{y}^n - \tilde{v}^{n-1}).$$

In Chapter 3, we have shown that $\|K\|_A < 1$. So $\|K'\|_{A'} < 1$ as well.

Since

$$x^n = A^{-1}u^{n-1} + \Delta t A^{-1}f^{n-1}$$

and

$$\tilde{x}^n = A^{-1}\tilde{u}^{n-1} + \Delta t A^{-1}\tilde{f}^{n-1},$$

we have

$$\begin{aligned} (\tilde{x}^n - x^n) + (u^n - \tilde{u}^n) &= K^m(\tilde{x}^n - x^n + u^{n-1} - \tilde{u}^{n-1}), \\ u^n - \tilde{u}^n &= (I - K^m)(x^n - \tilde{x}^n) + K^m(u^{n-1} - \tilde{u}^{n-1}), \\ u^n - \tilde{u}^n &= (I - K^m)A^{-1}(u^{n-1} - \tilde{u}^{n-1}) + \Delta t(I - K^m)A^{-1}(f^{n-1} - \tilde{f}^{n-1}) \\ &\quad + K^m(u^n - \tilde{u}^{n-1}), \\ u^n - \tilde{u}^n &= (A^{-1} + K^m(I - A^{-1}))(u^{n-1} - \tilde{u}^{n-1}) \\ &\quad + \Delta t(I - K^m)A^{-1}(f^{n-1} - \tilde{f}^{n-1}). \end{aligned} \tag{4.1}$$

Since

$$f(u_{ij}^n, v_{ij}^n) - f(\tilde{u}_{ij}^n, \tilde{v}_{ij}^n) = f_u(\zeta_{ij}^n, \eta_{ij}^n)(u_{ij}^n - \tilde{u}_{ij}^n) + f_v(\zeta_{ij}^n, \eta_{ij}^n)(v_{ij}^n - \tilde{v}_{ij}^n),$$

where $(\zeta_{ij}^n, \eta_{ij}^n)$ is between (u_{ij}^n, v_{ij}^n) and $(\tilde{u}_{ij}^n, \tilde{v}_{ij}^n)$, (4.1) can be expressed as

$$\begin{aligned} u^n - \tilde{u}^n &= (A^{-1} + K^m(I - A^{-1}))(u^{n-1} - \tilde{u}^{n-1}) + \Delta t(I - K^m)A^{-1}B_{11}^n(u^{n-1} - \tilde{u}^{n-1}) \\ &\quad + \Delta t(I - K^m)A^{-1}B_{12}^n(v^{n-1} - \tilde{v}^{n-1}), \end{aligned} \quad (4.2)$$

where B_{11}^n and B_{12}^n are diagonal matrices depending on the values of f_u and f_v .

Similarly,

$$\begin{aligned} v^n - \tilde{v}^n &= ((A')^{-1} + (K')^m(I - (A')^{-1}))(v^{n-1} - \tilde{v}^{n-1}) \\ &\quad + \Delta t(I - (K')^m)(A')^{-1}B_{21}^n(u^{n-1} - \tilde{u}^{n-1}) \\ &\quad + \Delta t(I - (K')^m)(A')^{-1}B_{22}^n(v^{n-1} - \tilde{v}^{n-1}), \end{aligned} \quad (4.3)$$

where B_{21}^n and B_{22}^n are diagonal matrices depending on the values of g_u and g_v .

Combining (4.2) and (4.3), we have

$$\begin{aligned} w^n - \tilde{w}^n &= \begin{pmatrix} A^{-1} + K^m(I - A^{-1}) & 0 \\ 0 & (A')^{-1} + (K')^m(I - (A')^{-1}) \end{pmatrix} (w^{n-1} - \tilde{w}^{n-1}) \\ &\quad + \Delta t \begin{pmatrix} (I - K^m)A^{-1}B_{11}^n & (I - K^m)A^{-1}B_{12}^n \\ (I - (K')^m)(A')^{-1}B_{21}^n & (I - (K')^m)(A')^{-1}B_{22}^n \end{pmatrix} (w^{n-1} - \tilde{w}^{n-1}). \end{aligned}$$

Because of the properties of $f_u, f_v, g_u,$ and $g_v,$

$$\left\| \left\| \begin{pmatrix} (I - K^m)A^{-1}B_{11}^n & (I - K^m)A^{-1}B_{12}^n \\ (I - (K')^m)(A')^{-1}B_{21}^n & (I - (K')^m)(A')^{-1}B_{22}^n \end{pmatrix} \right\| \right\| \leq C',$$

where $C' \geq 0$ is a constant independent of h and Δt . But A^{-1} is symmetric with respect to $\|\cdot\|_A$ and $\|A^{-1}\|_A \leq 1$, and moreover $\|K\|_A < 1$ and $\|K'\|_{A'} < 1$.

Thus if m is large enough, $(\|K\|_A)^m < C'\Delta t$ and $(\|K'\|_{A'})^m < C'\Delta t$. Thus

$$\begin{aligned} \|A^{-1} + K^m(I - A^{-1})\|_A &\leq \|A^{-1}\|_A + \|K^m(I - A^{-1})\|_A \\ &\leq \|A^{-1}\|_A + (\|K\|_A)^m \|I - A^{-1}\|_A \\ &\leq 1 + C'\Delta t. \end{aligned}$$

Similarly,

$$\| (A')^{-1} + K^m(I - (A')^{-1}) \|_{A'} \leq 1 + C' \Delta t.$$

Using lemma 4.1, we have

$$\left\| \left\| \begin{pmatrix} A^{-1} + K^m(I - A^{-1}) & 0 \\ 0 & (A')^{-1} + K^m(I - (A')^{-1}) \end{pmatrix} \right\| \right\| \leq 1 + C' \Delta t.$$

Set $C = 2C'$. Then

$$\begin{aligned} \| w^n - \tilde{w}^n \| &\leq (1 + C' \Delta t) \| w^{n-1} - \tilde{w}^{n-1} \| + C' \Delta t \| w^{n-1} - \tilde{w}^{n-1} \| \\ &= (1 + C \Delta t) \| w^{n-1} - \tilde{w}^{n-1} \| \\ &\leq (1 + C \Delta t)^n \| w^0 - \tilde{w}^0 \|. \end{aligned}$$

□

There is the well known Lax's equivalence theorem that guarantees that for a consistent finite difference scheme, stability is equivalent to convergence [51]. This theorem is only valid for linear equations. So, for our problem, we have to investigate the convergence of the scheme separately.

