NUMERICAL SOLUTIONS OF NONLINEAR PARABOLIC PROBLEMS USING COMBINED-BLOCK ITERATIVE METHODS

Yaxi Zhao

A Thesis Submitted to the
University of North Carolina at Wilmington in Partial Fulfillment
Of the Requirements for the Degree of
Master of Science

Department of Mathematics and Statistics

University of North Carolina at Wilmington

2003

Approved by

Advisory Committee

Chair

Accepted by

Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT

This paper is concerned with the block monotone iterative schemes of numerical solutions of nonlinear parabolic systems with initial and boundary condition in two dimensional space. By using the finite difference method, the system is discretized into algebraic systems of equations, which can be represented as block matrices. Two iterative schemes, called the block Jacobi scheme and the block Gauss-Seidel scheme, are introduced to solve the system block by block. The Thomas algorithm is used to solve tridiagonal matrices system efficiently. For each scheme, two convergent sequences starting from the initial upper and lower solutions are constructed. Under a sufficient condition the monotonicity of the sequences, the existence and the uniqueness of solution are proven. To demonstrate how these method work, the numerical results of several examples with different types of nonlinear functions and different types of boundary conditions are also presented.

DEDICATION

To my family with love.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Prof. Xin Lu, my advisor, for his guidance and great help on my thesis. I also want to thank Prof. Wei Feng and Prof. Matthew TenHuisen for serving on my thesis advisory committee and all faculty members in the Mathematics and Statistics Department who gave me valuable advice and help.

LIST OF TABLES

1 INTRODUCTION

1.1 Background and Motivation

The studies of many physical phenomena like heat dispersion, chemical reaction and population dynamics etc. lead to reaction diffusion equations of the nonlinear parabolic type (See [6] about classification of PDE). For example, consider a simple irreversible monoenzyme kinetics in a biochemical system in space $\Omega \in \mathbf{R^2}$

$$
\begin{aligned}
u_t - D\nabla^2 u &= \frac{-\sigma u}{1 + au + bu^2} \quad \text{in } \ (0,T] \times \Omega \\
BC: \ u(t,x,y) &= h(t,x,y) \quad \{t \in (0,T], \ (x,y) \in \partial\Omega\} \\
IC: \ u(0,x,y) &= g(x,y) \quad \text{in } \ \Omega
\end{aligned}
$$

where σ, a, and b are positive constants and functions $h, \ g$ are given.

Among all nonlinear PDEs only a few special types can be solved analytically. In most situations such as the above example, we investigate the existence and uniqueness of their solutions, and also need to employ some appropriate numerical algorithms by utilizing the speed and memory of digital computers to get close approximations. There are many iterative methods for solving the nonlinear parabolic system such as the Picard, Jacobi, Gauss-Seidel monotone iterative schemes.

The fundament of this paper, the monotone iterative method, has been widely used recently. The details of this method may be found in [1] by Pao. In [2] Pao sought the point-wise numerical solution of a semilinear parabolic equation. In [4], Lu extended this method to the time-delay parabolic system and proved that his monotone iterative scheme is quadratically convergent. Most monotone iterative schemes are of the point-wise Picard type, which is inefficient in two or higher dimensional space.

By combining block partitioning and monotone methods Pao developed two iterative schemes, namely the Block Jocobi and Gauss-Seidel monotone iterative schemes, for nonlinear elliptic equation in [3]. These new numerical schemes are much more efficient than point-wise numerical schemes.

1.2 Problem and Goal

Consider the nonlinear parabolic type system with boundary and initial conditions in two dimensional space,

$$\begin{aligned} u_t - (D^1 u_x)_x - (D^2 u_y)_y = f(u, x, y, t) \quad &\text{in } \Omega \times (0, T] \\ BC: \quad B[u] = h(x, y, t) \quad &\text{on } \partial\Omega \times (0, T] \\ IC: \quad u(x, y, 0) = g(x, y) \quad &\text{in } \Omega \end{aligned} \tag{1}$$

where the boundary operator is defined as:

$$B[u] = \alpha \frac{\partial u}{\partial \nu} + \beta u$$

$\frac{\partial u}{\partial \nu}$ is the outward normal derivative on $\partial\Omega$, and $f(u, x, y, t)$ is a C^1 function. $D^1 = D^1(x, y)$, $D^2 = D^2(x, y)$ are positive functions on $\Omega \cup \partial\Omega$. $\alpha \equiv \alpha(x, y)$, $\beta \equiv \beta(x, y)$. This paper extends Block Jocobi and Gauss-Seidel monotone iterative schemes into solving parabolic systems to improve the computational efficiency further.

First we discretize (1) by finite difference and represent the corresponding finite difference system in terms of matrices. By partitioning the the system with respect to row, the system can be represented by block matrices. To solve the finite difference system, we construct monotone iterative sequences, namely,upper or lower sequences starting from either upper or lower solution, respectively, by applying Jocobi or Gauss-Seidel method on block matrices. Each block matrix is in the form of $Ax = b$, where A is tridiagonal. We choose to use the Thomas algorithm to solve the tridiagonal block because of its well known efficiency. The monotone properties of upper and lower sequences, existence and uniqueness of solutions are proven for both Block Jocobi and Gauss-Seidel methods. Finally numerical simulations of some examples are given to demonstrate the efficiency of both new numerical schemes.

2 Finite Difference System

To describe the continuous domain Ω as discrete points, we discretize the domain into N column and evenly divide each column into pieces with the size m. Therefor, the number of points on each column is $M_j,$ where M is integer and $j \in (0, N+1)$. Let the size of mesh grid to be $m \times n$, $m = \min(\frac{1}{M_j})$ and $n = \frac{1}{N}$, where $M_j,\ N$ are positive integers indicating the number of pieces along x-direction and y-direction. The continuous bounded convex domain Ω in R^2 can be approximately describe as $(M+1) \times (N+1)$ discrete grids. Correspondingly. u is represented by $u_{i,j,k}$. According to the finite difference method, one can consider the first derivatives u_x, u_y as,

$$u_x = \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2m}, \quad u_y = \frac{u_{i,j+1,k} - u_{i,j-1,k}}{2m}$$

and the second partial derivatives u_{xx}, u_{yy} by central approximation as

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{m^2}[u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}]$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{1}{n^2}[u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}]$$

Suppose we are solving the DE on [0,t], we divide time t into P pieces, each of which has the same length p. By the forward difference method, u_t can be described as

$$u_t = \frac{u_{i,j,k} - u_{i,j,k-1}}{p}$$

Now consider the general nonlinear parabolic system (1),

$$u_t - (D^1 u_x)_x - (D^2 u_y)_y = f(u, x, y, t)$$

Substitute these derivatives into Eq. (1), and it becomes

$$\frac{u_{i,j,k}-u_{i,j,k-1}}{p}-\frac{D^1_{i+1,j}-D^1_{i-1,j}}{2m}\cdot\frac{u_{i+1,j,k}-u_{i-1,j,k}}{2m}-D^1_{i,j}\frac{u_{i+1,j,k}-2u_{i,j,k}+u_{i-1,j,k}}{m^2}$$
$$-\frac{D^2_{i,j+1}-D^2_{i,j-1}}{2n}\cdot\frac{u_{i,j+1,k}-u_{i,j-1,k}}{2n}-D^2_{i,j}\frac{u_{i,j+1,k}-2u_{i,j,k}+u_{i,j-1,k}}{n^2}=f_{i,j,k}$$

Simplify this,

$$4m^2n^2(u_{i,j,k}-u_{i,j,k-1})-n^2p(D^1_{i+1,j}-D^1_{i-1,j})(u_{i+1,j,k}-u_{i-1,j,k})$$
$$-4n^2pD^1_{i,j}(u_{i+1,j,k}-2u_{i,j,k}+u_{i-1,j,k})-m^2p(D^2_{i,j+1}-D^2_{i,j-1})(u_{i,j+1,k}-u_{i,j-1,k})$$
$$-4m^2pD^2_{i,j}(u_{i,j+1,k}-2u_{i,j,k}+u_{i,j-1,k})=4m^2n^2pf_{i,j,k}$$

Collect terms,

$$(4m^2n^2+8n^2pD^1_{i,j}+8m^2pD^2_{i,j}+4m^2n^2p\bar{\gamma})u_{i,j,k}$$
$$-(m^2pD^2_{i,j+1}-m^2pD^2_{i,j-1}+4m^2pD^2_{i,j})u_{i,j+1,k}$$
$$-(m^2pD^2_{i,j-1}-m^2pD^2_{i,j+1}+4m^2pD^2_{i,j})u_{i,j-1,k}$$
$$-(n^2pD^1_{i+1,j}-n^2pD^1_{i-1,j}+4n^2pD^1_{i,j})u_{i+1,j,k}$$
$$-(n^2pD^1_{i-1,j}-n^2pD^1_{i+1,j}+4n^2pD^1_{i,j})u_{i-1,j,k}=4m^2n^2pf_{i,j,k}$$

