FOUR POINT HIGH ORDER COMPACT ITERATIVE SCHEMES FOR THE SOLUTION OF THE HELMHOLTZ EQUATION

TENG WAI PING

UNIVERSITI SAINS MALAYSIA

2015

FOUR POINT HIGH ORDER

COMPACT ITERATIVE SCHEMES FOR THE

SOLUTION OF THE HELMHOLTZ EQUATION

by

TENG WAI PING

Thesis submitted in fulfillment of the requirements

for the Degree of

Master of Science

August 2015

SKEMA LELARAN PADAT

PERINGKAT TINGGI EMPAT TITIK BAGI

PENYELESAIAN PERSAMAAN HELMHOLTZ

oleh

TENG WAI PING

Tesis yang diserahkan untuk

memenuhi keperluan bagi

Ijazah Sarjana Sains

Ogos 2015

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude and respect to my supervisor, Prof. Dr. Norhashidah Mohd. Ali, for the continuous help and support throughout all the stages of my study and research, with her patience, motivation, enthusiasm, and immense knowledge. Her advice and insight were invaluable to me. Her guidance helped me all the time of research and writing of this thesis. I would also like to thank her for being an open person to ideas, and for encouraging and helping me to shape my interest and ideas. I attribute the level of my Masters degree to her encouragement and effort and without her this thesis, too, would not have been completed or written.

In addition, I would like to extend my special thanks to Prof. Jun Zhang for enlightening me with his research in high order solutions, and thank him for the stimulating discussions and insightful comments.

. Also, I am grateful to Fundamental Research Grant Scheme (203/PMATHS/6711321) which has provided the financial support needed to produce and complete my thesis.

Last but not the least, I would like to thank my family, for believing in me and supporting me throughout my life, for their continuous love and their supports in my decisions. Without whom I could not have made it here.

TABLES OF CONTENTS

LIST OF ABBREVIATIONS

PDE	Partial differential equation
MG	Multigrid method
MMGR	Multiscale Multigrid method combined with Richardson's extrapolation and operation based interpolation
EG	Explicit group
EDG	Explicit decoupled group
MEG	Modified explicit group
2D	two dimensional
FW	full-weighted
HW	half-weighted
FOC	fourth order compact
Ω^h	fine grid
Ω^{2h}	coarse grid

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

SKEMA LELARAN PADAT PERINGKAT TINGGI EMPAT TITIK BAGI PENYELESAIAN PERSAMAAN HELMHOLTZ

ABSTRAK

Teknik-teknik yang lebih baik diperoleh daripada beza terhingga dalam grid piawai dan grid putaran telah dibangunkan sejak beberapa tahun kebelakangan ini dalam menyelesaikan sistem linear yang terhasil daripada pendiskretan persamaan pembezaan separa (PDEs). Selain itu, satu sistem dengan peringkat kejituan yang lebih tinggi boleh dihasilkan daripada pendiskretan skema beza terhingga dengan menggunakan satu skim padat dengan kejituan peringkat empat yang dihasilkan daripada beza memusat dengan kejituan peringkat kedua. Dengan menggunakan beza terhingga padat ini, satu skim titik putaran dengan kejituan peringkat empat bagi persamaan Helmholtz dua dimensi (2D) yang baru terbentuk. Skim peringkat empat dalam grid piawai dan grid putaran boleh dikembangkan menjadi skim kumpulan ataupun sistem yang berperingkat empat. Sehubungan itu, kaedah multigrid berskala-multi digabungkan dengan ekstrapolasi Richardson diperkenalkan oleh Zhang [18] untuk menyelesaikan persamaan Poisson 2D. Dengan menggabungkan skim/sistem peringkat empat dan kaedah multigrid berskala-multi dengan ekstrapolasi Richardson dalam penyelesaian persamaan Helmholtz 2D, kejituan penyelesaian yang dianggarkan boleh diperbaik sehingga peringkat enam, dan walaupun dengan saiz grid yang lebih besar, kadar penumpuan dengan menggunakan kaedah lelaran ini adalah lebih cepat juga. Ujikaji berangka dijalankan pada skim putaran yang digabungkan dengan kaedah multigrid berskala-multi dan ekstrapolasi Richardson, dan hasilnya dibandingkan dengan kaedah-kaedah titik/kumpulan yang sedia ada dengan tatacara multigrid. Keputusan menunjukkan peningkatan dalam kadar penumpuan dan kecekapan lelaran skim yang baru digubal.

FOUR POINT HIGH ORDER COMPACT ITERATIVE SCHEMES FOR THE SOLUTION OF THE HELMHOLTZ EQUATION

ABSTRACT

Improved techniques derived from the standard and rotated finite difference operators have been developed over the last few years in solving linear systems that arise from the discretization of various partial differential equations (PDEs) [14]. Furthermore, a higher order system can be generated from discretization of the finite difference scheme by using the fourth order compact scheme generated from the second order central difference. By using compact finite differences, new standard and rotated point schemes with fourth order accuracy for the two-dimensional (2D) Helmholtz equation are formulated in this thesis. The fourth order point schemes in both standard and rotated grids can be further applied to formulate a fourth order system to be used as group iterative method in their respective grid. On the other hand, the multiscale multigrid method combined with Richardson's extrapolation is first introduced by Zhang [18] to solve the 2D Poisson equation. By combining all the fourth order schemes, multiscale multigrid method and Richardson's extrapolation in the solution of the 2D Helmholtz equation, the order of accuracy of the approximation can be improved up to sixth order, and with larger mesh size, the convergence rate of these iterative methods is faster as well. Numerical experiments are conducted on all the schemes combined with multiscale multigrid method and Richardson's extrapolation, and the results are compared with existing point and group methods solved by using the multigrid method. The results show the improvements in the convergence rate and the efficiency of the newly formulated iterative schemes/systems.

CHAPTER 1

PRELIMINARIES

1.1 Introduction

Most physical phenomena or engineering simulations, such as heat or fluid flow, human brain medical imaging, and global climate change, can be represented by mathematical formulation in the form of partial differential equations (PDEs). By considering finite difference approximations, a continuous problem in the form of a PDE can be changed into a discrete problem. The arising linear system, can be solved by using numerical methods. Numerical method, particularly the iterative methods, is a common place today to solve those scientific and engineering problems of great complexity, due to the high-speed computers that can solve the repeated arithmetic operations without getting tired and where the approximate answer is obtained from a sequence of improved estimates. Over the years, in order to solve the PDEs, mathematicians and engineers have strived to develop efficient and scalable algorithms that are more efficient or faster in terms of execution time, and at the same time, there is increasing demand for higher resolution simulations [15].