Theorem 4.2 *Assume that a solution (\tilde{u}, \tilde{v}) of (0.1) is of class C^4 , and let (u, v) denote the numerical solution for (0.1). Define the errors $E_u^n = \tilde{u}^n - u^n$ and $E_v^n = \tilde{v}^n - v^n$. Then for $T > 0$ fixed and any n such that $n\Delta t \leq T$, we have*

$$\| E_u^{n+1} \|_A \leq \| E_u^0 \|_A + \Delta t \sum_{j=0}^n (\| \tilde{f}^j - f^j \|_A + \epsilon^{j+1})$$

and

$$\| E_v^{n+1} \|_{A'} \leq \| E_v^0 \|_{A'} + \Delta t \sum_{j=0}^n (\| \tilde{g}^j - g^j \|_{A'} + \epsilon^{j+1}),$$

where $\epsilon^j = O(\Delta t + h^{3/2})$.

Proof: Again, we only establish the estimates for u , since the estimates for v follow similarly. Lemma 1.1 shows that

$$A\tilde{u}^{n+1} = \tilde{u}^n + \Delta t \tilde{f}^n + \Delta t R^{n+1},$$

where R^{n+1} is a vector whose components are r_{ij}^{n+1} as in Lemma 1.1. Thus

$$\tilde{u}^{n+1} = A^{-1}\tilde{u}^n + \Delta t A^{-1}\tilde{f}^n + \Delta t A^{-1}R^{n+1}.$$

Set $Ax^{n+1} = u^n + \Delta t f^n$. Then $x^{n+1} = A^{-1}u^n + \Delta t A^{-1}f^n$. Let K be the error operator defined in Chapter 3 and assume m V-cycles are performed at each time step. Then $x^{n+1} - u^{n+1} = K^m(x^{n+1} - u^n)$. Moreover

$$\begin{aligned} u^{n+1} &= (I - K^m)x^{n+1} + K^m u^n \\ &= (I - K^m)(A^{-1}u^n + \Delta t A^{-1}f^n) + K^m u^n \\ &= K^m(I - A^{-1})u^n + A^{-1}u^n + (I - K^m)\Delta t A^{-1}f^n \end{aligned}$$

and

$$\begin{aligned} \tilde{u}^{n+1} - u^{n+1} &= A^{-1}(\tilde{u}^n - u^n) - K^m(I - A^{-1})u^n + \Delta t A^{-1}(\tilde{f}^n - f^n) + K^m \Delta t A^{-1}f^n \\ &\quad + \Delta t A^{-1}R^{n+1} \\ &= A^{-1}(\tilde{u}^n - u^n) + K^m[\Delta t A^{-1}f^n - (I - A^{-1})u^n] + \Delta t A^{-1}(\tilde{f}^n - f^n) \\ &\quad + \Delta t A^{-1}R^{n+1}. \end{aligned}$$

Recall that $\|A^{-1}\|_A \leq 1$, $\|I - A^{-1}\|_A \leq 1$, and $\|K\|_A < 1$. Thus we may assume $(\|K\|_A)^m \leq (\Delta t)^2$. But then

$$\begin{aligned} \|\tilde{u}^{n+1} - u^{n+1}\|_A &\leq \|\tilde{u}^n - u^n\|_A + \Delta t \|\tilde{f}^n - f^n\|_A + (\|K\|_A)^m \|u^n\|_A \\ &\quad + (\|K\|_A)^m \Delta t \|f^n\|_A + \Delta t \|R^{n+1}\|_A \\ &\leq \|\tilde{u}^n - u^n\|_A + \Delta t \|\tilde{f}^n - f^n\|_A + (\Delta t)^2 \|u^n\|_A + (\Delta t)^3 \|f^n\|_A \\ &\quad + \Delta t \|R^{n+1}\|_A \\ &= \|\tilde{u}^n - u^n\|_A + \Delta t (\|\tilde{f}^n - f^n\|_A + \Delta t \|u^n\|_A + \epsilon^{n+1}), \quad (4.4) \end{aligned}$$

where $\epsilon^{n+1} = (\Delta t)^2 |||f^n|||_A + |||R^{n+1}|||_A + \Delta t |||u^n|||_A$. We have

$$\begin{aligned} |||R^{n+1}|||_A &= \max_{u \neq 0} \frac{A(R^{n+1}u, R^{n+1}u)}{A(u, u)} \\ &= \max_{u \neq 0} \frac{(AR^{n+1}u, R^{n+1}u)_J}{(Au, u)_J} \\ &= \max_{u \neq 0} \frac{(AR^{n+1}u, R^{n+1}u)_J (R^{n+1}u, R^{n+1}u)_J}{(R^{n+1}u, R^{n+1}u)_J (Au, u)_J}. \end{aligned}$$