Divide both sides by $4m^2n^2$,

$$(1+\frac{2pD^1_{i,j}}{m^2}+\frac{2pD^2_{i,j}}{n^2})u_{i,j,k}-(\frac{pD^2_{i,j+1}}{4n^2}-\frac{pD^2_{i,j-1}}{4n^2}+\frac{pD^2_{i,j}}{n^2})u_{i,j+1,k}$$
$$-(\frac{pD^2_{i,j-1}}{4n^2}-\frac{pD^2_{i,j+1}}{4n^2}+\frac{pD^2_{i,j}}{n^2})u_{i,j-1,k}-(\frac{pD^1_{i+1,j}}{4m^2}-\frac{pD^1_{i-1,j}}{4m^2}+\frac{pD^1_{i,j}}{m^2})u_{i+1,j,k}$$
$$-(\frac{pD^1_{i-1,j}}{4m^2}-\frac{pD^1_{i+1,j}}{4m^2}+\frac{pD^1_{i,j}}{m^2})u_{i-1,j,k}-u_{i,j,k-1}=pf_{i,j,k}$$

Rearrange terms,

$$\begin{aligned}
&[-(\frac{pD^1_{i-1,j}}{4m^2} - \frac{pD^1_{i+1,j}}{4m^2} + \frac{pD^1_{i,j}}{m^2})u_{i-1,j,k} + (1 + \frac{2pD^1_{i,j}}{m^2} + \frac{2pD^2_{i,j}}{n^2})u_{i,j,k} \\
&-(\frac{pD^1_{i+1,j}}{4m^2} - \frac{pD^1_{i-1,j}}{4m^2} + \frac{pD^1_{i,j}}{m^2})u_{i+1,j,k}] - (\frac{pD^2_{i,j-1}}{4n^2} - \frac{pD^2_{i,j+1}}{4n^2} + \frac{pD^2_{i,j}}{n^2})u_{i,j-1,k} \\
&-(\frac{pD^2_{i,j+1}}{4n^2} - \frac{pD^2_{i,j-1}}{4n^2} + \frac{pD^2_{i,j}}{n^2})u_{i,j+1,k} - u_{i,j,k-1} = pf_{i,j,k} \qquad (2)
\end{aligned}$$

If we let

$$b_{ij} = \frac{pD^1_{i-1,j}}{4m^2} - \frac{pD^1_{i+1,j}}{4m^2} + \frac{pD^1_{i,j}}{m^2}, \quad b'_{ij} = \frac{pD^1_{i+1,j}}{4m^2} - \frac{pD^1_{i-1,j}}{4m^2} + \frac{pD^1_{i,j}}{m^2}$$

$$c_{ij} = \frac{pD^2_{i,j-1}}{4n^2} - \frac{pD^2_{i,j+1}}{4n^2} + \frac{pD^2_{i,j}}{n^2}, \quad c'_{ij} = \frac{pD^2_{i,j+1}}{4n^2} - \frac{pD^2_{i,j-1}}{4n^2} + \frac{pD^2_{i,j}}{n^2}$$

$$a_{ij} = b_{ij} + b'_{ij} + c_{ij} + c'_{ij} + 1 = 1 + \frac{2pD^1_{i,j}}{m^2} + \frac{2pD^2_{i,j}}{n^2}$$

then we can rewrite Eq. (2) as

$$\begin{pmatrix} -b_{ij} & a_{ij} & -b'_{ij} \end{pmatrix} \begin{pmatrix} u_{i-1,j,k} \\ u_{i,j,k} \\ u_{i+1,j,k} \end{pmatrix} - c_{ij}u_{i,j-1,k} - c'_{ij}u_{i,j+1,k} - u_{i,j,k-1} = pf_{i,j,k} \qquad (3)$$

If the domain we are solving is not a rectangular box, at any moment k and for each fixed j, the number of grid points on x direction for each $k,\ j$ is not a constant, which depend on j. So i ranges form 0 to M_j including the boundary for each fixed $k,\ j$. Actually Eq. (3) represents a system of $M_j - 1$ equations for a fixed time and j.

Writing this system of equations in matrix form, it is

$$
\begin{pmatrix}
-b_{0,j} & a_{0,j} & -b'_{0,j} & . & . & . & 0 \\
0 & -b_{1,j} & a_{1,j} & -b'_{1,j} & . & . & \vdots \\
\vdots & . & \ddots & \ddots & \ddots & . & \vdots \\
\vdots & . & . & -b_{M_j-1,j} & a_{M_j-1,j} & -b'_{M_j-1,j} & 0 \\
0 & . & . & . & -b_{M_j,j} & a_{M_j,j} & -b'_{M_j,j}
\end{pmatrix}
\begin{pmatrix}
u_{-1,j,k} \\ u_{0,j,k} \\ \vdots \\ u_{M_j,j,k} \\ u_{M_j+1,j,k}
\end{pmatrix}
- \begin{pmatrix}
c_{0,j} & . & 0 \\
\vdots & \ddots & \vdots \\
0 & . & c_{M_j,j}
\end{pmatrix}
\begin{pmatrix}
u_{0,j-1,k} \\ \vdots \\ u_{M_j,j-1,k}
\end{pmatrix}
- \begin{pmatrix}
c'_{0,j} & . & 0 \\
\vdots & \ddots & \vdots \\
0 & . & c'_{M_j,j}
\end{pmatrix}
\begin{pmatrix}
u_{0,j+1,k} \\ \vdots \\ u_{M_j,j+1,k}
\end{pmatrix}
- \begin{pmatrix}
u_{0,j,k-1} \\ \vdots \\ u_{M_j,j,k-1}
\end{pmatrix}
= p \begin{pmatrix}
f_{0,j,k} \\ \vdots \\ f_{M_j,j,k}
\end{pmatrix}
$$

Notice that terms,$u_{-1,j,k},\ u_{M_j+1,j,k}$ and $u_{i,0,k},\ u_{i,N,k}$, are undefined. Those four terms are eliminated by applying the boundary condition on the extended exterior points and the points inside the domain. Then the system can be rewritten as the following.

$$
\begin{pmatrix}
a_{0,j} & -b'_{0,j} & . & . & 0 \\
-b_{1,j} & a_{1,j} & -b'_{1,j} & . & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & . & -b_{M_j-1,j} & a_{M_j-1,j} & -b'_{M_j-1,j} \\
0 & . & . & -b_{M_j,j} & a_{M_j,j}
\end{pmatrix}
\begin{pmatrix}
u_{0,j,k} \\ \vdots \\ u_{M_j,j,k}
\end{pmatrix}
- \begin{pmatrix}
c_{0,j} & . & 0 \\
\vdots & \ddots & \vdots \\
0 & . & c_{M_j,j}
\end{pmatrix}
\begin{pmatrix}
u_{0,j-1,k} \\ \vdots \\ u_{M_j,j-1,k}
\end{pmatrix}
- \begin{pmatrix}
c'_{0,j} & . & 0 \\
\vdots & \ddots & \vdots \\
0 & . & c'_{M_j,j}
\end{pmatrix}
\begin{pmatrix}
u_{0,j+1,k} \\ \vdots \\ u_{M_j,j+1,k}
\end{pmatrix}
$$

$$
-\begin{pmatrix} u_{0,j,k-1} \\ \vdots \\ u_{M_j,j,k-1} \end{pmatrix} = p \begin{pmatrix} f_{0,j,k} \\ \vdots \\ f_{M_j,j,k} \end{pmatrix} + G^*
$$

G^* is associated with coefficients $b_{0,j,k},\ b'_{0,j,k},\ b_{M_j,j,k},\ b'_{M_j,j,k}$, which are determined by the boundary condition. They could be zero if the boundary condition is of the Dirichlet type.

For each $j, k \quad j \in [0...N], k \in [1...P]$, let

$$
A_j = \begin{pmatrix} a_{0,j} & -b'_{0,j} & . & . & 0 \\ -b_{1,j} & a_{1,j} & -b'_{1,j} & . & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & . & -b_{M_j-1,j} & a_{M_j-1,j} & -b'_{M_j-1,j} \\ 0 & . & . & -b_{M_j,j} & a_{M_j,j} \end{pmatrix}, \quad \Gamma_j = \begin{pmatrix} p\bar{\gamma} & . & 0 \\ \vdots & \ddots & \vdots \\ 0 & . & p\bar{\gamma} \end{pmatrix}
$$

$$
U_{j,k} = \begin{pmatrix} u_{0,j,k} \\ \vdots \\ u_{M_j,j,k} \end{pmatrix}, \quad C_j = \begin{pmatrix} c_{0,j} & . & 0 \\ \vdots & \ddots & \vdots \\ 0 & . & c_{M_j,j} \end{pmatrix}
$$

$$
C'_j = \begin{pmatrix} c'_{0,j} & . & 0 \\ \vdots & \ddots & \vdots \\ 0 & . & c'_{M_j,j} \end{pmatrix}, \quad F_{j,k}(U_{j,k}) = p \begin{pmatrix} f_{0,j,k} \\ \vdots \\ f_{M_j,j,k} \end{pmatrix}
$$

A_j is a tridiagonal matrix with all diagonal entries greater than zero. A_j is invertible. Again, the sizes of matrices $U_{j,k},\ C,\ \Gamma$ depend on j. Then the above equation becomes

$$
A_j U_{j,k} - (C_j U_{j-1,k} + C'_j U_{j+1,k}) - U_{j,k-1} = F_{j,k}(U_{j,k}) + G^* \tag{4}
$$

This is equivalent to

$$A_jU_{j,k} = C_jU_{j-1,k} + C'_jU_{j+1,k} + U_{j,k-1} + F_{j,k}(U_{j,k}) + G^* \tag{5}$$

