NEW PARALLEL GROUP ACCELERATED OVERRELAXATION ALGORITHMS FOR THE SOLUTION OF 2-D POISSON AND DIFFUSION EQUATIONS

by

FOO KAI PIN

Thesis submitted in fulfillment of the requirements for the degree of Master of Science

December 2011

ACKNOWLEDGEMENTS

First and foremost, I would like to express my specially thank to my supervisors, Associate Professor Dr. Norhashidah Binti Mohd. Ali for her mentorship. I am honoured and lucky to be one of her students. She has given me a lot of helpful guidelines for my research experiments and providing a conducive environment for me to do this research. Her valuable advice on data interpretation, numerical scheme techniques, and continuous encouragement throughout the study are gratefully acknowledged.

I would like to extend my special thanks to my co-researchers, Mr. Ng Kok Fu and Mr. Khoo Kok Teong for their advisements on parallel computing and Message Passing Interface techniques.

I would also want to extend my gratitude to all the staffs in Aurora and Stealth, School of Computer Science for their gracious and invaluable helps.

I also would like to express my obligations and an appreciation to all the staff in School of Mathematics Science and Universiti Sains Malaysia for providing me with the opportunity and allowing me to use their facilities throughout my research.

Finally, I would like to thank my loved wife and family for their comforting words and encouragement. Thank you again to all of the above named and those who have not been mentioned for helping me in completing this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

AOR	Accelerated Over-Relaxation
EDG	Explicit Decoupled Group
EG	Explicit Group
MEDG	Modified Explicit Decoupled Group
MEG	Modified Explicit Group
MPI	Message Passing Interface
PDEs	Partial Differential Equations
SOR	Successive Over-Relaxation

LIST OF PUBLICATIONS & SEMINARS

K. P. Foo and N. H. M. Ali , Parallel Group Explicit Accelerated OverRelaxation Methods On Distributed Memory Multicomputer, Paper presented at Simposium Kebangsaan Sains Matematik Ke-16, 3-5 Jun 2008, Hotel Renassaince, Kota Bahru KELANTAN.

K. P. Foo and N. H. M. Ali , Parallel Group Explicit Accelerated OverRelaxation Methods On Distributed Memory Multicomputer, *Journal of Communication and Computer, September 2010, vol.7, No.9* ,ISSN 1548-7709,USA, 45–54.

N. H. M. Ali and K. P. Foo, Performance Analysis of the Modified Group AOR Algorithms In Elliptic PDEs, Paper in proceeding to be presented at International Conference on Computational and Information Sciences17th Dec 2010 – 19th Dec 2010, Chengdu, Sichuan, China

ALGORITMA PENGENDURAN BERLEBIHAN TERPECUT

KUMPULAN SELARI BARU BAGI

PENYELESAIAN PERSAMAAN POISSON 2-MATRA DAN RESAPAN

ABSTRAK

Kaedah beza terhingga biasanya digunakan untuk menyelesaikan persamaan pembezaan separa (PPS) yang timbul dalam bidang mekanik bendalir dan termodinamik. Namun demikian, pendiskretan PPS ini lazimnya menghasilkan suatu sistem persamaan linear yang besar dan jarang di mana skema ini mengambilkan masa yang panjang untuk menyelesaikan masalah.

Pembangunan dalam kaedah lelaran terpecut dan teknologi pengkomputeran selari berupaya untuk mengatasi masalah ini. Skema lelaran titik yang berasaskan pendiskretan lima titik biasa putaran lazimnya digunakan untuk menyelesaikan persamaan Poisson. Selain itu, skema lelaran blok atau berkumpulan, di mana titik-titik-titik grid dikelompokkan ke dalam blok atau kumpulan, didapati mengurangkan bilangan lelaran yang diperlukan dan masa pelaksanaan kerana penyelesaian bagi titik-titik grid dikemaskini dalam blok atau kumpulan tetapi bukan titik demi titik. Antara skema-skema lelaran berkumpulan, kaedah kumpulan tak tersirat (EG) dan kaedah kumpulan nyah pasangan tak tersirat (EDG) telah banyak dikaji dan terbukti bahawa menpunyai penumpuan yang lebih cepat berbanding dengan kaedah titik titik. Untuk mempercepatkan lagi penumpuan bagi kaedah ini, kaedah pemecutan yang biasa seperti kaedah pengenduran berlebihan berturut-turut (Successive OverRelaxation, SOR)dan kaedah pengenduran berlebihan terpecut (Accelerated

OverRelaxation, AOR) telah diterapkan ke dalam kaedah-kaedah ini dan telah mengurangkan bilangan lelaran yang diperlukan.

Misalnya, Martins et al. (2002) telah merumuskan kaedah kumpulan tak tersirat bagi AOR (EG (AOR)) di mana telah mengurangkan bilangan lelaran yang diperlukan berbanding dengan kaedah-kaedah lelaran titik AOR. Ali & Lee (2007) mengembangkan kaedah kumpulan nyah pasangan tak tersirat bagi AOR (EDG (AOR)) dalam penyelesaian persamaan pembezaan separa eliptik dengan menggunakan kaedah lima titik putaran. Pengambilan masa penumpuan bagi EDG (AOR) telah menunjukkan pengurangan masa pelaksanaan berbandingkan dengan EG (AOR) di mana EDG (AOR) memerlukan operasi aritmetik yang lebih rendah untuk menyelesaikan masalah. Baru ini, Rakhimov & Othman (2008) membangunkan MEG (AOR) (Modified EG (AOG)) pada tahun 2008 di mana hasil kaji mereka menunjukkan kemajuan jika berbanding dengan kaedah EDG (AOR).

