Department of Mathematics and Statistics

The New Variants of Modified Weighted Mean Iterative Methods for Fredholm Integro-Differential Equations

Elayaraja Aruchunan

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

December 2016

Declaration

To the best of my knowledge and belief, this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Elayaraja Aruchunan
December 2016

Abstract

The integro-differential equations are widely opted as mathematical models for many real world applications. In this research, linear integro-differential equations of Fredholm type (FIDEs) are considered and solved by proposed numerical methods. The primary objectives of this research are to introduce nine new variants of Weighted Mean iterative methods known as two-stage iterative methods. The new variants or the modified Weighted Mean (MWM) iterative methods are classified into three main groups which are the modified Arithmetic Mean (MAM), the modified Geometric Mean (MGM) and the modified Harmonic Mean (MHM) methods. To reduce the computational complexities in each group, techniques of the half-sweep and quarter-sweep iterations are applied to generate two new variants called as the Half-Sweep and the Quarter-Sweep MAM (MGM, MHM) methods, respectively. Convergence theorems of all the Half-Sweep and Quarter-Sweep MWM methods are proposed and their mathematical proofs are presented. Discretisation schemes based on the Central Difference method with 2-point, 3-point and 5-point Composite Closed Newton-Cotes formulae are used to transform the linear FIDEs to systems of algebraic equations. Effectiveness and efficiency of the proposed methods are compared in term of a number of iterations and usage of the CPU time. Computational complexity for all the proposed MWM iterative methods are analysed through numerical experiments via two boundary value problems of the second and fourth order FIDEs subjecting to Dirichlet and Neumann boundary conditions. The results indicate that the MHM are slightly better than the MGM method, and both MHM and MGM methods perform better than the MAM method. The implementation of the half-sweep and the quarter-sweep iteration techniques to the standard ones have reduced a number of iterations and CPU time. In addition, the Quarter-Sweep MWM methods are superior to other methods. Among the Quarter-Sweep MWM methods, the Quarter-Sweep MHM performs outstandingly in terms of the smallest number of iterations and the least CPU time.

List of publications during PhD candidature

- E. Aruchunan, B. Wiwatanapataphee and P. Jitsangiam. ”A New Variant of Arithmetic Mean Iterative Method for Fourth Order Integro-differential Equations Solution,” *2015 3rd IEEE International Conference on Artificial Intelligence, Modelling and Simulation (IEEE Computer Society)*, pp. 82-87, 2015. DOI 10.1109/AIMS.2015.24.
- E. Aruchunan, N. Khajohnsaksumeth and B. Wiwatanapataphee. ”A New Algorithm of Geometric Mean for Solving High-Order Fredholm Integro-differential Equations,” *Dependable, Autonomic and Secure Computing, 14th IEEE International Conference on Pervasive Intelligence and Computing, Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pp. 723-729, 2016. DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.128.
- E. Aruchunan, B. Wiwatanapataphee and X. Ge. ”An Efficient Dense-System Solver for Solving Fredholm Integral and Integro-differential Equations,” (Submitted).
- E. Aruchunan, B. Wiwatanapataphee and X. Ge. ”Modified Geometric Mean Method with Half-Sweep Iterative Technique for Solving High-Order Integro-differential Equations,” (Submitted).

Acknowledgements

All praise to lord Ganesha for His endless blessings, loves and enlightenments giving me the strength to complete this thesis very successfully.

First and foremost, my heartfelt thanks to my supervisor Professor Dr. Yonghong Wu for all his intellectual guidance, supervision and continuous professional supports throughout my studies.

My sincere gratitude to my co-supervisor Associate Professor Dr. Benchawan Wiwatanapataphee for all her constructive feedbacks and guidance to enhance my research skills.

I would like to thank my former supervisor Associate Professor Dr. Jumat Sulaiman who inspired me a lot in my early research career in applied mathematics.

A special thanks to my parents Mr. Aruchunan and Mrs. Letchumi for all their loves and supports throughout my education.

I also would like to thank the Department of Mathematics and Statistics administration staffs Cyril, Jeannie, Marcus and Petrina for their continuous helps and supports during my Ph.D. studies.

It is my radiant sentiment to place on record my best regards to my dearest friends Dr. Mohana Sundaram, Dr. Pawaton Kaemawichanurat, Dr. Nathnarong Khajohnsaksumeth, Dewi, Francisca, Juan, Lisa, Kathy and Amy for their endless moral supports during my time in Curtin University.

Finally, I would like to acknowledge International Postgraduate Research Scholarship (IPRS) and Australian Postgraduate Award (APA) for my Ph.D. studies in Australia.

Contents

CHAPTER 1

Introduction

1.1 Preliminaries

The studies of Integro-differential equations (IDEs) are growing fast and intensively due to its significance to many scientific, engineering and financial applications. The IDEs have been used to describe many physical phenomena such as Dirichlet problems, radiative equilibrium, elastic contact problems, heat transfer, diffusion process, astronomy, biology, potential theory, electrostatics, kinetics of nuclear reactors, mathematical neuroscience, nano-hydrodynamics, phenomena such as wind ripple in the desert, glass-forming process, biological process and option price. In many cases, the IDE models are significantly better than the partial differential equation models [19, 64, 69, 83, 92, 96, 100, 108, 117, 131–133].

Generally, the IDE models consist of equations in which the undetermined function exists under the combination form of the derivatives and the integral terms. Usually, such equations can be classified into Fredholm equations and Volterra equations which were named after the two leading mathematicians who have first studied them, Erik Ivar Fredholm and Vito Volterra. The Fredholm and Volterra types equations are encountered in the studies of real-life phenomena. The Volterra types of IDEs consist a variable at the upper bound of the region at the integral sign, while the bounds are fixed in the Fredholm type.