Definition 2.1 *For each $j,\ k,$ a column vector $\tilde{U}_{j,k} \equiv (\tilde{U}_{0,j,k}, ..., \tilde{U}_{M_j,j,k})' \in R^N$ is called the upper solution of (5), if*

$$A_j\tilde{U}_{j,k} \geq C_j\tilde{U}_{j-1,k} + C'_j\tilde{U}_{j+1,k} + \tilde{U}_{j,k-1} + F_{j,k}(\tilde{U}_{j,k}) + G^* \tag{6}$$

and $\hat{U}_{j,k} \equiv (\hat{U}_{0,j,k}, ..., \hat{U}_{M_j,j,k})' \in R^N$ is called the lower solution of (5) if

$$A_j\hat{U}_{j,k} \leq C_j\hat{U}_{j-1,k} + C'_j\hat{U}_{j+1,k} + \hat{U}_{j,k-1} + F_{j,k}(\hat{U}_{j,k}) + G^* \tag{7}$$

We say that $\tilde{U}_{jk}$ and $\hat{U}_{jk}$ are ordered if $\tilde{U} \geq \hat{U}$. At any time step k, given any ordered upper and lower solutions $\tilde{U}_{j,k} \equiv (\tilde{U}_{0,j,k}, ..., \tilde{U}_{M_j,j,k})'$, $\hat{U}_{j,k} \equiv (\hat{U}_{0,j,k}, ..., \hat{U}_{M_j,j,k})'$, we set

$$\begin{aligned} <\hat{U}, \tilde{U}> &\equiv \{U \in R^N; \hat{U} \leq \tilde{U}\}; \\ <\hat{U}_{j,k}, \tilde{U}_{j,k}> &\equiv \{U_{j,k} \in R^M; \hat{U}_{j,k} \leq \tilde{U}_{j,k}\}; \end{aligned} \tag{8}$$

Define

$\gamma_{ijk} \equiv \max\{-\frac{\partial f_{ijk}}{\partial u}(u_{ijk});\ \hat{u}_{ijk} \leq \tilde{u}_{ijk}\}$,

where $\hat{u}_{ijk}, \tilde{u}_{ijk}$ are the components of upper and lower solution respectively.

$$\gamma^+_{ijk} \equiv \max\{0, \gamma_{ijk}\}, \qquad\qquad \underline{\gamma}_{jk} \equiv \min\{\gamma^+_{ijk};\ i = 0, 1, ..., M\}$$

$\bar{\gamma}_{ijk}$ is any nonnegative function satisfying $\bar{\gamma}_{ijk} \geq \gamma^{+}_{ijk}$.

Define

$$\Gamma_j \equiv \text{diag}(p\bar{\gamma}_{0jk}, ..., p\bar{\gamma}_{M_j jk}),$$

then we have

$$F(\tilde{U}_{j,k}) - F(\hat{U}_{j,k}) + \Gamma_j(\tilde{U}_{j,k} - \hat{U}_{j,k}) \geq 0 \quad (9)$$

By adding $\Gamma_j U_{j,k}$ to both sides of Eq. (5), we get

$$(A_j + \Gamma_j)U_{j,k} = C_j U_{j-1,k} + C'_j U_{j+1,k} + U_{j,k-1} + F_{j,k}(U_{j,k}) + G^* + \Gamma_j U_{j,k} \quad (10)$$

No confusion should be raised that the upper solution $\tilde{U}_{j,k}$ and the lower solution $\hat{U}_{j,k}$ are still the upper and lower solutions of Eq. (10).

Let $\mathcal{U}_k$ be a column vector with $(N+1)$ block entries. For $j = 0...N$, after adding $\Gamma_k U_k$ to both side of Eq. (4), we can write the system of Eq. (4) in a more compact form. For $j = 0...N$, we can write the system of Eq. (4) in more compact form.

$$\mathcal{U}_k = \begin{pmatrix} \boxed{U_{0,k}} \\ \vdots \\ \boxed{U_{N+1,k}} \end{pmatrix}.$$

Similar to the way we deal with $b_{0,j}$ and $b'_{M_j,j}$, C_0 and C'_N are determined by the boundary condition along the y-direction. Again $C_0 U_{-1,k}$ and $C'_N U_{N+1,k}$ can be move to the right side as G'^*. Let $\mathcal{A}$ be the tridiagonal block matrix with diagonal sub-matrices $A_0 + \Gamma_0, ..., A_N + \Gamma_N$, off-diagonal sub-matrices $-C_1, ... - C_N$ and $-C'_0, ... -$

C'_{N-1}.

$$\mathcal{A} = \begin{pmatrix} \boxed{A_0+\Gamma_0} & \boxed{-C'_0} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \vdots & \ddots & \ddots & \ddots & \cdot & \cdot & \vdots \\ \vdots & \cdot & \cdot & \cdot & \boxed{-C_{N-1}} & \boxed{A_{N-1,k}+\Gamma_{N-1}} & \boxed{-C'_{N-1}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \boxed{-C_N} & \boxed{A_{N,k}+\Gamma_N} \end{pmatrix}$$

$$\mathcal{F}_k = (F_{0,k}, \ldots, F_{j-1,k}, F_{j,k}, F_{j+1,k}, \ldots F_{N,k})'$$

For each k, the whole system can be written as

$$\mathcal{A}\mathcal{U}_k = \mathcal{F}_k + \mathcal{U}_{k-1} + G_k^{'*} + \Gamma_k U_k \tag{11}$$

Without considering the boundary condityon, the tridiagonal matrix $A_j + \Gamma_j$ in Eq. (10) has pasitive entries on diagonal and negtive entries on offdiaganols for $n = 1...N-1$ and it is diagonally dominant. Moreover, $A_j + \Gamma_j$ is invertible when the boundary condition is either Dirichlet type or Rubin type. In fact, its eigenvalues have positive real parts (see [9]). For Neumann boundary condition, the eigenvalues have nonnegative real parts. In any case $(A_j + \Gamma_j)^{-1}$ exists and is positive.

3 The Block Jacobi Iterative Scheme

Based on Eq. (10), starting from either the initial upper solution $\tilde{U}$ or the initial lower solution $\hat{U}$, for $\forall\ k$ we formulate the Jacobi type of block iterative scheme to generate the sequence $\{U_{j,k}^{(r)}\}$.

$$(A_j+\Gamma_j)U_{j,k}^{(r)} = C_jU_{j-1,k}^{(r-1)} + C_j'U_{j+1,k}^{(r-1)} + U_{j,k-1} + F(U_{j,k}^{(r-1)}) + G^{*(r-1)} + \Gamma_jU_{j,k}^{(r-1)} \quad (12)$$

where $j = 0...N_i$, and $r = 0, 1, 2, ...$

It is easy to see that $A_j + \Gamma_j$ is a tridiagonal matrix with all entries on the diagonal greater than zero, so the inverse of $A_j + \Gamma_j$ exits and $(A_j + \Gamma_j)^{-1} > 0$. This equation can be solved by using the Thomas algorithm (see [5]). Starting form the upper solution $\tilde{U}$ or lower solution $\hat{U}$, $\overline{U}_k^{(0)} = \tilde{U}_k$ or $\underline{U}_k^{(0)} = \hat{U}_k$, we construct a sequence $\{\overline{U}_k^{(r)}\} = \{\overline{U}_{0k}^{(r)}, ..., \overline{U}_{N_ik}^{(r)}\}$ or $\{\underline{U}_k^{(r)}\} = \{\underline{U}_{0k}^{(r)}, ..., \underline{U}_{N_ik}^{(r)}\}$, which refers to the maximal sequence or the minimal sequence. The monotone properties of these sequences are given by the following lemma.

Lemma 3.1 *The maximal and minimal sequences* $\{\overline{U}_k^{(r)}\}$, $\{\underline{U}_k^{(r)}\}$ *given by Eq. 12 with* $\overline{U}_k^{(0)} = \tilde{U}_k$ *and* $\underline{U}_k^{(0)} = \hat{U}_k$ *possess the monotone property*

$$\hat{U}_k = \underline{U}_k^{(0)} \le \underline{U}_k^{(1)} \le ... \le \underline{U}_k^{(r)} \le \underline{U}_k^{(r+1)} \le \overline{U}_k^{(r+1)} \le \overline{U}_k^{(r)} \le ... \le \overline{U}_k^{(1)} \le \overline{U}_k^{(0)} = \tilde{U}_k \quad (13)$$

Moreover for each r, $\overline{U}_k^{(r)}$ *and* $\underline{U}_k^{(r)}$ *are ordered upper and lower solutions.*

Proof. Let $W_{jk}^{(0)} = \overline{U}_{jk}^{(0)} - \overline{U}_{jk}^{(1)} = \tilde{U}_{jk} - \overline{U}_{jk}^{(1)}$.

$$\begin{aligned}(A_j+\Gamma_j)W_{jk}^{(0)} &= (A_j+\Gamma_j)\tilde{U}_{jk} - (A_j+\Gamma_j)\overline{U}_{jk}^{(1)} \\ &\quad \text{By Equ. (12)} \\ &= (A_j+\Gamma_j)\tilde{U}_{jk} - [C_j\overline{U}_{j-1,k}^{(0)} + C_j'\overline{U}_{j+1,k}^{(0)} + U_{j,k-1} \\ &\quad +F(\overline{U}_{j,k}^{(0)}) + \overline{G}^{*(0)} + \Gamma_j\overline{U}_{j,k}^{(0)}]\end{aligned}$$

$$
\begin{aligned}
&= (A_j + \Gamma_j)\tilde{U}_{jk} - [C_j\tilde{U}_{j-1,k} + C'_j\tilde{U}_{j+1,k} + \tilde{U}_{j,k-1} \\
&\quad + F(\tilde{U}_{j,k}) + \tilde{G}^* + \Gamma_j\tilde{U}_{jk}] \\
&= A_j\tilde{U}_{jk} - [C_j\tilde{U}_{j-1,k} + C'_j\tilde{U}_{j+1,k} + \tilde{U}_{j,k-1} + F(\tilde{U}_{j,k}) + \tilde{G}^*]
\end{aligned}
$$