Dalam tesis ini, perumusan MEDG (AOR) (Modified EDG (AOR)) akan dibentangkan dalam penyelesaian persamaan Poisson dan persamaan resapan berperingkat masa. Prestasi bagi ujikaji berangka dan kompleksiti komputer akan dibincangkan dan dibandingkan dengan usaha yang sebelum ini. Akhirnya, kaedah-kaedah ini akan dilaksanakan ke atas gugusan bagi computer ingatan teragih dengan menggunakan persekitaran pengaturcaraan antara muka penghantaran mesej (Message-Passing Interface programming) untuk menentukan perbandingan bagi kecekapan kaedah-kaedah ini dengan menggunakan beberapa jenis saiz grid dan bilangan pemproses. Analisis skalabiliti akan juga ditunjukkan untuk membandingkan masa yang sebenar dengan masa yang diramalkan.

NEW PARALLEL GROUP ACCELERATED OVERRELAXATION ALGORITHMS FOR THE SOLUTION OF 2-D POISSON AND DIFFUSION EQUATIONS

ABSTRACT

Finite difference method is commonly used to solve partial differential equations (PDEs) which arise from fluid mechanics and thermodynamics problem. However, the discretization of these PDEs oftenly lead to large sparse linear systems which require large amount of execution times to solve.

The development in accelerated iterative techniques and parallel computing technologies can be utilized to surmount this problem. Point iterative schemes which are based on the standard five point discretization and the rotated five point discretization are commonly used to solve the Poisson equation. In addition, block or group iterative schemes where the mesh points are grouped into block have been shown to reduce the number of iterations and execution timings because the solution at the mesh points can be updated in groups or blocks instead of pointwise. Among these group iterative schemes, the Explicit Group (EG) method (Yousif and Evans, 1986) and Explicit Decoupled Group (EDG) method (Abdullah, 1990) have been extensively researched and have been shown to converge faster than their pointwise counterparts. In order to improve the rate of convergence of these methods, the common accelerated methods such as Successive OverRelaxation (SOR) method and Accelerated OverRelaxation (AOR) method may be applied to these methods and have been shown to reduce the number of iterations.

Martins et al. (2002) for example, formulated the Explicit Group AOR (EG (AOR)) method which was shown to decrease the number of iterations if compared with the point AOR iterative schemes. Ali and Lee (2007) developed the Explicit Decoupled AOR (EDG (AOR)) method in solving the elliptic partial differential equations by using the rotated 5-point AOR method. The gains in timings of EDG (AOR) method show lesser execution timings over the EG (AOR) method since required lower arithmetic operation to solve the problem. Recently, Rakhimov & Othman (2009) developed Modified EG (AOR) (MEG (AOR)) in 2009 where their experimental results show an improvement if compared with EDG (AOR) method.

In this thesis, the formulation of the Modified EDG (AOR) (MEDG (AOR)) method is presented in solving the Poisson and the time-dependent diffusion equation. The performance of the numerical experiments and the computation complexity will be discussed and compared with the previous works. Finally, these methods will be implemented on a cluster of distributed memory computer using Message-Passing interface programming environment to establish the comparison for the efficiency of these methods using several grid size and number of processors. The scalability analysis will also be presented to compare the actual timings with the predicted timings of these methods.

CHAPTER TWO

PRELIMINARIES

2.1 Classification Of PDEs

An ordinary differential equations (ODEs) is a mathematical equation for an unknown function which consists of function of only one independent variable that relates the values of the function itself whereas PDEs consists of derivatives of function that involves an unknown function (or functions) of two or more independent variables and their partial derivatives with respect to those variables. A PDEs for the function $u(x, y, z)$ is the form of

$$F\left(x, y, z, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial^3 u}{\partial x^3}, \frac{\partial^3 u}{\partial x^2 \partial y}, \frac{\partial^3 u}{\partial x \partial y^2}, \frac{\partial^3 u}{\partial y^3}, \mathbf{L} \quad \right) = 0 . \tag{2.1}$$

PDEs are used to formulate the solution of problems involving functions of several variables for most of the physical problems. The general form of PDEs of second order in two independent variables can be shown as

$$A(x,y)\frac{\partial^2 u}{\partial x^2} + B(x,y)\frac{\partial^2 u}{\partial x \partial y} + C(x,y)\frac{\partial^2 u}{\partial y^2} + D(x,y)\frac{\partial u}{\partial x} + E(x,y)\frac{\partial u}{\partial y} + F(x,y)u = G(x,y) , \tag{2.2}$$

where A, B, C, D, E, F and G are constant or independent of u. Therefore, the equation is homogeneous if $G(x, y)$ is zero for all values x and y.