The simplest finite difference approximation was derived by Euler and was then developed and applied by different scientists, from one-dimensional to higher systems, in their computational algorithms. The finite difference methods are easier to implement and higher order accuracy can be obtained by deriving higher order compact difference schemes [5], [18]. On the other hand, the algebraic solution of finite differences is usually point-wise. It is extended to group-wise by Evans and Biggins, who introduce the Explicit Group (EG) method [13]. To reduce the computational complexity during the iterative process and thus shorten the execution time with comparable order of accuracy, instead of considering the formulation in the standard grid (also known as the full-sweep approach), Abdullah [2] introduces the formulation of stencils and systems in the rotated grid. The four-point Explicit Decoupled Group (EDG) (also known as the half-sweep approach) and Modified EG (MEG) (also known

as the quarter-sweep approach) methods are introduced by Abdullah [2], and Abdullah & Othman [3] respectively, to solve the two-dimensional (2D) Poisson equation. By using the half-sweep approach, Akhir et al [6] obtained second-order approximations for the 2D Helmholtz equation, by using the Gauss Seidel method, and proved that this approach is faster than that of the full-sweep approach.

Multigrid method is a well-known iterative method to efficiently solve the resulting sparse linear systems arising from the finite difference discretizations, and is able to accelerate convergence and improve accuracy of the algebraic solution [4], [5], [10], [16]. Zhang solves the 2D Poisson and Convection Diffusion equations by introducing multiscale multigrid method combined with Richardson extrapolation and operator based interpolation, to improve the order of accuracy of the approximations [18].

In this thesis, the concept of the stencil formulation derived from the compact finite differences will be applied in both the standard and rotated grids, point- and group-wise. Several newly-formulated point- or group-wise fourth-order schemes will be applied to the multiscale multigrid method. This technique will yield approximations on two grids with different scales. Upon applying Richardson's extrapolation, approximations accurate up to the sixth order will be obtained.

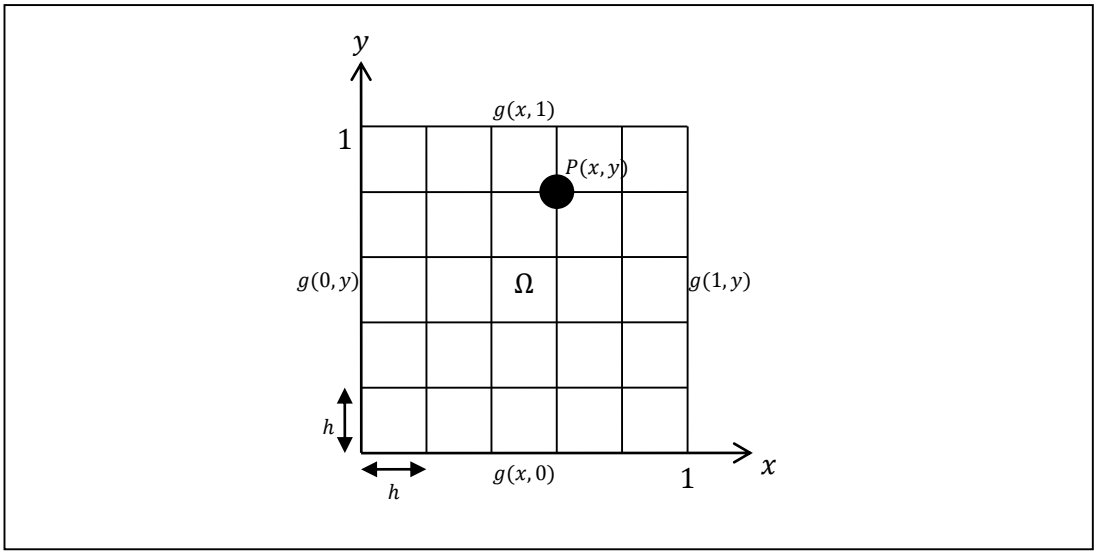

Figure 1.1: The continuous 2D solution domain.

1.2 Research Scope – Two-Dimensional Helmholtz Equation

The research scope is focused on the 2D elliptic equation, specifically the Helmholtz equation in the general form given by

$$u_{xx} + u_{yy} + k^2 u(x,y) = f(x,y), \qquad (x,y) \in \Omega \tag{1.1}$$

where Ω is a unit square solution domain $\Omega = [0,1] \times [0,1]$, with suitable Dirichlet boundary conditions defined on the boundary $\partial\Omega$, and satisfying the exact solution $u(x,y) = g(x,y)$ for each point $(x,y) \in \partial\Omega$. The exact solution $u(x,y)$ and the forcing function $f(x,y)$ are assumed to be sufficiently smooth and have the necessary continuous partial derivatives up to certain orders. The solution domain Ω is discretized uniformly in the x and y directions. The subinterval distance is denoted as $h = 1/n$, where n is the number of uniform intervals along the x- and y-axes. The 2D mesh points are (x_i, y_j) with $x_i = ih$ and $y_j = jh$, $0 \leq i,j \leq n$. The number of internal mesh points is given by m^2, where $m = n - 1$. The solutions of m^2 internal mesh points (x,y) of Eq. (1.1) can be approximated by various finite difference schemes in several ways, see Figure 1.1, as well as §3.1 and §4.1 for the standard and rotated grids respectively.

All the existing methods in both standard and rotated grid, namely the standard five- and nine-point stencil, EG $O(h^2)$, rotated five-point stencil and EDG $O(h^2)$, see Algorithms 3.1, 3.2, 3.3 and 4.1 respectively, as well as the newly formulated schemes, i.e. the EG $O(h^4)$, rotated nine-point stencil, EDG $O(h^2)$ and EDG $O(h^4)$, see Algorithms 3.4, 4.2, 4.3 and 4.4 respectively, will be run by using multigrid method, see Algorithm 2.1. The schemes of fourth-order accuracy, namely the standard nine-point stencil, rotated nine-point stencil, EG $O(h^4)$ and EDG $O(h^4)$ will be further applied with multiscale multigrid method combined with Richardson's extrapolation and operator based interpolation, see Algorithms 2.2 and 2.3, to retrieve sixth order accurate solution, which is the ultimate result desired in this research.

All the algorithms will be implemented by using C++ and run individually in different grid sizes of 8, 16, 32, 64 and 128. Note that, for all the four group methods, i.e. the EG and EDG of second and fourth order, there will always be grouped points and ungrouped points, see §2.1. Naturally, the grouped points will be solved by using the respective methods, but for the ungrouped points, point method will be used to tackle them. On the other hand, for all the methods in the rotated grid, there will be only certain points involved in the iterative process. The rest of the points will be solved by using direct solutions. In other words, the solution will be computed by using the standard five-point formula, see Algorithms 4.1, 4.2, 4.3 and 4.4.

Throughout the experiment, four important parameters are used to determine if the result obtained is satisfactory or not, i.e. the number of iterations, execution time, error and order of accuracy. The details of the parameters used will be elaborated in Chapter 5.

A summary of the research scope is given in Table 2.1 below. Altogether there will be a total of 8 different schemes, applied to 2 different methods. The respective expected order of accuracy of the approximations obtained and the algorithm to refer to are stated.

Table 2.1: Summary of the research scope.