Because $\tilde{U}_{j,k}$ is the upper solution, from the Definition 2.1, the right hand side is greater than zero. We have $(A_j + \Gamma_j)W_{jk}^{(0)} \geq 0$. Also because $(A_j + \Gamma_j)^{-1} > 0$, then $W_{jk}^{(0)} \geq 0$ and $\overline{U}_{jk}^{(0)} \geq \overline{U}_{jk}^{(1)}$. In the same fashion we can show $\underline{U}_{jk}^{(0)} \leq \underline{U}_{jk}^{(1)}$.
Let $W_{jk}^{(1)} = \overline{U}_{jk}^{(1)} - \underline{U}_{jk}^{(1)}$

$$
\begin{aligned}
(A_j + \Gamma_j)W_{jk}^{(1)} &= (A_j + \Gamma_j)\overline{U}_{jk}^{(1)} - (A_j + \Gamma_j)\underline{U}_{jk}^{(1)} \\
&= C_j\overline{U}_{j-1,k}^{(0)} + C'_j\overline{U}_{j+1,k}^{(0)} + \overline{U}_{j,k-1} + F(\overline{U}_{j,k}^{(0)}) + \overline{G}^{*(0)} + \Gamma_j\overline{U}_{j,k}^{(0)} \\
&\quad - C_j\underline{U}_{j-1,k}^{(0)} - C'_j\underline{U}_{j+1,k}^{(0)} - \underline{U}_{j,k-1} - F(\underline{U}_{j,k}^{(0)}) - \underline{G}^{*(0)} - \Gamma_j\underline{U}_{j,k}^{(0)} \\
&= C_j\tilde{U}_{j-1,k} + C'_j\tilde{U}_{j+1,k} + \tilde{U}_{j,k-1} + F(\tilde{U}_{j,k}) + \tilde{G}^* + \Gamma_j\tilde{U}_{j,k} \\
&\quad - C_j\hat{U}_{j-1,k} - C'_j\hat{U}_{j+1,k} - \hat{U}_{j,k-1} - F(\hat{U}_{j,k}) - \hat{G}^* - \Gamma_j\hat{U}_{j,k} \\
&= C_j(\tilde{U}_{j-1,k} - \hat{U}_{j-1,k}) + C'_j(\tilde{U}_{j+1,k} - \hat{U}_{j+1,k}) + (\tilde{U}_{j,k-1} - \hat{U}_{j,k-1}) \\
&\quad + F(\tilde{U}_{j,k}) - F(\hat{U}_{j,k}) + (\tilde{G}^* - \hat{G}^*) + \Gamma_j(\tilde{U}_{j,k} - \hat{U}_{j,k})
\end{aligned}
$$

Since $< \tilde{U}_{j,k}, \hat{U}_{j,k} >$ are ordered, so $\tilde{U}_{j-1,k} \geq \hat{U}_{j-1,k}$. According to inequality (9) and the nonnegative property of C and Γ the right hand side of the above equation is greater than zero. Then we have

$$\overline{U}_{jk}^{(1)} \geq \underline{U}_{jk}^{(1)}$$

In the same fashion, by mathematical induction, we have $\overline{U}_{jk}^{(r)} \geq \overline{U}_{jk}^{(r+1)}$, $\underline{U}_{jk}^{(r)} \leq \underline{U}_{jk}^{(r+1)}$ and $\overline{U}_{jk}^{(r)} \geq \underline{U}_{jk}^{(r)}$. Putting these together, we have

$$\hat{U}_k = \underline{U}_k^{(0)} \leq \underline{U}_k^{(1)} \leq ... \leq \underline{U}_k^{(r)} \leq \underline{U}_k^{(r+1)} \leq \overline{U}_k^{(r+1)} \leq \overline{U}_k^{(r)} \leq ... \leq \overline{U}_k^{(1)} \leq \overline{U}_k^{(0)} = \tilde{U}_k$$

QED.

Based this monotonicity lemma, we have the following convergence theorem.

Theorem 3.1 *Let $\tilde{U}_{j,k}, \hat{U}_{j,k}$ be a pair of ordered upper and lower solutions of Eq. (4). Then the sequences $\{\overline{U}_k^{(r)}\} = \{\overline{U}_{0k}^{(r)}, ..., \overline{U}_{N_i k}^{(r)}\}$, $\{\underline{U}_k^{(r)}\} = \{\underline{U}_{0k}^{(r)}, ..., \underline{U}_{N_i k}^{(r)}\}$ given by Eq.(12) with $\overline{U}^{(0)} = \tilde{U}$, $\underline{U}^{(0)} = \overline{U}$ converge monotonically to solutions $\overline{U}_k$ and $\underline{U}_k$ of Eq. (4), respectively. Moreover*

$$\hat{U}_k \leq ... \leq \underline{U}_k^{(r)} \leq ... \leq \underline{U}_k \leq \overline{U}_k \leq ... \leq \overline{U}_k^{(r)} \leq ... \leq \tilde{U}_k \tag{14}$$

and if $U_k^* \in < \tilde{U}_{j,k}, \hat{U}_{j,k} >$ is the solution of Eq. (4) then

$$\underline{U}_k \leq U_k^* \leq \overline{U}_k$$

Proof. By Lemma 3.1 we know that $\{\overline{U}_k^{(r)}\}$ is monotone decreasing and it is bounded below by $\hat{U}_k$. From [8], a bounded monotone sequence must have a limit, say $\lim_{r\to\infty} \overline{U}_k^{(r)} = \overline{U}_k$. So $\overline{U}_k \leq \overline{U}_k^{(r)}$. Similarly we have $\lim_{r\to\infty} \underline{U}_k^{(r)} = \underline{U}_k$. Letting $m \to \infty$, $\overline{U}_k$ and $\underline{U}_k$ are solutions of Eq. (4). For $\forall r = 0, 1, ...,$ $\underline{U}_k^{(r)}$ and $\overline{U}_k^{(r)}$ are ordered and those two sequences are monotone, $\underline{U}_k \leq \overline{U}_k$. Now if U_k^* is a solution in the sector $< \hat{U}_k, \tilde{U}_k >$, then $U_{j,k}^*, \hat{U}_{j,k}$ are ordered upper and lower solutions. Using $\overline{U}_{j,k}^{(0)} = U_{j,k}^*$ and $\underline{U}_{j,k}^{(0)} = \hat{U}_{j,k}$ theorem 3.1 Ineq (14) tells that $\underline{U}_k \leq U_{j,k}^*$. Similarly, it is easy to get $U_{j,k}^* \leq \overline{U}_k$. So

$$\hat{U}_k \leq ... \leq \underline{U}_k^{(r)} \leq ... \leq \underline{U}_k \leq U_k^* \leq \overline{U}_k \leq ... \leq \overline{U}_k^{(r)} \leq ... \leq \tilde{U}_k \tag{15}$$

QED.

The following theorem shows that under a certain condition the finite system has a unique solution.

Theorem 3.2 (Uniqueness) *Let*

$$\sigma \equiv \max\{\frac{\partial f_{i,j,k}}{\partial u}(u_{i,j,k});\ \hat{u}_{i,j,k} < u_{i,j,k} < \tilde{u}_{i,j,k}\},$$

If the conditions in Theorem 3.1 hold and $\sigma \leq p^{-1}$*, then* $\overline{U}_k = \underline{U}_k$ *and it is the unique solution of Eq. (4).*

Proof. Let $V_k = \overline{U}_k - \underline{U}_k$. When $k = 1$, $V_1 = \overline{U}_1 - \underline{U}_1 \geq 0$.

Substitute it in Eq. (11)

$$\begin{aligned} \mathcal{A}V &= F(\overline{U}_1) - F(\underline{U}_1) + \overline{U}_0 - \underline{U}_0 + \overline{G}_0^{'*} - \underline{G}_0^{'*} \\ &\quad \text{recall that } F = pf \text{ and when } k = 0 \text{ the initial condition applies.} \\ &= p[f(\overline{U}_1) - f(\underline{U}_1)] \ \leq \ p\sigma(\overline{U}_1 - \underline{U}_1) \ = \ p\sigma V_1 \end{aligned}$$

If $p\sigma \leq 1$, then $(A - I)V_1 \leq (A - p\sigma)V_1 \leq 0$

$(A - I)^{-1} \geq 0 \ \Rightarrow \ V_1 \leq 0$. Because V_1 can not be > 0 and < 0 and the same time, so $V_1 = 0$.

When $k = 2, 3...$, following the same derivation, by induction, we can prove that $V_k = 0,\ \forall k \in N$. That is $\underline{U}_k = \overline{U}_k$.

QED.