In recent years, there has been a growing interest in linear and nonlinear IDEs and their applications. The Fredholm IDEs of the second kind are commonly seen in numerous scientific applications such as the theory of signal processing and neural networks [13]. In the most cases, obtaining analytical solutions of the Fredholm IDEs are difficult particularly when the problems are high order IDEs. Therefore, as an alternative choice to solve such complex problems physically and geometrically with the minimum amount of mathematical operations, numerical methods are the best. [79].

1.2 Fredholm Integro-Differential Equations with Initial Conditions

Consider the general form of Nth order linear Fredholm Integro-differential equations (FIDEs)

$$\sum_{i=0}^{N} \alpha_i \psi^{(i)}(x) - \gamma \int_a^b K(x,\xi)\psi(\xi)d\xi = f(x), \quad a \leq x \leq b \tag{1.1}$$

subjected to Dirichlet boundary conditions

$$\psi(a) = \psi_a \tag{1.2}$$

and

$$\psi(b) = \psi_b \tag{1.3}$$

and Neumann boundary conditions

$$\psi^{(i)}(a) = \psi_{a_i}, \quad i = 0, 1, 2, \ldots, (r-1) \tag{1.4}$$

and

$$\psi^{(i)}(b) = \psi_{b_i}, \quad i = r, (r+1), \ldots, (N-1), \tag{1.5}$$

where ψ_a, ψ_b, ψ_{a_i} for $0 \leq i \leq (r-1)$ and ψ_{b_i} for $r \leq i \leq (N-1)$ are real finite constants [103], and equations (1.1) can be expressed in equivalent linear system form as

$$A\widehat{\psi} = \widehat{f}, \tag{1.6}$$

which can be written more distinctly as

$$A\widehat{\psi} = (D - W)\widehat{\psi} = \widehat{f}, \tag{1.7}$$

where A is an integro-differential operator given by

$$A = D - W. \tag{1.8}$$

Equation (1.7) can be rewritten as

$$A\widehat{\psi} = (D - W)\widehat{\psi} = D\widehat{\psi} - W\widehat{\psi} = \widehat{f} \tag{1.9}$$

with

$$D\widehat{\psi} = \sum_{i=0}^{N} \alpha_i \psi^{(i)}(x), \tag{1.10}$$

and

$$W\widehat{\psi} = \gamma \int_a^b K(x,\xi)\psi(\xi)d\xi, \tag{1.11}$$

in which $K(x,\xi)$ is a smooth kernel continuous function in $L^2([a,b]\times[a,b])$ space, α_i, γ, a and b are given constants and $\widehat{\psi}(x)$ is an unknown vector function to be determined.

The definitions of the linear differential operator $D\widehat{\psi}$ and the integral operator $W\widehat{\psi}$ of IDEs are given below.

Definition 1.2.1. ([102]) *Let D be a linear differential operator with degree m whose coefficients be continuous functions of t: $D = a_m(t)D^m + a_{m-1}(t)D^{m-1} + \cdots + a_1(t)D + a_0(t)I$. That is $D(x) = a_m(t)D^m(x) + a_{m-1}(t)D^{m-1}(x) + \cdots + a_1(t)D(x) + a_0(t)(x)$, in common $a_m(t)$ is written as a_m which can be expressed as $D(x) = a_m x^{(m)} + a_{m-1}x^{(m-1)} + \cdots + a_1 x' + a_0 x$.*

Definition 1.2.2. ([66]) *Let $W : X \to Y$ be an operator from a normed space X into a normed space Y, the equation $A\widehat{\psi} = \widehat{f}$ is known as well-posed if the function W is bijective (onto and one to one) and its inverse $W^{-1} : Y \to X$ is continuous. Otherwise, the equation is known as ill-posed.*

The conditions for the existence and uniqueness of the solution of Eq. (1.1) have been investigated by many researchers [45, 111, 112].

1.2.1 Types of Integro-differential Equations

(i) **Fredholm Integro-differential Equations (FIDEs)** [100]
The FIDE

$$\sum_{i=0}^{N} \alpha_i \psi^{(i)}(x) - \gamma \int_a^b K(x,\xi)\psi(\xi)d\xi = f(x) \tag{1.12}$$

is called as Fredholm type as the limits of lower and upper bounds of the region for the integral are fixed.

(ii) **Volterra Integro-differential Equations (VIDEs)** [119]

The VIDE

$$\sum_{i=0}^{N} \alpha_i \psi^{(i)}(x) - \gamma \int_{a}^{x} K(x,\xi)\psi(\xi)d\xi = f(x) \tag{1.13}$$

is called as Volterra type as the limit of lower bound is given while the upper bound is a variable of x.

(iii) **Renewal Integro-differential Equations** [32]

The Renewal IDE

$$\sum_{i=0}^{N} \alpha_i \psi^{(i)}(x) - \gamma \int_{a}^{x} K(x-\xi)\psi(\xi)d\xi = f(x). \tag{1.14}$$

(iv) **Abel's Integro-differential Equations** [115]

The Abel's type IDE

$$\sum_{i=0}^{N} \alpha_i \psi^{(i)}(x) - \gamma \int_{a}^{x} \frac{\psi(\xi)}{(x-\xi)^{\beta}}d\xi = f(x). \tag{1.15}$$

(v) **Cauchy Integro-differential Equations** [1]

The Cauchy type IDE

$$\sum_{i=0}^{N} \alpha_i \psi^{(i)}(x) - \gamma \int_{\Gamma} \frac{\psi(\xi)}{(x-\xi)^{\beta}}d\xi - \int_{\Gamma} K(x-\xi)\psi(\xi)d\xi = f(x) \tag{1.16}$$

where Γ is open or closed in R^2.

