

# 國立交通大學

應用數學系

數學建模與科學計算碩士班



不均勻擴散問題之 A. D. I. 法的研究

A Numerical Study of A.D.I. Methods  
to Anisotropic Diffusion Problems

研 究 生：陳昱丞

指 導 教 授：賴明治 教授

中 華 民 國 九 十 八 年 九 月

# 不均勻擴散問題之 A. D. I. 法的研究

## A Numerical Study of A.D.I. Methods to Anisotropic Diffusion Problems

研究生：陳昱丞

Student : Yu-Chen Chen

指導教授：賴明治

Advisor : Ming-Chih Lai



Submitted to Department of Applied Mathematics College of Science,  
Institute of Mathematical Modeling and Scientific Computing,  
National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Applied Mathematics

September 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年九月

## 誌 謝

打從大學開始直到研究所畢業，這是一條漫長求學的路程。這其中或多或少有經歷些困難與瓶頸甚至迷惘。而今能夠如願順利取得學位，首先要感謝的是我的父母親給予的支持以及鼓勵。感謝他們耐心等待我達到自己期待的成就，並且適時給出意見和提醒，陪我朝著自己的目標前進。同時也感謝弟妹的加油打氣，我也祝福他們能夠跟我一樣有順利的求學過程。

再者感謝賴明治教授，不厭其煩的在做研究的路途上給我指引方向。無論是討論中或在課堂裡，他都期待著學生培養起洞察問題的能力，勉勵我們要多花心思去想問題，徹底了解問題之後，積極的解決問題。老師總是要求我們要把不懂的事情說明白，其實背後隱藏著很多功夫要我們自己去鍛鍊。老師的一番話我銘記在心，我會嘗試並努力實行。

很慶幸有一個溫馨的研究團隊。感謝同窗哲維、義閔、永潔、露結，在課堂上的照顧，並肩學習。我會記得一起討論作業，腦力激盪的那一刻。感謝同研究室的夥伴昆霖、裕昇、州哥、秉恆，有你們在的研究室是歡樂輕鬆的，讓我在緊湊的研究生生活中有緩衝的時候。我會記得一起討論金融風暴，看足球，討論種植物的專屬話題。另外感謝組合組與分析組的阿華田、油油、天兵、松育、士慶、慧茶、彥寧、新同學、文昱、瑞毅、平日的照顧。感謝同門師兄小豪、仲尹、鈺傑、先皓、麻將，在論文撰寫還有口試期間的鼎力相助，幫我打氣產生不少信心。尤其感謝慧茶同學與仲尹學長在論文撰寫期間的幫忙跟贊助，其繁瑣的校稿訂正與撰寫，真是辛苦你們了。

在口試準備期間，感謝張書銘教授、黃聰明教授，與賴明治教授的百忙之中的指點，審校論文，讓我的畢業論文更加完整嚴謹。也感謝您們在口試期間給予的指正跟批評，以及對我的肯定。

另外要感謝系上傳恆霖教授、陳秋媛教授對我的關心，讓我在求學過程中充滿了鼓勵。最後感謝棒球隊教練黃杉盈教練七年來的指教，除了教會我棒球之外，也教我許多看事情的觀點，待人處世的方法。球場是我釋放壓力的地方，也是另一個獲得成就的來源，感謝教過我打球的邱毛、浩呆、小吉，陳教練，李教練、鄭教練，謝謝你們讓我學到更多。還有辛苦的學弟，替我做雜事，撿球抬水搬東西整理球場樣樣來。感謝一起拼鬥的隊友：范銘隆、拉紐，跟我一起像瘋子一樣扛起了一段交大乙組棒球的江山。

# 不均勻擴散問題之 A. D. I. 法的研究

學生：陳昱丞

指導教授：賴明治 教授

國立交通大學應用數學系數學建模與科學計算碩士班

## 摘要

微觀世界中存在著不均勻擴散現象，此現象在科學應用領域中占有重要地位。本論文應用 A. D. I. 法研究不均勻擴散方程式，我們除了討論 A. D. I. 法的收斂速度以及無條件穩定性之外，並且將其數值結果與疊代法所得數值結果互相比較。由於不均勻擴散方程式中的擴散係數可分為常係數與變係數的形式，因此離散之後所得的線性系統亦可分為常係數與變係數的型態。我們利用疊代法中常用的 C. G. 法與 B. I. C. G. 法分別處理。基於特殊的線性系統結構，A. D. I. 法在計算速度上遠勝於疊代法。

# A Numerical Study of A.D.I. Methods to Anisotropic Diffusion Problems

Student : Yu-Chen Chen      Advisor : Ming-Chih Lai

Institute of Mathematical Modeling and Scientific Computing,  
National Chiao Tung University,  
1001 Ta Hsueh Road, Hsinchu 30050,  
Taiwan.

## Abstract

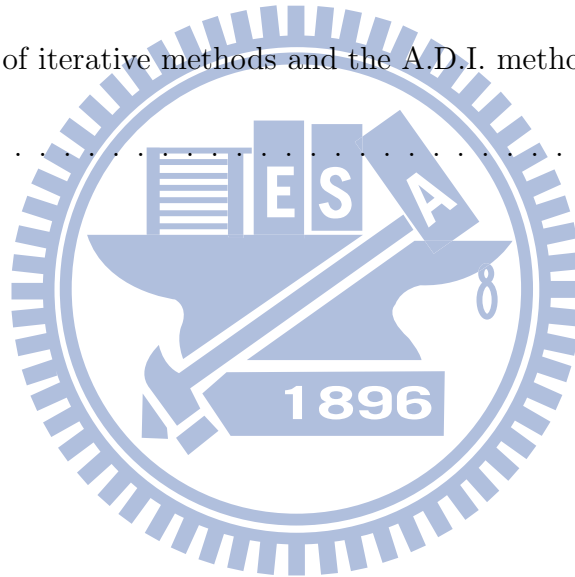
Anisotropic diffusion is a kind of microscopic phenomenon. It plays an important role in a lot of scientific applications. We use A.D.I. method which is efficient to solve anisotropic diffusion problems. We study the order of accuracy and unconditionally stable convergence of the method, and compare it with preconditioned iterative methods. Since diffusivity of anisotropic diffusion equations can be constant and variable type. We choose conjugate gradient method to deal with the constant type equation and biconjugate gradient method to solve the general type. Because of the special structure of linear systems, A.D.I. method outperforms iterative methods of CPU time.

**Keywords:** A.D.I. method; Conjugate gradient method; Biconjugate gradient method; dirichlet boundary condition; anisotropic diffusion problems; precondition; iterative methods

# Contents

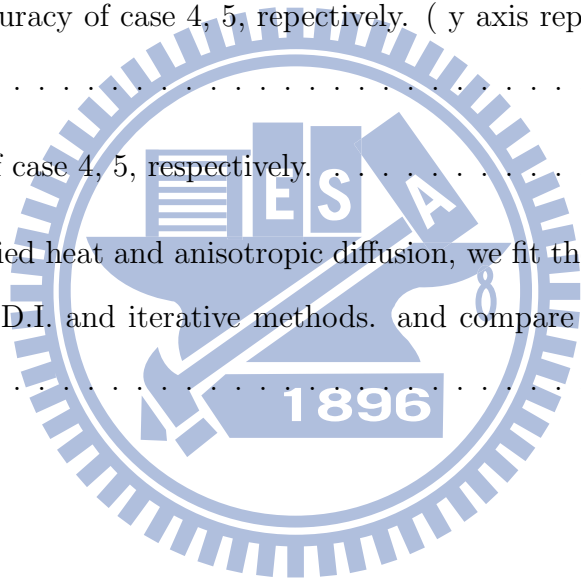
Abstract (in Chinese)	i
Abstract (in English)	ii
Contents	iii
List of Figures	v
<b>1 Introduction</b>	<b>1</b>
<b>2 The Alternating Direction Implicit method (A.D.I.)</b>	<b>3</b>
2.1 The anisotropic diffusion problems . . . . .	3
2.2 Discretization of A.D.I. method . . . . .	4
2.3 Stability for A.D.I. method . . . . .	7
2.3.1 von Neumann analysis for the anisotropic diffusion problems .	8
2.3.2 von Neumann analysis for general anisotropic diffusion problems	10
2.4 Apply A.D.I. method to anisotropic diffusion problems . . . . .	12
<b>3 Conjugate gradient method for anisotropic diffusion problem</b>	<b>15</b>
3.1 Steepest Descent method . . . . .	15
3.2 Introduction of conjugate gradient method (C.G. method) . . . . .	19

3.3	Implementation of the conjugate gradient method for anisotropic diffusion problems . . . . .	22
3.3.1	Discretization by Crank-Nicolson scheme . . . . .	22
3.3.2	The biconjugate gradient method . . . . .	26
3.3.3	The preconditioned conjugate gradient method . . . . .	27
3.3.4	The preconditioned biconjugate gradient method . . . . .	28
<b>4</b>	<b>Numerical results</b>	<b>30</b>
4.1	Numerical results for heat equations . . . . .	30
4.2	Comparison of iterative methods and the A.D.I. method . . . . .	32
4.3	Conclusion . . . . .	36
	<b>Reference</b>	<b>38</b>



# List of Figures

1	The quadratic form with a positive definite matrix has minimal extreme value. . . . .	16
2	We use regular uniform domain and collect the columns to be a new column. . . . .	24
3	Comparison of order of accuracy for case 1, 2, 3, respectively. (y-axis is order of accuracy, x-axis is number of grid point) . . . . .	33
4	CPU time of case 1, 2, 3, respectively. . . . .	34
5	Order of accuracy of case 4, 5, respectively. ( y axis represent order of accuracy) . . . . .	35
6	CPU time of case 4, 5, respectively. . . . .	36
7	A case satisfied heat and anisotropic diffusion, we fit the curve of data solved by A.D.I. and iterative methods. and compare with the CPU time. . . . .	37





# 1 Introduction

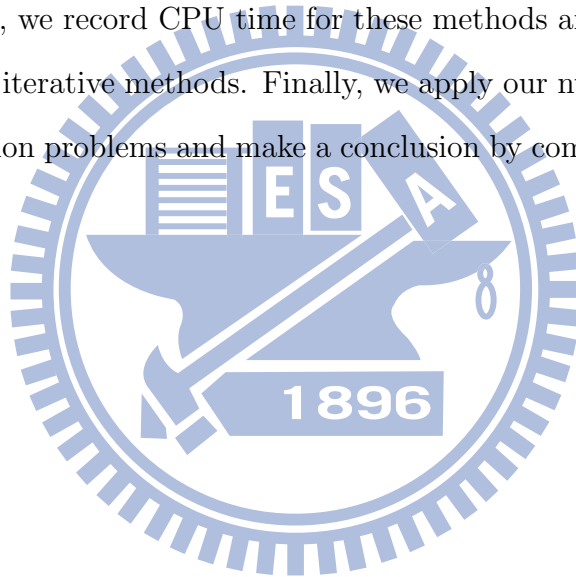
As it known, the anisotropic diffusion [5] is a kind of microscopic phenomenon in the random thermal motion on a microscopic scale. A lot of applications of anisotropic diffusion in physics, chemistry, biotechnology, and engineering. It is a net transport of particles from higher concentration to lower concentration. Because of anisotropic diffusion, the material mix gradually. The mixing process which can be described by Fick's law will attempt to the equilibrium state. By solving the anisotropic diffusion problems could help us to simulate the whole behavior of diffusion.

In section 2, we introduce the alternating direction implicit (A.D.I.) method [9] which is known as the finite-difference implicit-type algorithm. A.D.I. method has the advantage of ensuring a more efficient formulation and calculation than other implicit methods in the case of multidimensional problems. We discretize the anisotropic diffusion equation by Crank-Nicolson scheme and split the finite difference time-domain problem by Douglas-Rachford method which is a kind of A.D.I. method. By using Taylor series, we discuss the order of accuracy of the A.D.I. method which is first order in time and second order in space. Next, we illustrate that the two level finite-difference method is unconditionally stable.

Next, we introduce some direct methods solving large sparse linear system of equations. Direct methods take too much time and limited memory in computing process. In section 3, the conjugate gradient method [7] can be regarded as the iterative method to solve the anisotropic diffusion problems. When the diffusivity is constant type and the matrix of linear system is symmetric positive definite, the problem will be dealt by conjugate gradient (C.G.) method. If the diffusivity is variable type, we employ bi-conjugate gradient (B.I.C.G.) [12] method. Both B.I.C.G.

and C.G. are iterative methods, but the first one does not take advantage of the symmetric property, it requires the transpose of the original matrix. In other words, B.I.C.G. is more general than C.G. method. Generally speaking, one employs iterative methods to include preconditioning, if the matrix is ill-conditioned. C.G. and B.I.C.G. is highly susceptible to rounding errors. We use incomplete LU factorization [15] which is often used as a preconditioner, especially for large sparse matrix.

The last section contains numerical results about implementing of A.D.I. method, C.G. method and biconjugate gradient method. Firstly, we consider a heat equation which is a simpler problem than the anisotropic diffusion problem. The numerical results show that A.D.I. , C.G. and B.I.C.G methods converge in the second order of accuracy. In addition, we record CPU time for these methods and notice that A.D.I. is more efficient than iterative methods. Finally, we apply our numerical methods for the anisotropic diffusion problems and make a conclusion by comparison of numerical results.