4 The Gauss-Seidel Iterative Scheme

Based on Eq. (10), we can construct the block Gauss-Seidel iterative scheme:

$$(A_j + \Gamma_j)U_{j,k}^{(r)} = C_j U_{j-1,k}^{(r)} + C_j U_{j+1,k}^{(r-1)} + U_{j,k-1} + F(U_{j,k}^{(r-1)}) + G^{*(r-1)} + \Gamma_j U_{j,k}^{(r-1)} \quad (16)$$

Denote the sequence again by $\{\overline{U}_k^{(m)}\} = \{\overline{U}_{0k}^{(m)}, ..., \overline{U}_{N_i k}^{(m)}\}$ when $\overline{U}_k^{(0)} = \tilde{U}_k$ and $\{\underline{U}_k^{(m)}\} = \{\underline{U}_{0k}^{(m)}, ..., \underline{U}_{N_i k}^{(m)}\}$ when $\underline{U}_k^{(0)} = \hat{U}_k$, and refer to them as the maximal and minimal sequences, respectively. The following lemma gives an analogous result as in Lemma 3.1.

Lemma 4.1 *The maximal and minimal sequences $\{\overline{U}_k^{(m)}\}$, $\{\underline{U}_k^{(m)}\}$ given by (16) with $\overline{U}_k^{(0)} = \tilde{U}_k$ and $\{\underline{U}_k^{(m)}\}$ possess the same monotone property (13). Moreover, for each r $\{\overline{U}_k^{(m)}\}$, and $\{\underline{U}_k^{(m)}\}$ are ordered upper and lower solutions.*

Proof. Let $W_{jk}^{(0)} = \overline{U}_{jk}^{(0)} - \overline{U}_{jk}^{(1)} = \tilde{U}_{jk} - \overline{U}_{jk}^{(1)}$.

$$\begin{aligned}
(A_j + \Gamma_j)W_{jk}^{(0)} &= (A_j + \Gamma_j)\tilde{U}_{jk} - (A_j + \Gamma_j)\overline{U}_{jk}^{(1)} \\
&\quad \text{By Equ. (16)} \\
&= (A_j + \Gamma_j)\tilde{U}_{jk} - [C_j\overline{U}_{j-1,k}^{(1)} + C_j'\overline{U}_{j+1,k}^{(0)} + U_{j,k-1} + F(\overline{U}_{j,k}^{(0)}) \\
&\quad + \overline{G}^{*(0)} + \Gamma_j \overline{U}_{j,k}^{(0)}] \\
&= (A_j + \Gamma_j)\tilde{U}_{jk} - [C_j\overline{U}_{j-1,k}^{(1)} + C_j'\tilde{U}_{j+1,k} + \tilde{U}_{j,k-1} + F(\tilde{U}_{j,k}) \\
&\quad + \tilde{G}^{*} + \Gamma_j \tilde{U}_{jk}] \\
&= A_j\tilde{U}_{jk} - [C_j\overline{U}_{j-1,k}^{(1)} + C_j'\tilde{U}_{j+1,k} + \tilde{U}_{j,k-1} + F(\tilde{U}_{j,k}) + \tilde{G}^{*}]
\end{aligned}$$

Because $\tilde{U}_{j,k}$ is the upper solution, from the Definition 2.1, we have

$$(A_j + \Gamma_j)W_{jk}^{(0)} \geq C_j\tilde{U}_{j-1,k} - C_j U_{j-1,k}^{(1)} = C_j W_{j-1,k}^{(0)}$$

When $j=0,\ C_0=0$, $(A_j+\Gamma_j)W_{0k}^{(0)}\geq 0$. Because $\text{inv}(A_j+\Gamma_j)>0$, $W_{0k}^{(0)}\geq 0$.
When $j=1$, $(A_j+\Gamma_j)W_{1k}^{(0)}\geq C_jW_{0,k}^{(0)}, \Rightarrow, W_{1k}^{(0)}\geq 0$
By induction, $W_{jk}^{(0)}\geq 0$, that is $\overline{U}_{jk}^{(0)}\geq \overline{U}_{jk}^{(1)}$. Similarly we can show $\underline{U}_{jk}^{(0)}\leq \underline{U}_{jk}^{(1)}$.
Then let $W_{jk}^{(1)}=\overline{U}_{jk}^{(1)}-\underline{U}_{jk}^{(1)}$

$$\begin{aligned}
(A_j+\Gamma_j)W_{jk}^{(1)} &= (A_j+\Gamma_j)\overline{U}_{jk}^{(1)}-(A_j+\Gamma_j)\underline{U}_{jk}^{(1)}\\
&= C_j\overline{U}_{j-1,k}^{(1)}+C_j'\overline{U}_{j+1,k}^{(0)}+\overline{U}_{j,k-1}+F(\overline{U}_{j,k}^{(0)})+\overline{G}^{*(0)}+\Gamma_j\overline{U}_{j,k}^{(0)}\\
&\quad -C_j\underline{U}_{j-1,k}^{(1)}-C_j'\underline{U}_{j+1,k}^{(0)}-\underline{U}_{j,k-1}-F(\underline{U}_{j,k}^{(0)})-\underline{G}^{*(0)}-\Gamma_j\underline{U}_{j,k}^{(0)}\\
&= C_j\overline{U}_{j-1,k}^{(1)}+C_j'\tilde{U}_{j+1,k}+\tilde{U}_{j,k-1}+F(\tilde{U}_{j,k})+\tilde{G}^*+\Gamma_j\tilde{U}_{j,k}\\
&\quad -C_j\underline{U}_{j-1,k}^{(1)}-C_j'\hat{U}_{j+1,k}-\hat{U}_{j,k-1}-F(\hat{U}_{j,k})-\hat{G}^*-\Gamma_j\hat{U}_{j,k}\\
&= C_j(\overline{U}_{j-1,k}^{(1)}-\underline{U}_{j-1,k}^{(1)})+C_j'(\tilde{U}_{j+1,k}-\hat{U}_{j+1,k})+(\tilde{U}_{j,k-1}-\hat{U}_{j,k-1})\\
&\quad +F(\tilde{U}_{j,k})-F(\hat{U}_{j,k})+(\tilde{G}^*-\hat{G}^*)+\Gamma_j(\tilde{U}_{j,k}-\hat{U}_{j,k})\\
&= C_jW_{j-1,k}^{(1)}+C_j'(\tilde{U}_{j+1,k}-\hat{U}_{j+1,k})+(\tilde{U}_{j,k-1}-\hat{U}_{j,k-1})\\
&\quad +F(\tilde{U}_{j,k})-F(\hat{U}_{j,k})+(\tilde{G}^*-\hat{G}^*)+\Gamma_j(\tilde{U}_{j,k}-\hat{U}_{j,k})
\end{aligned}$$

Since $<\tilde{U}_{j,k},\hat{U}_{j,k}>$ are ordered, so $\tilde{U}_{j+1,k}\geq\hat{U}_{j+1,k}$ and $\tilde{U}_{j-1,k}\geq\hat{U}_{j-1,k}$. According to inequality (9) and the nonnegative property of C and Γ the right hand side of above equation is greater than $C_jW_{j-1,k}^{(1)}$. Then we have

$$(A_j+\Gamma_j)W_{jk}^{(1)}\geq C_jW_{j-1,k}^{(1)}$$

When $j=0$, $C_0=0$, $(A_j+\Gamma_j)W_{jk}^{(1)}\geq 0, \Rightarrow\ W_{0k}^{(1)}\geq 0$
When $j=1$, $(A_j+\Gamma_j)W_{1k}^{(1)}\geq C_jW_{0,k}^{(1)}, \Rightarrow\ W_{1k}^{(1)}\geq 0$
By induction, $W_{jk}^{(1)}\geq 0$, that is $\overline{U}_{jk}^{(1)}\geq \underline{U}_{jk}^{(1)}$ By induction again, $\forall r,\ \overline{U}_{jk}^{(r)}\geq\underline{U}_{jk}^{(r)}$.
In the same fashion, by mathematic induction, we have $\overline{U}_{jk}^{(m)}\geq\overline{U}_{jk}^{(m+1)}$, $\underline{U}_{jk}^{(m)}\leq$

$\underline{U}_{jk}^{(m+1)}$ and $\overline{U}_{jk}^{(m)} \geq \underline{U}_{jk}^{(m)}$. Putting these together, we have

$$\hat{U}_k = \underline{U}_k^{(0)} \leq \underline{U}_k^{(1)} \leq ... \leq \underline{U}_k^{(r)} \leq \underline{U}_k^{(r+1)} \leq \overline{U}_k^{(r+1)} \leq \overline{U}_k^{(r)} \leq ... \leq \overline{U}_k^{(1)} \leq \overline{U}_k^{(0)} = \tilde{U}_k$$

QED.

Theorem 4.1 *Let the conditions in Theorem 3.1 hold. Then the sequences* $\{\overline{U}_k^{(m)}\}$, $\{\underline{U}_k^{(m)}\}$ *given by (16) with* $\overline{U}_k^{(0)} = \tilde{U}_k$ *and* $\{\underline{U}_k^{(m)}\}$ *converge monotonically to their respective solutions* $\overline{U}$ *and* $\underline{U}$, they satisfy the same relation (14). Moreover if U^* is any solution of Eq. (4) in $< \hat{U},\ \tilde{U} >$, then $\underline{U} \leq U^* \leq \overline{U}$.

Proof. The proof exactly follows the same steps as the proof of Thm 3.1.