Grid rotation	Point/Group	Scheme	Method	Order	Algorithm	Remark
Standard	Point	Five-point	MG	2	3.1 / 2.1	Existing
		Nine-point	MG	4	3.2 / 2.1	Existing
			MMG	6	3.2 / 2.2 / 2.3	New
	Group	EG $O(h^2)$	MG	2	3.3 / 2.1	Existing
		EG $O(h^4)$	MG	4	3.4 / 2.1	New
			MMG	6	3.4 / 2.2 / 2.3	New
Rotated	Point	Five-point	MG	2	4.1 / 2.1	Existing
		Nine-point	MG	4	4.2 / 2.1	New
			MMG	6	4.2 / 2.2 / 2.3	New
	Group	EDG $O(h^2)$	MG	2	4.3 / 2.1	Existing
		EDG $O(h^4)$	MG	4	4.4 / 2.1	New
			MMG	6	4.4 / 2.2 / 2.3	New

1.3 Problem Statement and Research Objective

The existing point- and group-wise schemes, in both standard and rotated grid are usually accurate up to fourth order, and the computational complexity increases with the stencils involved. Therefore, the main objectives of this research are:-

- To formulate the fourth-order four-point EG method, the fourth-order rotated nine-point stencil, and the fourth-order four-point EDG method in solving the 2D Helmholtz equation.
- To examine the application of multigrid method on all the stencils compared to that of multiscale multigrid method combined with Richardson's extrapolation on all the fourth order methods to obtain approximations up to sixth order.
- To compare the point- and group-wise methods in both standard and rotated grid, in terms of execution time and order of accuracy.
- To analyse the computational complexity of the developed methods.

1.4 Thesis Organization

The outline of this thesis is organized in the following way. Chapter 1 gives a brief idea on the overall concept for the thesis, including the research scope covering the 2D Helmholtz equation, as well as the research objective.

Chapter 2 covers the study on the system of linear equations and their solution. From these studies, we know that for large enough solution domain Ω^h, iterative solutions are suitable and more economic compared to direct solutions. The literature review of iterative methods used, such as Gauss Seidel, multigrid and multiscale multigrid method will be discussed.

The formulation of the finite differences in the standard grid, the standard five- and nine-point stencil as well as the EG $O(h^2)$ method for the 2D Helmholtz equation is widely known. From these, the detailed description on deriving the compact finite difference scheme and the EG $O(h^4)$ method will be given in Chapter 3. The implementation of each scheme will be given in algorithm form, and the details, such as the different restriction and interpolation operator used will be described with reason.

Chapter 4 is similar to Chapter 3, but gives information about the rotated grid. Since it is slightly newer than the standard grid, the derivation of the rotated finite differences will be elaborated, stencil and system formation, as well as the implementation of the methods, especially those newly-formulated methods, namely the rotated nine-point stencil, EDG $O(h^2)$ and EDG $O(h^4)$ method, which will be presented in §4.2, §4.3, §4.4 and §4.5 respectively.

In Chapter 5, the experimental results and analysis in terms of computational complexity of all the algorithms mentioned in Chapter 3 and 4 will be presented. The approximations obtained by different methods used will be compared and discussed.

Conclusions and remarks will be made in Chapter 6 from the analysis and some suggestions for future work on the samples studied will be considered.

CHAPTER 2

ITERATIVE METHODS

Most of the problems of elliptic PDEs will produce a large and sparse coefficient matrix when the partial derivatives are approximated using finite difference. To be exact, the application of the of the equations to each internal mesh points will result in a large and sparse system of linear algebraic equations as follow

$$Au = f \tag{2.1}$$

where A and f are a square nonsingular matrix with a column matrix, respectively, and u is a column matrix. Relatively, the matrix A becomes sparser when the number of equations increases. Thus iterative methods, which attempt to solve a problem by finding successive approximations to the solution starting from an initial guess, can be employed due to its efficiency in terms of computer memory requirements. They can be programmed to take advantage of the zeros in the coefficient matrix, are self-correcting in nature, and their very structure easily permits modifications such as under- and over-relaxation. To be useful the iteration must converge, but it is not considered to be effective unless the convergence is rapid. The level of accuracy of the approximate solution is pre-determined.

The purpose of this first half of this chapter is to give a literature review of the basic iterative methods (e.g. Jacobi, Gauss Seidel, SOR, multigrid method), which also fall naturally into two categories, point and group. A short description about the group method as an extension to the point iterative method is given in §2.1. The type of iterative methods discussed are the Gauss Seidel and multigrid methods, in which the multigrid method is presented in §2.2. Then in §2.3, the multiscale multigrid method introduced by Zhang [18] will be reviewed. Note that iterative methods are called relaxation (or smoothing) methods if they are used for the purpose of error smoothing in multigrid, see §2.2.1.

2.1 Point and Group Iterative Methods in the Standard and Rotated Grid

Point iterative methods are characterized by the explicit nature of calculation in which, at any one time only a single equation of the linear system is treated in each successive approximation. In other words, it intends to find the solution point-by-point in each iteration. For example, the formulations of point methods in the standard grid are called the standard five- and nine-point stencil, see Chapter 3; while that in the rotated grid are called the rotated five- and nine-point stencil, see Chapter 4.

An extension from the point iterative methods is the group iterative methods. The motivation in a group iterative method is to group a certain number of individual equations (points) and treat each group implicitly, i.e. similar to the way a single point is treated in the point iterative methods. In other words, it computes the solution of several linear systems in each iteration simultaneously. Since the value of the points around the block is used to calculate the value of the points inside the block, then the number of iterations using group iterative method is reduced, compared to that of the point iterative method. Despite the faster convergence rate, its computational complexity increases, as the number of points in a block is too large, which result in the increased number of arithmetic operations, and thus the execution time is high. *This is why only the four-point group is considered.* Given n number of discrete grid, where $m = n - 1$, such that the discrete grid consists of m^2 interior points. Then the number of four-point-groups in the grid Ω is given as $N = \left\lfloor \frac{m}{2} \right\rfloor^2$. For example, if the grid size $n = 8$, there will be 9 groups, see Figure 2.1,

$$N = \left\lfloor \frac{8-1}{2} \right\rfloor^2 = \left\lfloor \frac{7}{2} \right\rfloor^2 = 3^2 = 9 \text{ groups}$$

The group iterative methods in the standard grid are called EG method [13], while that in the rotated grid are called EDG method [2], [6], [12]. Note that even number of n will produce ungrouped points, see Figure 2.1.