## 2 The Alternating Direction Implicit method (A.D.I.)

In section 2, firstly, we describe a standard problem of the anisotropic diffusion problem. Secondly, we introduce an efficient and simple numerical method for solving parabolic equations, especially on regular domains. By using Taylor series, we show that the scheme is consistent with the partial differential equation. The accuracy is first order in time and second order in space. Thirdly, we illustrate the implementation of A.D.I. method. We will demonstrate some numerical results in the last section, and give a concise conclusion.

### 2.1 The anisotropic diffusion problems

Anisotropic diffusion problems arise in widespread range of scientific fields such as oil-reservoir simulation, plasma physics, image processing, semiconductor modeling, biology, and hydro-geology etc. Numerical simulation plays an important role in wild applications. When implement various numerical methods on this type of problem, one needs to find an approximation of  $u$ , which is the solution of the following two dimensional anisotropic diffusion equation. [9]

$$\frac{\partial u}{\partial t} = \nabla \cdot (\beta \nabla u) \text{ in } \Omega \times (0, T] \quad (2.1)$$

with the initial condition

$$u(x, y, 0) = u_0(x, y), (x, y) \in \Omega$$

and the dirichlet boundary condition

$$u(x, y, t) = g(x, y, t), (x, y) \in \partial\Omega, t \in (0, T],$$

where  $\beta = \begin{pmatrix} \tilde{a}(x, y) & \tilde{b}(x, y) \\ \tilde{b}(x, y) & \tilde{c}(x, y) \end{pmatrix}$  is a  $2 \times 2$  symmetric coefficient matrix, subject to  $\tilde{a} > 0$ ,  $\tilde{c} > 0$ , and  $\tilde{b}^2 - \tilde{a}\tilde{c} < 0$ .

## 2.2 Discretization of A.D.I. method

As noted before, the anisotropic diffusion problems are difficult and expensive when solved by all kinds of direct methods or iterative methods. In other words, they may cost much time and memory at each time step. If we want to solve the problems efficient, the expensive method may not suitable. A.D.I. method is an approach which reduce two-dimensional problem to a succession of a system of one-dimensional problems. Now, we start our work to extend Eqn.(2.1) as follows.

$$\frac{\partial u}{\partial t} = \nabla \cdot (\beta \nabla u) = \nabla \cdot \left( \begin{pmatrix} \tilde{a} & \tilde{b} \\ \tilde{b} & \tilde{c} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \right) = \nabla \cdot \begin{pmatrix} \tilde{a}u_x + \tilde{b}u_y \\ \tilde{b}u_x + \tilde{c}u_y \end{pmatrix}.$$

Obviously, we extend the above equation which contains four terms.

$$\frac{\partial u}{\partial t} = au_{xx} + bu_{xy} + bu_{yx} + cu_{yy}.$$

Notice that the right-hand side includes  $\nabla^2 u$ , and two mixed terms  $u_{xy}$  and  $u_{yx}$ . In order to simplify the equation, we assume  $u \in \mathcal{C}^2$ , then we have  $u_{xy} = u_{yx}$ . Thus we obtain.

$$u_t = au_{xx} + 2bu_{xy} + cu_{yy}. \quad (2.2)$$

In the beginning of discretization, we omit the mixed derivative term to reduce the difficulties of designing the scheme of equation. Therefore we consider heat equation first and look for a higher order of accuracy approximation of time derivative. Crank-Nicolson scheme is one kind of traditional scheme which has high order of accuracy as we want. We start with the formula for  $u_t$  evaluated at  $t_{k+1/2}$ . By using Taylor series, we obtain two equations (2.3) and (2.4).

$$u^{n+1} = u^{n+1/2} + \frac{k}{2}u_t + \frac{k^2}{4}u_{tt} + O(k^3), \quad (2.3)$$

$$u^n = u_{n+1/2} - \frac{k}{2}u_t + \frac{k^2}{4}u_{tt} + O(k^3). \quad (2.4)$$

We combine equations (2.3) and (2.4) such that the second derivative of time vanished. Therefore we have a second order accuracy scheme (2.5) and we will use it to approximate the time derivative.

$$u_t = \frac{u^{n+1} - u^n}{k} + O(k^2). \quad (2.5)$$

We undertake to approximate the derivatives respect to space. By using Taylor series, we have

$$u_{i+1} = u_i + hu_x + h^2u_{xx} + h^3u_{xxx} + O(h^4), \quad (2.6)$$

$$u_{i-1} = u_i - hu_x + h^2u_{xx} - h^3u_{xxx} + O(h^4). \quad (2.7)$$

Take summation of eqn.(2.6) and (2.7), the first derivative in space can be vanished, and let the summation divided by  $h^2$  immediately. We product eqn.(2.8) which converges in the second order accuracy. Base on the same techniques, we derive eqn.(2.9).

$$u_{xx} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2), \quad (2.8)$$

$$u_{yy} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + O(h^2). \quad (2.9)$$

Next, we will derive the finite difference scheme for the first derivative respect to space in  $x$  and  $y$  directions. We subtract (2.7) from (2.6) to keep the term  $u_x$ , and divide by  $2h$  immediately, and we obtain eqn. (2.8).

Similarly, we produce the approximation for first order derivative in  $y$  direction. To simplify, we introduce some notations.

$$u_x = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + O(h^2), \quad (2.10)$$

$$u_y = \frac{u_{i,j+1} - u_{i,j-1}}{2h} + O(h^2). \quad (2.11)$$

Let  $\delta_x^2$ ,  $\delta_y^2$ ,  $H_x$ , and  $H_y$  be linear operators [10]. For convenience, we define

$$\begin{aligned}\delta_x^2 u &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, \\ \delta_y^2 u &= \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}, \\ H_x u &= \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \\ H_y u &= \frac{u_{i,j+1} - u_{i,j-1}}{2h}.\end{aligned}$$

For the approximation in space, we employ the same ideas which has been used in the Crank-Nicolson scheme at  $t_{k+1/2}$ . Thus we obtain the equation below.

$$\begin{aligned}u^{n+1} - u^n &= \frac{ka}{2}(\delta_x^2 u^{n+1} + \delta_x^2 u^n) + \frac{kc}{2}(\delta_y^2 u^{n+1} + \delta_y^2 u^n) + O(k^3 + kh^2), \\ (1+a)(1+b) &= 1 + a + b + ab.\end{aligned}\tag{2.12}$$

It is quiet difficult to solve the block tri-diagonal linear system by direct methods. However, base on formula (2.12), we add  $k^2 ac \delta_x^2 \delta_y^2 u^{n+1}/4$  to both side of above equation and re-write it as follows.

$$\left(1 - \frac{ka}{2}\delta_x^2\right)\left(1 - \frac{kc}{2}\delta_y^2\right)u^{n+1} = \left(1 + \frac{ka}{2}\delta_x^2\right)\left(1 + \frac{kc}{2}\delta_y^2\right)u^n + \frac{k^2 ac}{4}\delta_x^2 \delta_y^2 (u^{n+1} - u^n) + O(k^3 + kh^2).\tag{2.13}$$

Since  $u^{n+1} - u^n = ku_t + O(k^3)$  and the  $k^2$  factor, the second term of eqn.(2.13) is higher order term. We can omit the term without effecting the order of accuracy of original numerical scheme. From now on, we have already illustrated the detail of discretization of heat equation.

$$\left(1 - \frac{ka}{2}\delta_x^2\right)\left(1 - \frac{kc}{2}\delta_y^2\right)u^{n+1} = \left(1 + \frac{ka}{2}\delta_x^2\right)\left(1 + \frac{kc}{2}\delta_y^2\right)u^n + O(k^3 + kh^2).\tag{2.14}$$

Now, we try to deal with the mixed derivative. According to the Douglas-Rachford method [9], we put the mixed term on right-hand side of eqn. (2.14). In other words,

we treat it as the information that we have already known.

$$\begin{aligned}
u_{xy} &= \frac{1}{2h} \left[ \frac{u_{i+1,j+1} - u_{i+1,j-1}}{2h} - \frac{u_{i-1,j+1} + u_{i-1,j-1}}{2h} \right] + O(h), \\
&= \frac{1}{4h^2} [u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}] + O(h), \\
&= H_x H_y u + O(h).
\end{aligned}$$

We insert the mixing derivative term into eqn. (2.14), then we have.

$$\left(1 - \frac{ka}{2}\delta_x^2\right)\left(1 - \frac{kc}{2}\delta_y^2\right)u^{n+1} = \left(1 + \frac{ka}{2}\delta_x^2\right)\left(1 + \frac{kc}{2}\delta_y^2\right)u^n + 2bkH_xH_yu^n + O(k^3 + kh^2 + kh). \quad (2.15)$$

By the way, the scheme is consistent with the anisotropic diffusion equation. The order of accuracy is first order in time and second order in space.

We split above equation into two level time-domain scheme by Douglas-Rachford method.

$$\left(1 - \frac{1}{2}ka\delta_x^2\right)u_{i,j}^* = \left(1 + \frac{1}{2}ka\delta_x^2 + kc\delta_y^2 + 2kbH_xH_y\right)u_{i,j}^n, \quad (2.16)$$

$$\left(1 - \frac{1}{2}kc\delta_y^2\right)u_{i,j}^{n+1} = u_{i,j}^* - \frac{1}{2}kc\delta_y^2u_{i,j}^n. \quad (2.17)$$

Notice that each step requires the value  $u_{i,j}^*$ . We should regard  $u_{i,j}^*$  as intermediate or temporary value. A.D.I. method deal with intermediate values by boundary condition [10]. The problem about temporary values will illustrate in implementation of A.D.I method.

### 2.3 Stability for A.D.I. method

The Douglas-Rachford method is unconditionally stable. It can be easily checked by von Neumann analysis for two dimensional problems. Since von Neumann analysis can not deal with variable coefficient case, we show the variable type as well as constant type referring to [13].

### 2.3.1 von Neumann analysis for the anisotropic diffusion problems

Before we study the stability of A.D.I. method, we apply von Neumann analysis of finite difference schemes which is used to check the stability of finite difference schemes. By using von Neumann analysis, we can give necessary and sufficient conditions for the stability of A.D.I. method.

We consider simple case whose coefficient matrix is constant type. By using von Neumann analysis for two dimensional equation [9], we replace  $u_{i,j}^n$  by  $\lambda^n e^{im\theta} e^{jn\phi}$ ,  $\theta = \eta h$ ,  $\phi = \xi h$ ,  $\eta$  and  $\xi$  which are arbitrary real numbers,  $0 \leq \theta, \phi \leq \pi$ .

Where  $\lambda$  is amplification factor. We will perform von Neumann analysis for A.D.I method. The method is stable if and only if  $|\lambda| \leq 1$ .

$$\begin{aligned}\delta_x^2 u^n &= \frac{e^{i\theta} - 2 + e^{-i\theta}}{h^2} \lambda^n e^{im\theta} e^{jn\phi} = -\frac{4}{h^2} \sin^2 \frac{\theta}{2} \lambda^n e^{im\theta} e^{jn\phi}, \\ \delta_y^2 u^n &= \frac{e^{j\phi} - 2 + e^{-j\phi}}{h^2} \lambda^n e^{im\theta} e^{jn\phi} = -\frac{4}{h^2} \sin^2 \frac{\phi}{2} \lambda^n e^{im\theta} e^{jn\phi}, \\ H_x H_y u^n &= \frac{(e^{i\theta} e^{j\phi} - e^{i\theta} e^{-j\phi} - e^{-i\theta} e^{j\phi} + e^{-i\theta} e^{-j\phi})}{4h^2} \lambda^n e^{im\theta} e^{jn\phi} = -\frac{1}{h^2} \lambda^n e^{im\theta} e^{jn\phi} \sin \theta \sin \phi.\end{aligned}$$

By using equations above, eqn. (2.15) can be re-written as follows.

$$\left(1 + \frac{2ak}{h^2} \sin^2 \frac{\theta}{2}\right) \left(1 + \frac{2ck}{h^2} \sin^2 \frac{\phi}{2}\right) \lambda = \left[\left(1 - \frac{2ak}{h^2} \sin^2 \frac{\theta}{2}\right) \left(1 - \frac{2ck}{h^2} \sin^2 \frac{\phi}{2}\right) - \frac{2bk}{h^2} \sin \theta \sin \phi\right].$$

Then we have the absolute value of  $\lambda$ .

$$|\lambda| = \frac{\left|(1 - \frac{2ak}{h^2} \sin^2 \frac{\theta}{2}\right) \left(1 - \frac{2ck}{h^2} \sin^2 \frac{\phi}{2}\right) - \frac{2bk}{h^2} \sin \theta \sin \phi}{\left|(1 + \frac{2ak}{h^2} \sin^2 \frac{\theta}{2}\right) \left(1 + \frac{2ck}{h^2} \sin^2 \frac{\phi}{2}\right)}.$$

To simplify, we replaced  $k/h^2$  by  $r$ .

$$|\lambda| = \frac{\left|(1 - 2ar \sin^2 \frac{\theta}{2}\right) \left(1 - 2cr \sin^2 \frac{\phi}{2}\right) - 2br \sin \theta \sin \phi}{\left|(1 + 2r \sin^2 \frac{\theta}{2}\right) \left(1 + 2cr \sin^2 \frac{\phi}{2}\right)}.$$