Theorem 3.2 (Uniqueness) *Let*

$$\sigma \equiv \max\{\frac{\partial f_{i,j,k}}{\partial u}(u_{i,j,k});\ \hat{u}_{i,j,k} < u_{i,j,k} < \tilde{u}_{i,j,k}\},$$

If the conditions in Theorem 3.1 hold and $\sigma \leq p^{-1}$, *then* $\overline{U}_k = \underline{U}_k$ *and it is the unique solution of Eq. (4).*

5 Applications and Numerical Results

In this section, several numerical results are given by applying the block monotone iterative methods. It is shown that the computational error tends to zero by decreasing the mesh size. Considering the complexity of the program, the examples are solved only on a rectangular domain. The problems with irregular shapes can be solved in the same fashion. The programming environment is chosen in MATLAB because of its excellence of matrix manipulating.

Example 1.

Consider the IBVP problem on a unit square $\Omega = \{(x, y),\ 0 \le x \le 1,\ 0 \le y \le 1\}$.

$$u_t - \Delta u = f(u, x, y, t)$$

$$BC:\ u(0, y, t) = u(1, y, t) = u(x, 0, t) = u(x, 1, t) = 0$$

$$IC: u(x, y, 0) = 100 \sin \pi x \sin \pi y$$

with nonlinear function $f(u, x, y, t) = u(1 - u) + q(x, y, t)$, where
$q(x, y, t) = 200e^{-t} \sin \pi x \sin \pi y(-1 + \pi^2 + 50e^{-t} \sin \pi x \sin \pi y)$.
The analytical solution can be found as $100e^{-t} \sin \pi x \sin \pi y$.
The first step of solving the nonlinear system is to find the upper and lower solutions.

$$u_t - \Delta u = u(1 - u) + (100e^{-2\pi^2 t} \sin \pi x \sin \pi y)^2 - 100e^{-2\pi^2 t} \sin \pi x \sin \pi y \le 12000$$

The solution of the linear parabolic system

$$u_t - \Delta u = 12000$$

with the same boundary condition and initial value is the upper solution of corresponding nonlinear system. It is also easy to verify that zero is the lower solution. If the point-wise ℓ_2 norm of the two sequences is small enough,

$$\|\overline{U}^{(r)} - \underline{U}^{(r)}\| \leq \epsilon, \text{ where } \epsilon \text{ is any positive real number,}$$

then iterations are terminated at r^{th} step. Either the upper solution of the lower solution can be regarded as the approximation of the true solution. Tab(1) and Tab(2) show the maximal and the minimal solutions and the error rate as long as the number of iteration when $t = 1$ with mesh size 0.1×0.1. Tab(3) and Tab(4) contain the solutions on some fixed points with different time t. Tab(5) and Tab(6) demonstrate the monotone property of the two iterative methods.

Example 2.

Consider this model describing the enzyme kinetics

$$\begin{aligned} u_t - \Delta u &= \frac{-u}{1+u} + \frac{e^{-t}\sin \pi x \sin \pi y}{1 + e^{-t}\sin \pi x \sin \pi y} + (2\pi^2 - 1)e^{-t}\sin \pi x \sin \pi y \\ BC:\ u &= 0 \\ IC:\ u &= \sin \pi x \sin \pi y \end{aligned}$$

The numerical results given in Tab(7) through Tab(12) are similar to Tab(1) through Tab(6).

Example 3.

Consider a parabolic DE with the Neumann type of boundary condition:

$$u_t - \Delta u = u(1-u) + (2\pi^2 - 2)e^{-t}\cos \pi x \cos \pi y + (e^{-t}\cos \pi x \cos \pi y)^2$$

$$BC: \quad u_x(0, y, t) = u_x(1, y, t) = u_y(x, 0, t) = u_y(x, 1, t) = 0$$

$$IC: u(x, y, 0) = \cos \pi x \cos \pi y$$

The analytical solution can be found as $e^{-t} \cos \pi x \cos \pi y$, and $f_u = 1 - 2u$

$$\underline{c} = \max\{-f_u\} = max(2u - 1) = 1$$

(2, -2) is a pair of upper and lower solutions.

The results are given in Tab(13) and Tab(14).

Not only is the block iterative method designed for nonlinear problems, it can be used for solving linear problems as well. In this case, by imposing Γ_j to be 0, starting from any initial guess the iterative sequences approaches the true solution.

Example 4.

$$u_t - \Delta u = (2\pi^2 - 1)100e^{-t} \sin \pi x \sin \pi y$$

$$BC: \ u(0, y, t) = u(1, y, t) = u(x, 0, t) = u(x, 1, t) = 0$$

$$IC: u(x, y, 0) = 100 \sin \pi x \sin \pi y$$

For the comparison, the BC, IC and analytical solution are chosen be be the same as Example 1 except the the reaction function f. The results are shown in Tab(16) and Tab(17).

Table 1: Results of The Block Jacobi Method for Example 1

	(x,y)	0.2, 0.2	0.2, 0.4	0.4, 0.2	0.4, 0.4	0.5, 0.5	Iteration /error
M= 10	max	12.7587	20.6337	20.6338	33.3675	36.8848	62
P=10	min	12.7577	20.6326	20.6325	33.3662	36.8835	3.0E-3
M=20	max	12.7250	20.5864	20.5864	30.3038	36.8181	140
P=20	min	12.7245	20.5858	20.5857	33.3031	36.8174	9.30E-4
M=40	max	12.7151	20.5725	20.5725	33.2846	36.7981	410
P=40	min	12.7099	20.5650	20.5650	33.2750	36.7979	3.21E-4
	true	12.7099	20.5650	20.5650	33.2750	36.7879	

Table 2: Results of The Block Gauss-Seidel Method for Example 1

	(x,y)	0.2, 0.2	0.2, 0.4	0.4, 0.2	0.4, 0.4	0.5, 0.5	Iteration /error
M= 10	max	12.7588	20.6336	20.6337	33.3672	36.8844	38
P=10	min	12.7573	20.6325	20.6323	33.3663	36.8838	3.0E-3
M=20	max	12.7251	20.5864	20.5865	30.3038	36.8180	86
P=20	min	12.7244	20.5858	20.5857	33.3031	36.8175	9.36E-4
M=40	max	12.7152	20.5725	20.5725	33.2850	36.7984	221
P=40	min	12.7149	20.5721	20.5721	33.2846	36.7981	3.21E-4
	true	12.7099	20.5650	20.5650	33.2750	36.7879	

Table 3: Solutions by Using The Jacobi Method for Example 1

	(x,y)	0.2, 0.2	0.2, 0.4	0.4, 0.2	0.4, 0.4	0.5, 0.5
	max	28.3507	45.8516	45.8515	74.1549	81.9801
t=0.2	min	28.3504	45.8513	45.8512	74.1535	81.9774
	max	25.6577	41.4960	41.4960	67.1109	74.1838
t=0.3	min	25.6573	41.4952	41.4953	67.1091	74.1829
	max	23.2200	37.5527	37.5526	60.7312	67.1364
t=0.4	min	23.2192	37.5520	37.5522	60.7311	67.1352
	max	21.0143	33.9853	33.9851	54.9612	60.7557
t=0.5	min	21.0130	33.9839	33.9841	54.9602	60.7552
	max	19.0180	30.7567	30.7565	49.7391	54.9831
t=0.6	min	19.0169	30.7554	30.7556	49.7382	54.9823
	max	17.2116	27.8351	27.8350	45.0138	49.7594
t=0.7	min	17.2105	27.8338	27.8340	45.0129	49.7586
	max	15.5768	25.1913	25.1912	40.7380	45.0326
t=0.8	min	15.5758	25.1901	25.1902	40.7370	45.0316
	max	14.0975	22.7988	22.7988	36.8688	40.7553
t=0.9	min	14.0965	22.7976	22.7977	36.8677	40.7543
	max	12.7588	20.6339	20.6338	33.3675	36.8848
t=1	min	12.7578	20.6326	20.6326	33.3662	36.8836

Table 4: Solutions by Using The Gauss-Seidel Method for Example 1

Time	(x,y)	0.2, 0.2	0.2, 0.4	0.4, 0.2	0.4, 0.4	0.5, 0.5
	max	28.3512	45.8325	45.8523	74.1569	81.974
t=0.2	min	28.3498	45.8518	45.8518	74.1568	81.9774
	max	25.6578	41.4955	41.4955	67.1092	74.1859
t=0.3	min	25.6564	41.4950	41.4950	67.1091	74.1859
	max	23.2201	37.5527	37.5528	60.7317	67.1355
t=0.4	min	23.2188	37.5522	37.5521	60.7315	67.1353
	max	21.0142	33.9849	33.9849	54.9608	60.7557
t=0.5	min	21.0128	33.9842	33.9841	54.9605	60.7556
	max	19.0180	30.7564	30.7564	49.7389	54.9829
t=0.6	min	19.0165	30.7556	30.7554	49.7385	54.9826
	max	17.2116	27.8348	27.8349	45.0136	49.7592
t=0.7	min	17.2100	27.8339	27.8337	45.0131	49.7588
	max	15.5770	25.1911	25.1913	40.7378	45.0324
t=0.8	min	15.5751	25.1900	25.1897	40.7370	45.0318
	max	14.0975	22.7986	22.7987	36.8685	40.7550
t=0.9	min	14.0962	22.7977	22.7975	36.8678	40.7545
	max	12.7589	20.6336	20.6338	33.3673	36.8845
t=1	min	12.7574	20.6326	20.6323	33.3663	36.8838