Since $A = A_J$ are SPD with respect to $(\cdot, \cdot)_J$, $(Au, u)_J \geq \lambda_{\min}(A)(u, u)_J$ and

$$\frac{(AR^{n+1}u, R^{n+1}u)_J}{(R^{n+1}u, R^{n+1}u)_J} \leq \|A\|_J.$$

Therefore $|||R^{n+1}|||_A \leq \sqrt{\|A\|_J} \|R^{n+1}\|_J$ and

$$\|R^n\|_J^2 = \sum_{x_{ij} \in \Omega} h^2 (r_{ij}^n)^2 + \sum_{x_{ij} \in \partial\Omega} \gamma_i \gamma_j h^2 (r_{ij}^n)^2 = \left(O(\Delta t + h^{3/2})\right)^2,$$

since there are only $O(1/h)$ boundary points where $r_{ij}^n = O(\Delta t + h)$. Thus $\epsilon^{n+1} = \Delta t |||f^n|||_A + |||R^{n+1}|||_A + \Delta t |||u^n|||_A = O(\Delta t + h^{3/2})$. This together with (4.4) gives

$$|||\tilde{u}^{n+1} - u^{n+1}|||_A \leq |||\tilde{u}^0 - u^0|||_A + \Delta t \sum_{j=0}^n (|||\tilde{f}^j - f^j|||_A + \epsilon^{j+1})$$

and

$$|||E_u^{n+1}|||_A \leq |||E_u^0|||_A + \Delta t \sum_{j=0}^n (|||\tilde{f}^j - f^j|||_A + \epsilon^{j+1}).$$

□

Corollary 4.1 Let $E^n = \begin{pmatrix} E_u^n \\ E_v^n \end{pmatrix}$ with E_u^n and E_v^n defined in Theorem 4.2. If the partial derivatives f_u , f_v , g_u , and g_v are continuous and uniformly bounded, then for $T > 0$ such that $n\Delta t = T$, we have

$$|||E^n||| \leq e^{CT} (|||E^0||| + e^n),$$

where $e^n = TO(\Delta t + h^{3/2})$.

Proof: From the properties of f_u , f_v , g_u , and g_v and the proof in Theorem 4.1, we obtain

$$\begin{pmatrix} \tilde{f}^j - f^j \\ \tilde{g}^j - g^j \end{pmatrix} = \begin{bmatrix} B_{11}^j & B_{12}^j \\ B_{21}^j & B_{22}^j \end{bmatrix} (\tilde{w}^j - w^j),$$

with

$$\begin{bmatrix} B_{11}^j & B_{12}^j \\ B_{21}^j & B_{22}^j \end{bmatrix} \leq C.$$

Here B_{11} , B_{12} , B_{21} , and B_{22} are defined in Theorem 4.1 and the constant C is independent of h and Δt . Therefore

$$\left\| \begin{pmatrix} \tilde{f}^j - f^j \\ \tilde{g}^j - g^j \end{pmatrix} \right\| \leq \|\tilde{w}^j - w^j\| = C \|E^j\|$$

and

$$\|\tilde{f}^j - f^j\|_A + \|\tilde{g}^j - g^j\|_{A'} \leq C \|E^j\|.$$

Hence

$$\|E^n\| \leq \|E^0\| + C\Delta t \sum_{j=0}^{n-1} \|E^j\| + \Delta t \sum_{j=0}^{n-1} e^{j+1}.$$

Let $e^n = \Delta t \sum_{j=1}^n e^j = TO(\Delta t + h^{3/2})$. Then $e^j \leq e^n$ for $0 \leq j \leq n$ and

$$\begin{aligned} \|E^n\| &\leq \|E^0\| + e^n + C\Delta t \sum_{j=0}^{n-1} \|E^j\| \\ &= \|E^0\| + e^n + C\Delta t \|E^{n-1}\| + C\Delta t \sum_{j=0}^{n-2} \|E^j\| \\ &\leq \|E^0\| + e^n + C\Delta t (\|E^0\| + e^{n-1} + C\Delta t \sum_{j=0}^{n-2} \|E^j\|) + C\Delta t \sum_{j=0}^{n-2} \|E^j\| \\ &\leq \|E^0\| + e^n + C\Delta t (\|E^0\| + e^n + C\Delta t \sum_{j=0}^{n-2} \|E^j\|) + C\Delta t \sum_{j=0}^{n-2} \|E^j\| \\ &= (1 + C\Delta t)(\|E^0\| + e^n) + (1 + C\Delta t)(C\Delta t \sum_{j=0}^{n-2} \|E^j\|) \\ &= (1 + C\Delta t)(\|E^0\| + e^n + \sum_{j=0}^{n-2} \|E^j\|) \\ &\leq (1 + C\Delta t)^{n-1} (\|E^0\| + e^n + C\Delta t \|E^0\|) \end{aligned}$$

$$\begin{aligned}
&\leq (1 + C\Delta t)^n(\|E^0\| + e^n) \\
&\leq \exp^{Cn\Delta t}(\|E^0\| + e^n) \\
&= \exp^{CT}(\|E^0\| + e^n).
\end{aligned}$$

□

Remark: The number m of V-cycles applied to (2.1) is assumed to be large in the proof above in order to obtain the stability and convergence theory. However, in the practical computations, one or two V-cycles performed at each time step are sufficient for the desired accuracy. This efficient scheme will be demonstrated in the next chapter.

CHAPTER 5

Experimental Results

In this chapter, two reaction-diffusion systems are used to demonstrate the efficiency of the fully implicit finite difference discretization with V-cycle as a solver. The first example is a substrate-inhibition reaction-diffusion mechanism experimentally studied by Thomas(1976). It has been shown to be able to form patterns on animal coats [42]. First, we do a linear stability analysis on a rectangular domain to predict the possible pattern on this domain. Then a numerical simulation is carried out to verify this estimated pattern. Finally, we use this system to generate spots and stripes by numerical simulations. To demonstrate that this numerical scheme is valid for other shapes of domains in practice, the pattern on a cone surface is also carried out.

There are many other systems which has been used to study spatial patterns [3] [26] [42]. All these systems are capable of generating animal coat markings. The second example we use is the Schnakenberg(1979) reaction [41]. It is used to generate spots, horizontal and vertical stripes in our numerical computations.