(vi) **Wiener-Hopf Integro-differential Equations** [39]

The Wiener-Hopf type IDE

$$\sum_{i=0}^{N} \alpha_i \psi^{(i)}(x) - \gamma \int_{0}^{\infty} K(x-\xi)\psi(\xi)d\xi = f(x) \tag{1.17}$$

where $K(z) \in L_2[-\infty, \infty]$, under the condition

$$\int_{-\infty}^{\infty} \mid K(z) \mid dz < \infty. \tag{1.18}$$

(vii) **Volterra-Fredholm Integro-differential Equations (VFIDEs)** [117]

The VFIDE

$$\sum_{i=0}^{N} \alpha_i \psi^{(i)}(x) - \gamma_1 \int_a^x K_1(x,\xi)\psi(\xi)d\xi + \gamma_2 \int_a^b K_2(x,\xi)\psi(\xi)d\xi = f(x). \tag{1.19}$$

1.2.2 Types of Kernel for Integro-differential Equations

The classification of IDEs are based on the kernel of the integral. If the kernel $K(x, \xi)$ is continuous in $L_2(a, b)$, or at least the discontinuity of the kernel such that the double integral

$$\int_a^b \int_a^b K^2(x,\xi)dxd\xi < \infty \tag{1.20}$$

has a finite value, then equation (1.20) called Fredholm type, otherwise it is likely to be singular. Following are various types of kernel in the integral [6]

(i) If the kernel has the Carleman function form

$$K(x,\xi) = \frac{A(x,\xi)}{\mid x - \xi \mid^{\beta}}, 0 \leq \beta < 1 \tag{1.21}$$

or the logarithmic form

$$K(x,\xi) = A(x,\xi) \ln \mid x - \xi \mid, \tag{1.22}$$

where $A(x, \xi)$ is a smooth function, then the integral has a weak singularity.

(ii) If the integral has a singular type of kernel, then it may be expressed as

$$K(x,\xi) = \frac{B(x,\xi)}{x - \xi}, \tag{1.23}$$

where the numerator $B(x, \xi)$ is a differentiable function of x and ξ. The kernel of equation (1.23) is called a Cauchy kernel.

(iii) When the integral has a strong singularity, the kernel will be in the form of

$$K(x, \xi) = \frac{C(x, \xi)}{(x-\xi)^2}, \tag{1.24}$$

where $C(x, \xi)$ is a differentiable function of x and ξ. Some other types of integral kernels are as follows:

(iv) The symmetric kernel

$$K(x, \xi) = K(\xi, x); \tag{1.25}$$

(v) The skew-symmetric kernel

$$K(x, \xi) = -K(\xi, x); \tag{1.26}$$

(vi) The Hermitian kernel

$$K(x, \xi) = \overline{K}(\xi, x); \tag{1.27}$$

(vii) The degenerate kernel

$$K(x, \xi) = \sum_{i=1}^{n} a_i(x) b_i(\xi). \tag{1.28}$$

1.3 Problem statement

As described in the previous literature, FIDEs are one of the most practical and widely used IDEs which are extensively studied by researchers with particular focus emphasised on the first, second and fourth orders. The reason is that the IDEs give significantly better models than PDEs due to the characteristics in the equations [132].

In physics, a heat conduction occurrence, which is represented by a parabolic partial differential equation by having an infinite heat propagation speed, is considered a puzzling contradiction. In fact, the past influences of the material property to the present is better understood if the heat propagation is modelled by an integro-differential equation [7].

Apart from that, the system with generated dense matrix or a full matrix from the approximated FIDEs is incredibly expensive to solve by direct methods especially when the system is large. The main difficulties with IDEs formulations are the amount of memory needed to store the system matrix and the time consuming process.

Moreover, the applications of the numerical methods to discretise and solve FIDEs also predominantly lead to dense linear systems which can be incredibly expensive in terms of computational complexity as well as time-consuming [27]. Moreover, the existing numerical methods need high mathematical procedures to solve such dense linear system, especially when the order of mesh size is large.

In practice, the storage requirements could easily exceed the memory of computer, and the computational time of solution may also be unacceptable. Therefore, to solve IDEs more efficiently, effort must be devoted for solving a large dense linear system. Indeed, to execute this type of problems, an efficient and quicker method with a favourable accuracy is crucial.

Previously, Fedotov [9] studied on quadrature-difference method in solving linear IDEs. His method was based on the full-sweep iteration which required large computational complexity and a lot of execution time.

Besides, numerous works have been carried out on the development of more advanced and efficient methods for solving FIDE such as Wavelet-Galerkin [3], Lagrange interpolation [92], Taylor polynomial [63] and Adomian's decomposition [14].

The previous studies require more mathematical operations to achieve these results without a proper methodical way to solve the equation (1.1). Moreover, they are not accurate and usually they are developed for some special types of FIDEs [103]. Therefore, it is important to develop efficient dense-system solvers for both sequential and parallel computers to solve all types FIDE models numerically.

1.4 Methods of Investigation

This research aims to introduce nine new variants of iterative methods under the conventional Weighted Mean (WM) iterative methods which are also known as Two-Staged iterative methods. The WM iterative methods to be considered in this research are the families of Arithmetic Mean (AM) and Geometric Mean (GM) iterative methods.

The family of AM which are the standard Arithmetic Mean [129], Half-Sweep Arithmetic Mean (HSAM) [55] and Quarter-Sweep Arithmetic Mean (QSAM) iterative methods [53], and similarly the family of GM which are the standard

Geometric Mean [58], Half-Sweep Geometric Mean (HSGM) [58] and Quarter-Sweep Geometric Mean (QSGM) [49] iterative methods. The standard AM and GM methods are also known as Full-Sweep Arithmetic Mean (FSAM) and Full-Sweep Geometric Mean (FSGM) methods, respectively.