Next work will discuss the relations between  $a, b, c, \theta, \phi, r$  which make  $|\lambda| \leq 1$ . We multiply denominator on both sides, and take square.

$$|(1 - 2ar \sin^2 \frac{\theta}{2})(1 - 2cr \sin^2 \frac{\phi}{2}) - 2br \sin \theta \sin \phi| \leq |(1 + 2r \sin^2 \frac{\theta}{2})(1 + 2cr \sin^2 \frac{\phi}{2})|,$$

$$|(1 - 2ar \sin^2 \frac{\theta}{2})(1 - 2cr \sin^2 \frac{\phi}{2}) - 2br \sin \theta \sin \phi|^2 \leq |(1 + 2r \sin^2 \frac{\theta}{2})(1 + 2cr \sin^2 \frac{\phi}{2})|^2,$$

We subtract right hand side from left hand side and apply difference of two squares.

$$|(1 - 2ar \sin^2 \frac{\theta}{2})(1 - 2cr \sin^2 \frac{\phi}{2}) - 2br \sin \theta \sin \phi|^2 - |(1 + 2r \sin^2 \frac{\theta}{2})(1 + 2cr \sin^2 \frac{\phi}{2})|^2 \leq 0,$$

$$\begin{aligned} & [(1 - 2ar \sin^2 \frac{\theta}{2})(1 - 2cr \sin^2 \frac{\phi}{2}) - 2br \sin \theta \sin \phi + (1 + 2r \sin^2 \frac{\theta}{2})(1 + 2cr \sin^2 \frac{\phi}{2})] \times \\ & [(1 - 2ar \sin^2 \frac{\theta}{2})(1 - 2cr \sin^2 \frac{\phi}{2}) - 2br \sin \theta \sin \phi - (1 + 2r \sin^2 \frac{\theta}{2})(1 + 2cr \sin^2 \frac{\phi}{2})] \leq 0. \end{aligned}$$

Finally, we obtain a inequality as follows.

$$[1 + 4acr^2 \sin^2 \frac{\theta}{2} \sin^2 \frac{\phi}{2} - br \sin \theta \sin \phi][a \sin^2 \frac{\theta}{2} + c \sin^2 \frac{\phi}{2} + \frac{b}{2} \sin \theta \sin \phi] \geq 0$$

For simplification, we let

$$\lambda_1 = 1 + 4acr^2 \sin^2 \frac{\theta}{2} \sin^2 \frac{\phi}{2} - br \sin \theta \sin \phi, \text{ and } \lambda_2 = a \sin^2 \frac{\theta}{2} + c \sin^2 \frac{\phi}{2} + \frac{b}{2} \sin \theta \sin \phi.$$

Obviously,  $|\lambda| \leq 1$  if and only if  $\lambda_1 \lambda_2 \geq 0$ .

$$\begin{aligned} \lambda_1 &= 1 + 4acr^2 \sin^2 \frac{\theta}{2} \sin^2 \frac{\phi}{2} - br \sin \theta \sin \phi \\ &= 1 + 4acr^2 \sin^2 \frac{\theta}{2} \sin^2 \frac{\phi}{2} - 4br \sin \frac{\theta}{2} \cos \frac{\theta}{2} \sin \frac{\phi}{2} \cos \frac{\phi}{2} \\ &= 1 + [2r\sqrt{ac} \sin \frac{\theta}{2} \sin \frac{\phi}{2} - \frac{b}{\sqrt{ac}} \cos \frac{\theta}{2} \cos \frac{\phi}{2}]^2 - \frac{b^2}{ac} \cos^2 \frac{\theta}{2} \cos^2 \frac{\phi}{2}. \end{aligned}$$

In the above three equations, we use addition formula and complete the square.

$$\lambda_1 = [2r\sqrt{ac} \sin \frac{\theta}{2} \sin \frac{\phi}{2} - \frac{b}{\sqrt{ac}} \cos \frac{\theta}{2} \cos \frac{\phi}{2}]^2 + \frac{ac - b^2}{ac} \cos^2 \frac{\theta}{2} \cos^2 \frac{\phi}{2} + 1 - \cos^2 \frac{\theta}{2} \cos^2 \frac{\phi}{2}$$

By the constraint of anisotropic diffusion equation, we have  $b^2 - ac < 0$ . Thus the second term will be positive.

$$\begin{aligned}
\lambda_1 &= [2r\sqrt{ac}\sin\frac{\theta}{2}\sin\frac{\phi}{2} - \frac{b}{\sqrt{ac}}\cos\frac{\theta}{2}\cos\frac{\phi}{2}]^2 + \frac{ac-b^2}{ac}\cos^2\frac{\theta}{2}\cos^2\frac{\phi}{2} + 1 - \cos^2\frac{\theta}{2}\cos^2\frac{\phi}{2} \\
&= [2r\sqrt{ac}\sin\frac{\theta}{2}\sin\frac{\phi}{2} - \frac{b}{\sqrt{ac}}\cos\frac{\theta}{2}\cos\frac{\phi}{2}]^2 + \frac{ac-b^2}{ac}\cos^2\frac{\theta}{2}\cos^2\frac{\phi}{2} + \sin^2\frac{\theta}{2} + \cos^2\frac{\theta}{2}(1 - \cos^2\frac{\phi}{2}) \\
&= [2r\sqrt{ac}\sin\frac{\theta}{2}\sin\frac{\phi}{2} - \frac{b}{\sqrt{ac}}\cos\frac{\theta}{2}\cos\frac{\phi}{2}]^2 + \frac{ac-b^2}{ac}\cos^2\frac{\theta}{2}\cos^2\frac{\phi}{2} + \sin^2\frac{\theta}{2} + \cos^2\frac{\theta}{2}\sin^2\frac{\phi}{2} \geq 0
\end{aligned}$$

Finally, we find the third and fourth terms of  $\lambda_1$  are both non-negative. Therefore, we have a conclusion that  $\lambda_1$  must be positive. Similarly, we deal with  $\lambda_2$ .

$$\begin{aligned}
\lambda_2 &= a\sin^2\frac{\theta}{2} + c\sin^2\frac{\phi}{2} + \frac{b}{2}\sin\theta\sin\phi \\
&= a\sin^2\frac{\theta}{2} + c\sin^2\frac{\phi}{2} + 2b\sin\frac{\theta}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2}\cos\frac{\phi}{2} \\
&= [\sqrt{a}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \frac{b}{\sqrt{a}}\cos\frac{\theta}{2}\sin\frac{\phi}{2}]^2 + a\sin^2\frac{\theta}{2}\sin^2\frac{\phi}{2} + c\sin^2\frac{\phi}{2} - \frac{b^2}{a}\cos^2\frac{\theta}{2}\sin^2\frac{\phi}{2} \\
&= [\sqrt{a}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \frac{b}{\sqrt{a}}\cos\frac{\theta}{2}\sin\frac{\phi}{2}]^2 + \frac{ac-b^2}{a}\cos^2\frac{\theta}{2}\sin^2\frac{\phi}{2} + (a+c)\sin^2\frac{\theta}{2}\sin^2\frac{\phi}{2} \geq 0.
\end{aligned}$$

Notice that the constraint  $b^2 - ac < 0$  plays an important role both in the calculation process of  $\lambda_1$  and  $\lambda_2$ .

No matter what  $a, b, c, \theta, \phi, \gamma$  be,  $\lambda_1 \geq 0, \lambda_2 \geq 0$ . The result implies  $\lambda_1\lambda_2 \geq 0$ . In other words,  $|\lambda| \leq 1$  is unconditional. Thus A.D.I. method is also unconditional stable when the anisotropic diffusion equation has a constant coefficient matrix.

### 2.3.2 von Neumann analysis for general anisotropic diffusion problems

Next, we show that the general situation for the anisotropic diffusion problems. The von Neumann analysis fails to deal with variable type. Fortunately, we find that the important reference which be made by Widlund[13]. Therefore, the stability of A.D.I. method in general situation will be done. We rewrite the equation (2.15) as follows.

$$(1 - \frac{ka}{2}\delta_x^2 - \frac{kc}{2}\delta_y^2 + \frac{k^2ac}{4}\delta_x^2\delta_y^2)u^{n+1} = (1 + \frac{ka}{2}\delta_x^2 + \frac{kc}{2}\delta_y^2 + \frac{k^2ac}{4}\delta_x^2\delta_y^2 + 2bkH_xH_y)u^n.$$

For convenience, we define notations below and rewrite the equation.

$$P = 1 - (1 - \frac{1}{2}ka\delta_x^2)(1 - \frac{1}{2}kc\delta_y^2) = \frac{1}{2}ka\delta_x^2 + \frac{1}{2}kc\delta_y^2 - \frac{1}{4}ack^2\delta_x^2\delta_y^2,$$

$$Q = (1 + \frac{1}{2}ka\delta_x^2)(1 + \frac{1}{2}kc\delta_y^2) + 2kbH_xH_y - 1 = \frac{1}{2}ka\delta_x^2 + \frac{1}{2}kc\delta_y^2 + \frac{1}{4}ack^2\delta_x^2\delta_y^2 + 2kbH_xH_y,$$

$$u_{i,j}^{n+1} - u_{i,j}^n = \bar{P}u_{i,j}^{n+1} + \bar{Q}u_{i,j}^n.$$

Now, we replace some terms which refer to widlund's notations.

$$u_{i\pm 1,j} - u_{i,j} = \pm 2i \sin \frac{\theta}{2} e^{\pm i \frac{\theta}{2}},$$

$$u_{i,j\pm 1} - u_{i,j} = \pm 2i \sin \frac{\phi}{2} e^{\pm i \frac{\phi}{2}},$$

$$u_{i+1,j} - u_{i-1,j} = 2i \sin \theta,$$

$$u_{i,j+1} - u_{i,j-1} = 2i \sin \phi.$$

$$u_{i,j}^{n+1} - u_{i,j}^n = \bar{P}u_{i,j}^{n+1} + \bar{Q}u_{i,j}^n,$$

$$\bar{P} = -\frac{2ka}{h^2} \sin^2 \frac{\theta}{2} - \frac{2kc}{h^2} \sin^2 \frac{\phi}{2} - \frac{4ack^2}{h^4} \sin^2 \frac{\theta}{2} \sin^2 \frac{\phi}{2},$$

$$\bar{Q} = -\frac{2ka}{h^2} \sin^2 \frac{\theta}{2} - \frac{2kc}{h^2} \sin^2 \frac{\phi}{2} + \frac{4ack^2}{h^4} \sin^2 \frac{\theta}{2} \sin^2 \frac{\phi}{2} - \frac{2kb}{h^2} \sin \theta \sin \phi,$$

$$u_{i,j}^{n+1} - \bar{P}u_{i,j}^{n+1} = +\bar{Q}u_{i,j}^n + u_{i,j}^n \Rightarrow (1 - \bar{P})u_{i,j}^{n+1} = (1 + \bar{Q})u_{i,j}^n \Rightarrow u_{i,j}^{n+1} = \frac{1 + \bar{Q}}{1 - \bar{P}}u_{i,j}^n.$$

It is only necessary to show that  $\frac{|1+\bar{Q}|}{|1-\bar{P}|} \leq 1$ . We find

$$\frac{|1 + \bar{Q}|}{|1 - \bar{P}|} = |\lambda| = \frac{|(1 - \frac{2ak}{h^2} \sin^2 \frac{\theta}{2})(1 - \frac{2ck}{h^2} \sin^2 \frac{\phi}{2}) - \frac{2bk}{h^2} \sin \theta \sin \phi|}{|(1 + \frac{2ak}{h^2} \sin^2 \frac{\theta}{2})(1 + \frac{2ck}{h^2} \sin^2 \frac{\phi}{2})|}.$$

Since the absolutely value of amplification factor of constant type is the same to variable type, so we can repeat the same work as well as the case with constant coefficient matrix. Therefore A.D.I method is unconditionally stable with variable coefficient matrix.

## 2.4 Apply A.D.I. method to anisotropic diffusion problems

To implement A.D.I. method on a square domain  $\Omega = \{(x, y) | 0 \leq x, y \leq 1\}$ , we begin with a grid consisting of points  $(x_i, y_j)$ , given by  $x_i = ih$ , and  $y_j = jh$  for  $i, j = 0, 1, \dots, N$  respectively. In time direction we have  $t = nk, 1 \leq n \leq N - 1$ , where  $k = 1/N$ .