For each numerical simulation, the computer program terminates when the numerical solutions u and v reach their equilibrium. In the simulation of the pattern on a tail, it requires two V-cycles at each time step for the numerical solution to converge to the equilibrium. Each of other simulations requires only one V-cycle at each

time step for the numerical solution to converge. Thus at the end of this chapter, we compare the running time needed in using the multigrid technique with that needed in using a classical iterative method on the fine grid alone. The results show that multigrid technique is much more efficient.

5.1 The Thomas System

The example we consider here is taken from [41], the Thomas system. It is a substrate inhibition system:

$$\begin{cases} U_t = D_u \Delta U + F(U, V) \\ V_t = D_v \Delta V + G(U, V), \end{cases}$$

where

$$F(U, V) = k_1 - k_2 U - \frac{k_5 UV}{k_6 + k_7 U + k_8 U^2}$$

and

$$G(U, V) = k_3 - k_4 V - \frac{k_5 UV}{k_6 + k_7 U + k_8 U^2}.$$

Here U and V are the concentrations of two chemical species with U a substrate and V an inhibitor. The constants D_u and D_v are positive diffusion coefficients and $F(U, V)$ and $G(U, V)$ are the kinetics with the positive rate constants k_s . The nondimensional system is given by

$$\begin{cases} u_t = \Delta u + \gamma f(u, v) \\ v_t = d \Delta v + \gamma g(u, v), \end{cases}$$

where

$$f(u, v) = a - u - \frac{\rho uv}{1+u+Ku^2}$$

and

$$g(u, v) = \alpha(b - v) - \frac{\rho uv}{1+u+Ku^2},$$

with zero flux boundary condition $\frac{\partial u}{\partial n} \Big|_{\partial \Omega} = \frac{\partial v}{\partial n} \Big|_{\partial \Omega} = 0$ and initial condition $u(x, y, 0)$ and $v(x, y, 0)$ given on Ω .

Let us choose Ω as $[0, 2.1] \times [0, 1.2]$ and parameter values $d = 10$, $\gamma = 20$, $\alpha = 1.5$, $K = 0.1$, $\rho = 18.5$, $a = 92$, and $b = 64$. The parameter values determine a steady state $u_s = 9.934$ and $v_s = 9.289$. To form inhomogeneous spatial patterns, a reaction diffusion system must undergo diffusion driven instability or Turing instability. The uniform steady state is stable to small disturbance when the diffusion terms are absent but unstable to small spatial disturbance in the presence of diffusion. With the above parameter values, linear theory gives the range of modes k^2 that are driven unstable:

$$10.5 \leq k^2 = \pi^2 \left(\frac{m^2}{2.1^2} + \frac{n^2}{1.2^2} \right) \leq 10.25,$$

which means

$$1.0637 \leq \frac{m^2}{2.1^2} + \frac{n^2}{1.2^2} \leq 1.9507,$$

where m and n are integers. Details in deriving the range of unstable modes can be found in [41]. The above range of unstable wavenumbers admits only the wavenumber $m = 2$, $n = 1$. The solution which involves exponentially growing modes about the uniform steady state u_s and v_s is given as

$$C_{2,1} e^{\lambda(k^2)t} \cos \frac{2\pi x}{2.1} \cos \frac{\pi y}{1.2}, \quad (5.2)$$

where $C_{2,1}$ is determined by initial conditions. Figure 5.1 is the pattern obtained from (5.2) to predict the pattern that could be formed. The values on the dark regions are greater than zero. The top graph is obtained with $C_{2,1} < 0$ and the bottom with $C_{2,1} > 0$. Which of these two solutions is obtained thus depends on the bias in the initial conditions.

Turing(1952) suggested that, under certain conditions, the homogeneous steady state was unstable to small spatial perturbations and the stable nonuniform spatial patterns could evolve by diffusion driven instability. In the numerical experiments, the initial conditions are taken as random perturbations confined to a small region about the steady state. Figure 5.2 shows the concentrations in the morphogen u .

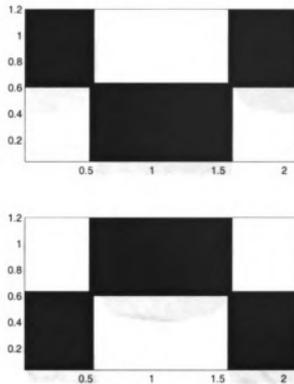


Figure 5.1: Possible patterns estimated by linear stability analysis.

The dark regions represent $u < u_s$. There are two patterns formed by using different seeds in the function calls of random number generator. They agree with the patterns suggested by linear stability analysis.

The next three figures are obtained by numerical simulations of different patterns generated with different parameter values of γ and different geometry. Each example is computed until the equilibrium is reached.

The computing details of the simulation are as follows. In Figure 5.3, the pattern (top) of morphogen concentration $u < u_0 = 10$ in system (5.1) is computed on a grid of 65×257 points with $\gamma = 8$. The graph of the residual (bottom) shows that morphogen u reaches the equilibrium at $t \approx 400$.

In Figure 5.4, the pattern of morphogen concentration $u < u_0 = 10$ in system (5.1) is computed on a grid of 65×225 points with $\gamma = 48$. The graph of the residual shows that morphogen u reaches the equilibrium at $t \approx 60$.

In Figure 5.5, the pattern of morphogen concentration $u < u_0 = 10$ in system

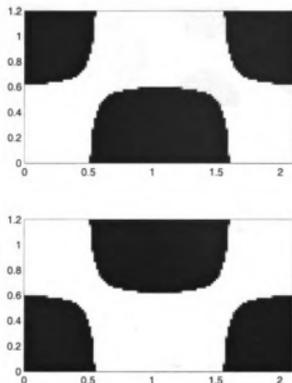


Figure 5.2: Turing's patterns obtained by numerical simulation.