Aforementioned, the WM (i.e. AM and GM) iterative methods are modified with some relevant mathematical procedures in order to improve the convergence rate and the overall performances for solving the large dense linear systems associated with the numerical solution of FIDEs.

Furthermore, based on the relationship between modified Arithmetic and Geometric Mean, three new iterative methods are developed and introduced as family of modified Harmonic Mean iterative methods. The fundamental idea of modified Harmonic Mean iterative method is from the Harmonic Mean operator which was studied by Fujii [43]. Figure 1.2 presents a modified Weighted Mean iterative methods diagram.

Finally, all the proposed iterative methods are further investigated by implementing half- and quarter-sweep iteration techniques associated with the generated dense linear systems arising from FIDEs.

1.5 Objectives of Research

In this research, mathematical procedures with convergence theorems and proofs of three families of modified WM iterative methods are presented. The families of modified Weighted Mean (MWM) are

(a) Family of modified Arithmetic Mean (MAM) - standard or Full-Sweep modified Arithmetic Mean (FSMAM), Half-Sweep modified Arithmetic Mean (HSMAM) and Quarter-Sweep modified Arithmetic Mean (QSMAM);

(b) Family of modified Geometric Mean (MGM) - standard or Full-Sweep modified Geometric Mean (FSMGM), Half-Sweep modified Geometric Mean (HSMGM) and Quarter-Sweep modified Geometric Mean (QSMGM);

(c) Family of modified Harmonic Mean (MHM) - standard or Full-Sweep modified Harmonic Mean (FSMHM), Half-Sweep modified Harmonic Mean (HSMHM) and Quarter-Sweep modified Harmonic Mean (QSMHM).

The proposed MWM iterative methods are implemented from the WM methods to solve linear systems of FIDEs. The objectives of this research are as follows:

(i) To investigate the effectiveness of the standard or full-, half- and quarter-sweep iteration techniques with Central Difference-Composite Closed Newton Cotes (CD-CCNC) combination sets approximation schemes on the second and fourth order linear FIDEs.

(ii) To formulate and implement the new proposed MWM iterative methods for solving corresponding generated CD-CCNC systems of the second and fourth order linear FIDEs.

(iii) To analyse the efficiency of the new proposed MWM iterative methods for solving the generated CD-CCNC dense linear systems of the second and fourth order linear FIDEs.

1.6 Scope of Research

In this research, numerical solutions for two linear FIDEs (i.e the second and fourth order FIDEs) problems are considered with the following characteristics:

(i) well-posed problems;

(ii) nonhomogeneous case (i.e. $f(x) \neq 0$);

(iii) limits of integration with $a = 0$ (lower bound) and $b = 1$ (upper bound);

(iv) the smooth type of kernel in the integral part.

The discretisation schemes which are considered in this research are Central Difference and Composite Closed Newton-Cotes as depicted in Figure 1.1. Meanwhile, the proposed MWM iterative methods are illustrated in Figures 1.2.

1.7 Outline of the Thesis

The organisation of the thesis is as follows

(i) In Chapter 2, literature reviews on the real world challenges of IDEs are discussed. Following that, the summary of some important iterative methods for solving the generated dense linear systems of IDEs are presented. A brief review of the development and application of the Two-Stage iterative methods and computational complexity reduction techniques are also included.

(ii) In Chapter 3, the modified Weighted Mean iterative methods are developed The techniques of the computational complexity reduction based on half- and quarter-sweep iteration techniques are described in detail and applied on the CD-CCNC discretisation schemes and followed by the proposed MWM iterative methods. Subsequently, all the new variants of the proposed MWM iterative methods are presented with their corresponding convergence theorems and the proofs.

(iii) Chapter 4 presents numerical solutions for the second and fourth order linear FIDEs. Performances of the existing methods and the proposed MWM methods are investigated by solving tested problems. Numerical results and discussions are also presented at the end of each section. The computational complexity of the proposed MWM iterative methods are also analysed.

(iv) In Chapter 5, the achievements and findings of the research are concluded based on the objectives in Section 1.5. The overall conclusions and recommendations for future research works are also suggested.