2.2 Multigrid Method

Instead of using the basic iterative methods (e.g. Jacobi, Gauss Seidel or SOR method), the multigrid method is opted, because it is one of the fastest iterative methods, and most effective in solving a system of linear equations [4], [5], [15], [16], [17]. The basic concept of a multigrid method consists of the smoother and coarse grid correction, and it iterates on a successively coarser grids until the convergence is reached. The smoother is the specific iterative method used to smooth high frequency error; while the coarse grid principle is motivated where a smoothed error term is well approximated in a coarse grid, uses information from the coarse grids to approximate low frequency error. In fact, a coarse grid procedure is substantially less expensive (fewer grid points) than a fine grid procedure. Both smoothing and coarse grid principles will be combined, where the steps involved are restriction, smoothing and interpolation, which will be performed in between one complete cycle of a recursive multigrid algorithm to improve the approximation. In general, a multigrid algorithm consists of the smoother, residual computation, restriction, coarsest level processing, and interpolation.

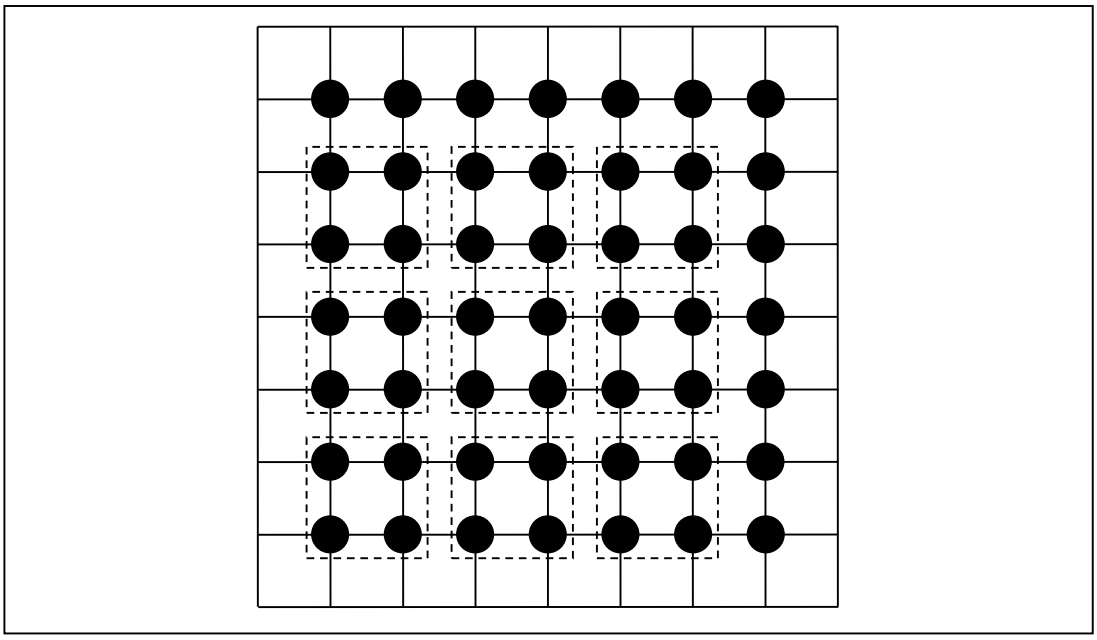

Figure 2.1: Number of four-point groups in a solution domain.

2.2.1 Smoother

The smoother as a component in a multigrid algorithm, see §2.2.6, is the chosen iterative method, which is applied to discrete elliptic problems, see Chapter 3 and Chapter 4, to smooth the errors or residuals in order to accelerate convergence of any approximation.

The errors usually consist of two components, i.e. the smooth errors at low frequency, and the oscillating errors at high frequency. After several iterations, the oscillating errors can be reduced, but the smooth errors remain. The smooth errors in the fine grid Ω^h can be seen more oscillating in the coarse grid. So, the smooth errors (which are less efficient with iterations in the fine grid) is expected to decrease rapidly if the iterations are performed in the coarse grid. This is the main reason why multigrid is used to improve the efficiency of the basic iterative algorithms (e.g. Gauss Seidel, SOR) [16].

The chosen method as the error smoother is the Gauss Seidel method, where the solution at each point (i, j) is improved in each iteration by using the latest approximations. The residual must be smoothed before it is transferred to the coarser grid Ω^{2h}. This is because the residual transferring process from the fine grid to the coarser grid will cause low frequency errors that will couple with the high frequency errors, and thus result in slower convergence of the point/group iterative method. This happens because the problem is changed from the finer to the coarser grid where an approximation is obtained with lower cost due to decreased number of grid points. The low frequency error is the error at the finer grid. Using the multigrid method, the high frequency error is smoothed in the fine grid, while the low frequency error is injected and smoothed at the coarser grid. The smoothed errors are interpolated back to the coarser grid before the grid correction operation is performed. The selection of smoothing scheme is important to ensure the errors are smoothed properly before transferring to the coarser grid, and vice versa [17].

In a complete V-cycle multigrid, see Algorithm 2.1, the only difference between point and group-wise scheme is only at each smoothing process, where the interior points of each hierarchical grid are treated either point-wise or group-wise.

2.2.2 Residual Computation

The residual equation plays an important role in the multigrid concept [17]. Consider the system of linear equations obtained from the discretization of the Helmholtz equation, written in matrix form as Eq. (2.1), can be approximated by a sequence of system of linear equations in the discrete form

$$A^i u^i = f^i, \qquad (x, y) \in \Omega^i$$

where $i = h, 2h, 4h, \dots, 2^d h$. Here, we are only interested in getting the approximation v^h at the fine grid, while the smoothed errors in the coarse grid will be used to improve the approximation.

There are two important measurements for v^h as an approximation to the exact solution u^h. One of them is the error of estimate, defined as

$$e^h = u^h - v^h \tag{2.2.1}$$

The error is also a vector. The size of an error vector is a standard measurement of any vector norm. The norm used for this purpose in this thesis is the maximum norm

$$\|e\|_\infty = \max_{i \geq 1, j \leq n} |e_{i,j}|, \tag{2.2.2}$$

Since the exact solution u is usually known, the error e is also accessible. However, in Ω^h, the residual r, that represents how v approximates u, is given by

$$r^h = f^h - A^h v^h$$

Rearranging it, we have

$$Av = f - r$$

$$Au - Av = f - (f - r) \qquad \text{(from the original equation } Au = f\text{)}$$

$$A^h(u^h - v^h) = r$$

$$A^h e^h = r^h \qquad (2.2.3)$$

Eq. (2.2.3) is known as the residual equation. Both Eq. (2.1) and Eq. (2.2.3) have the same form. Eq. (2.2.3) connects the error and residual parameters, while Eq. (2.1) connects the approximation and the one value of a given function. As a result of these similarities, any conditions imposed on Eq. (2.1) can be applied to Eq. (2.2.3). Eq. (2.2.3) will be used extensively, especially in coarse grid correction step, where the residual must be computed at each level before it can be restricted to the coarser grid.