The linear PDEs can be classified into three categories,

a) Elliptic PDEs if $B^2 - 4AC < 0$. The best known elliptic equations are

i. Poisson equation: $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$, and

ii. Laplace equation: $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$.

b) Parabolic PDEs if $B^2 - 4AC = 0$. The best known parabolic equations are

i. Heat equation: $\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}$ where a^2 is a constant, and

ii. Convection-diffusion equation: $\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} - b \frac{\partial u}{\partial x}$ where a^2 and b are constants.

c) Hyperbolic PDEs if $B^2 - 4AC > 0$. The best known hyperbolic equations are

i. Wave equation: $\frac{\partial^2 u}{\partial t^2} = b^2 \frac{\partial^2 u}{\partial x^2} 0$ where b^2 is a constant.

The solutions of the same category of PDEs have the same characteristic and solving method. Therefore, we will need to justify the type of the PDEs before using the correct numerical method.

The boundary of elliptic PDEs are normally composed in certain condition. For example, Figure 2.1 shows that the boundary conditions are specified around a closed region in a rectangle shape.

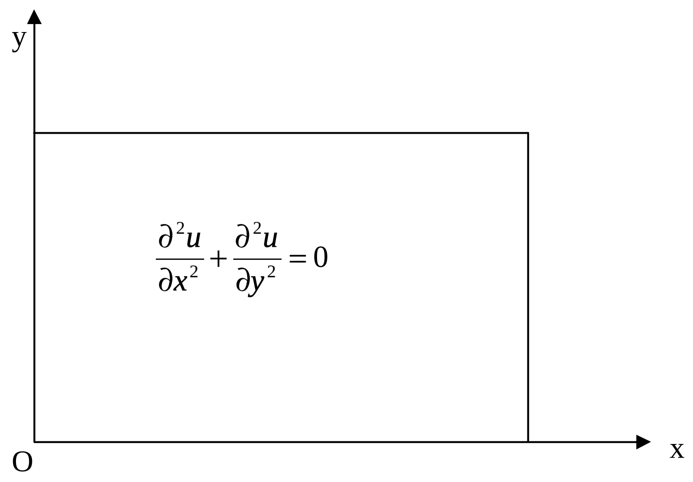
y
$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$
O
x

Figure 2.1: Domain for an elliptic PDEs.

For the parabolic PDEs, initial values or initial boundary values are provided. For example, Figure 2.2 shows that the boundary conditions are specified on the side of the open region. The solutions will move forward towards the open side.

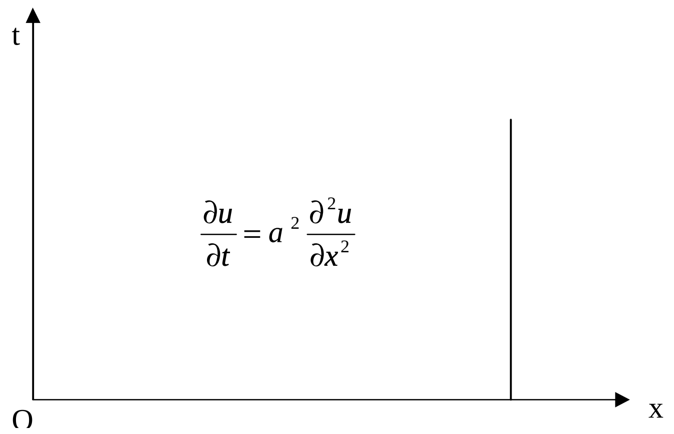
t
$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}$
O
x

Figure 2.2: Domain for a parabolic PDEs.

Generally, the initial condition and boundary condition can be specified in three ways:

a) Dirichlet boundary condition, where the unknown values of function *u* are given on each point of boundary for the domain.

$$u(0,t) = K_1, \qquad\qquad u(L,t) = K_2.$$

b) Neumann boundary condition, where the values of normal derivative are given on boundary for the domain.

$$\frac{\partial u(0,t)}{\partial x} = K_1, \qquad \frac{\partial u(L,t)}{\partial x} = K_2.$$

c) Robin's boundary condition, which is the combination of Dirichlet boundary condition and Neumann boundary condition where the values of function u and normal derivative are given on boundary for the domain.

$$u(0,t) = K_1, \qquad u(L,t) = K_2.$$

d) Cauchy initial condition, where the values of function u and its derivative are given from the beginning.

$$u(x,0) = K_1, \qquad \frac{\partial u(x,0)}{\partial t} = K_2.$$

2.2 Finite Difference method

Most of the PDEs are too complicated to be solved analytically. Therefore, finite difference method is used to replace partial derivative of PDEs to transform PDEs to algebraic equation system. The first step of the finite difference method is to divide domain solution into discrete grid. For example, the solution domain with square-shaped is divided to discrete grid as shown below:

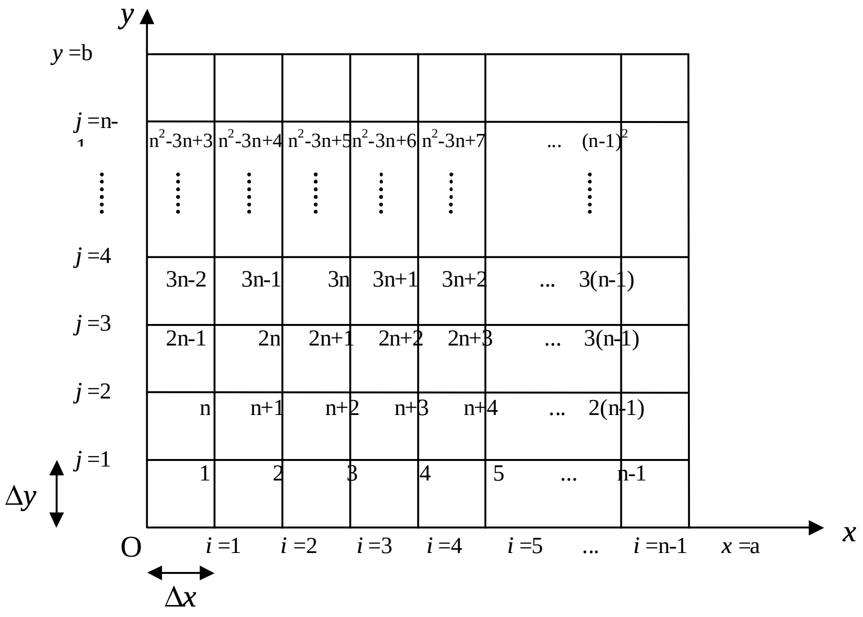

Figure 2.3: Discretized solution domain with square-shaped.

Each intersection point of these lines is naming as mesh point or grid. Here, the square-shaped domain involves $(n-1)^2$ internal mesh points. The values of x and y for each point are represented as $\Delta x_i = i\Delta x$ and $\Delta y_j = j\Delta y$ where $1 \le i, j \le n-1$. The value of u of the (i,j)th node will be written as $u_{i,j}$.

For example,

$$u(x_i, y_j) = u(i\Delta x, j\Delta y) = u_{i,j},$$

$$u(x_{i+1}, y_j) = u(x_i + \Delta x, y_j) = u((i+1)\Delta x, j\Delta y) = u_{i+1,j}.$$

Taylor series expansion is the most suitable method in order to obtain the approach used for the finite difference equations. The Taylor series of a function f in a neighbourhood of x_0 can be written as

$$f(x) = f(x_0) + \frac{(x-x_0)}{1!} f^{'}(x_0) + \frac{(x-x_0)^2}{2!} f^{''}(x_0) + \frac{(x-x_0)^3}{3!} f^{'''}(x_0)$$

$$+ \mathbf{L} + \frac{(x-x_0)^n}{n!} f^{(n)}(x_0) + \mathbf{L},$$

where function f and its derivatives are continuous on the closed interval [a,b]. This series will be called as Maclaurin series if x_0=0.

By expanding Taylor series with two variables like $u(x_i, y_{j+1})$ on the point (x_i, y_j), which is shown as below:

$$u(x_i, y_j + \Delta y) = u(x_i, y_j) + \frac{\Delta y}{1!}\frac{\partial u(x_i, y_j)}{\partial y} + \frac{(\Delta y)^2}{2!}\frac{\partial^2 u(x_i, y_j)}{\partial y^2} + \frac{(\Delta y)^3}{3!}\frac{\partial^3 u(x_i, y_j)}{\partial y^3} + \mathbf{L}$$

$$u_{i,j+1} = u_{i,j} + \frac{\Delta y}{1!}\frac{\partial u(x_i, y_j)}{\partial y} + \frac{(\Delta y)^2}{2!}\frac{\partial^2 u(x_i, y_j)}{\partial y^2} + \frac{(\Delta y)^3}{3!}\frac{\partial^3 u(x_i, y_j)}{\partial y^3} + \mathbf{L}\,. \tag{2.3}$$

Rearrange equation (2.3), we obtain

$$\frac{\partial u(x_i, y_j)}{\partial y} = \frac{u_{i,j+1} - u_{i,j}}{\Delta y} - \frac{(\Delta y)^2}{2!}\frac{\partial^2 u(x_i, y_j)}{\partial y^2} - \frac{(\Delta y)^3}{3!}\frac{\partial^3 u(x_i, y_j)}{\partial y^3} - \frac{(\Delta y)^4}{4!}\frac{\partial^4 u(x_i, y_j)}{\partial y^4}\mathbf{L}$$

$$= \frac{u_{i,j+1} - u_{i,j}}{\Delta y} + \mathrm{O}(\Delta y), \tag{2.4}$$

where $\mathrm{O}(\Delta y)$ denotes term containing higher powers of Δy. Assuming that $\mathrm{O}(\Delta y)$ is negligible if compared with lower powers of Δy,

$$\frac{\partial u(x_i, y_j)}{\partial y} \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta y}. \tag{2.5}$$

Equation (2.5) is called the forward difference formula.

Substitute Δy with $-\Delta y$ in equation (2.3), another new equation can be formed as

$$u(x_i, y_j - \Delta y) = u(x_i, y_j) + \frac{-\Delta y}{1!}\frac{\partial u(x_i, y_j)}{\partial y} + \frac{(-\Delta y)^2}{2!}\frac{\partial^2 u(x_i, y_j)}{\partial y^2} + \frac{(-\Delta y)^3}{3!}\frac{\partial^3 u(x_i, y_j)}{\partial y^3} + \mathbf{L}$$

or

$$u_{i,j-1} = u_{i,j} + \frac{-\Delta y}{1!}\frac{\partial u(x_i, y_j)}{\partial y} + \frac{(-\Delta y)^2}{2!}\frac{\partial^2 u(x_i, y_j)}{\partial y^2} + \frac{(-\Delta y)^3}{3!}\frac{\partial^3 u(x_i, y_j)}{\partial y^3} + \mathbf{L}\,. \qquad (2.6)$$