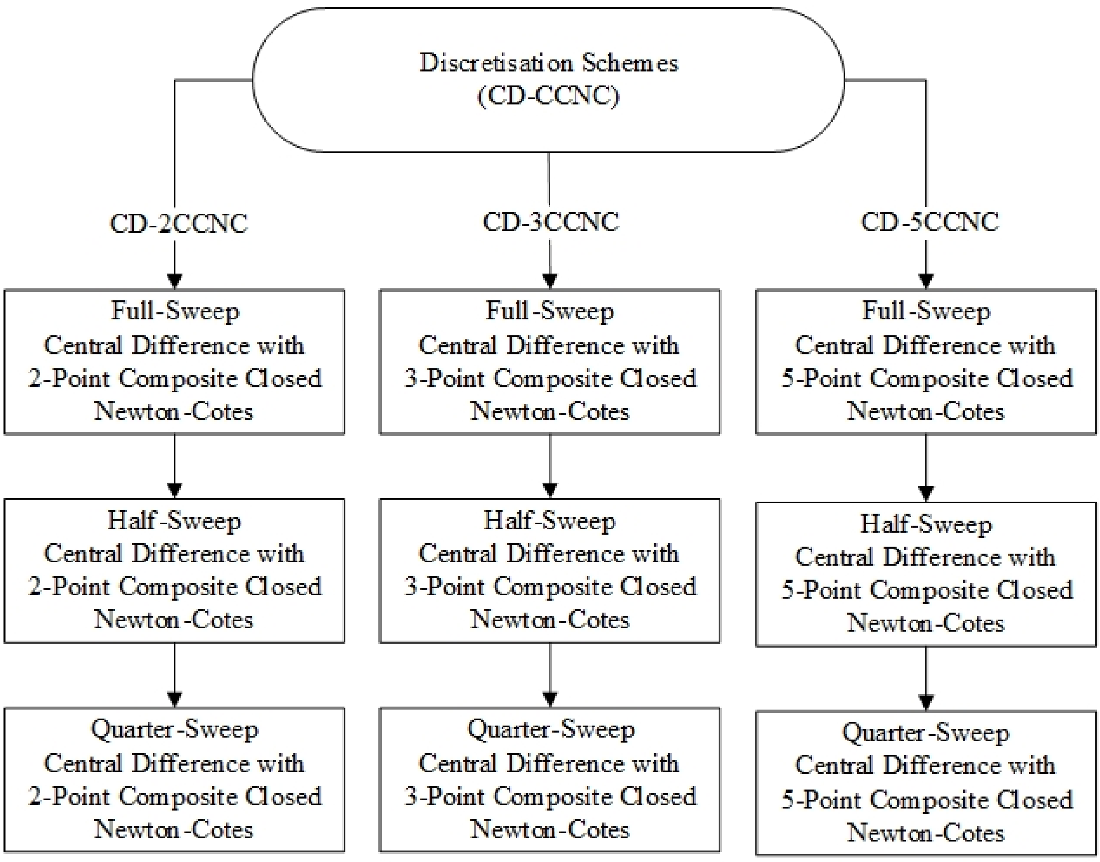

Figure 1.1: Scope of discretisation schemes

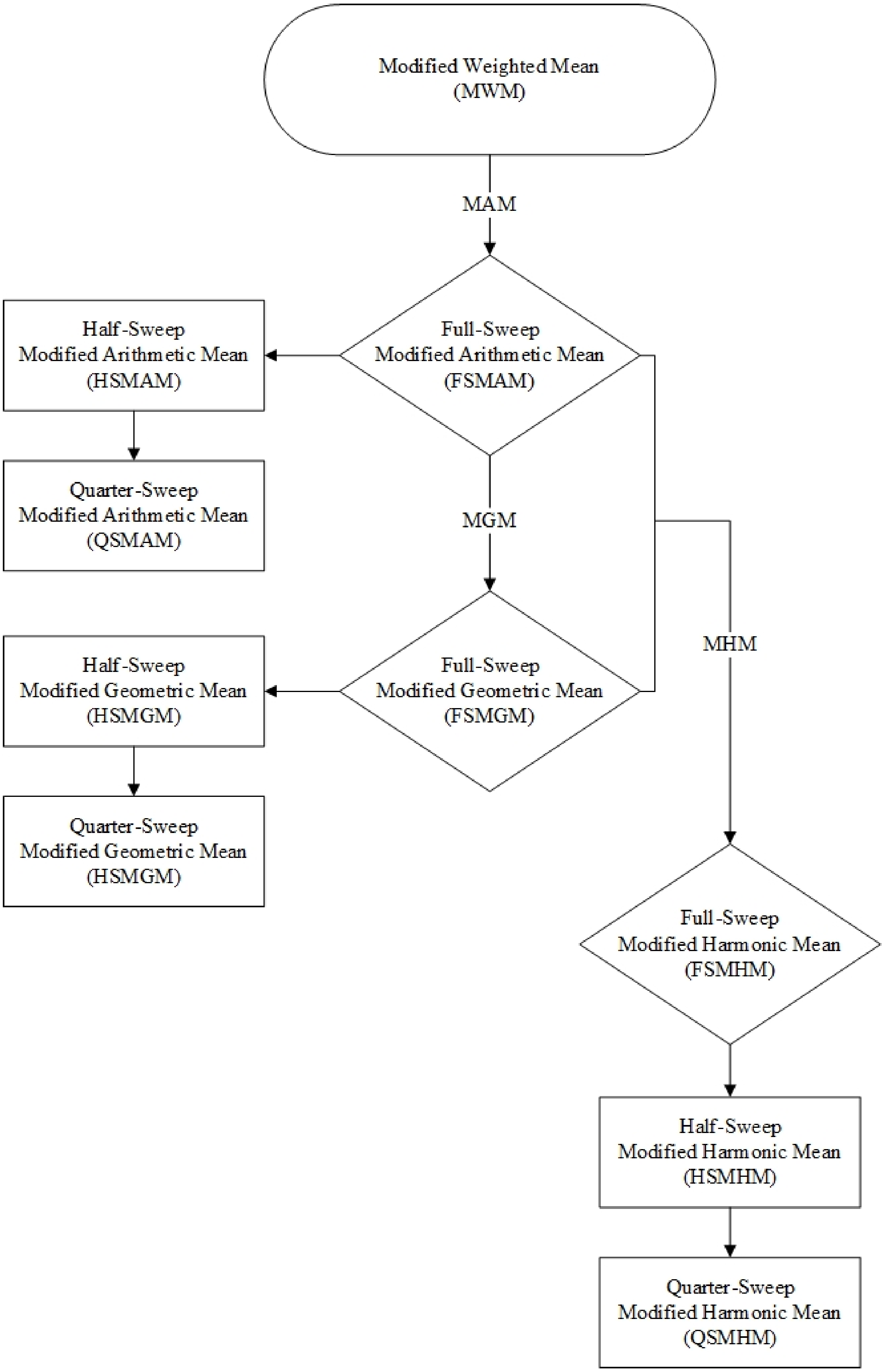

Figure 1.2: Scope of the proposed iterative methods

CHAPTER 2

Background

2.1 General

In this chapter, the importance and real world applications of IDEs are discussed in detail. Some important achievements of the iterative methods for solving the generated linear systems arising from IDEs are highlighted. Further, some relevant numerical approximation schemes to approximate IDEs are also summarised. A brief review on the development of the Two-Stage iterative methods and the applications of computational complexity reduction techniques are also included.