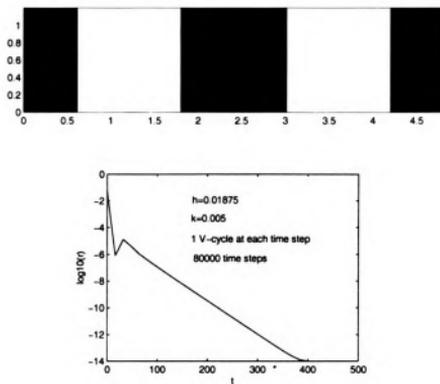


Figure 5.3: Numerical simulation of Turing pattern of (5.1) and residual plot with $\gamma = 8$.

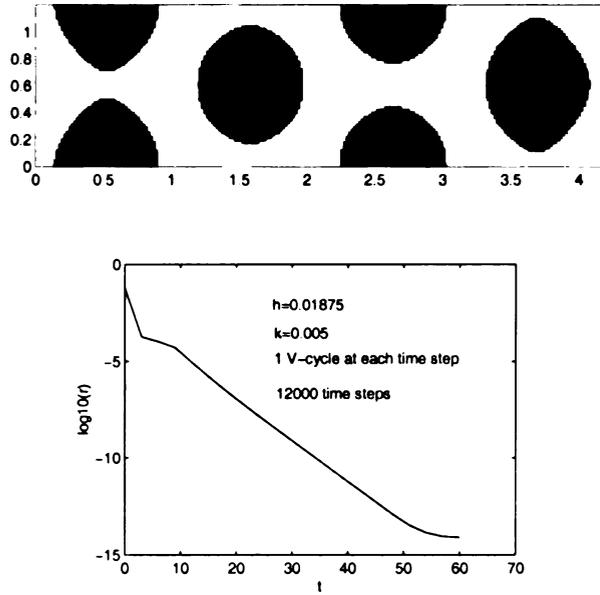


Figure 5.4: Numerical simulation of Turing pattern of (5.1) and residual plot with $\gamma = 48$.

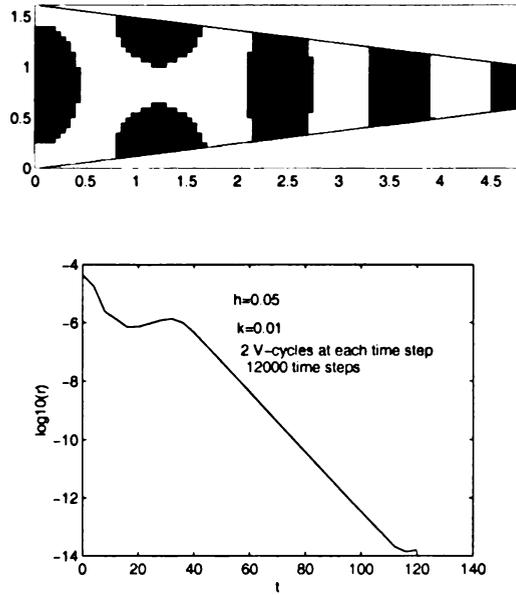


Figure 5.5: Numerical simulation of Turing pattern of (5.1) and residual plot on a cone surface.

(5.1) is computed on a grid of 65×225 points with $\gamma = 35$. The graph of the residual shows that morphogen u reaches the equilibrium at $t \approx 120$.

Rings of pattern typical of many spotted animal tails are generated when the reaction diffusion domains are tapering cylinders. Rings are at the tip and spots are obtained as the circumference increases. Although the domain is not rectangular, V-cycle algorithm could still be applied analogously.

5.2 The Schnakenberg System

The main forms of patterns generated by a reaction-diffusion system are spots (Figure 5.4 and Figure 5.7), rings (Figure 5.5), horizontal stripes (Figure 5.8), and vertical stripes (Figure 5.3 and Figure 5.6). In this section, we choose another reaction kinetics. It is the simplest class of two-species reaction mechanism studied by Schnakenberg(1979) in limit cycle solutions of two-species reaction systems. It has also been shown to be able to generate animal coat markings [41]. Its nondimensionalization gives

$$\begin{cases} u_t = \Delta u + \gamma f(u, v) \\ v_t = d\Delta v + \gamma g(u, v), \end{cases}$$

where

$$f(u, v) = a - u + u^2v$$

and

$$g(u, v) = b - u^2v,$$

(5.3)

with parameter values $d = 50$, $a = 0.2$, and $b = 2$. Thus the uniform steady state is $u_0 = 2.2$ and $v_0 = 0.413$.

Next we demonstrate the V-cycle algorithm with this reaction-diffusion system. We change the scale factor γ and the domain to obtain different forms of pattern. The computing details follow.

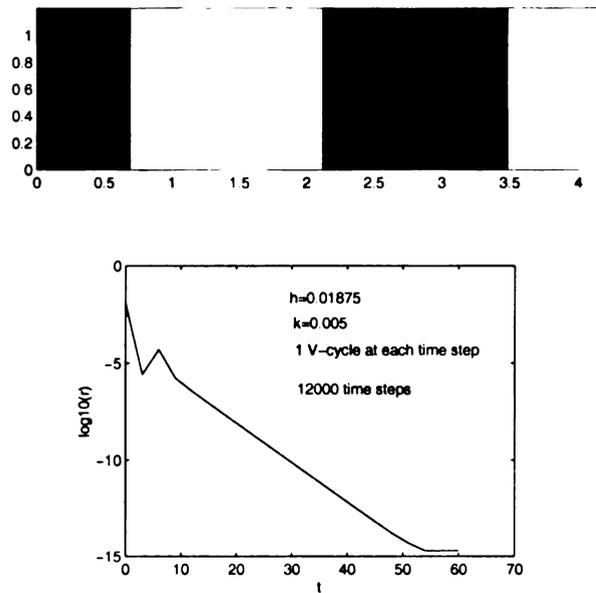


Figure 5.6: Numerical simulation of Turing pattern of (5.3) and residual plot of with $\gamma = 15$.