Table 5: Jacobi Iterations for Example 1 When $t = 1$

Iteration	(x,y)	0.2, 0.2	0.2, 0.4	0.4, 0.2	0.4, 0.4	0.5, 0.5
	max	23.5151	33.1348	30.4532	43.9541	47.6547
4	min	6.5165	11.7833	11.6565	21.4181	24.4394
	max	15.5955	24.0227	23.4375	36.3808	39.9234
8	min	10.3685	17.5200	17.5800	29.7344	33.2629
	max	13.5687	21.6245	21.4705	34.2908	37.7820
12	min	11.9612	19.6311	19.6914	32.2927	35.8367
	max	12.9978	20.9287	20.8866	33.6495	37.1553
16	min	12.5077	20.3219	20.3496	33.0476	36.5761
	max	12.8301	20.7220	20.7101	33.4529	36.9665
20	min	12.6813	20.5397	20.5480	33.2713	36.7924
	max	12.7799	20.6600	20.6566	33.3930	36.9092
24	min	12.7348	20.6042	20.6075	33.3381	36.8566
	max	12.7649	20.6414	20.6403	33.3749	36.8918
28	min	12.7511	20.6244	20.6254	33.3582	36.8759
	max	12.7603	20.6357	20.6354	33.3694	36.8866
32	min	12.7561	20.6305	20.6308	33.3642	36.8817
	max	12.7590	20.6340	20.6339	33.3677	36.8850
36	min	12.7576	20.6324	20.6325	33.3660	36.8834

Table 6: Gauss-Seidel Iterations for Example 1 When $t = 1$

Iteration	(x,y)	0.2, 0.2	0.2, 0.4	0.4, 0.2	0.4, 0.4	0.5, 0.5
	max	14.6396	22.4451	22.3711	33.8552	38.0858
6	min	10.2580	18.1873	17.4931	30.7225	34.6686
	max	13.5149	21.2943	21.3519	33.9135	37.3119
8	min	11.6470	19.6611	19.3505	32.4074	36.1168
	max	13.0550	20.8753	20.9182	33.5702	37.0404
10	min	12.3005	20.2608	20.1357	33.0189	36.6142
	max	12.8723	20.7222	20.7437	33.4426	36.9415
12	min	12.5779	20.4931	20.4446	33.2401	36.7879
	max	12.8015	20.6660	20.6755	33.3950	36.9052
14	min	12.6890	20.5810	20.5625	33.3205	36.8495
	max	12.7746	20.6453	20.6492	33.3773	36.8919
16	min	12.7321	20.6139	20.6068	33.3449	36.8716
	max	12.7644	20.6377	20.6393	33.3708	36.8871
18	min	12.7484	20.6261	20.6234	33.3606	36.8796
	max	12.7606	20.6349	20.6355	33.3683	36.8853
20	min	12.7546	20.6306	20.6295	33.3649	36.8825
	max	12.7592	20.6339	20.6341	33.3675	36.8846
22	min	12.7569	20.6322	20.6318	33.3660	36.8836

Table 7: Results of The Block Jacobi Method for Example 2

	(x,y)	0.1, 0.1	0.2, 0.2	0.3, 0.3	0.4, 0.4	0.5, 0.5	Iteration /error
M= 10	max	0.0357	0.1291	0.2447	0.3381	0.3738	60
P=10	min	0.0355	0.1285	0.2434	0.3364	0.3719	1.09E-2
M=20	max	0.0353	0.1279	0.2422	0.3348	0.3701	195
P=20	min	0.0352	0.1275	0.2416	0.3338	0.3691	3.30E-3
M=40	max	0.0352	0.1274	0.2414	0.3336	0.3688	457
P=40	min	0.0352	0.1272	0.2410	0.3331	0.3683	1.10E-3
	true	0.0351	0.1271	0.2408	0.3328	0.3679	

Table 8: Results of The Block Gauss-Seidel Method for Example 2

	(x,y)	0.2, 0.2	0.2, 0.4	0.4, 0.2	0.4, 0.4	0.5, 0.5	Iteration /error
M= 10	max	0.0358	0.1294	0.2449	0.3382	0.3737	31
P=10	min	0.0355	0.1285	0.2434	0.3364	0.3719	1.09E-2
M=20	max	0.0354	0.1279	0.2423	0.3348	0.3701	99
P=20	min	0.0352	0.1275	0.2416	0.3338	0.3691	3.30E-3
M=40	max	0.0352	0.1274	0.2414	0.3336	0.3688	290
P=40	min	0.0352	0.1272	0.2410	0.3331	0.3683	1.10E-3
	true	0.0351	0.1271	0.2408	0.3328	0.3679	

Table 9: Solutions by Using The Jacobi Method for Example 2

	(x,y)	0.1, 0.1	0.2, 0.2	0.3, 0.3	0.4, 0.4	0.5, 0.5
	max	0.0791	0.2862	0.5422	0.7493	0.8285
t=0.2	min	0.0789	0.2855	0.5410	0.7476	0.8265
	max	0.0717	0.2593	0.4912	0.6789	0.7505
t=0.3	min	0.0715	0.2586	0.4899	0.6771	0.7486
	max	0.0649	0.2348	0.4448	0.6146	0.6795
t=0.4	min	0.0647	0.2341	0.4435	0.6129	0.6776
	max	0.0587	0.2125	0.4026	0.5564	0.6151
t=0.5	min	0.0585	0.2118	0.4013	0.5546	0.6132
	max	0.0532	0.1932	0.3644	0.5036	0.5568
t=0.6	min	0.0530	0.1917	0.3631	0.5018	0.5548
	max	0.0481	0.1741	0.3298	0.4558	0.5040
t=0.7	min	0.0479	0.1734	0.3286	0.4541	0.5020
	max	0.0436	0.1576	0.2986	0.4126	0.4562
t=0.8	min	0.0434	0.1569	0.2973	0.4109	0.4542
	max	0.0394	0.1427	0.2703	0.3735	0.4129
t=0.9	min	0.0392	0.1420	0.2690	0.3718	0.4110
	max	0.0357	0.1291	0.2447	0.3381	0.3738
t=1	min	0.0355	0.1285	0.2434	0.3364	0.3719

Table 10: Solutions by Using The Gauss-Seidel Method for Example 2

	(x,y)	0.1, 0.1	0.2, 0.2	0.3, 0.3	0.4, 0.4	0.5, 0.5
	max	0.0792	0.2864	0.5425	0.7495	0.8284
t=0.2	min	0.0789	0.2855	0.5409	0.7476	0.8265
	max	0.0718	0.2595	0.4915	0.6790	0.7505
t=0.3	min	0.0715	0.2586	0.4899	0.6771	0.7486
	max	0.0650	0.2350	0.4450	0.6148	0.6795
t=0.4	min	0.0647	0.2341	0.4434	0.6129	0.6776
	max	0.0588	0.2127	0.4028	0.5565	0.6150
t=0.5	min	0.0585	0.2118	0.4013	0.5546	0.6132
	max	0.0533	0.1926	0.3647	0.5037	0.5567
t=0.6	min	0.0530	0.1917	0.3631	0.5018	0.5548
	max	0.0482	0.1743	0.3301	0.4560	0.5039
t=0.7	min	0.0479	0.1734	0.3286	0.4541	0.5020
	max	0.0436	0.1578	0.2988	0.4127	0.4561
t=0.8	min	0.0434	0.1569	0.2973	0.4109	0.4543
	max	0.0395	0.1429	0.2705	0.3736	0.4128
t=0.9	min	0.0392	0.1420	0.2690	0.3718	0.4110
	max	0.0358	0.1294	0.2449	0.3382	0.3737
t=1	min	0.0355	0.1285	0.2434	0.3364	0.3719

Table 11: Jacobi Iterations for Example 2 When $t = 1$

Iteration	(x,y)	0.1, 0.1	0.2, 0.1	0.3, 0.3	0.4, 0.4	0.5, 0.5
	max	0.42173	1.33959	2.44566	3.03826	3.39981
6	min	0.02085	0.07539	0.142688	0.19704	0.21778
	max	0.17016	0.59088	1.14583	1.52015	1.73049
12	min	0.02943	0.10647	0.20164	0.27859	0.30797
	max	0.09027	0.31931	0.61882	0.83587	0.94511
18	min	0.03299	0.11936	0.22608	0.31241	0.34538
	max	0.05824	0.20810	0.39979	0.54555	0.61125
24	min	0.03446	0.12470	0.23623	0.32645	0.36091
	max	0.04499	0.16179	0.30864	0.42392	0.47177
30	min	0.03508	0.12692	0.24044	0.33229	0.36737
	max	0.03948	0.14248	0.27075	0.37318	0.41377
36	min	0.03533	0.12784	0.24219	0.33471	0.37005
	max	0.03719	0.13444	0.25502	0.35206	0.38968
42	min	0.03544	0.12823	0.24292	0.33571	0.37116
	max	0.03625	0.13110	0.24849	0.34326	0.37968
48	min	0.03548	0.12838	0.24322	0.33613	0.37162
	max	0.03585	0.12971	0.24578	0.33961	0.37553
54	min	0.03550	0.128455	0.24335	0.33631	0.37181