2.2 Real World Challenges of Integro-Differential Equations

In recent years, the mathematical models of IDEs have motivated a number of research works on the development of IDEs as it naturally arises in many real-world applications. The origin of the IDEs study are traced to the work of Abel, Lotka, Fredholm, Malthus, Verhulst and Volterra on many problems such as mechanics, mathematical biology and economics [123].

The importance of Fredholm type IDEs were described in [105] where they arise in real-world applications such as scientific and engineering fields. Subsequently, the IDEs are widely seen in financial areas.

In financial mathematics, IDEs naturally arise in the study of stochastic processes with jumps, which is encountered more precisely in Lévy process (refer to [46, 107]). The fact is that the asset prices often have a sudden changes in the market fluctuations so jump processes become increasingly popular for the modelling. Rama and Ekaterina [108] have explored the precise link between option price in exponential Lévy model and the related partial integro-differential equations in the case of European option and option with a single or double barrier.

In Ecology studies, Brownian motion are often considered to locate abundant prey for the optimal search theory. Therefore, it anticipates that predators should adopt search strategies based on long jumps where prey is sparse and distributed unpredictably [93]. Thus, the reaction-diffusion type of IDEs arises naturally in such population dynamics problems.

In the electromagnetic fields, Beurden [72] has focused the studies on the interaction between electromagnetic field and objects. In this research, the author has formulated an IDE model of the interaction between three different elements of objects with the electromagnetic field.

Other challenges of IDEs problems are usually seen in the material science where the stress-induced atomic diffusion along surfaces and grain boundaries in polycrystalline solids. The structural material which is subjected to high temperature creep conditions which is frequently failed by the growth and coalescence of grain boundary cavities caused by stress-induced grain boundaries diffusion [120].

In the dynamic problems, Mariani [71] has conducted the study of strong and global solutions for a quasi linear parabolic IDE system arising in Hydrodynamic models for charged transport in semiconductors. In this studies, the fundamental model for charged transport in a semiconductor device is the semi classical Boltzmann equation in the parabolic band approximation. Basically, this equation is combined with Poisson's equation to model the electric field created by the electrons moving within the semiconductor devices.

2.3 Numerical Methods for Integro-Differential Equations

In many industrial applications, numerical methods play a significant role for solving IDEs problems. In recent years, numerical methods for solving IDEs have been amassed in the books such as Delves [68], Wazwaz [13] and these works still extended by researchers around the globe.

Wazwaz [12] has applied the decomposition method to determine the boundary value problems for higher order IDEs. Hosseini and Shahmorad [116] studied the Tau method with arbitrary polynomial bases to solve FIDEs. Maleknejad [62] has proposed rationalized Haar functions method while Arikoglu and Ozkol [2] have worked on the differential transform technique to solve boundary value problems for IDEs. There are many other methods such as Wavelet-Galerkin method [3], variational iteration method (VIM) [114], Homotopy perturbation method (HPM) [40,41], Taylor collocation [10], Taylor polynomial [63], Legendre Wavelet [84], Lagrange interpolation [92], sine-cosine Wavelet [121], collocation [109], Compact finite difference [59], Sinc-Collocation [15], wavelet Petrov-Galerkin [65] Quarter-Sweep Gauss-Seidel [28], Quarter-Sweep Conjugate Gradient [27] and Order-optimal [101] are developed to solve some special type of IDEs.

2.4 Discretisation Schemes

In this section, some of important developments of approximation schemes for IDEs are discussed. There are a number of numerical discretisation techniques that have been carried out to approximate the IDEs such as Taylor Collocation method [10], Taylor Polynomial method [63], Compact Finite Difference method [59], quadrature-difference method [9], etc. The literature in the following section emphasises more on Finite Difference and quadrature approximation methods due to their stability in generating dense linear systems by approximating IDEs.

2.4.1 Finite Difference-Quadrature Approximation Schemes

Over the last few decades, Finite Difference approximation has been widely used for solving many mathematical models such as partial differential equations (PDEs), ordinary differential equations (ODEs), etc. The error bounds studies of Finite Difference for elliptic problems were successfully derived by Gerschgorin [20] via Laplace's equation.

Meanwhile, quadrature methods are initiated to discretise the integral terms in the IEs and then later they have been extended to IDEs to approximate the integral term. The quadrature methods are ways to make it easier to execute on a computer program, nevertheless, the error analysis is more complex than the projection methods [61].

Earlier, Vainikko [36] used quadrature-difference methods to approximate the linear IDEs. The convergence of the quadrature-difference methods for general assumptions have been proved based on the compact approximation of operators. These results have been used as a justification of the method of mechanical quadrature for the solution of integral equations.

Styś [60] adopted Finite Difference-Gauss quadrature numerical approximation schemes to approximate different order of IDEs systems with the global error. Then, the generated algebraic system is successfully solved by using an efficient implicit iterations.

Zhao and Corless [59] proposed a sixth order compact Finite Difference formula for solving the second order IDEs with different boundary conditions and the numerical experiments to confirm that their compact Finite Difference method can obtain fifth order of accuracy. They have adjusted compact Finite Difference method for the solutions of the first order IDEs and a system of IDEs. Their proposed method also has been investigated on nonlinear IDEs and an unsplit kernel of IDEs. The superiority of the compact Finite Difference method is a high order of accuracy for solving IDEs problems, and the time complexity is $O(N)$ to solve the matrix equations.