2.2.3 Restriction

The main purpose of this process is to assign values from the fine grid Ω^h to the coarse grid Ω^{2h}. At each level of the solution domain, the residual r is calculated and transferred to the iterative points in the coarser grid Ω^{2h}, by using a restriction operator defined as $R_h^{2h}: \Omega^h \rightarrow \Omega^{2h}$. The new residual is then defined as $r^{2h} = R_h^{2h} r^h$ in the new grid, where $r^h \in \Omega^h$ and $r^{2h} \in \Omega^{2h}$. A new system of linear equations $A^{2h} e^{2h} = r^{2h}$ is formed in Ω^{2h}, where e^{2h} is the error value to be searched and will be smoothed by the Gauss Seidel method, see §2.2.1 and §2.2.2. Each residual must be smoothed properly before transferring to Ω^{2h}. This process is continued until the coarsest grid, where the equation formed is solved to obtain the error approximate value, see §2.2.4.

The different methods in Chapter 3 and Chapter 4 will require different restriction and interpolation operators to ensure the grid transfer, from the fine grid Ω^h to the coarse grid Ω^{2h},

or vice versa, can be performed more efficiently. There are two types of restriction operators (see Appendix A), namely the full-weighted (FW) and the half-weighted (HW) [17].

In stencil notation, the FW restriction operator used in the standard grid (see §3.3 and §3.5) is given as

$$R_h^{2h} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.2.4}$$

while that for the rotated grid (see §4.3 and §4.5) is given as

$$R_h^{2h} = \frac{1}{16}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 2 & 0 \\ 1 & 0 & 4 & 0 & 1 \\ 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.2.5}$$

The FW restriction operator is higher in accuracy, but the computational complexity is also relatively higher (and thus the longer execution time). When the accuracy is the priority, this operator will be opted.

On the other hand, the accuracy of the HW operator is good, and the execution time required is shorter compared to FW due to the lower computational complexity. In stencil notation, for the standard grid (see §3.2 and §3.4)

$$R_h^{2h} = \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.2.6}$$

and for the rotated grid (see §4.2 and §4.4)

$$R_h^{2h} = \frac{1}{8}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{2.2.7}$$

2.2.4 Coarsest Level Processing

The system in the coarsest level must be solved exactly. In ideal cases, the coarsest grid consists of one grid point. If the system is small enough, a single (or, more generally a few) sweeps of a direct method or some relaxation/smoothing methods (if it has sufficiently good convergence properties) may serve as a solver. Any other method may be chosen as well, as long as the process does not add significantly to the work count.

2.2.5 Interpolation

The interpolation, also known as the reverse grid transfer operation, transfers the smoothed residuals/errors e^{2h} from the coarse grid Ω^{2h} to the fine grid Ω^h within a V-cycle multigrid, see Algorithm 2.1, by using linear displacement and bilinear interpolation, where the interpolation operator is denoted as $P_{2h}^h: \Omega^{2h} \to \Omega^h$. In other words, a grid function $u^{2h} \in \Omega^{2h}$ gets interpolated to $u^h := P_{2h}^h u^{2h}$, where $u^h \in \Omega^h$ [17].

For the algorithms in the standard grid, see Chapter 3, since all the interior points are involved in the iteration, the derivation of the bilinear interpolation can be obtained when the stencil entries correspond to weights in a distribution process. Each interior points in the coarse grid will make contribution to the neighbouring points in the fine grid, as follow:-

$$\begin{aligned}
&v_{2i,2j}^h = v_{i,j}^{2h} && \text{for } 1 \le i,j \le n_c - 1 \\
&v_{2i+1,2j}^h = \tfrac{1}{2}(v_{i,j}^{2h} + v_{i+1,j}^{2h}) && \text{for } 0 \le i \le n_c - 1;\ 1 \le j \le n_c - 1 \\
&v_{2i,2j+1}^h = \tfrac{1}{2}(v_{i,j}^{2h} + v_{i,j+1}^{2h}) && \text{for } 1 \le i \le n_c - 1;\ 0 \le j \le n_c - 1 \\
&v_{2i+1,2j+1}^h = \tfrac{1}{4}(v_{i+1,j}^h + v_{i-1,j}^h + v_{i,j+1}^h + v_{i,j-1}^h) \\
&\qquad\qquad && \text{for } 1 \le i,j \le n_c - 1
\end{aligned} \qquad (2.2.8)$$

where n_c is the grid size of the coarser grid Ω^{2h}.

As for the algorithms in the rotated grid, see Chapter 4, based on Figure 4.3, only the ○ (and ● for some case) points are involved in the iteration, and thus the reverse-transfer process. To reverse-transfer the smoothed errors and values of the ○ (and ● for some case) points calculated, the linear displacement used to transfer the ○ (and ● for some case) points from the coarser grid Ω^{2h} to the finer grid Ω^{h} is as follow:-

$$
\begin{aligned}
&v_{2i,2j}^{h} = v_{i,j}^{2h} && \text{for } 1 \le i,j \le n_c - 1 \text{ both even or both odd} \\
&\text{and the bilinear interpolation is given by} \\
&v_{2i+2,2j}^{h} = \tfrac{1}{2}(v_{i,j}^{2h} + v_{i+2,j}^{2h}) \\
&\qquad\qquad \text{for } 1 \le i,j \le n_c - 2 \text{ both even or both odd} \\
&v_{2i,2j+2}^{h} = \tfrac{1}{2}(v_{i,j}^{2h} + v_{i,j+2}^{2h}) \\
&\qquad\qquad \text{for } 1 \le i,j \le n_c - 2 \text{ both even or both odd} \\
&v_{i,j}^{h} = \tfrac{1}{4}(v_{i+1,j+1}^{h} + v_{i+1,j-1}^{h} + v_{i-1,j+1}^{h} + v_{i-1,j-1}^{h}) \\
&\qquad\qquad\qquad \text{for all } 1 \le i,j \le n_c - 1 \text{ odd}
\end{aligned}
\qquad (2.2.9)
$$

where n_c is the grid size of the coarser grid Ω^{2h}.

Note that, due to the fact that the focus of this research is on the derivation of approximation with higher order accuracy, there is another method of interpolation used in this thesis, which is called the bicubic interpolation [18], see §2.3.2.

2.2.6 Multigrid Algorithm

Multigrid method operates in a sequence of solution domains with different sizes. The solution domain Ω^h is discretized and approximated in a sequence of discrete grids, starting from the finest grid Ω^h to a sequence of coarser grids $\Omega^{2h}, \Omega^{4h}, \ldots, \Omega^{2^{(d-1)}h}$ until the coarsest grid $, \Omega^{2^d h}$, finally go back to Ω^h. This is a cycle, and d indicates the depth. The grid Ω^h is known as the fine grid, while the remaining discrete grids are known as a sequence of coarse grids.

In a V-cycle multigrid, the iterations are performed where the approximation v^h is smoothed by using the appointed initial value by the smoothing scheme, see §2.2.1. Then the residual, see §2.2.2, is calculated and transferred to Ω^{2h} using a specific restriction operator, see §2.2.3. Every time the residual is transferred to the coarser grid, error smoothing is performed to obtain better approximations, and this process is continued until the coarsest grid, where the residual equation is solved directly, see §2.2.4. The reverse transfer process is performed by using interpolation, see §2.2.5, followed by the grid correction process, where the old approximation v^h is added with the smoothed error value e^h to obtain the latest error value. The smoothing operation is performed on the previously corrected approximation to get a new approximation, and one V-cycle is completed. This iteration is repeated until the convergence criteria are met, see Algorithm 2.1.