Firstly, we solve the temporary value  $u^*$  by implicit method. We write down the left hand side of eqn. (2.16) as follows. Obviously, the matrix which corresponds to the linear system is tri-diagonal.

$$-\frac{ka_{i,j}}{2h^2}u_{i-1,j}^* + \left(1 + \frac{ka_{i,j}}{h^2}\right)u_{i,j}^* - \frac{ka_{i,j}}{2h^2}u_{i+1,j}^*.$$

The right hand side will be written below, it contains known information.

$$\left(1 + \frac{1}{2}ka\delta_x^2 + kc\delta_y^2 + 2kbH_xH_y\right)u_{i,j}^n.$$

We assume that  $r = k/h^2$  and reduce the equation. We have implicit part

$$-0.5ra_{i,j}u_{i-1,j}^* + (1 + ra_{i,j})u_{i,j}^* - 0.5ra_{i,j}u_{i+1,j}^*.$$

and information part.

$$\begin{aligned} & (1 - ra_{i,j} - 2rc_{i,j})u_{i,j}^n + 0.5ra_{i,j}(u_{i+1,j}^n + u_{i-1,j}^n) + rc_{i,j}(u_{i,j+1}^n + u_{i,j-1}^n) + 0.5rb_{i,j} \\ & (u_{i+1,j+1}^n - u_{i-1,j+1}^n - u_{i+1,j-1}^n + u_{i-1,j-1}^n). \end{aligned}$$

For  $1 \leq j \leq N - 1$ , we rewrite above equation for each  $j$  in matrix form, and solve the

linear system  $A_j x_j = r_j$ , where  $A_j$  is a tri-diagonal square matrix with order  $N - 1$ .

$$A_i = \begin{pmatrix} 1 + ka_{1,j} & -0.5ka_{1,j} & & & \\ -0.5ka_{2,j} & 1 + ka_{2,j} & -0.5ka_{2,j} & & \\ & \ddots & \ddots & \ddots & \\ & & -0.5ka_{N-1,j} & 1 + ka_{N-1,j} & \end{pmatrix}, x_j = \begin{pmatrix} u_{1,j}^* \\ \vdots \\ u_{i,j}^* \\ \vdots \\ u_{N-1,j}^* \end{pmatrix},$$

$$r_i = \begin{pmatrix} u_{1,j}^n + 0.5ka_{1,j}\delta_x^2 u_{1,j}^n + kc_{1,j}\delta_y^2 u_{1,j}^n + 0.5kb_{1,j}H_x H_y u_{1,j}^n \\ \vdots \\ u_{i,j}^n + 0.5ka_{i,j}\delta_x^2 u_{i,j}^n + kc_{i,j}\delta_y^2 u_{i,j}^n + 0.5kb_{i,j}H_x H_y u_{i,j}^n \\ \vdots \\ u_{N-1,j}^n + 0.5ka_{N-1,j}\delta_x^2 u_{N-1,j}^n + kc_{N-1,j}\delta_y^2 u_{N-1,j}^n + 0.5kb_{N-1,j}H_x H_y u_{N-1,j}^n \end{pmatrix}$$

$$+ \begin{pmatrix} 0.5a_{1,j}u_{1,j}^* \\ 0 \\ \vdots \\ 0 \\ 0.5a_{N-1,j}u_{N-1,j}^* \end{pmatrix}.$$

Notice  $r_j$  composed of two terms, the second term is consisted of boundary condition. Unfortunately,  $u^*$  is undefined. Here is a problem arising, how to deal with the boundaries about  $u^*$ ? Obviously, the term  $u^*$  is contained both in eqn. (2.16) and eqn. (2.17). Generally, we combine two equations and vanish the derivative terms about  $u^*$ . Therefore,  $u^*$  will represent by  $u^{n+1}$  and  $u^n$ . We want to follow the same tactic to deal with our equation. Consider that eqn. (2.17)

$$\left(1 - \frac{1}{2}kc\delta_y^2\right)u_{i,j}^{n+1} = u_{i,j}^* - \frac{1}{2}kc\delta_y^2 u_{i,j}^n.$$

The terms of left hand side associate with  $u^{n+1}$  only, and the right hand side contains  $u^*$  and some terms associate with  $u^n$ .  $u^*$  can be expressed by combination of  $u_{i,j}^n$  and  $u_{i,j}^{n+1}$  as the equation below, so the boundary condition  $u_{i,j}^*$  becomes easier to implement.

$$u_{i,j}^* = \left[1 - \frac{1}{2}kc_{i,j}\delta_y^2\right]u_{i,j}^{n+1} - \frac{1}{2}kc_{i,j}\delta_y^2 u_{i,j}^n.$$

Thomas is applied to solve the tri-diagonal system. We obtain  $u^*$ .

Similarly, we solve  $u^{n+1}$ . By eqn. (2.17)

$$-\frac{kc_{i,j}}{2h^2}u_{i-1,j}^{n+1} + \left(1 + \frac{kc_{i,j}}{h^2}\right)u_{i,j}^{n+1} - \frac{kc_{i,j}}{2h^2}u_{i+1,j}^{n+1} = u_{i,j}^* - \frac{kc_{i,j}}{2h^2}u_{i,j-1}^n + kc_{i,j}u_{i,j}^n - \frac{kc_{i,j}}{2h^2}u_{i,j+1}^n.$$

For  $1 \leq i \leq N-1$ , we have

$$\begin{aligned} \bar{A}_i \bar{x}_i = \bar{b}_i \bar{A}_i &= \begin{pmatrix} 1 + kc_{i,1} & -0.5kc_{i,1} & & & \\ -0.5ka_{i,2} & 1 + ka_{i,2} & -0.5ka_{i,2} & & \\ & \ddots & \ddots & \ddots & \\ & & -0.5ka_{i,N-1} & 1 + ka_{i,N-1} & \end{pmatrix}, \bar{x}_i = \begin{pmatrix} u_{i,1}^{n+1} \\ \vdots \\ u_{i,j}^{n+1} \\ \vdots \\ u_{i,N-1}^{n+1} \end{pmatrix}, \\ \bar{r}_i &= \begin{pmatrix} u_{i,1}^n + 0.5ka_{i,1}\delta_x^2 u_{i,1}^n + kc_{i,1}\delta_y^2 u_{i,1}^n + 0.5kb_{i,1}H_x H_y u_{i,1}^n \\ \vdots \\ u_{i,j}^n + 0.5ka_{i,j}\delta_x^2 u_{i,j}^n + kc_{i,j}\delta_y^2 u_{i,j}^n + 0.5kb_{i,j}H_x H_y u_{i,j}^n \\ \vdots \\ u_{i,N-1}^n + 0.5ka_{i,N-1}\delta_x^2 u_{i,N-1}^n + kc_{i,N-1}\delta_y^2 u_{i,N-1}^n + 0.5kb_{i,N-1}H_x H_y u_{i,N-1}^n \end{pmatrix} \\ &+ \begin{pmatrix} 0.5a_{i,1}u_{i,1}^{n+1} \\ 0 \\ \vdots \\ 0 \\ 0.5a_{i,N-1}u_{i,N-1}^{n+1} \end{pmatrix}. \end{aligned}$$

We deal with tri-diagonal linear system by using Thomas algorithm again.

### 3 Conjugate gradient method for anisotropic diffusion problem

In section 3 we introduce some iterative methods and preconditioner. Then apply the iterative methods to anisotropic diffusion problems. Finally, we will show the numerical results

#### 3.1 Steepest Descent method

Steepest descent method [7] is one kind of iterative method which generally converges to the solution and global in nature. Nearly, for any starting initial value will give convergence. Because C.G. method is originated from steepest descent method, we introduce the steepest descent method before we study C.G method. Consider the linear system of equation.

$$Ax = b, \tag{3.1}$$

where  $A^{n \times n}$  is a large sparse matrix which is symmetric positive definite.  $b \in \mathbb{R}^{n \times 1}$  is a known vector, and  $x \in \mathbb{R}^{n \times 1}$  is solution of the linear system. If we solve eqn. (3.1) by fully implicit scheme, we will face to save CPU time and limited memory for computing when we invert the matrix. In order to avoid the expensive process, it is by no means of inverting the fully implicit matrix by direct methods. Thus we will change our tactic and turn to take advantage of iterative methods.

In the beginning of study the steepest descent method, we consider the quadratic form which is simply defined by

$$g(x) = \frac{1}{2} \langle x, Ax \rangle - \langle x, b \rangle, \tag{3.2}$$

where  $x \in \mathbb{R}^{n \times 1}$  are arbitrary vector,  $A \in \mathbb{R}^{n \times n}$  is defined in eqn. (3.1) and  $\langle, \rangle$  is usual inner product. We demonstrate that the solution of linear system problem (3.1)

is equivalent to the solution of minimizing problem (3.2) by the detail as follows. Let  $v \neq 0 \in \mathbb{R}^{n \times 1}$  be a fixed vector and  $t$  is a real number, then think about

$$\begin{aligned}
 g(x + tv) &= \frac{1}{2} \langle x + tv, Ax + tAv \rangle - \langle x + tv, b \rangle \\
 &= \frac{1}{2} \langle x, Ax \rangle + \frac{t}{2} \langle x, Av \rangle + \frac{t^2}{2} \langle v, Av \rangle + \frac{t}{2} \langle v, Ax \rangle - \langle x, b \rangle \\
 &\quad - t \langle v, b \rangle \\
 &= g(x) + \frac{t^2}{2} \langle v, Av \rangle + t \langle v, Ax - b \rangle .
 \end{aligned} \tag{3.3}$$

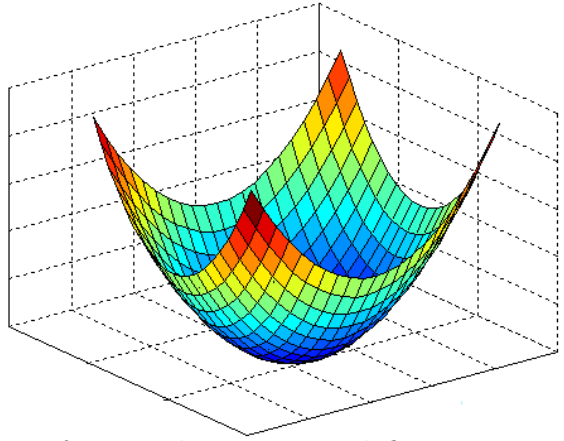


Figure 1: The quadratic form with a positive definite matrix has minimal extreme value.

The extension (3.3) can be regarded as function of  $t$ , we suppose that

$$h(t) = g(x) + \frac{t^2}{2} \langle v, Av \rangle + t \langle v, Ax - b \rangle .$$

Since  $A$  is positive-definite, with the special structure, and the leading coefficient  $\langle v, Av \rangle$  is always positive when  $v \neq 0$ . Since the extreme value will occur at critical point. The first derivative respect to  $t$  as follows.

$$h'(x) = t \langle v, Av \rangle + \langle v, Av - b \rangle = 0,$$



$$t = -\frac{\langle v, Ax - b \rangle}{\langle v, Av \rangle}.$$

After replacing  $t$  into eqn. (3.3) we obtain

$$g(x + tv) = g(x) - \frac{\langle v, Ax \rangle^2}{2 \langle v, Av \rangle},$$

$$\forall t, t \neq 0 \Rightarrow g(x + tv) \geq g(x). \quad (3.4)$$

Now, we show the equivalence of linear system (3.1) and minimization problem (3.2)

If there exists certain vector  $\bar{x}$  which satisfies the linear system  $Ax = b$ , we have  $A\bar{x} = b$ . Therefore,  $t = A\bar{x} - b = 0$ . The equality of (3.4) holds, in other words,  $g(\bar{x})$  is minimal. On the other way, the equality of (3.4) holds when  $t = 0$  implies  $b - Ax = 0$ . We find a solution of the linear system.

In the method of steepest descent, we give an arbitrary vector  $x$  and then approximate to the minimal value step by step. We take a series of steps  $x_1, x_2, \dots$ , until we are satisfied with the numerical solution close enough to exact solution. By theorem, we have the direction of greatest increase in the value  $g(x)$  is the direction given by  $\nabla g(x)$ . In other words, the decreasing rate is maximal along the opposite direction of gradient.

$$g(x) = \frac{1}{2} \langle x, Ax \rangle - \langle x, b \rangle$$

$$= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j - \sum_{i=1}^n x_i b_i.$$

And then we calculate the gradient.

$$\frac{\partial g(x)}{\partial x_j} = \sum_{i=1}^n a_{ji} x_i - b_j,$$

$$\nabla g(x) = \left[ \frac{\partial g(x)}{\partial x_1}, \frac{\partial g(x)}{\partial x_2}, \dots, \frac{\partial g(x)}{\partial x_n} \right]^T = Ax - b = -(b - Ax) = -r,$$

Based on the advantage of greatest decreasing rate, we will solve eqn. (3.2) rather than eqn. (3.1). Thus the method of steepest descent will start from an initial vector

$x^0$ , and then define the initial search direction  $r^0 = -\nabla g(x^0) = b - Ax^0$ . The following steps are recursively given by

$$x^{k+1} = x^k + \alpha_k r^k, k = 0, 1, 2, \dots$$

$\alpha_k$  are parameters which will be chosen so that  $g(x)$  can be minimized. We add  $b$  to both hand sides after multiplying on both hand sides.

$$-Ax^{k+1} = -Ax^k - \alpha_k Ar^k,$$

$$b - Ax^{k+1} = b - Ax^k - \alpha_k Ar^k.$$

Since  $-\nabla g(x^k) = b - Ax^k$ , thus search direction is also given as follows.

$$r^{k+1} = r^k - \alpha_k Ar^k, k = 0, 1, 2, \dots$$

A line search is a procedure that minimizing  $g(x)$  by the way of choosing  $\alpha_k$ , we initiate our calculation from the equation.

$$g(x^{k+1}) = g(x^k + \alpha_k r^k) = g(x^k) - \alpha_k \langle r^k, r^k \rangle + \frac{1}{2} \alpha_k^2 \langle r^k, Ar^k \rangle.$$

The above equation can be regarded as function of  $\alpha_k$ . The leading coefficient  $\frac{1}{2} \langle r^k, Ar^k \rangle$  is positive. By partial derivative respect to  $\alpha_k$ , the minimal value occurs at critical point.

$$\partial g(x^{k+1}) / \partial \alpha_k = -\langle r^k, r^k \rangle + \alpha_k \langle r^k, Ar^k \rangle.$$

Setting  $\partial g(x^{k+1}) / \partial \alpha_k = 0$ , we find that  $\alpha_k$  given by

$$\alpha_k = \frac{\langle r^k, r^k \rangle}{\langle r^k, Ar^k \rangle}.$$

We check that consecutive residuals are orthogonal for the choice of

$$\begin{aligned} \langle r^{k+1}, r^k \rangle &= \langle r^k - \alpha_k Ar^k, r^k \rangle = \langle r^k, r^k \rangle - \alpha_k \langle r^k, Ar^k \rangle \\ &= \langle r^k, r^k \rangle - \frac{\langle r^k, r^k \rangle}{\langle r^k, Ar^k \rangle} \langle r^k, Ar^k \rangle = 0. \end{aligned}$$

We collect the steps of the steepest descent method as follows.