Dehghan and Saadatmandi [73] introduced the Chebyshev Finite Difference method to investigate linear and nonlinear second order FIDEs. This method helped to transform the IDEs to a set of algebraic equations. This approach can also be regarded as a nonuniform Finite Difference schemes. They claimed that this approach can solve the problem effectively and produce satisfactory results with a least number of collocation points.

Zerarka [18] presented a numerical approach based on the generalised integro-differential quadratic method, and applied to weakly singular Volterra and Fredholm IDE of linear case. The quadrature method which was implemented for integration part seems to be a powerful alternative, and consequently gives an accurate solution to the considered problems.

Basically, all of the above discussed discretisation methods are based on the Full-Sweep iteration, they are highly complex to compute the solutions even numerically, and they require a huge amount of memory for the data storage. Therefore, to reduce the complex mathematical procedures of the discretisation schemes, complexity-reduction techniques are essential.

2.4.2 Newton-Cotes Quadrature Schemes

Quadrature methods are generally considered extremely useful and important to solve the model problems that involve integrals. Numerical quadrature refers to the approximation of an integral $\int \psi(\xi)d\xi$ by a discrete summation $\sum B_j\psi(\xi_j)$ over points ξ_j with some weights B_j. There are many methods of numerical quadrature corresponding to different choices of points ξ_j and weights B_j , with varying degrees of accuracy for various types of functions $\psi(\xi)$.

In this subsection, the quadrature formulae are discussed to approximate definite integral which is in the form of

$$Q(\psi) = \int_a^b \psi(\xi)d\xi \tag{2.1}$$

for a continuous function ψ over the interval $[a, b]$ with $a < b$ and a weighted sum

$$Q_n(\psi) = \sum_{j=0}^{n} B_j \psi(\xi_j) \tag{2.2}$$

with $n+1$ distinct quadrature points $\xi_0, \xi_1, \cdots, \xi_{n-1}, \xi_n \in [a, b]$ and quadrature weights $B_0,\ B_1,\ \cdots,\ B_{n-1},\ B_n \in \mathbb{R}$. A quadrature formula is obtained by integrating polynomial instead of the integrand ψ, that is by approximating

$$\int_a^b \psi(\xi)d\xi \approx \int_a^b (L_n\psi)(\xi)d\xi, \tag{2.3}$$

where $L_n : C[a, b] \to P_n$ represents the polynomial interpolation operator with interpolation points $\xi_0,\ \xi_1,\ \cdots,\ \xi_{n-1},\ \xi_n$. Note that both the integral Q and the quadrature formula Q_n represent linear operators from $C[a, b]$ into $\mathbb{R}$. The well-known quadrature methods include Newton-Cotes, Gaussian and ClenshawCurtis quadrature. Based on the interpolatory quadrature formulae, further discussions thus be restricted into Newton-Cotes quadrature formulae. The following Theorems describe an equivalent definition for Newton-Cotes quadrature formulae.

Theorem 2.1. *The polynomial interpolatory quadrature of order N defined by*

$$Q_n(\psi) = \int_a^b (L_n\psi)(\xi)d\xi \tag{2.4}$$

is of the form (2.2) with the weights given by

$$B_j = \frac{1}{\varrho_{n+1}(\xi_j)} \int_a^b \frac{\varrho_{n+1}(\xi)}{\xi - \xi_j}, j = 0, 1, \cdots, n-1, n \tag{2.5}$$

where $\varrho_{n+1}(\xi_j) = (\xi - \xi_0)\cdots(\xi - \xi_n)$

Proof. Refer Theorem 9.1 in [110] □

Theorem 2.2. *Given $n+1$ distinct quadrature points $\xi_0, \xi_1, \cdots, \xi_{n-1}, \xi_n \in [a, b]$, the interpolatory quadratures (2.4) of order n is uniquely determined by its prop-*

erty of integrating all polynomials $p \in P_n$ exactly, i.e., by the property

$$\sum_{j=0}^{n} B_j p(\xi_j) = \int_a^b p(\xi) d\xi \tag{2.6}$$

for all $p \in P_n$.

Proof. Refer Theorem 9.2 in [110] □

Theorem 2.3. *The polynomial interpolatory quadrature of order n with equidistant quadrature points*

$$\xi_j = a + jh \tag{2.7}$$

and step size $h = \frac{b-a}{n}$ is called the Newton-Cotes quadrature formula of order n. Its weights are given by

$$B_j = h \frac{(-1)^{n-j}}{j!(n-j)!} \int_0^n \prod_{k=0, k\neq j}^{n} (\xi - k) d\xi, j = 0, 1, \cdots, n-1, n \tag{2.8}$$

and have the symmetry property $B_j = B_{n-j}, j = 0, 1, \cdots, n-1, n$.

Proof. Refer Theorem 9.3 in [110] □

Fundamentally, the formulas for Newton-Cotes quadrature can be divided into two groups that are open and closed formulas. Besides, its degree of exactness, a Newton-Cotes quadrature formulas can also be qualified by its order of infinitesimal with respect to the integration step size h, which is defined as the maximum integer m such that $\mid Q(\psi) - Q_n(\psi) \mid = O(h^m)$. Therefore, the following Theorems hold for Newton-Cotes quadrature formulas

Theorem 2.4. *For any Newton-Cotes formula equivalent to an even value of n, the following quadrature error, $E_n(\psi)$ characterization holds*

$$\epsilon_n(\psi) = \frac{M_n}{(n+2)!} h^{n+3} \psi^{(n+2)}(\eta) \tag{2.9}$$

provided $\psi \in C^{n+2}([a, b])$ where $\eta \in (a, b)$ and

$$M_n = \begin{cases} \int_0^n \varsigma_{n+1}(\xi) d\xi < 0 & \textit{for closed formula,} \\ \int_{-1}^{n+1} \varsigma_{n+1}(\xi) d\xi > 0 & \textit{for open formula.} \end{cases} \tag{2.10}$$

having defined $\varsigma_{n+1}(\xi) = \prod_{i=0}^{n} (\xi - i)$.