Rearrange equation (2.6) in term of $\frac{\partial u(x_i, y_j)}{\partial y}$. Same as the forward difference formula, we assume that $\mathrm{O}(\Delta y)$ is negligible if compared with lower powers of Δy .The backward difference formula can be written as

$$\frac{\partial u(x_i, y_j)}{\partial y} \approx \frac{u_{i,j} - u_{i,j-1}}{\Delta y}. \qquad (2.7)$$

By subtracting equation (2.7) from equation (2.5), we get

$$u(x_i, y_j + \Delta y) - u(x_i, y_j - \Delta y) = \frac{2\Delta y}{1!}\frac{\partial u(x_i, y_j)}{\partial y} + \frac{2(\Delta y)^3}{3!}\frac{\partial^3 u(x_i, y_j)}{\partial y^3} + \mathbf{L}$$

or

$$u_{i,j+1} - u_{i,j-1} = \frac{2\Delta y}{1!}\frac{\partial u(x_i, y_j)}{\partial y} + \frac{2(\Delta y)^3}{3!}\frac{\partial^3 u(x_i, y_j)}{\partial y^3} + \mathbf{L}\,. \qquad (2.8)$$

Rearrange equation (2.8) in term of $\frac{\partial u(x_i, y_j)}{\partial y}$,

$$\frac{\partial u(x_i, y_j)}{\partial y} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} + \frac{(\Delta y)^3}{6}\frac{\partial^3 u(x_i, y_j)}{\partial y^3} + \mathbf{L}$$

$$= \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} + \mathrm{O}(\Delta y^2)\,. \qquad (2.9)$$

Similarly, we assume that $\mathrm{O}(\Delta y^2)$ is negligible if compared with lower powers of Δy .The central difference formula can be written as

$$\frac{\partial u(x_i, y_j)}{\partial y} \approx \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y}. \tag{2.10}$$

Forward difference formula, backward difference formula and central difference formula for $\frac{\partial u(x_i, y_j)}{\partial x}$ can be obtained by using same method.

$$\frac{\partial u(x_i, y_j)}{\partial x} = \frac{u_{i,j+1} - u_{i,j}}{\Delta x} + \mathrm{O}(\Delta x), \qquad \text{forward difference formula.} \tag{2.11}$$

$$\frac{\partial u(x_i, y_j)}{\partial x} = \frac{u_{i,j} - u_{i,j-1}}{\Delta x} + \mathrm{O}(\Delta x), \qquad \text{backward difference formula.} \tag{2.12}$$

$$\frac{\partial u(x_i, y_j)}{\partial x} = \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta x} + \mathrm{O}(\Delta x^2), \qquad \text{central difference formula.} \tag{2.13}$$

By adding equation (2.5) with equation (2.7), we can obtain,

$$\frac{\partial^2 u(x_i, y_j)}{\partial x^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta x^2} + \mathrm{O}(\Delta x^2). \tag{2.14}$$

Equation (2.14) is called central difference formula for the second order partial derivative $\frac{\partial^2 u(x_i, y_j)}{\partial x^2}$.

With the same method, central difference formula for the second order partial derivative $\frac{\partial^2 u(x_i, y_j)}{\partial y^2}$ can be written as,

$$\frac{\partial^2 u(x_i, y_j)}{\partial y^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} + \mathrm{O}(\Delta y^2). \tag{2.15}$$

2.3 Point iterative method

We use stationary iterative methods to solve the linear equation,

$$A\underline{x} = \underline{b}, \tag{2.16}$$

where A is a given matrix and b is a given vector. Stationary iterative methods can be written as

$$\underline{x}^{(k+1)} = G\underline{x}^{(k)} + \underline{g}, \tag{2.17}$$

where neither G nor $\underline{g}$ depends upon the iteration count k. We decompose the matrix A into

$$A = D - L - U, \tag{2.18}$$

where D is a block diagonal matrix, L is a lower triangular matrix and U is an upper triangular matrix obtained from the matrix A.

Three types of stationary iterative methods will be introduced in the next section.

For Jacobi method, equation (2.16) can be rewritten as

$$D\underline{x} = (L + U)\underline{x} + \underline{b}. \tag{2.19}$$

By assuming D^{-1} exist, multiplying both sides of (2.19) by D^{-1},

$$\underline{x} = D^{-1}(L + U)\underline{x} + D^{-1}\underline{b}. \tag{2.20}$$

The Jacobi iterative method can be defined as

$$\underline{x}^{(k+1)} = G\underline{x}^{(k)} + \underline{g}, \tag{2.21}$$

where

$$G = D^{-1}(L+U) \text{ and } \underline{g} = D^{-1}\underline{b}. \tag{2.22}$$

For Gauss-Seidel (GS) method, each the updated values will be used as they are available. Hence, equation (2.16) can be rewritten as

$$(D-L)\underline{x} = U\underline{x} + \underline{b}. \tag{2.23}$$

The GS method is defined as

$$D\underline{x}^{(k+1)} = L^{(k+1)} + U\underline{x}^{(k)} + \underline{b}. \tag{2.24}$$