1. Choose an arbitrary initial vector  $x^0$  and  $r^0 = b - Ax^0$ , and tolerance is given.

2. for  $k=0,1,2,\dots$  ,  $\alpha_k = \frac{\langle r^k, r^k \rangle}{\langle r^k, Ar^k \rangle}$ ,

$$x^{k+1} = x^k + \alpha_k r^k,$$

$$r^{k+1} = r^k - \alpha_k Ar^k.$$

3. If  $|x^{k+1} - x^k| < \text{tolerance}$ , the solution can be approximated by  $x^{k+1}$

else return to step 2.

### 3.2 Introduction of conjugate gradient method (C.G. method)

The conjugate gradient method [7] of Hestenes and Stiefel was originally developed as a direct method to solve positive definite matrix of linear systems. In this section, we employed the C.G. method as an iterative method. The C.G. method is one of the most prominent and efficient algorithms for the numerical solution of particular linear systems, namely the systems whose matrix is symmetric positive definite. Since the C.G. method is a kind of iterative method, so it can be applied to deal with large sparse systems which can not be deal with by direct method such as cholesky decomposition, Gauss-Seidel, and Jacobi methods. The idea of the C.G. method is from minimization of the quadratic forms. Moreover, preconditioning is a technique in further acceleration. C.G. method is most popular and efficient iterative method for solving large sparse systems of the form.

The conjugate method can be regarded as the modification of the steepest descent method. The most difference is that C.G. method modifies the search direction, and then it becomes more efficient method. We start with an arbitrary  $x^0$ , then we have

$p^0 = r^0 = b - Ax^0$ . Therefore, the iterative steps will be defined as follows.

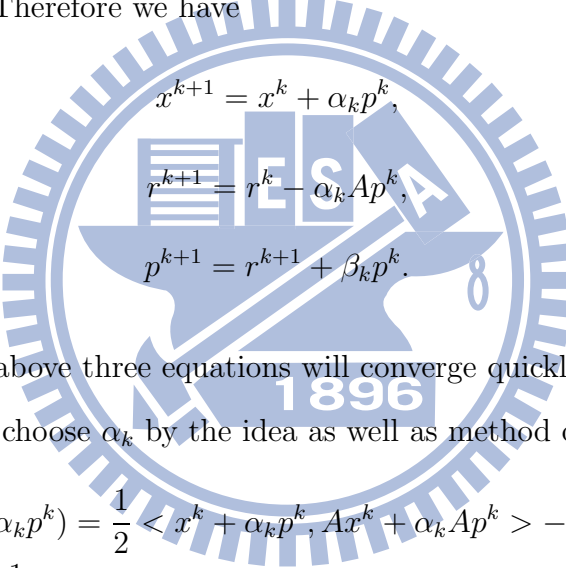
$$x^{k+1} = x^k + \alpha_k p^k, \quad (3.5)$$

$$p^k = r^k + \gamma_k(x^k - x^{k-1}). \quad (3.6)$$

Notice the vector  $p^k$  is called new search direction to the k-th iteration which is the modification of steepest descent method  $p^k$  is linear combination of original defined search direction  $r^k$  and difference between consecutive steps is  $x^k - x^{k-1}$ . We re-write eqn. (3.6)

$$p^k = r^k + \gamma_k(x^k - x^{k-1}) = r^k + \gamma_k \alpha_{k-1} p^{k-1}.$$

Then let  $\gamma_k \alpha_{k-1} = \beta_{k-1}$ . The above equation will become  $p^{k+1} = r^{k+1} + \beta_k p^k$ . We collect the formulas. Therefore we have



$$\begin{aligned} x^{k+1} &= x^k + \alpha_k p^k, \\ r^{k+1} &= r^k - \alpha_k A p^k, \\ p^{k+1} &= r^{k+1} + \beta_k p^k. \end{aligned}$$

We wish that the above three equations will converge quickly once  $\alpha_k$  and  $\beta_k$  are determined. Now, we choose  $\alpha_k$  by the idea as well as method of steepest descent.

$$\begin{aligned} g(x^{k+1}) &= g(x^k + \alpha_k p^k) = \frac{1}{2} \langle x^k + \alpha_k p^k, Ax^k + \alpha_k A p^k \rangle - \langle x^k + \alpha_k p^k, b \rangle \\ &= g(x^k) + \frac{1}{2} \alpha_k^2 \langle p^k, A p^k \rangle - \alpha_k \langle p^k, r^k \rangle. \end{aligned} \quad (3.7)$$

Then we calculate the partial derivative respect to  $\alpha_k$ .

$$\partial g(x^{k+1}) / \partial \alpha_k = \alpha_k \langle p^k, A p^k \rangle - \langle p^k, r^k \rangle.$$

Set  $\partial g(x^{k+1}) / \partial \alpha_k = 0$ , we find that  $\alpha_k$  can be given by

$$\alpha_k = \frac{\langle p^k, r^k \rangle}{\langle p^k, A p^k \rangle}.$$

We re-write eqn. (3.7)

$$g(x^{k+1}) = g(x^k) - \frac{\langle p^k, r^k \rangle^2}{2 \langle p^k, Ap^k \rangle}. \quad (3.8)$$

We illustrate the idea why we let the search direction  $p_0 = r_0$ , according to eqn. (3.8). If we choose the initial search direction  $p_0$  equal to  $r_0$ , we can decrease the value  $g(x^1)$  than  $g(x^0)$  as follows.

$$g(x^1) = g(x_0) - \frac{\langle p^0, r^0 \rangle^2}{2 \langle p^0, Ap^0 \rangle}.$$

Next, we determine  $\beta_k$  by studying eqn. (3.8).

As we know  $g(x^{k+1}) = g(x^k) - \frac{\langle p^k, r^k \rangle^2}{2 \langle p^k, Ap^k \rangle}$ . In order to minimize the value of each step, we discuss the minus term  $-\frac{\langle p^k, r^k \rangle^2}{2 \langle p^k, Ap^k \rangle}$ .

The numerator part :

$$\begin{aligned} \langle p^k, r^k \rangle &= \langle r^k + \beta_k p^{k-1}, r^k \rangle \\ &= \langle r^k, r^k \rangle + \beta_{k-1} \langle p^{k-1}, r^k \rangle, \\ \langle p^{k-1}, r^k \rangle &= \langle p^{k-1}, r^{k-1} - \alpha_{k-1} Ar^{k-1} \rangle \\ &= \langle p^{k-1}, r^{k-1} \rangle - \alpha_{k-1} \langle p^{k-1}, Ar^{k-1} \rangle \\ &= \langle p^{k-1}, r^{k-1} \rangle - \frac{\langle p^{k-1}, r^{k-1} \rangle}{\langle p^{k-1}, Ar^{k-1} \rangle} \langle p^{k-1}, Ar^{k-1} \rangle = 0. \end{aligned} \quad (3.9)$$

In order to make the equation simple, we replace (3.9) into (3.10) we have the

$\langle p^k, r^k \rangle = \langle r^k, r^k \rangle$ , and re-write eqn. (3.10) as  $g(x^{k+1}) = g(x^k) - \frac{\langle r^k \rangle^2}{2 \langle p^k, Ap^k \rangle}$ .

The denominator part : Since  $p^k = r^k + \beta_{k-1} p^{k-1}$ .

$$\begin{aligned} \langle p^k, Ap^k \rangle &= \langle r^k + \beta_{k-1} p^{k-1}, A(r^k + \beta_{k-1} p^{k-1}) \rangle \\ &= \langle r^k, Ar^k \rangle + 2\beta_{k-1} \langle r^k, Ap^{k-1} \rangle + \beta_{k-1}^2 \langle p^{k-1}, Ap^{k-1} \rangle, \end{aligned}$$

$$\frac{\partial}{\partial \beta_{k-1}} [\langle r^k, Ar^k \rangle + 2\beta_{k-1} \langle r^k, Ap^{k-1} \rangle + \beta_{k-1}^2 \langle p^{k-1}, Ap^{k-1} \rangle] = 0,$$

$$\beta_{k-1} = -\frac{\langle r^k, Ap^{k-1} \rangle}{\langle p^{k-1}, Ap^{k-1} \rangle}. \text{ Equivalence, } \beta_k = -\frac{\langle r^{k+1}, Ap^k \rangle}{\langle p^k, Ap^k \rangle}. \quad (3.11)$$

By minimizing the denominator, C.G. method converges more quickly. Next, we describe by saying that consecutive search directions are conjugate. By eqn.(3.11)

$$\begin{aligned} \langle r^{k+1}, Ap^k \rangle + \beta_k \langle p^k, Ap^k \rangle &= 0. \\ \Rightarrow \langle r^{k+1} + \beta_k p^k, Ap^k \rangle &= 0, \\ \Rightarrow \langle p^{k+1}, Ap^k \rangle &= 0. \end{aligned}$$

### 3.3 Implementation of the conjugate gradient method for anisotropic diffusion problems

We consider the anisotropic diffusion equation with constant coefficient matrix.

$$\frac{\partial u}{\partial t} = \nabla \cdot (\beta \nabla u),$$

where  $\beta = \begin{pmatrix} \tilde{a} & \tilde{b} \\ \tilde{b} & \tilde{c} \end{pmatrix}$ ,  $\tilde{a} > 0$ ,  $\tilde{c} > 0$ , and  $\tilde{b}^2 - \tilde{a}\tilde{c} < 0$ ,  $u(x, y, 0) = u_0(x, y)$ ,  $(x, y) \in \Omega$ ,  $u(x, y, t) = g(x, y, t)$ ,  $(x, y) \in \partial\Omega, t \in (0, T]$  Because of constant coefficient matrix, we can choose suitable discretization such that the linear system  $Ax = b$  has a symmetric positive definite matrix.

#### 3.3.1 Discretization by Crank-Nicolson scheme

We extend the anisotropic diffusion equation into simple form as follows.

$$\frac{\partial u}{\partial t} = \nabla \cdot (\beta \nabla u) \Rightarrow u_t = au_{xx} + 2bu_{xy} + cu_{yy}.$$

Beginning with the formula  $u_t = \frac{u^{n+1} - u^n}{k} + O(k^2)$  for  $u_t$  evaluated at  $t + 1/2$ , and using the same idea to deal with  $u_{xx}$ ,  $u_{yy}$  and  $u_{xy}$ . We have

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{k} = \frac{1}{2}a\delta_x^2 u_{i,j}^{n+1} + \frac{1}{2}a\delta_x^2 u_{i,j}^n + \frac{1}{2}c\delta_y^2 u_{i,j}^{n+1} + \frac{1}{2}c\delta_y^2 u_{i,j}^n + \frac{1}{2}bH_x H_y u_{i,j}^{n+1} + \frac{1}{2}bH_x H_y u_{i,j}^n,$$

$$u_{i,j}^{n+1} - u_{i,j}^n = \frac{k}{2}a\delta_x^2 u_{i,j}^{n+1} + \frac{k}{2}a\delta_x^2 u_{i,j}^n + \frac{k}{2}c\delta_y^2 u_{i,j}^{n+1} + \frac{k}{2}c\delta_y^2 u_{i,j}^n + \frac{k}{2}bH_x H_y u_{i,j}^{n+1} + \frac{k}{2}bH_x H_y u_{i,j}^n.$$

We separate  $u^{n+1}$  from  $u^n$ , and then

$$\left(1 - \frac{k}{2}a\delta_x^2 - \frac{k}{2}c\delta_y^2 - \frac{k}{2}bH_x H_y\right)u_{i,j}^{n+1} = \left(1 + \frac{k}{2}a\delta_x^2 + \frac{k}{2}c\delta_y^2 + \frac{k}{2}bH_x H_y\right)u_{i,j}^n.$$

To simplify, we multiply two on both sides.

$$(2 - ka\delta_x^2 - kc\delta_y^2 - kbH_x H_y)u_{i,j}^{n+1} = (2 + ka\delta_x^2 + kc\delta_y^2 + kbH_x H_y)u_{i,j}^n. \quad (3.12)$$

We extend the eqn. (3.12) into two parts. To simplify, we define some notations.

$$\tilde{d} = 2 + 2ka + 2kc, \tilde{e} = 2 - 2ka - 2kc, \tilde{a} = ka, \tilde{b} = kb, \tilde{c} = kc.$$

The left hand side of eqn. (3.12)

$$(2 + 2ka + 2kc)u_{i,j}^{n+1} - ka(u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1}) - kc(u_{i,j-1}^{n+1} + u_{i,j+1}^{n+1}) - kb(u_{i+1,j+1}^{n+1} - u_{i-1,j+1}^{n+1} - u_{i+1,j-1}^{n+1} + u_{i-1,j-1}^{n+1}).$$

By notations which is defined above, left part becomes

$$\tilde{d}u_{i,j}^{n+1} - \tilde{a}(u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1}) - \tilde{c}(u_{i,j-1}^{n+1} + u_{i,j+1}^{n+1}) - \tilde{b}(u_{i+1,j+1}^{n+1} - u_{i-1,j+1}^{n+1} - u_{i+1,j-1}^{n+1} + u_{i-1,j-1}^{n+1}).$$

The right hand side of eqn. (3.12) which is

$$(2 - 2ka - 2kc)u_{i,j}^n + ka(u_{i+1,j}^n + u_{i-1,j}^n) + kc(u_{i,j-1}^n + u_{i,j+1}^n) + kb(u_{i+1,j+1}^n - u_{i-1,j+1}^n - u_{i+1,j-1}^n + u_{i-1,j-1}^n).$$

turns to

$$\tilde{e}u_{i,j}^n + \tilde{a}(u_{i+1,j}^n + u_{i-1,j}^n) + \tilde{c}(u_{i,j-1}^n + u_{i,j+1}^n) + \tilde{b}(u_{i+1,j+1}^n - u_{i-1,j+1}^n - u_{i+1,j-1}^n + u_{i-1,j-1}^n).$$

Obviously, the fully implicit scheme will need to solve nine points at the same time.