Algorithm 2.1: A V-Cycle Multigrid algorithm.

1. Pre-smooth $A^h v^h = f^h$ in the finest domain using the desired stencil.
2. Compute the residual $r^h \leftarrow f^h - A^h v^h$ and set $e^{2h} \leftarrow 0$.
3. Restrict $r^{2h} \leftarrow R_h^{2h} r^h$.
4. If coarsest grid, solve directly $A^{2h} e^{2h} = r^{2h}$. Otherwise, go back to step 1.
5. Improve the error and transfer back to the fine grid using interpolation and grid correction $v^h \leftarrow v^h + P_{2h}^h e^{2h}$.
6. Post-smooth $A^h v^h = f^h$ using the desired stencil.

2.3 Multiscale Multigrid Method Combined with Richardson's Extrapolation and Operator Based Interpolation

Zhang [18] developed the multiscale multigrid method and combined it with Richardson's Extrapolation and some high order interpolation to obtain a sixth order approximation for the 2D Poisson equation. This method, described in §2.3.1, and shown in Figure 2.2, will be applied by using only the fourth order stencils/systems in the standard and rotated grid, see Chapter 3 and Chapter 4.

2.3.1 Multiscale Multigrid Method

The multiscale multigrid method, similar to the full multigrid method, is used to elevate the order of accuracy of the computed solution. The major advantage of this method is that it has an optimal computational cost similar to that of a full multigrid, and can bring us the converged fourth order solutions on two grids with different scales, see Figure 2.2. The gray circle indicates the unconverged solution $u^{4h} \in \Omega^{4h}$, and the black circles are the fourth order converged solutions $u^{2h} \in \Omega^{2h}$ and $u^{h} \in \Omega^{h}$. The solid lines going downwards and upwards represent the restriction and interpolation process respectively. Note that the interpolation within V-cycles is the bilinear interpolation. After each V-cycle, bicubic interpolation, see §2.3.2, is used to interpolate the points to a finer grid, which is shown in the figure as double lines.

Unlike the full multigrid method in which the process starts from the coarsest grid, the multiscale multigrid method is initiated by running one or two cycles of V-cycle to get a better approximation, then the interpolated coarse grid solution is used as the initial guess for the fine-grid V-cycle. Therefore, relative to the full multigrid method, this algorithm will need fewer number of multigrid cycles than running the V-cycle on Ω^{h} and Ω^{2h} separately to get the converged fourth order solutions u^{h} and u^{2h}.

By solving any fourth-order scheme (in the standard or rotated grid, or point or group method), upon completing Algorithm 2.2, two sets of fourth-order converged solutions will be obtained, and applied with the Richardson extrapolation technique, see §4.3.1, to compute a sixth order accurate solution $\tilde{u}_{i,j}^{2h}$ on Ω^{2h}. Finally, the operator based interpolation, see §4.3.2, will be combined to make the remaining points in the fine grid to be all converged to sixth order.

Algorithm 2.2: Multiscale multigrid method

1. Run the V-Cycle multigrid on Ω^{4h} to get an approximate solution u^{4h}.
2. Use bicubic interpolation Eq. (2.3.1) to interpolate u^{4h} to Ω^{2h}, $u^{2h} = P_{4h}^{2h} u^{4h}$.
3. Relax $L^{2h}u^{2h} = f^{2h}$, and use u^{2h} as the initial guess to run the V-Cycle multigrid on Ω^{2h} until it converges. Now, the converged fourth order solution u^{2h} will be obtained.
4. Use bicubic interpolation Eq. (2.3.1) to interpolate u^{2h} to Ω^{h}, $u^{h} = P_{2h}^{h} u^{2h}$.
5. Relax on $L^{h}u^{h} = f^{h}$, and use u^{h} as the initial guess to run the V-Cycle multigrid on Ω^{h} until it converges. Now, the converged fourth order solution u^{h} will be obtained.

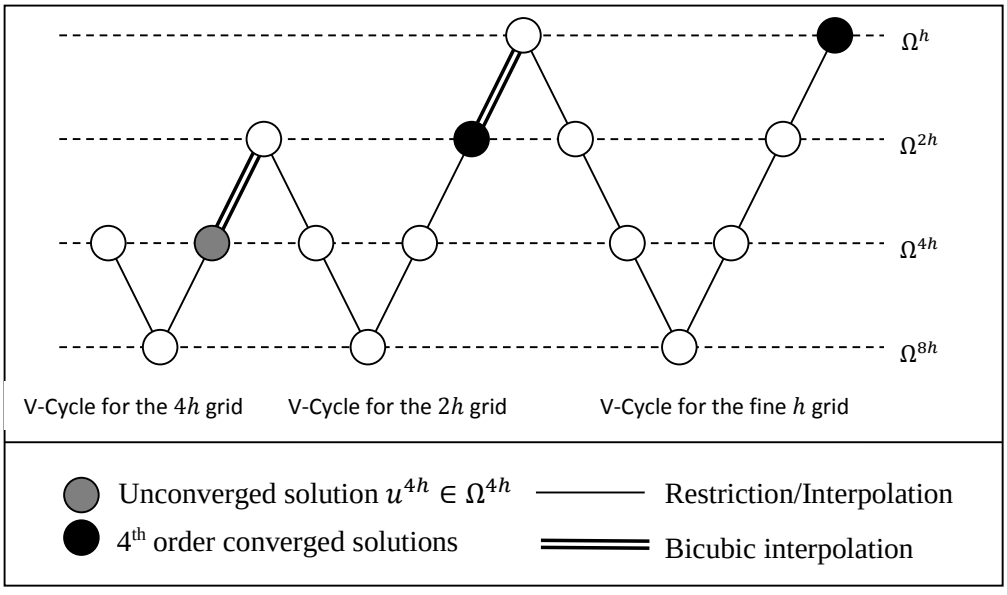

Figure 2.2: Representation of the multiscale multigrid method.