Multiplying both sides of equation (2.24) by $(D-L)^{-1}$

$$\underline{x}^{(k+1)} = (D-L)^{-1}U\underline{x}^{(k)} + (D-L)^{-1}\underline{b}. \tag{2.25}$$

Equation (2.25) can be defined as

$$\underline{x}^{(k+1)} = G\underline{x}^{(k)} + \underline{r}, \tag{2.26}$$

where

$$G = (D-L)^{-1}U \text{ and } \underline{r} = (D-L)^{-1}\underline{b}. \tag{2.27}$$

The Successive OverRelaxation (SOR) method is a modified version of GS method where an acceleration parameter w is used to accelerate the rate of the convergence. Let $\underline{x}^{(k+1)}$ be the vector obtained from the GS method,

$$\underline{x}^{(k+1)} = D^{-1}L^{(k+1)} + D^{-1}U\underline{x}^{(k)} + D^{-1}\underline{b}. \tag{2.28}$$

The extrapolation factor w are introduced into equation (2.28),

$$\underline{x}^{(k+1)} = w\overset{-}{\underline{x}}^{(k+1)} + (1-w)\underline{x}^{(k)}. \tag{2.29}$$

Substitute $\overset{-}{\underline{x}}^{(k+1)}$ from equation (2.24) into equation (2.29) and we get

$$D\underline{x}^{(k+1)} = w(L^{(k+1)} + U\underline{x}^{(k)} + \underline{b}) + (1-w)D\underline{x}^{(k)}. \tag{2.30}$$

Equation (2.30) can be rewritten as

$$(I - wD^{-1}L)\underline{x}^{(k+1)} = (wD^{-1}U + (1-w)I)\underline{x}^{(k)} + wD^{-1}\underline{b}. \tag{2.31}$$

Multiplying both sides of equation (2.31) by $(I - wD^{-1}L)^{-1}$ since $I - wD^{-1}L$ is non-singular for any choice of w,

$$\underline{x}^{(k+1)} = L_w\underline{x}^{(k)} + (I - wD^{-1}L)^{-1}wD^{-1}\underline{b}, \tag{2.32}$$

where

$$L_w = (I - wD^{-1}L)^{-1}(wD^{-1}U + (1-w)I). \tag{2.33}$$

and L_w is the SOR iteration matrix.

When $w = 1$, GS method will be obtained.

As we mention in Chapter 1.2, the five point iterative scheme is the foundation of other finite difference iterative methods. Therefore, we will concentrated on standard five point iterative method and rotated five point iterative method in this thesis for the point iterative method.

2.3.1 Standard five point (SOR) iterative method for elliptic PDEs

Consider the second order elliptic equation which is the Poisson equation,

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x, y), \tag{2.34}$$

with Dirichlet boundary conditions and function f are given. Here, we assume that the domain is the unit square. Assume that the grid spacing $h = 1/n$ with $x_i = ih$ and $y_j = jh$ where $(i, j = 0,1,2, \ldots, n)$. Equation (2.34) can be approximated at the point (x_i, y_j) based on (2.14) and (2.15). We get

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} \approx f_{i,j}. \tag{2.35}$$

Simplify (2.35) will get

$$u_{i,j} \approx \frac{1}{4}(u_{i,j-1} + u_{i,j+1} + u_{i+1,j} + u_{i-1,j} - h^2 f_{i,j}), \tag{2.36}$$

which is known as standard five point iterative scheme and is the most common used in solving Poisson equation.

By applying the SOR iterative scheme (2.29) into equation (2.36), we get standard five point (SOR) iterative method for elliptic PDEs,

$$u_{i,j}^{(k+1)} = (1-w)u_{i,j}^{(k)} + \frac{w}{4}(u_{i-1,j}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k)} - h^2 f_{i,j}). \tag{2.37}$$

2.3.2 Rotated five point (SOR) iterative method for elliptic PDEs

By rotating the i-plane axis and the j-plane axis clockwise by 45^{o} which surrounds the point, the computational molecule will becomes as Figure 2.4.

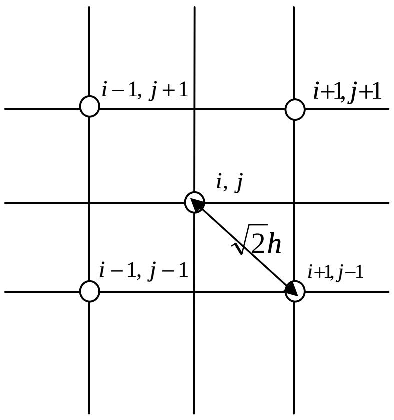

Figure 2.4: The computational molecule by rotating the i-plane axis and the j-plane axis clockwise by 45^{o}.

The transformations are shown as below,

- $i, j \pm 1$ is taken place by $i \pm 1, j \pm 1$,
- $i \pm 1, j$ is taken place by $i \pm 1, j \mp 1$,

and h is taken place by $\sqrt{2}h$.