We solve the equation by fully implicit method. Re-write the eqn. (3.12) in matrix form

$$\mathbf{A}\mathbf{u} = \mathbf{B}\mathbf{x} + \mathbf{R}.$$

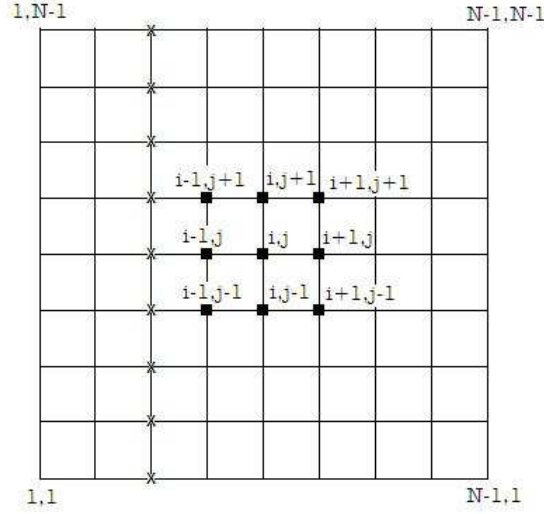


Figure 2: We use regular uniform domain and collect the columns to be a new column.

The vector  $A\mathbf{u}$  is corresponding to the left hand side part of eqn. (3.12), where  $\mathbf{u} \in \mathbb{R}^{(N-1)^2 \times 1}$  is an unknown vector which represented the value of next time step. We take each column from domain and compose them to be a new vector which is denoted by

$$\mathbf{u} = [u^1 \ u^2 \ \dots \ u^{N-1}]^T,$$

where  $u^i = [u_{i,1}^{n+1}, u_{i,2}^{n+1}, \dots, u_{i,N-1}^{n+1}]^T$ .

Therefore we have the matrix

$$A = \begin{pmatrix} D & U & & & \\ L & \ddots & \ddots & & \\ & \ddots & \ddots & U & \\ & & & L & D \end{pmatrix},$$

where  $A \in \mathbb{R}^{(N-1)^2 \times (N-1)^2}$  is block tri-diagonal and positive definite matrix. The three block matrices are also tri-diagonal.

$$D = \begin{pmatrix} \tilde{d} & -\tilde{c} & & & \\ -\tilde{c} & \ddots & \ddots & & \\ & \ddots & \ddots & -\tilde{c} & \\ & & & -\tilde{c} & \tilde{d} \end{pmatrix}, U = \begin{pmatrix} \tilde{a} & -\tilde{b} & & & \\ \tilde{b} & \ddots & \ddots & & \\ & \ddots & \ddots & -\tilde{b} & \\ & & & \tilde{b} & \tilde{a} \end{pmatrix}, L = \begin{pmatrix} \tilde{a} & \tilde{b} & & & \\ -\tilde{b} & \ddots & \ddots & & \\ & \ddots & \ddots & \tilde{b} & \\ & & & -\tilde{b} & \tilde{a} \end{pmatrix}$$

Notice that  $U^T = L$ , so the matrix has the property of symmetricization.



The left hand side part of eqn. (3.12) is represented by vector  $B\mathbf{x}$ , where  $\mathbf{x} \in \mathbb{R}^{(N-1)^2 \times 1}$  is a known vector which take the information for solving next time step.

$$B = \begin{pmatrix} D' & U' & & \\ L' & \ddots & \ddots & \\ & \ddots & \ddots & U' \\ & & L' & D' \end{pmatrix}, \text{ where } B \text{ is also a block tri-diagonal matrix, and}$$

$$D' = \begin{pmatrix} \tilde{e} & -\tilde{c} & & \\ -\tilde{c} & \ddots & \ddots & \\ & \ddots & \ddots & \tilde{c} \\ & & -\tilde{c} & \tilde{e} \end{pmatrix}, U' = \begin{pmatrix} -\tilde{a} & \tilde{b} & & \\ -\tilde{b} & \ddots & \ddots & \\ & \ddots & \ddots & \tilde{b} \\ & & -\tilde{b} & -\tilde{a} \end{pmatrix}, L' = \begin{pmatrix} -\tilde{a} & -\tilde{b} & & \\ \tilde{b} & \ddots & \ddots & \\ & \ddots & \ddots & -\tilde{b} \\ & & \tilde{b} & -\tilde{a} \end{pmatrix}$$

The column vector  $\mathbf{R} \in \mathbb{R}^{(N-1)^2 \times 1}$  contains boundary condition.

$$\mathbf{R} = [R^1, R^2, \dots, R^{N-1}]^T,$$

$$R^1 = \begin{pmatrix} \tilde{a}u_{0,1}^{n+1} + \tilde{c}u_{1,0}^{n+1} + \tilde{b}u_{0,0}^{n+1} - \tilde{b}u_{2,1}^{n+1} - \tilde{b}u_{0,2}^{n+1} \\ \tilde{a}u_{0,2}^{n+1} + \tilde{b}u_{0,1}^{n+1} - \tilde{b}u_{0,3}^{n+1} \\ \vdots \\ \tilde{a}u_{i-1,j}^{n+1} + \tilde{b}u_{i-1,j-1}^{n+1} - \tilde{b}u_{i-1,j+1}^{n+1} \\ \vdots \\ \tilde{a}u_{0,N-2}^{n+1} + \tilde{b}u_{0,N-3}^{n+1} - \tilde{b}u_{0,N-1}^{n+1} \\ \tilde{a}u_{0,N}^{n+1} + \tilde{c}u_{1,N-1}^{n+1} + \tilde{b}u_{0,N-1}^{n+1} - \tilde{b}u_{2,N-1}^{n+1} - \tilde{b}u_{0,N+1}^{n+1} \end{pmatrix},$$

$$R^{N-1} = \begin{pmatrix} \tilde{a}u_{N,1}^{n+1} + \tilde{c}u_{N-1,0}^{n+1} + \tilde{b}u_{N-2,0}^{n+1} + \tilde{b}u_{N,2}^{n+1} - \tilde{b}u_{N,0}^{n+1} \\ \tilde{a}u_{N,2}^{n+1} + \tilde{b}u_{N,1}^{n+1} + \tilde{b}u_{N,3}^{n+1} \\ \vdots \\ \tilde{a}u_{i+1,j}^{n+1} - \tilde{b}u_{i+1,j-1}^{n+1} + \tilde{b}u_{i+1,j+1}^{n+1} \\ \vdots \\ \tilde{a}u_{N,N-2}^{n+1} - \tilde{b}u_{N,N-3}^{n+1} + \tilde{b}u_{N,N-1}^{n+1} \\ \tilde{a}u_{N,N-1}^{n+1} + \tilde{c}u_{N-1,N}^{n+1} + \tilde{b}u_{N,N}^{n+1} - \tilde{b}u_{N,N-2}^{n+1} - \tilde{b}u_{N-2,N}^{n+1} \end{pmatrix},$$

$$R^i = \begin{pmatrix} \tilde{c}u_{i,0}^{n+1} + \tilde{b}u_{i-1,0}^{n+1} - \tilde{b}u_{i+1,0}^{n+1} \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \tilde{c}u_{i,N}^{n+1} + \tilde{b}u_{i+1,N}^{n+1} - \tilde{b}u_{i-1,N}^{n+1} \end{pmatrix}, 2 \leq i \leq N-2.$$

We apply conjugate gradient method to deal with the anisotropic diffusion problem with constant coefficient matrix. Next, we consider the matrix whose entries are variable type, and we will use biconjugate gradient method to deal with the problem.

### 3.3.2 The biconjugate gradient method

Different from conjugate gradient method, biconjugate gradient method does not require symmetric matrix, but conjugate transpose  $A^T$ . The algorithm is as follows.

1. Choose a initial vector  $x^0$ ,  $r^0 = b - Ax^0$  and select  $\tilde{r}^0$  such that  $\langle r^0, \tilde{r}^0 \rangle \neq 0$ .
2. Set  $p^0 = r^0$  and  $\tilde{p}^0 = \tilde{r}^0$ ,
3. For  $k = 0, 1, 2, \dots$

$$\alpha_k = \frac{\langle r^k, \tilde{r}^k \rangle}{\langle p^k, A\tilde{p}^k \rangle},$$

$$x^{k+1} = x^k + \alpha_k p^k,$$

$$\tilde{x}^{k+1} = \tilde{x}^k + \alpha_k \tilde{p}^k,$$

$$r^{k+1} = r^k - \alpha_k A p^k,$$

$$\tilde{r}^{k+1} = \tilde{r}^k - \alpha_k A^T \tilde{p}^k,$$

$$\beta_k = \frac{\langle r^{k+1}, \tilde{r}^{k+1} \rangle}{\langle r^k, \tilde{r}^k \rangle},$$

$$p^{k+1} = r^{k+1} + \beta_k \tilde{p}^k,$$

$$\tilde{p}^{k+1} = \tilde{r}^{k+1} + \beta_k \tilde{p}^k, \text{ if convergence ends.}$$

### 3.3.3 The preconditioned conjugate gradient method

A technique resulting in further acceleration of the conjugate gradient method is the preconditioned conjugate gradient method.[3] The basic idea of the preconditioned conjugate method is to replace the system.

$$Ax = b$$

by

$$C^{-1}AC^{-1}(Cx) = C^{-1}b.$$

Since  $A$  is large sparse matrix, we apply incomplete  $LU$  factorization which is a kind of precondition. The factorization is used to solve sparse square matrices. Incompleting  $LU$  factorization produces a unit lower triangular matrix  $L$ , an upper triangular matrix  $U$ , and residuals  $R$ .

$$A = LU - R$$

We let  $C = L$ , and  $C^{-1}AC^{-1}$  is a matrix for which the conjugate gradient method converges faster than it does with  $A$  itself. We define

$$\tilde{A} = C^{-1}AC^{-1}$$

$$\tilde{x} = Cx$$

$$\tilde{b} = C^{-1}b$$

Since  $\tilde{A}$  is symmetric positive definite, then we apply conjugate gradient method to the linear system  $\tilde{A}\tilde{x} = \tilde{b}$ .