2.3.2 Bicubic Interpolation

The bilinear method takes the closest four diagonal points, and averages their values to produce the approximation for the middle point. Bicubic interpolation, in contrast, takes not only the four closest diagonal points, but their closest points as well, for a total of 16 points. Since this method makes use of more data, its results are generally smoother, thus it is opted in this research which focus on derivation of high order accuracy approximation, specifically during the multiscale multigrid method, see Algorithm 2.2. The bicubic interpolation [18] from the coarser grid Ω^{2h} to the finer grid Ω^{h} is given as follows:

$$
\begin{aligned}
v_{2i,2j}^{h} &= v_{i,j}^{2h} \\
v_{2i+1,2j}^{h} &= \tfrac{1}{16}\left(9v_{i,j}^{2h} + 9v_{i+1,j}^{2h} - v_{i+2,j}^{2h} - v_{i-1,j}^{2h}\right) \\
v_{2i,2j+1}^{h} &= \tfrac{1}{16}\left(9v_{i,j}^{2h} + 9v_{i,j+1}^{2h} - v_{i,j+2}^{2h} - v_{i,j-1}^{2h}\right) \\
v_{2i+1,2j+1}^{h} &= \tfrac{1}{24}(9v_{i,j}^{2h} + 9v_{i+1,j}^{2h} + 9v_{i,j+1}^{2h} + 9v_{i+1,j+1}^{2h} - v_{i-1,j-1}^{2h} \\
&\quad - v_{i-1,j}^{2h} - v_{i-1,j+1}^{2h} - v_{i-1,j+2}^{2h} - v_{i,j-1}^{2h} - v_{i,j+2}^{2h} - v_{i+1,j-1}^{2h} \\
&\quad - v_{i+1,j+2}^{2h} - v_{i+2,j-1}^{2h} - v_{i+2,j}^{2h} - v_{i+2,j+1}^{2h} - v_{i+2,j+2}^{2h})
\end{aligned}
\qquad (2.3.1)
$$

For the points at the boundary,

$$
\begin{aligned}
v_{1,2j}^{h} &= \tfrac{1}{17}\left(9v_{0,j}^{2h} + 9v_{1,j}^{2h} - v_{2,j}^{2h}\right) \\
v_{2n_c-1,2j}^{h} &= \tfrac{1}{17}\left(9v_{n_c-1,j}^{2h} + 9v_{n_c,j}^{2h} - v_{n_c-2,j}^{2h}\right) \\
v_{2i,1}^{h} &= \tfrac{1}{17}\left(9v_{i,0}^{2h} + 9v_{i,1}^{2h} - v_{i,2}^{2h}\right) \\
v_{2i,2n_c-1}^{h} &= \tfrac{1}{17}\left(9v_{i,n_c-1}^{2h} + 9v_{i,n_c}^{2h} - v_{i,n_c-2}^{2h}\right) \\
v_{2i+1,1}^{h} &= \tfrac{1}{4}\left(v_{i,0}^{2h} + v_{i+1,0}^{2h} + v_{i,1}^{2h} + v_{i+1,1}^{2h}\right)
\end{aligned}
$$

$$v_{2i+1,2n_c-1}^{h} = \frac{1}{4}\left(v_{i,n_c-1}^{2h} + v_{i+1,n_c-1}^{2h} + v_{i,n_c}^{2h} + v_{i+1,n_c}^{2h}\right)$$

$$v_{1,2j+1}^{h} = \frac{1}{4}\left(v_{0,j}^{2h} + v_{0,j+1}^{2h} + v_{1,j}^{2h} + v_{1,j+1}^{2h}\right)$$

$$v_{2n_c-1,2j+1}^{h} = \frac{1}{4}\left(v_{n_c-1,j}^{2h} + v_{n_c-1,j+1}^{2h} + v_{n_c,j}^{2h} + v_{n_c,j+1}^{2h}\right)$$

where $1 \leq i, j \leq n_c - 1$, and n_c is the grid size of the coarser grid Ω^{2h}.

2.3.3 Richardson's Extrapolation

After obtaining the converged approximation at the coarse and fine grid, Richardson's extrapolation will be considered, where the general form can be written as

$$\tilde{u}_{i,j}^{2h} = \frac{2^p u_{2i,2j}^{h} - u_{i,j}^{2h}}{2^p - 1}$$

where p is the order of accuracy of the different stencil/system used. After the extrapolation, the order of accuracy will be increased to $p + 2$. For example, if a five-point stencil is used, then $p = 2$. Since the approximation correct up to the sixth order is desired, this step is only applied together with all the fourth order methods, including the point and group methods in both the standard and rotated grids, as a comparison in different aspects. At this point, two sets of approximation converged to fourth order, i.e. at the fine and coarse grid will be obtained, and $p = 4$. Therefore, the Richardson extrapolation formula used is

$$\tilde{u}_{i,j}^{2h} = \frac{16u_{2i,2j}^{h} - u_{i,j}^{2h}}{15} \qquad (2.3.2)$$

The grid points which are involved in the iterative process at both the finest grid Ω^h and coarse grid Ω^{2h} will be applied with Eq. (2.3.2), and directly interpolated for only once as $\tilde{u}_{2i,2j}^{h} = \tilde{u}_{i,j}^{2h}$ and it keeps the sixth order accuracy. For the remaining points, the operator based interpolation scheme is used.

2.3.4 Operator Based Interpolation

A mesh refinement interpolation strategy is used to interpolate the sixth order accurate solution from Ω^{2h} to Ω^{h}. The 2D grid points is divided into four groups *(even, even)*, *(odd, odd)*, *(even, odd)*, and *(odd, even)*, see Figure 2.3. After Richardson's extrapolation, the *(even, even)* indexed grid points on Ω^{h} will be directly interpolated as $u^{h}_{2i,2j} = u^{2h}_{i,j}$, i.e. the *(even, even)* points are not involved in the iteration. For the remaining grid points, the operator based interpolation scheme is applied with the most current approximations, including the sixth order *(even, even)* points, to improve and achieve the sixth order accuracy. The operator based interpolation scheme is an iterative procedure. To establish the relationship between the values of the approximation at the *(odd, odd)*, *(even, odd)*, and *(odd, even)* grid points by using the sixth order approximation at the *(even, even)* points, some fourth order schemes will be used. The iteration will continue until the 2-norm of the correction vector is reduced to below a certain tolerance, see Algorithm 2.3.

Algorithm 2.3: Operator based interpolation iteration combined with the Richardson extrapolation technique.

1. Update every *(even, even)* grid point on Ω^{h}.

   By using the fourth order converged approximations in the coarse and fine grid, i.e. $\tilde{u}^{2h,k}_{i,j}$ and $\tilde{u}^{h,k}_{2i,2j}$, we first compute $\tilde{u}^{2h,k+1}_{i,j}$ by Eq. (2.3.2). This set of sixth order approximation will be directly interpolated to obtain $\tilde{u}^{h,k+1}_{2i,2j}$.