In Figure 5.6, the pattern of morphogen concentration $u < u_0 = 2.2$ in system (5.3) is computed on a grid of 65×225 points with $\gamma = 15$. The graph of the residual shows that morphogen u reaches the equilibrium at $t \approx 54$.

In Figure 5.7, the pattern of morphogen concentration $u < u_0 = 2.2$ in system (5.3) is computed on a grid of 65×217 points with $\gamma = 30$. The graph of the residual shows that morphogen u reaches the equilibrium at $t \approx 100$.

In Figure 5.8, the pattern (top) of morphogen concentration $u < u_0 = 2.2$ in system (5.3) is computed on a grid of 65×225 points with $\gamma = 50$. The graph of the residual (bottom) also shows that morphogen u reaches the equilibrium at $t \approx 26$.

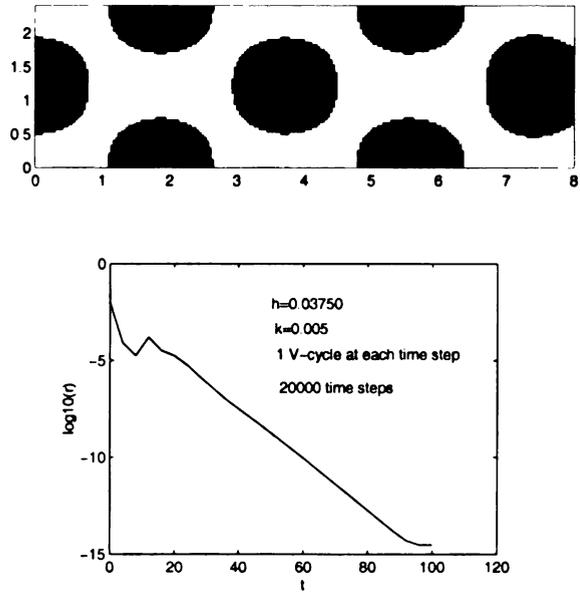


Figure 5.7: Numerical simulation of Turing pattern of (5.3) and residual plot with $\gamma = 30$.

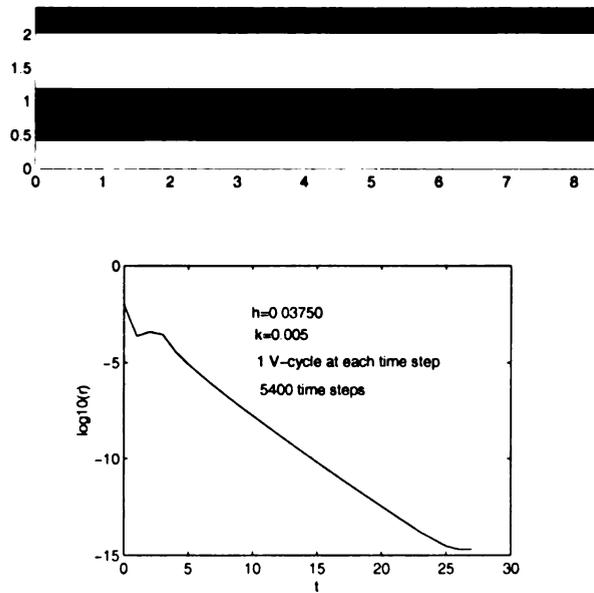


Figure 5.8: Numerical simulation of Turing pattern of (5.3) and residual plot with $\gamma = 50$.

5.3 Comparison

We end this thesis by demonstrating the robustness of the V-cycle algorithm. We compute the solution of system (5.1) with $\gamma = 25$ on the domain $[0, 1.6] \times [0, 1.6]$. We discretize (5.1) with different spatial and time step sizes. Both the V-cycle algorithm and direct iterative method using red-black Gauss-Seidel iteration are applied to each case. In each case, the solution is obtained when the morphogen concentrations u and v reach their equilibrium. The number of iterations at each step, the number of time steps needed for the morphogen concentration to reach its inhomogeneous steady state, and the running time for each case are recorded in the following tables. Notice that when the spatial step size in each direction is refined by half (see Table 1 and Table 2) in order to keep the same ratio of $\frac{\Delta t}{h^2}$, time step size is changed into a factor of $\frac{1}{4}$. Comparing the running time between these two cases it does not show the power of the V-cycle algorithm. Since the fully implicit finite difference discretization is unconditionally stable, we could exploit this advantage by increasing the step size in time to reduce the running time. Thus in Table 3, we double the time step size. Then only one V-cycle is needed, so the running time for V-cycle algorithm is reduced by about half. But the direct iterative method suffers from its limitations because of the change of the ratio $\frac{\Delta t}{h^2}$. Thus the number of iterations at each time step must be increased in order to have convergence. Even if the fully implicit finite difference scheme is unconditionally stable, the direct iterative method seems to be unable to take advantage of it. In Table 4, we can see when the spatial grid size is even finer – the running time for direct iterative method is almost 10 times that needed for V-cycle algorithm.

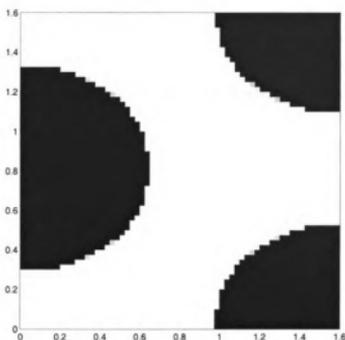


Figure 5.9: Numerical simulation of Turing pattern on a square.

| $h = 0.05$ $\Delta t = 0.01$ | number of iterations at each time step | number of time steps | CPU time |
|----------------------------------|---|-------------------------|---------------|
| Red-black Gauss-Seidel iteration | 25 | 3000 | 39.78 seconds |
| V-cycle | 1 | 3000 | 10.8 seconds |

Table 5.1: Comparison between single grid and multigrid with $h = 0.05$ and $\Delta t = 0.01$.