Table 12: Gauss-Seidel Iterations for Example 2 When $t = 1$

Iteration	(x,y)	0.1, 0.1	0.2, 0.1	0.3, 0.3	0.4, 0.4	0.5, 0.5
	max	0.59801	1.63046	2.51533	3.10614	3.40204
3	min	0.01483	0.05895	0.11963	0.17486	0.20353
	max	0.24148	0.78721	1.39098	1.82822	1.92930
6	min	0.02519	0.09478	0.18551	0.26405	0.30020
	max	0.13375	0.44808	0.78948	1.01598	1.04488
9	min	0.030755	0.11329	0.21800	0.30547	0.34193
	max	0.08067	0.27257	0.48424	0.62945	0.65615
12	min	0.03344	0.12194	0.23260	0.32337	0.35940
	max	0.05523	0.19080	0.34653	0.46071	0.49153
15	min	0.03463	0.12573	0.23887	0.33095	0.36670
	max	0.043945	0.155036	0.28718	0.38898	0.42238
18	min	0.03514	0.12733	0.24152	0.33414	0.36976
	max	0.03910	0.13980	0.26206	0.35879	0.39341
21	min	0.03536	0.12801	0.24264	0.33547	0.37103
	max	0.03706	0.13338	0.25150	0.34613	0.38128
24	min	0.03545	0.12829	0.24310	0.33603	0.37157
	max	0.03620	0.13069	0.24707	0.34083	0.37621
27	min	0.03548	0.12841	0.24330	0.33626	0.37179

Table 13: Results of The Block Jacobi Method for Example 3 When t=1

	(x,y)	0, 0	0.25, 0.25	0.5, 0.5	0.75, 0.25	1, 0	Iteration /error
M= 8	max	0.3691	0.1819	-0.0053	-0.1926	-0.3799	
P=10	min	0.3663	0.1791	-0.0082	-0.1956	-0.3830	2.61E-2
M=20	max	0.3685	0.1838	-0.0008	-0.1854	-0.3701	
P=20	min	0.3672	0.1826	-0.0021	-0.1867	-0.3714	4.10E-4
M=40	max	0.3690	0.1848	0.0007	-0.1835	-0.3677	
P=40	min	0.3667	0.1825	-0.0017	-0.1859	-0.3702	3.6E-4
	true	0.3679	0.1839	0	-0.1839	-0.3679	

Table 14: Results of The Block Gauss-Seidel Method for Example 3 When t=1

	(x,y)	0, 0	0.25, 0.25	0.5, 0.5	0.75, 0.25	1, 0	Iteration /error
M= 8	max	0.3690	0.1818	-0.0054	-0.1927	-0.3800	
P=10	min	0.3664	0.1792	-0.0080	-0.1954	-0.3828	2.67E-2
M=20	max	0.3685	0.1838	-0.0008	-0.1855	-0.3701	
P=20	min	0.3673	0.1826	-0.0020	-0.1867	-0.3714	4.20E-4
M=40	max	0.3682	0.1840	-0.0002	-0.1843	-0.3685	
P=40	min	0.3676	0.1834	-0.0008	-0.1850	-0.3692	1.3E-4
	true	0.3679	0.1839	0	-0.1839	-0.3679	

Table 15: Results of The Jacobi and The Gauss-Seidel Method for Example 4 When t=1

	(x,y)	0.1, 0.1	0.2, 0.2	0.3, 0.3	0.4, 0.4	0.5, 0.5	Iteration /error
M= 10	J	3.5517	12.8503	24.3440	33.6425	37.1943	56/1.1E-2
P=10	G	3.5522	12.8526	24.3491	33.6507	37.2042	31/1.13E-2
M=20	J	3.5216	12.7413	24.1374	33.3571	36.8787	154/2.5E-3
P=20	G	3.5230	12.7470	24.1492	33.3747	36.8995	85/3.0E-3
M=40	J	3.5095	12.6976	24.0547	33.2428	36.7524	390/9.66E-4
P=40	G	3.5128	12.7101	24.0795	33.2787	36.7938	215/1.56E-4
	true	3.5129	12.7099	24.0780	33.2750	36.7879	

Table 16: Comparison of the Number of Iteration of All Examples

	Method	Ex. 1	Ex. 2	Ex. 3	Ex. 4
M=10	J	62	66	208	56
P=10	G	38	31	118	31
M=20	J	140	195	399	154
P=20	G	86	99	213	85
M=40	J	410	457	814	390
P=40	G	221	290	420	215

Table 17: Comparison of Error of All Examples

	Method	Ex. 1	Ex. 2	Ex. 3	Ex. 4
M=10	J	3.0E-3	1.09E-2	2.61E-2	1.1E-2
P=10	G	3.0E-3	1.09E-2	2.67E-2	1.13E-2
M=20	J	9.3E-4	3.3E-3	4.1E-4	2.5E-3
P=20	G	9.36E-3	3.3E-3	4.2E-4	3.0E-3
M=40	J	3.21E-4	1.1E-3	1.6E-4	9.66E-4
P=40	G	3.21E-4	1.1E-3	1.3E-4	1.56E-4

6 Discussions

6.1 Conclusions

Based on the numerical results from the four examples, we have following observations and comments:

1. *Monotone and convergence property*

   In Tab(5), Tab(6), Tab(11) and Tab(12), it is shown that the for a fixed time the upper iterative sequence starts from the upper solution and decreases to true solution monotonically, and the lower iterative sequence starts form the lower solution and increasingly converges to the unique solution. Actually, this monotone convergence property holds for every mesh point and any time t, no matter how the grid size and time interval are chosen. In all tables, it is also shown that the upper sequence and the lower sequence are ordered, which are cogent to the Lemma 3.1 and Lemma 4.1. Obviously the way the finite system is formatted affects the computation accuracy. The numerical results are consistent with theoretical properties given in Thm 3.1 and Thm 4.1.

2. *Times of iteration*

   The times of iteration depends on how far the initial upper solution and lower solution are away from the true solution. Suppose the discrete domain has $M \times N$ points and P points for time it needs n steps to get the solution satisfying given threshold. If we double the points in each dimension it needs $3n$ steps approximately to get the same threshold.

3. *Computation efficiency and comparison between two methods*

   In each iteration, unlike the traditional point wise method solving the finite difference system directly, the block Jacobi and Gauss-Seidel method solves it block by block. Each block is a tridiagonal system representing equations on one row(x-direction). So it can be solved by the fastest method, the Thomas Algorithm(See [5]). By using this algorithm a $M \times M$ block can be solved with about $3M$ operations. To solve the whole system(N blocks), it only needs $3MN$ operations. Comparing to M^2N^2 operations needed for the point wise method the advantage of block methods is obvious.

4. *Comparison between two methods*

   From Tab(1), Tab(2), Tab(7), Tab(8) and Tab(16) we see that starting from the same upper solution and lower solution the number of iteration of Gauss-Seidel method is dramatically (about 50%) less than that of Jacobi method. This is because Gauss-Seidel method uses previously computed results as soon as they are available.

5. *Error analysis*

   The error comes from two parts, one from the discrete finite system and another from the round-off errors. The errors are reduced by choosing the smaller mesh size and the shorter time interval as shown in Tab(1), Tab(2), Tab(7) and Tab(8).

6. *Effect of boundary conditions*

   Example 3 has the Neumann type boundary condition that makes the problem more complicate. It requires more iterations.

7. *Solving for linear problems*

   It is commonly known that solving nonlinear problems needs more work than linear ones. But with this method, comparing the different columns in Tab(16) and Tab(17), it shows that the costs of solving nonlinear and linear systems are at the same level. This is a major advantage of the block monotone method.

6.2 Future Studies

1. The convergent rates of both block iterative schemes need to be investigated theoretically. Specially, it is important to relate the convergent rates between a linear problem and a nonlinear problem.

   The block monotone method can only solve the problem on a convex domain. This paper only gives the examples on the rectangle box. Basically, a problem with irregular convex shapes can be dealt with the same way but there are some issues of how and where to choose grid lines to get the best approximation.

2. How can we extend the block monotone method to three or higher dimensional space?

3. The relationship among the mesh size, length of time interval and number of iterations that discussed in above section 6.1.3 is only concluded from observation. The more detailed theoretical analysis and quantitative numerical computational will be helpful of showing the efficiency of block monotone methods.

REFERENCES

[1] C.V.Pao, *Nonlinear Parabolic and Elliptic Equations*, Plenum Press, New York and London, 1992.

[2] C.V.Pao, "Numerical Methods For Semilinear Parabolic Equations", *Social for Industrial and Applied Mathematics*, V 24, #1, 1987.

[3] C.V.Pao, "Block Monotone Iterative Methods for Numerical Solutions of Nonlinear Elliptic Equations", *Numer. Math.*, V72 239-262, 1995

[4] Xin Lu, "Monotone method and convergence acceleration for finite-difference solutions of parabolic problems with time delays", *Numerical methods for partial differential equations*, V11(6), 591-602, 1995

[5] G.D.Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Method*, 2nd Edition, Claredon Press, Oxford, 1985.

[6] David Bleecker & George Csordas, Basic Partial Differential Equations, *American Mathematical Society*, 1997.

[7] R. Barrett and M. Berry and T. F. Chan and J. Demmel and J. Donato and J. Dongarra and V. Eijkhout and R. Pozo and C. Romine and H. Van der Vorst "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods", 2nd Edition, *SIAM*, 1994.

[8] R. G. Bartle & D. R. Sherbert, *Introduction to Real Analysis, 3rd Edition*, 3rd Edition, John Wiley & Sons, 1999.

[9] R. S. Varga *Matrix Iterative Analysis*, Prentice-Hall Englewood Cliffs, NJ 1962.