2. Update every *(odd, odd)* grid point on Ω^{h} by using a fourth order scheme.

3. Update every *(odd, even)* grid point on Ω^{h} by using a fourth order scheme.

4. Update every *(even, odd)* grid point on Ω^{h} by using a fourth order scheme.

5. Check the convergence. If not converged, repeat the iteration (i.e. go to step 2).

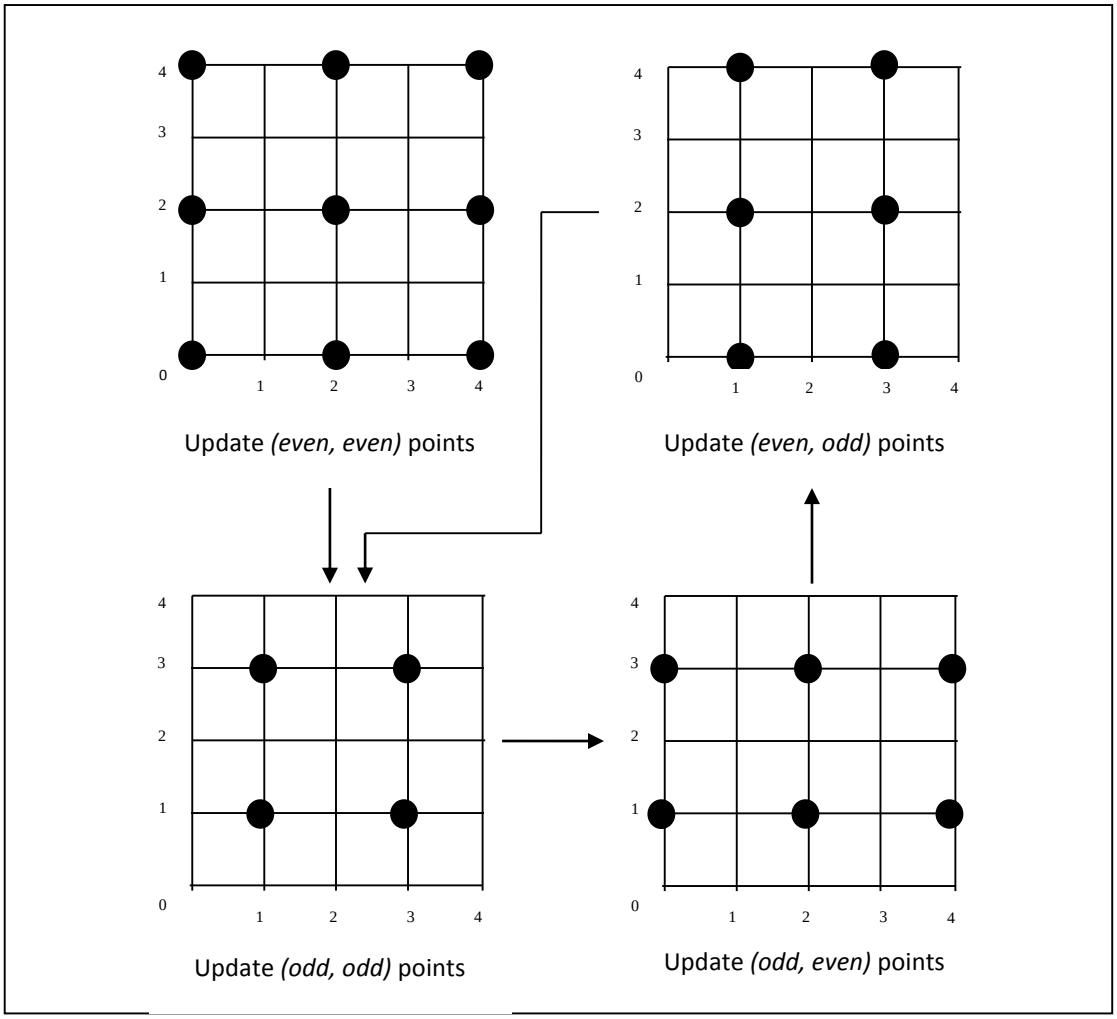

Figure 2.3: Illustration of the operator based interpolation scheme for a 5×5 fine grid.

CHAPTER 3

STANDARD GRID

The study on system of linear equations and their solution are described briefly in Chapter 2. From these studies, we know that for large enough Ω^h, iterative solutions are suitable and more economic compared to direct solutions.

Consequently, Chapter 3 consists of the formulation of stencil and system as well as the implementation of the algorithms in the standard grid, where §3.1 gives the finite differences in the standard grid, and details will be given in the formulation of the fourth order compact (FOC) scheme which is generated from the second order central differences. Then, the formulation of the five- and nine-point stencil of order $O(h^2)$ and $O(h^4)$ respectively, which are the result of discretization of the 2D Helmholtz Eq. (1.1), are given in §3.2 and §3.3 respectively. Note that instead of the standard nine-point stencil, the FOC scheme is applied to the derivatives, therefore the nine-point stencil is called the compact nine-point scheme. The implementation of the algorithm is given in their respective subsections. Then, §3.4 and §3.5 is on the formulation of system of group methods, derived from the five- and nine-point stencils, respectively. The second- and fourth-order system derived are called the EG method, denoted as EG (h^2) and EG $O(h^4)$ respectively. The compact nine-point stencil and the EG $O(h^4)$ method, both of fourth order accurate, will be applied together with multiscale multigrid method combined with Richardson's extrapolation.

By applying the newly formulated EG $O(h^4)$ with multiscale multigrid method with Richardson's extrapolation and operator based interpolation, sixth order approximations will be obtained, within the shortest execution time, relative to the point method in the standard grid.

3.1 Finite Differences for the Standard Grid

The finite difference of the standard grid is well-known, and thus, we will skip the elaboration in deriving them, but simply listing down some that we will use.

$$\delta_x^2 u_{i,j} = u_{xx} = \frac{1}{h^2}\left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right) \tag{3.1.1}$$

Eq. (3.1.1) is the *central difference* approximation for the standard grid in the x-direction, with truncation error $O(h^2)$. Note that, the notation $O(h^k)$ represents the *truncation error* of this approximation, where it is read (the term of) order h with the power k, denotes terms containing kth order and higher powers of h, and can be interpreted to mean that, when h is small enough, the term behaves essentially like a constant times h^k. From [8], by using Taylor series expansion, we have

$$\begin{aligned}
\delta_x^2 u_{i,j} &= u_{xx} + \frac{h^2}{12}u_{xxxx} + \frac{h^4}{360}u_{xxxxxx} + O(h^6) \\
&= \left(1 + \frac{h^2}{12}\partial_x^2\right)u_{xx} + O(h^4) \\
&= \left(1 + \frac{h^2}{12}\delta_x^2\right)u_{xx} + O(h^4) \qquad \text{where } f_{xx} = \delta_x^2 f + O(h^2)
\end{aligned}$$

$$\Longrightarrow u_{xx} = \left(1 + \frac{h^2}{12}\delta_x^2\right)^{-1}\delta_x^2 u_{i,j} \tag{3.1.2}$$

Eq. (3.1.2) is known as the *fourth order compact (FOC) approximation*. By using the same concept, the equations of u in the y-direction can be obtained, where

$$\delta_y^2 u_{i,j} = u_{yy} = \frac{1}{h^2}\left(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}\right) \tag{3.1.3}$$

is the *central difference* for the second order derivatives, with truncation error $O(h^2)$; and

$$u_{yy} = \left(1 + \frac{h^2}{12}\delta_y^2\right)^{-1}\delta_y^2 u_{i,j} \tag{3.1.4}$$

is the *FOC approximation* for the second order derivatives, with truncation error $O(h^4)$.