Therefore, equation (2.34) can be derived from the rotated five point finite difference approximation (Dahlquist and Bjorck 1974),

$$\frac{u_{i-1,j-1} - 2u_{i,j} + u_{i-1,j+1}}{2h^2} + \frac{u_{i+1,j-1} - 2u_{i,j} + u_{i+1,j+1}}{2h^2} \approx f_{i,j}\,, \tag{2.38}$$

and we obtain

$$u_{i,j} \approx \frac{1}{4}(u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i+1,j+1} - 2h^2 f_{i,j})\,. \tag{2.39}$$

By applying the SOR iterative scheme (2.29) into Equation (2.39), we get rotated five point (SOR) iterative method for elliptic PDEs,

$$u_{i,j}^{(k+1)} = (1-w)u_{i,j}^{(k)} + \frac{w}{4}(u_{i-1,j-1}^{(k+1)} + u_{i-1,j+1}^{(k)} + u_{i+1,j-1}^{(k+1)} + u_{i+1,j+1}^{(k)} - 2h^2 f_{i,j}) \,. \qquad (2.40)$$

2.4 Block Iterative method

Block iterative methods involve the update of the values for a block of points at a time which are different with point iterative methods. Therefore, a block of equation system has to be solved at a time.

Consider the linear equation $A\underline{x} = \underline{b}$ can be written as

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1q} \\ A_{21} & A_{22} & \cdots & A_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ A_{q1} & A_{q1} & \cdots & A_{qq} \end{bmatrix} \begin{bmatrix} \underline{X}_1 \\ \underline{X}_2 \\ \vdots \\ \underline{X}_q \end{bmatrix} = \begin{bmatrix} \underline{B}_1 \\ \underline{B}_2 \\ \vdots \\ \underline{B}_q \end{bmatrix}, \qquad (2.41)$$

where A is a square matrix of order n. A_{ij} is submatrix of order $n_i \times n_j (n_1 + n_2 + \cdots + n_q = n)$. $\underline{X}_i$ and $\underline{B}_i$ are the subvectors of order n_i.

Matrix A can be decomposed into

$$A = D - (L + U)\,, \qquad (2.42)$$

where D is a block diagonal matrix, $-L$ is a lower triangular matrix and $-U$ is an upper triangular matrix obtained from the matrix A.

$$D = \begin{bmatrix} A_{11} & & & & \\ & A_{22} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & A_{qq} \end{bmatrix}, -U = \begin{bmatrix} 0 & A_{12} & A_{13} & \cdots & A_{1q} \\ & 0 & A_{23} & \cdots & A_{2q} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & A_{q-1,q} \\ & & & & 0 \end{bmatrix}$$

$$, -L = \begin{bmatrix} 0 & & & & \\ A_{21} & 0 & & & \\ A_{31} & A_{32} & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ A_{q1} & A_{q2} & \cdots & A_{q,q-1} & 0 \end{bmatrix}. \tag{2.43}$$

Therefore, the block Jacobi iterative method can be defined as

$$A_{ii} \underline{X}_i^{(k+1)} = -\sum_{\substack{i=1 \\ j \neq i}}^{q} A_{ij} \underline{X}_j^{(k)} + \underline{B}_i , \qquad 1 \le i \le q , \tag{2.44}$$

or

$$\underline{X}_i^{(k+1)} = \sum_{\substack{i=1 \\ j \neq i}}^{q} G_{ij} \underline{X}_j^{(k)} + \underline{C}_i , \qquad 1 \le i \le q , \tag{2.45}$$

where

$$G_{ij} = \begin{cases} -A_{ij}^{-1} A_{ij}, & if\ i \neq j \\ 0, & if\ i = j \end{cases},$$

and

$$\underline{C}_i = A_{ij}^{-1} B_j , \qquad 1 \le i \le q . \tag{2.46}$$

Equation (2.44) can be rewritten in general form as,

$$\underline{X}^{(k+1)} = G^{(p)} \underline{X}^{(k)} + \underline{C}^{(p)} , \tag{2.47}$$

where $G^{(p)} = (D^{(p)})^{-1}[L^{(p)} + U^{(p)}]$ is the block Jacobian matrix and $\underline{C}^{(p)} = (D^{(p)})^{-1}\underline{B}$,

$$D^{(p)} = diag_{(p)}(A) \quad . \tag{2.48}$$

The block GS iterative method can be written as

$$A_{ii}\underline{X}_i^{(k+1)} = -\sum_{j=1}^{i-1} A_{ij}\underline{X}_j^{(k+1)} - \sum_{j=i+1}^{q} A_{ij}\underline{X}_j^{(k)} + \underline{B}_i, \qquad 1 \le i \le q, \tag{2.49}$$

or

$$\underline{X}_i^{(k+1)} = \sum_{ij=1}^{i-1} G_{ij}\underline{X}_j^{(k+1)} + \sum_{j=i+1}^{q} G_{ij}\underline{X}_j^{(k)} + \underline{C}_i, \qquad 1 \le i \le q, \tag{2.50}$$

where G_{ij} and $\underline{C}_i$ are as given in (2.46).

Equation (2.50) can be rewritten in general form as,

$$\underline{X}^{(k+1)} = G^{(p)}\underline{X}^{(k)} + (D^{(p)} - L^{(p)})^{-1}\underline{B}, \tag{2.51}$$

where $G^{(p)} = (D^{(p)} - L^{(p)})^{-1}U^{(p)}$ is the block GS matrix. (2.52)

For the block SOR iterative method, we get

$$\underline{X}^{(k+1)} = L_w{}^{(p)}\underline{X}^{(k)} + w(I - wG^{(p)})^{-1}\underline{C}^{(p)}, \tag{2.53}$$

where $L_w{}^{(p)} = [I - w(D^{(p)})^{-1}L^{(p)}]^{-1}[w(D^{(p)})^{-1}U^{(p)} + (1-w)I]$ is the block GS matrix ,

$\underline{C}^{(p)} = (D^{(p)})^{-1}\underline{B}$, and w is the relaxation parameter.