| $h = 0.025$ $\Delta t = 0.0025$ | number of iterations at each time step | number of time steps | CPU time |
|------------------------------------|---|-------------------------|----------------|
| Red-black Gauss-Seidel iteration | 22 | 12000 | 607.54 seconds |
| V-cycle | 1 | 15000 | 219.18 seconds |

Table 5.2: Comparison between single grid and multigrid with $h = 0.025$ and $\Delta t = 0.0025$.

| $h = 0.025$ $\Delta t = 0.005$ | number of iterations at each time step | number of time steps | CPU time |
|-----------------------------------|---|-------------------------|----------------|
| Red-black Gauss-Seidel iteration | 45 | 5600 | 569.86 seconds |
| V-cycle | 1 | 7000 | 103.43 seconds |

Table 5.3: Comparison between single grid and multigrid with $h = 0.025$ and $\Delta t = 0.005$.

| $h = 0.0125$ $\Delta t = 0.0025$ | number of iterations at each time step | number of time steps | CPU time |
|-------------------------------------|---|-------------------------|-----------------|
| Red-black Gauss-Seidel iteration | 85 | 11000 | 8205.23 seconds |
| V-cycle | 1 | 14000 | 857.52 seconds |

Table 5.4: Comparison between single grid and multigrid with $h = 0.0125$ and $\Delta t = 0.0025$.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley, New York, 1978.
- [2] R. E. Bank and C. C. Douglas, *Sharp estimates for multigrid rates of convergence with general smoothing and acceleration*, SIAM J. Numer. Anal., 22(1985), pp. 617-633.
- [3] J. Bard, *A model for generating aspects of zebra and other mammalian coat patterns*, J. Theoret. Biol., 93(1981), pp. 363-385.
- [4] D. Braess and W. Hackbusch, *A new convergence proof for the multigrid method including the V cycle*, SIAM J. Numer. Anal., 20(1983), pp. 967-975.
- [5] J. H. Bramble, *Multigrid methods*, Longman Scientific & Technical, 1993.
- [6] J. H. Bramble and J. E. Pasciak, *The analysis of smoother for multigrid algorithm*, Math. Comp., 58(1992), pp. 467-488.
- [7] J. H. Bramble J. E. Pasciak, *New convergence estimates for multigrid algorithms*, Math. Comput., 49(1987), pp. 311-329.
- [8] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for multigrid algorithms without regularity assumption*, Math. Comput., 57(1991), pp. 23-45.
- [9] J. H. Bramble, J. E. Pasciak, and J. Xu, *The analysis of multigrid algorithm for nonsymmetric and indefinite problems*, Math. Comput., 51(1988), pp. 389-414.
- [10] J. H. Bramble, J. E. Pasciak, and J. Xu, *Guide to multigrid development*, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, New York, 1982, pp. 220-312.
- [11] N. F. Britton, *Reaction-Diffusion Equations and Their Application to Biology*, Academic Press, New York, 1986.

- [12] R. Bulirsch and J. Stoer, *Introduction to Numerical Analysis*, Springer-Verlag, 1992
- ✓ [13] C. Chiu, F. C. Hoppensteadt, and W. Jäger, *Analysis and computer simulation of accretion patterns in patterns in bacterial culture*, J. of Math. Biology, (1994) 32, pp. 841-855.
- [14] C. Chiu and N. Walkington, *An ADI method for Reaction-diffusion systems and Hysteric Reaction-Diffusion Torpor-Systems*, SIAM J. Numer. Anal., (1997) 34, No.3, pp. 1185-1206.
- [15] C. Chiu and N. Walkington, *Analysis of hysteric reaction-diffusion systems*, Quarterly of Appl. Math., (1998) LVI, No.1, pp. 89-106.
- [16] C. C. Douglas, *Abstract multi-grid with application to elliptic boundary-value problems*, in Elliptic Problem Solvers II, G. Birkhoff and A. Schoenstadt, eds., Academic Press, New York, 1984, pp. 453-466.
- [17] C. C. Douglas, *Multi-grid with application to elliptic boundary-value problems*, SIAM J. Numer. Anal., 21 (1984), pp. 236-254.
- [18] C. C. Douglas and J. Douglas, Jr., *A unified convergence theory for abstract multigrid or multilevel algorithms, serial and parallel*, SIAM J. Numer. Anal., 30 (1993), pp. 136-158.
- ✓ [19] A. Gierer and H. Meinhardt, *A theory of biological pattern formation*, Kybernetik, 12(1972), pp. 30-39.
- [20] A. Greenbaum, *Analysis of a multigrid method as an iterative technique for solving linear systems*, SIAM J. Numer. Anal., 21 (1984), pp. 473-485.
- ✓ [21] D. Hoff *A Stability and Convergence of Finite Difference Method for Systems of Nonlinear Reaction-Diffusion Equations*, SIAM J. Numer. Anal., 12 (1978), pp. 1161-1177.
- [22] G. Horton and S. Vandewalle, *A space-time multigrid method for parabolic partial differential equations*, SIAM J. Sci. Comput., 16 (1995), pp. 848-864.
- ✓ [23] Y. Keshet and L. A. Segel, *Pattern formation in aspects*, Modeling of Patterns in Space and Time, W. Jager, ed., Springer-Verlag, Berlin, pp. 188-197.