1. Start with a initial vector  $\tilde{x}^0$ . Initial search direction  $\tilde{r}^0 = \tilde{b} - \tilde{A}\tilde{x}^0$ .
2. For  $k = 0, 1, 2, \dots$ ,

$$\tilde{\alpha}_k = \frac{\langle \tilde{p}^k, \tilde{r}^k \rangle}{\langle \tilde{p}^k, \tilde{A}\tilde{p}^k \rangle},$$

$$\tilde{x}^{k+1} = \tilde{x}^k + \tilde{\alpha}_k \tilde{p}^k,$$

$$\tilde{r}^{k+1} = \tilde{r}^k - \tilde{\alpha}_k \tilde{A}\tilde{p}^k,$$

$$\tilde{p}^{k+1} = \tilde{p}^k - \tilde{\beta}_k \tilde{p}^k,$$

$$\tilde{\beta}_k = -\frac{\langle \tilde{r}^{k+1}, \tilde{A}\tilde{p}^k \rangle}{\langle \tilde{p}^k, \tilde{A}\tilde{p}^k \rangle}, \text{ if convergence ends.}$$

### 3.3.4 The preconditioned biconjugate gradient method

We apply the same idea to the precondition of biconjugate gradient method. The algorithm

1. Choose initial vector  $x^0$ , two vectors  $\tilde{x}^0, \tilde{b}$  and a preconditioner  $M, M$  can be  $I$ .
2.  $r^0 = b^0 - Ax^0, \tilde{r}^0 = \tilde{b}^0 - A^T \tilde{x}^0$ .
3.  $p^0 = M^{-1}x^0, \tilde{p}^0 = (M^{-1})^* \tilde{r}^0$ .
4. For  $k = 0, 1, 2, \dots$

$$\alpha_k = \frac{\langle \tilde{r}^k, M^{-1}r^k \rangle}{\langle \tilde{p}^k, Ap^k \rangle},$$

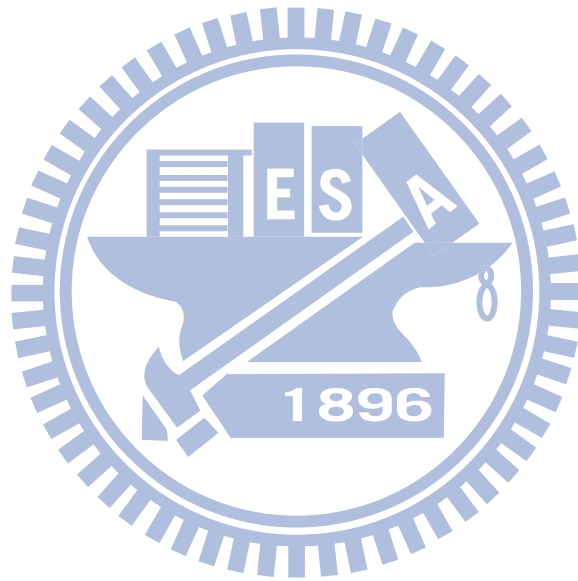
$$x^{k+1} = x^k + \alpha_k p^k, \tilde{x}^{k+1} = \tilde{x}^k + \tilde{\alpha}_k \tilde{p}^k,$$

$$r^{k+1} = r^k - \alpha_k Ap^k, \tilde{r}^{k+1} = \tilde{r}^k - \tilde{\alpha}_k A^T \tilde{p}^k,$$

$$\beta_k = \frac{\langle \tilde{r}^{k+1}, M^{-1}r^{k+1} \rangle}{\langle \tilde{r}^k, M^{-1}r^k \rangle},$$

$$p^{k+1} = M^{-1}r^{k+1} + \beta_k p^k, \tilde{p}^{k+1} = M^{-1}\tilde{r}^{k+1} + \tilde{\beta}_k \tilde{p}^k.$$

Notice that the terms  $r^k$  and  $\tilde{r}^k$  satisfy with  $r^k = b - Ax^k$ ,  $\tilde{r}^k = \tilde{b} - A\tilde{x}^k$ . And  $x^k$  and  $\tilde{x}^k$  are solutions corresponding to  $Ax = b$  and  $A^T\tilde{x} = \tilde{b}$ .



## 4 Numerical results

In this section, we discuss some numerical simulations performed by A.D.I. and iterative methods to characterize the advantages and disadvantages of these two different kinds of approach. To analyze the order of accuracy of previous algorithms, we firstly consider two dimensional test cases with no mix-derivative term which is heat equation, then constant and variable coefficients diffusivity. We will also demonstrate the CPU time to compare the efficiency of each method.

### 4.1 Numerical results for heat equations

In order to compare these two numerical methods, we consider two dimensional heat equation.

$$u_t = au_{xx} + cu_{yy}, u(0) = u_0, \partial u|_{\Omega} = g(t),$$
$$u \in C^2, \text{ and } \Omega = \{(x, y) | 0 \leq x, y \leq 1\}, 0 \leq t \leq T.$$

The first example is a two dimensional parabolic differential equation with initial and dirichlet boundary conditions.

$$u_t = u_{xx} + u_{yy}, 0 < x, y < 1 \text{ and } t > 0,$$

$$u(x, y, 0) = e^{x+y}, 0 \leq x, y \leq 1,$$

$$u(0, y, t) = e^{2t+y}, u(1, y, t) = e^{2t+y+1},$$

$$u(x, 0, t) = e^{2t+x}, u(x, 1, t) = e^{2t+x+1}.$$

The exact solution is

$$u(x, y, t) = e^{x+y+2t}.$$

No matter what the coefficient matrix be, A.D.I. method works. The discretized matrix of above equation is constant and symmetric positive definite. We employ

C.G. method then list a table about order of accuracy and CPU time immediately. Using exact solution, we compute 2-norm relative errors and calculation is run up to time  $T = 1$ .

Conjugate Gradient method				A.D.I method		
$\Delta t = h$	relative error	time	order	relative error	time	order
1/20	3.3598e-005	0.17		1.3355e-005	0.06	
1/40	8.0106e-006	0.29	2.0684	3.1992e-006	0.18	2.0616
1/80	1.9545e-006	2.43	2.0351	7.8148e-007	0.86	2.0334
1/160	4.8263e-007	35.90	2.0178	1.9303e-007	5.56	2.0174

Next, we consider another test equation with variable matrix which is not symmetric but positive definite and apply B.I.C.G. method.

$$u_t = x^2 u_{xx} + y^2 u_{yy}, 0 < x, y < 1 \text{ and } t > 0$$

$$u(x, y, 0) = x^2 y + x y^2, 0 \leq x, y \leq 1$$

$$u(0, y, t) = 0, u(1, y, t) = e^{2t}(y^2 + y)$$

$$u(x, 0, t) = 0, u(x, 1, t) = e^{2t}(x^2 + x)$$

The exact solution is  $u(x, y, t) = e^{2t}(x^2 y + x y^2)$ . The numerical results are as follows.

Bi-Conjugate Gradient method				A.D.I method		
$\Delta t = h$	relative error	time	order	relative error	time	order
1/20	2.1829e-005	0.20		2.1785e-005	0.06	
1/40	5.0507e-006	0.71	2.1117	5.0486e-006	0.16	2.1094
1/80	1.2156e-006	7.85	2.0548	1.2155e-006	0.88	2.0544
1/160	2.9814e-007	126.53	2.0276	2.9821e-007	5.44	2.0271

Finally, we test the third example and analyze the result.

$$u_t = \frac{2}{3}(1+x+y)^2 u_{xx} + \frac{2}{3}(1+x+y)^2 u_{yy}, 0 < x, y < 1 \text{ and } t > 0,$$

$$u(x, y, 0) = (1+x+y)^{1.5}, 0 \leq x, y \leq 1,$$

$$u(0, y, t) = e^t(1+y)^{1.5}, u(1, y, t) = e^t(2+y)^{1.5},$$

$$u(x, 0, t) = e^t(1 + x)^{1.5}, u(x, 1, t) = e^t(2 + x)^{1.5},$$

The exact solution is  $u(x, y, t) = e^t(1 + x + y)^{1.5}$ .

Bi-Conjugate Gradient method				A.D.I method		
$\Delta t = h$	relative error	time	order	relative error	time	order
1/20	2.2056e-006	0.15		8.3317e-007	0.06	
1/40	5.3968e-007	0.50	2.0310	2.0462e-007	0.18	2.0257
1/80	1.3306e-007	5.21	2.0200	5.0611e-008	0.85	2.0154
1/160	3.3048e-008	41.25	2.0095	1.2580e-008	5.55	2.0084

We analyze the examples previously by figures. The discussion contains order of accuracy and CPU time. Since heat equation contains pure second derivative terms respect to space. When we evaluate it at  $t_{k+1/2}$  by Crank-Nicolson scheme, A.D.I. and iterative methods will converge in the second order of accuracy. Figure 3 shows that A.D.I., C.G. and B.I.C.G method satisfy the expected results. Next, we want to figure out which method is more efficient. For each method, the grid sizes are 1/20, 1/40, 1/80, and 1/160. There are four kinds of grid points, we construct polynomials which are degree 3 to fit the data. We notice the iterative methods need more and more time in computing process, but the increasing rate of CPU time of A.D.I is quiet small. It seems linearly.

## 4.2 Comparison of iterative methods and the A.D.I. method

Mix-derivative term is the only difference between heat and anisotropic diffusion equation. In the following examples, We will illustrate how the mix-derivative term effects these methods.

$$u_t = au_{xx} + 2bu_{xy} + cu_{yy}, 0 < x, y < 1 \text{ and } t > 0, \text{ where } \beta = \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix},$$

$$u(x, y, 0) = e^{x+y}, 0 \leq x, y \leq 1,$$

$$u(0, y, t) = e^{y+2t}, u(1, y, t) = e^{1+y+2t},$$



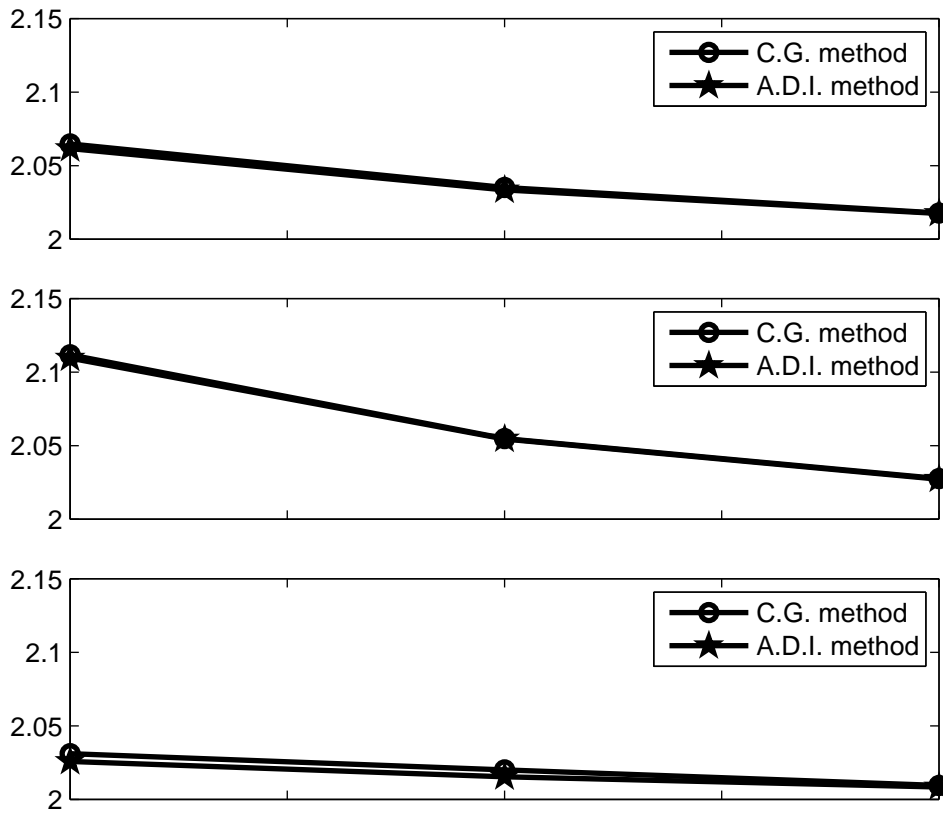


Figure 3: Comparison of order of accuracy for case 1, 2, 3, respectively. (y-axis is order of accuracy, x-axis is number of grid point)

$$u(x, 0, t) = e^{x+2t}, u(x, 1, t) = e^{x+1+2t}.$$

The exact solution is  $u(x, y, t) = e^{x+y+2t}$ .

Bi-Conjugate Gradient method				A.D.I method		
$\Delta t = h$	relative error	time	order	relative error	time	order
1/20	2.2459e-004	0.25		8.3664e-004	0.06	
1/40	5.7107e-005	0.89	1.9755	4.0900e-004	0.16	1.0325
1/80	1.4421e-005	7.32	1.9855	2.0468e-004	0.90	1.0200
1/160	3.6279e-006	53.37	1.9910	1.0007e-004	5.63	1.0110

The second case is as follows.

$$u_t = au_{xx} + 2bu_{xy} + cu_{yy}, 0 < x, y < 1 \text{ and } t > 0,$$

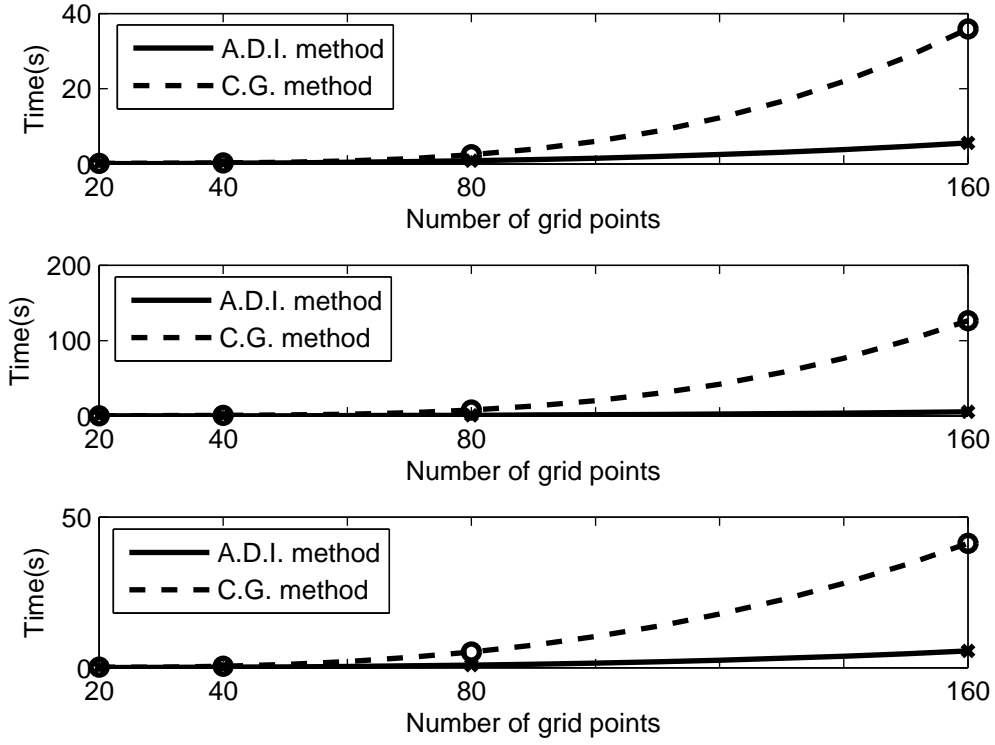


Figure 4: CPU time of case 1, 2, 3, respectively.