For the block iterative method, it can be extended to blocks of one line or more lines (or groups) which contain a group of mesh points. The block iterative methods which are constructed with group by group are called as group iterative method. The EG method and EDG method are the most common four point group iterative method recently which are introduced as alternative numerical methods for the solution of elliptic PDEs because of the high rate of convergence and the satisfying computation complexity.

2.4.1 Explicit group SOR (EG(SOR)) method for elliptic PDEs

Consider Equation (2.34) as model problem, we assume that any group of four points in solving a domain by using standard five point iterative scheme (2.36). A (4x4) system of equation is produced which is shown as:

$$\begin{bmatrix} 4 & -1 & 0 & -1 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i+1,j+1} \\ u_{i,j+1} \end{bmatrix} = \begin{bmatrix} u_{i-1,j} + u_{i,j-1} - h^2 f_{i,j} \\ u_{i+2,j} + u_{i+1,j-1} - h^2 f_{i+1,j} \\ u_{i+2,j+1} + u_{i+1,j+2} - h^2 f_{i+1,j+1} \\ u_{i-1,j+1} + u_{i,j+2} - h^2 f_{i,j+1} \end{bmatrix}. \tag{2.54}$$

This 4x4 matrix in equation (2.54) can be inverted to

$$\begin{bmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i+1,j+1} \\ u_{i,j+1} \end{bmatrix} = \frac{1}{24} \begin{bmatrix} 7 & 2 & 1 & 2 \\ 2 & 7 & 2 & 1 \\ 1 & 2 & 7 & 2 \\ 2 & 1 & 2 & 7 \end{bmatrix} \begin{bmatrix} u_{i-1,j} + u_{i,j-1} - h^2 f_{i,j} \\ u_{i+2,j} + u_{i+1,j-1} - h^2 f_{i+1,j} \\ u_{i+2,j+1} + u_{i+1,j+2} - h^2 f_{i+1,j+1} \\ u_{i-1,j+1} + u_{i,j+2} - h^2 f_{i,j+1} \end{bmatrix}. \tag{2.55}$$

The EG method is then defined as,

$$\begin{aligned}
u_{i,j} &= \frac{1}{24}(7b_1 + s_2 + b_4) \\
u_{i+1,j} &= \frac{1}{24}(7b_2 + s_1 + b_3) \\
u_{i,j+1} &= \frac{1}{24}(7b_3 + s_1 + b_2) \\
u_{i+1,j+1} &= \frac{1}{24}(7b_4 + s_2 + b_1)
\end{aligned} \quad , \tag{2.56}$$

where

$$\begin{aligned}
b_1 &= u_{i-1,j} + u_{i,j-1} - h^2 f_{i,j} \\
b_2 &= u_{i+2,j} + u_{i+1,j-1} - h^2 f_{i+1,j} \\
b_3 &= u_{i-1,j+1} + u_{i,j+2} - h^2 f_{i,j+1} \qquad & s_1 &= 2\times(b_1 + b_4) \\
b_4 &= u_{i+2,j+1} + u_{i+1,j+2} - h^2 f_{i+1,j+1} \qquad & s_2 &= 2\times(b_2 + b_3)
\end{aligned} \quad , \tag{2.57}$$

which was developed by Yousif and Evans (1986).

By applying the SOR iterative scheme (2.29) into equation (2.56) until (2.57), we get EG (SOR) method which is shown as below:

$$\begin{aligned}
u_{i,j}^{(k+1)} &= \frac{w}{24}(7b_1 + s_2 + b_4) + (1-w)u_{i,j}^{(k)} \\
u_{i+1,j}^{(k+1)} &= \frac{w}{24}(7b_2 + s_1 + b_3) + (1-w)u_{i+1,j}^{(k)} \\
u_{i,j+1}^{(k+1)} &= \frac{w}{24}(7b_3 + s_1 + b_2) + (1-w)u_{i,j+1}^{(k)} \\
u_{i+1,j+1}^{(k+1)} &= \frac{w}{24}(7b_4 + s_2 + b_1) + (1-w)u_{i+1,j+1}^{(k)}
\end{aligned} \quad , \tag{2.58}$$

where

$$\begin{aligned}
b_1 &= u_{i-1,j}^{(k)} + u_{i,j-1}^{(k)} - h^2 f_{i,j} \\
b_2 &= u_{i+2,j}^{(k+1)} + u_{i+1,j-1}^{(k)} - h^2 f_{i+1,j} \\
b_3 &= u_{i-1,j+1}^{(k)} + u_{i,j+2}^{(k+1)} - h^2 f_{i,j+1} \qquad & s_1 &= 2\times(b_1 + b_4) \\
b_4 &= u_{i+2,j+1}^{(k+1)} + u_{i+1,j+2}^{(k+1)} - h^2 f_{i+1,j+1} \qquad & s_2 &= 2\times(b_2 + b_3)
\end{aligned} \quad . \tag{2.59}$$