- [24] C.C. Kuo and B.C. Levy, *Two-Color Fourier Analysis of the Multigrid Method with Red-Black Gauss-Seidel Smoothing*, Appl. Math. Comput., 29 (1989), pp. 69-87.
- [25] S. A. Levy, *Non-uniform stable solutions to reaction-diffusion equations: applications to ecological pattern formation*, Pattern Formation by Dynamic Systems and Pattern Recognition, H. Haken, ed., Springer-Verlag, Berlin, pp. 210-222.
- [26] S. A. Levin and L. A. Segel, *Pattern Generation in Space and Aspect*, SIAM Review, 27 (1985), pp. 45-67.
- [27] N. Li, J. Steiner and S. Tang, *Convergence and stability analysis of an explicit finite difference method for 2-dimensional reaction-diffusion equations*, J. Austral. Math. Soc. Ser. B, 35 (1993), pp. 223-243.
- [28] P. Lötstedt and B. Gustafsson, *Fourier analysis of multigrid methods for general systems of PDEs*, Math. Comp. 60 (1993) pp. 473-493.
- [29] J. F. Maitre and F. Musy, *Multigrid methods: convergence theory in a variational framework*, SIAM J. Numer. Anal., 21 (1984), pp. 657-671.
- [30] J. F. Maitre and F. Musy, *Multigrid methods for Symmetric Variational Problems: A General Theory and Convergence Estimates for Usual Smoothers*, Appl. Math. Comput, 21 (1987), pp. 21-43.
- [31] J. Mandel, *Multigrid convergence for nonsymmetric, indefinite variational problems and one smoothing step*, Appl. Math. Comput., 19 (1986), pp. 201-216.
- [32] J. Mandel, S. McCormick, and R. Bank, *Variational multigrid theory*, Multigrid Methods, (S. McCormick, ed.), SIAM (Society for industrial and Applied Mathematics), Philadelphia, PA, 1987, pp. 131-178.
- [33] J. Mandel, *Algebraic Study of Multigrid methods for symmetric, definite problems*, Appl. Math. Comput., 25 (1988), pp. 39-56.
- [34] J. Mandel, S. McCormick, and J. Ruge, *A algebraic theory for multigrid method for variational problem*, SIAM J. Numer. Anal., 25 (1988), pp. 91-110.
- [35] S. F. McCormick, *An algebraic interpretation of multigrid methods*, SIAM J. Numer. Anal., 19 (1982), pp. 548-560.
- [36] S. F. McCormick, *Multigrid methods for variational problems: further results*, SIAM J. Numer. Anal., 21 (1984), pp. 255-263.

- [37] S. F. McCormick, *Multigrid methods for variational problems: general theory for the V-cycle*, SIAM J. Numer. Anal., 22 (1985), pp. 634-643.
- [38] S. McCormick and J. Ruge, *Multigrid methods for variational problems*, SIAM J. Numer. Anal., 19 (1982), pp. 924-929.
- ✓ [39] H. Meinhardt, *Models of Biological Pattern Formation*, Academic Press, New York, 1982. Methods Partial Differential Eq, 15 (1999), pp. 201-214.
- ✓ [40] R. E. Mickens, *Nonstandard finite difference schemes for reaction-diffusion equations*, Numer Methods Partial Differential Eq, 15 (1999), pp. 201-214.
- [41] J. D. Murray, *Mathematical Biology, Biomathematics Texts*, Springer-Verlag, 1989.
- [42] J. D. Murray, *A Per-pattern Formation Mechanism for Animal Coat Markings*, J. Theor. Biol., 88 (1981), pp. 161-199.
- [43] R. A. Nicolaides, *On multigrid convergence in the indefinite case*, Math., Comp. 32 (1978), pp. 1082-1086.
- [44] J. M. Ortega, *Matrix Theory: A Second Course*, Plenum Press, New York, 1987.
- ✓ [45] C. V. Pao, *Accelerated monotone iterative methods for finite difference equations of reaction-diffusion*, Numer. Math., 79 (1998), pp. 621-281.
- ✓ [46] C. V. Pao, *Numerical analysis of coupled systems of nonlinear parabolic equations*, SIAM J. Numer Anal, 36 (1999), pp. 393-416.
- [47] S. V. Parter, *Estimates for multigrid methods based on red-black Gauss-Seidel smoothings*, Numer. Math., 52 (1988), pp. 701-723.
- [48] J. I. Ramos, *Linearization methods for reaction-diffusion equation: 1-D problems*, Appl. Math. and Comput., 88 (1997), pp. 199-244.
- ✓ [49] J. I. Ramos, *Linearization methods for reaction-diffusion equation: multidimensional problems*, Appl. Math. and Comput., 88 (1997), pp. 225-254.
- [50] R. D. Reitz, *A Study of Numerical Methods for Reaction-Diffusion Equations*, SIAM. J. Sci. Stat. Comput., 2 (1981), pp. 95-106.
- ✓ [51] R. D. Richtnyer and K. W. Morton, *Difference methods for initial value problems*, Interscience, New York, 1967.

- [52] J. Smoller, *Shock Waves and Reaction-Diffusion Equations.*, Berlin Heidelberg New York Tokyo, Springer 1983.
- [53] J. C. Strikwerd, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, Calif, 1989.
- [54] U. Trottenberg and K. Stuben, *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications. Multigrid Methods (Proceedings) Lecture Notes in Mathematics. Ed. W. Hackbusch and U. Trottenberg.* Spring-Verlag, New York, 1982, pp. 1-176.
- [55] J. Wang, *Convergence analysis of multigrid algorithms for nonselfadjoint and indefinite elliptic problems*, SIAM J. Numer. Anal., 30 (1993), pp. 275-285.
- [56] Y.M. Wang, *Petrov-Galerkin methods of nonlinear reaction-diffusion equations*, Appl. Math. Comput., 96 (1998), pp. 209-236.
- [57] P. Wesseling, *Theoretical and practical aspects of a multigrid method*, SIAM J. Sci. Statist. Comput., 4 (1982), pp. 387-407.
- [58] D. D. Young, *Iterative Solution of Large Linear Systems*, Academic, 1971.
- [59] F. Zhang, *Matrix Theory: Basic Results and Techniques*, Springer-Verlag, New York, 1999.
- [60] J. Zhang, *Acceleration of Five-point Red-Black Gauss-Seidel in Multigrid for Poisson Equation*, Appl. Math. Comput., 80 (1996), pp. 73-93.

MICHIGAN STATE LIBRARIES



3 1293 02174 1685