$$a = c = 12\cos^2[\pi(x+y)/6]/\pi^2, b = -3\cos^2[\pi(x+y)/6]/\pi^2,$$

$$u(x, y, 0) = \tan[\pi(x+y)/6], 0 < x, y < 1,$$

$$u(0, y, t) = e^t \tan[\pi y/6], u(1, y, t) = e^t \tan[\pi(1+y)/6],$$

$$u(x, 0, t) = e^t \tan[\pi y/6], u(x, 1, t) = e^t \tan[\pi(1+y)/6].$$

The exact solution is  $u(x, y, t) = e^t \tan[\pi(x+y)/6]$

Bi-Conjugate Gradient method				A.D.I method		
$\Delta t = h$	relative error	time	order	relative error	time	order
1/20	1.0265e-004	0.19		1.1431e-004	0.07	
1/40	2.4869e-005	0.55	2.0453	6.3867e-005	0.14	0.8398
1/80	6.1116e-006	3.52	2.0247	3.3412e-005	0.89	0.9347
1/160	1.5142e-006	38.08	2.0130	1.7051e-005	6.01	0.9705

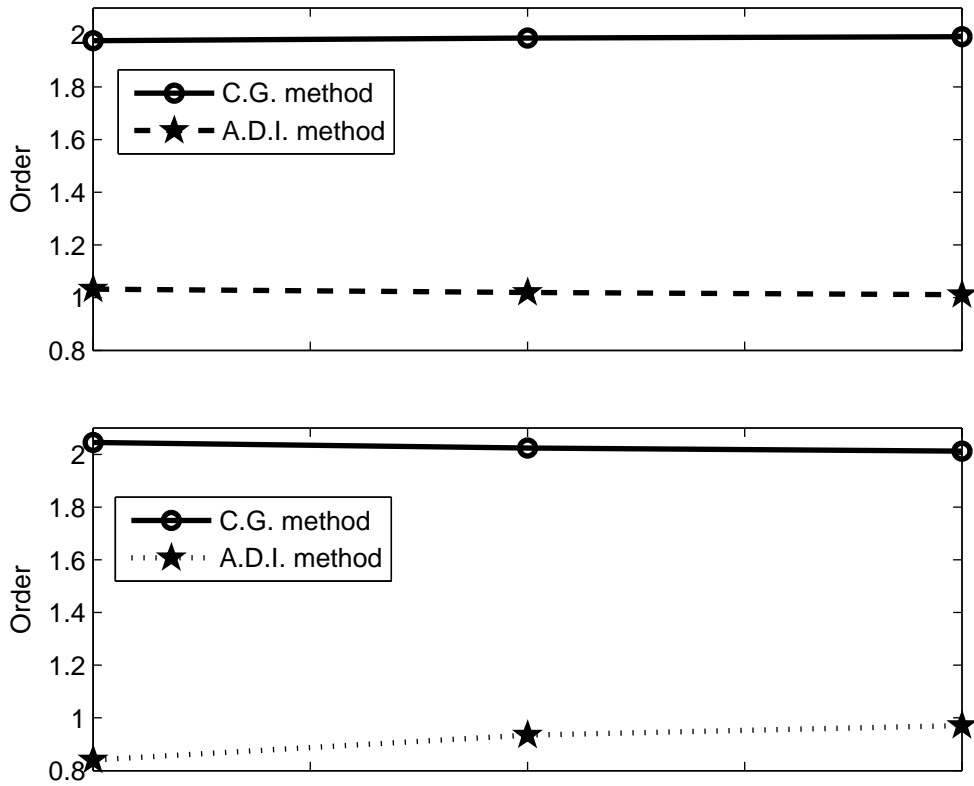


Figure 5: Order of accuracy of case 4, 5, respectively. ( y axis represent order of accuracy)

To the A.D.I. method, we see that the order of accuracy is first order only since we deal with the mixed derivative term as explicit type. Iterative methods still maintain the second order of accuracy, in spite the mix-derivative part makes block tri-diagonal matrix more complex. Mix-derivative term does not effect CPU time of A.D.I. method because the structure of tri-diagonal linear system do not change at all. By fitting the data, we can see the tendency of increasing of CPU time, and the result is given by figure 6.

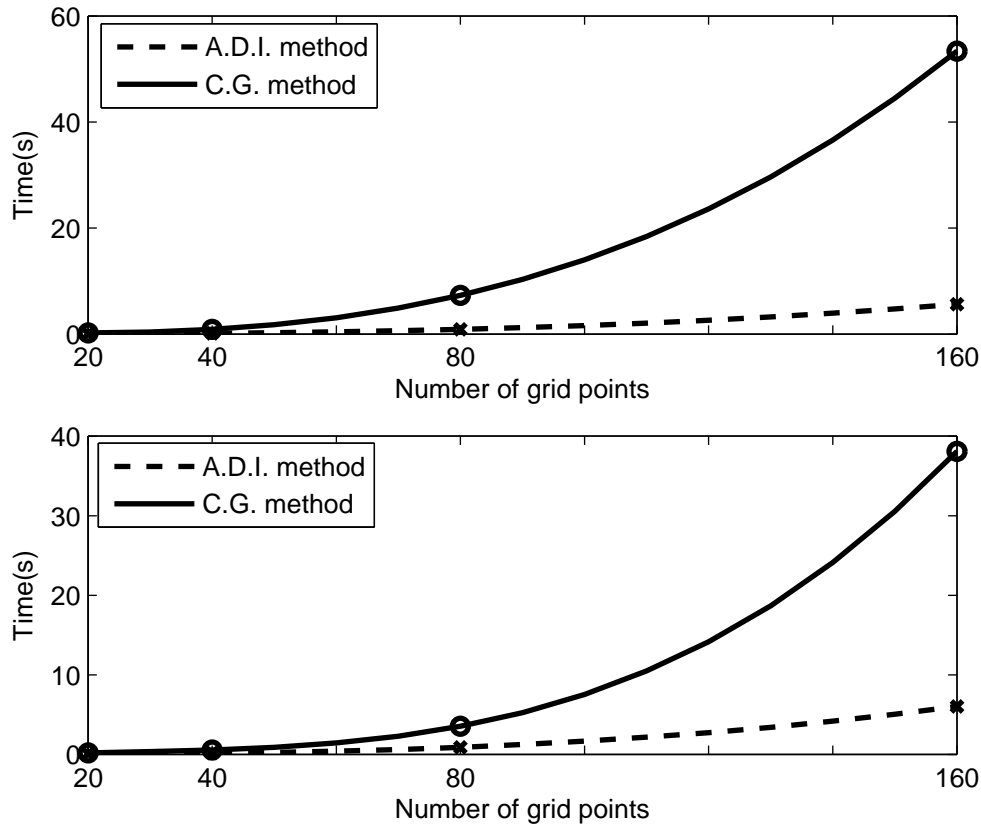


Figure 6: CPU time of case 4, 5, respectively.

### 4.3 Conclusion

By using A.D.I. and iterative methods to solve heat equations without mix-derivative term, the second order of accuracy has shown in figure 3. In calculating process, iterative methods cost more time than A.D.I. method, the defect will react to efficiency of simulating. Since A.D.I method solves two tri-diagonal systems during a single time step, the time consuming of anisotropic diffusion problem is as well as heat equation. As we employ iterative methods to the same problem, mix-derivative term makes coefficients of matrix complex. Therefore iterative methods take more time to deal with the large sparse matrix. If we need to refine the grid to search for higher precise simulations, A.D.I. method can outperform iterative method with a

speedy calculating.

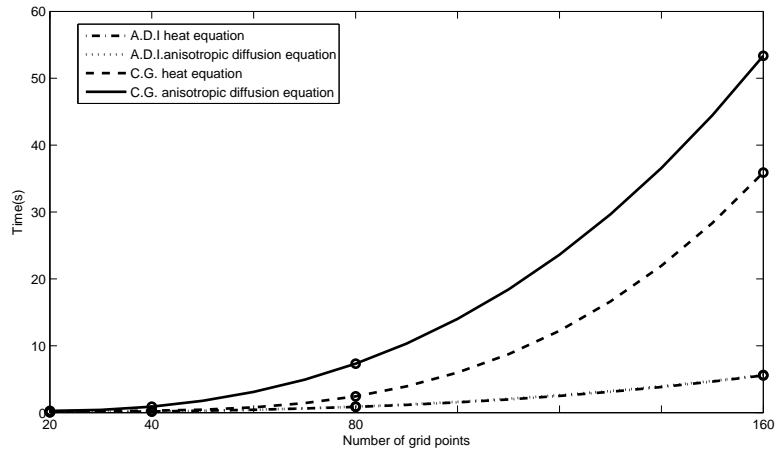
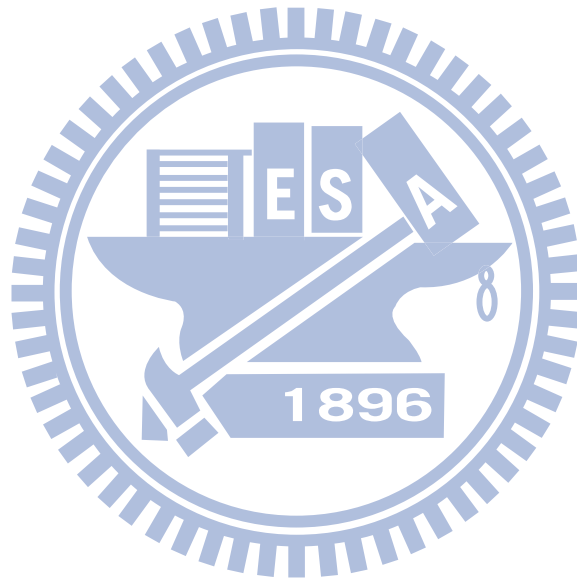


Figure 7: A case satisfied heat and anisotropic diffusion, we fit the curve of data solved by A.D.I. and iterative methods. and compare with the CPU time.



## References

- [1] Michele Benzi, C. D. Meyer and Miroslav Tuma, "A sparse approximation inverse preconditioner for the conjugate gradient method", SIAM J. Sci. Comput. Vol.7, No.5, September (1996) pp. 1135-1149
- [2] R. M. Beam and R. F. Watming, "Alternating direction implicit methods for parabolic equation with a mixed derivative", SIAM J. Sci. Stat. Comput. Vol.1, No.1 March (1980).
- [3] P. Concus, G. H. Golub and G. Meurant, "Block preconditioning for the conjugate gradient method", SIAM J. Sci. Stat. Comput. Vol.6, No.1, January (1985).
- [4] J. (Jr.) Douglas and J. E. Gunn, "A general formulation of alternating direction methods. Part 1. Parabolic and hyperbolic problems", Num Math., Vol.6 (1964), pp. 428-453.
- [5] U. Diewald, T. PreuBer, and M. Rumpf, "Anisotropic diffusion in vector field visualization on Euclidean domains and surfaces", Ieee Transactions On Visualization and Computer Graphics, Vol. 6, No. 2, April-June (2000).
- [6] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization", SIAM J. Optimization, Vol.2, No.1, February (1992) pp. 21-42.
- [7] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems", Journal of Research of the National Bureau of Standards. Vol.49, No.6, December (1952) Research Paper 2379.

- [8] A. R. Mitchell and G. Fairweather "Improved forms of the alternating direction methods of Douglas, Peaceman and Rachford for solving parabolic and elliptic equations", Num. Math., Vol.6 (1964), pp. 285-292.
- [9] S. McKee and A. R. Mitchell, "Alternating direction methods for parabolic equations in two space dimensions with a mixed derivative", The Computer Journal. Vol.13, No.1, March (1970).
- [10] S. McKee and A. R. Mitchell, "Alternating direction methods for parabolic equations in three space dimensions with a mixed derivative", The Computer Journal. Vol.14, No.3, February (1970).
- [11] D. W. Peaceman and H. H. Rachford, Jr., "The numerical solution of parabolic and elliptic differential equations", J. Soc. Indust. Appl. Math. Vol.3, No.1, March (1955). pp. 28-41
- [12] T. K. Sarkar, "On the Application of the Generalized BiConjugate Gradient Method", Journal of Electromagnetic Waves and Applications, Vol.1, No.3, (1987). pp. 223-242.
- [13] O. B. Widlund, "On stability of certain difference schemes", Maths. Comp., Vol.5, (1965). pp. 201-210.
- [14] E. L. Wachspress and G. J. Habetler. J, "An alternating direction implicit iteration technique" , Soc. Indust. Appl. Math. Vol.8, No.2, June (1960).
- [15] Li Wang, Jun Zhang, "A new stabilization strategy for incomplete LU preconditioning of indefinite matrices", Applied Mathematics and Computation 144 (2003) pp. 75-87.