From (2.9), it turns out that the degree of accuracy is equal to $N + 1$ and the order of infinitesimal is $n + 3$.

Similarly, for odd values of n, the following error characterization holds

$$\epsilon_n(\psi) = \frac{K_n}{(n+1)!} h^{n+2} \psi^{(n+1)}(\eta) \tag{2.11}$$

provided $\psi \in C^{n+1}([a, b])$ where $\eta \in (a, b)$ and

$$K_n = \begin{cases} \int_0^n \varsigma_{n+1}(\xi) d\xi < 0 & \text{for closed formula,} \\ \int_{-1}^{n+1} \varsigma_{n+1}(\xi) d\xi > 0 & \text{for open formula.} \end{cases} \tag{2.12}$$

The degree of exactness is thus equal to n and the order of infinitesimal is $n + 2$.

Proof. Refer Theorem 9.2 in [16] □

In numerical integration, to increase the accuracy, it is more practical to use composite Newton-Cotes quadrature formula instead of using higher order formula. The formula are obtained by subdividing the interval of integration and then supplying a fixed formula with low interpolation order to each of the subintervals. The common procedure consists of partitioning the integration interval $[a, b]$ into n subintervals $\Omega_k = [\xi_k, \xi_{k+1}]$ such that $\xi_k = a + kh$ where $h = \frac{b-a}{n}$ for $k = 0, 1, \cdots, n-1, n$. Then, over each subinterval, an interpolatory formula with nodes $\{\xi_j^{(k)}, 0 \le j \le n\}$ and weight $\{B_j^{(k)}, 0 \le j \le n\}$ is used. Since

$$Q(\psi) = \int_a^b \psi d\xi = \sum_{k=0}^{N-1} \int_{\Omega_k} \psi(\xi) d\xi, \tag{2.13}$$

a composite interpolatory quadrature formula is obtained by replacing $Q(\psi)$ with

$$Q_{n,N}(\psi) = \sum_{k=0}^{N-1} \sum_{j=0}^{n} B_j^{(k)} \psi(\xi_j^{(k)}). \tag{2.14}$$

By using the same notation as in Theorem 2.4, the following convergence result holds for composite formula.

Theorem 2.5. *Let a composite Newton-Cotes formula, with n even, be used. If $\psi \in C^{n+2}([a, b])$, then*

$$\epsilon_{n,N}(\psi) = \frac{b-a}{(n+2)!} \frac{M_n}{(n+2)^{n+3}} h^{n+2} \psi^{n+2}(\eta) \tag{2.15}$$

where $\eta \in (a, b)$. Therefore, the quadrature error is an infinitesimal in h of order $n + 2$ and the formula has degree of accuracy equal to $n + 1$.

For a composite Newton-Cotes formula, with n odd, if $\psi \in C^{n+2}([a,b])$.

$$\epsilon_{n,N}(\psi) = \frac{b-a}{(n+1)!}\frac{K_n}{n^{n+3}}h^{n+1}\psi^{n+1}(\eta), \tag{2.16}$$

where $\eta \in (a,b)$. Thus, the quadrature error is an infinitesimal in h of order $n+1$ and the formulae has degree of accuracy equal to n.

Proof. Refer Theorem 9.3 in [16]. □

2.5 Boundary Conditions

The boundary conditions involved in IDEs are the constraining values of the function at some specific values of the independent variables for differential term. Therefore, to have a complete solution, boundary conditions are essential for each order of the equations. Hence, the first order IDEs has one boundary condition or initial condition. Meanwhile, for the second and fourth order IDEs, respectively two and four boundary conditions are needed in order to determine the solution of the problems.

2.5.1 Dirichlet Boundary Condition

The Dirichlet type boundary condition or also known as a fixed boundary condition is usually seen in many engineering applications which is named after Johann Peter Gustav Lejeune Dirichlet (1805-1859) [5]. The Dirichlet boundary condition is the first type of boundary condition that defines the values on the border of the problem domain. The condition is an interior of a given region that hold the prescribed values on the boundary of the region. The question of determining solution to such equations is known as the Dirichlet problem. For an ordinary differential equation

$$\psi'' + \psi = 0, \tag{2.17}$$

the Dirichlet boundary conditions on an interval $[a,b]$ can be expressed as $\psi(a) = \alpha$ and $\psi(b) = \beta$ where α and β are given values or known functions [5].

2.5.2 Neumann Boundary Condition

The Neumann (or second-type) boundary condition is named after Carl Neumann [5]. Basically, Neumann boundary condition specifies the value of a normal derivative, or some combination of derivatives, along a boundary surface. They arise in problems where a flux has been specified on a boundary of the domain. For an ordinary differential equation

$$\psi'' + \psi = 0, \tag{2.18}$$

the Neumann boundary conditions on the interval $[a, b]$ take the form: $\psi'(a) = \alpha$ and $\psi'(b) = \beta$ where α and β are given values or known functions. There are many other boundary conditions which could be used, most of which have a physical interpretation. For example, Cauchy boundary condition or the mixed boundary condition is known as the combination of the Dirichlet and Neumann boundary conditions [5].

2.5.3 Robin Boundary Condition

The Robin (or third type) boundary condition is named after Victor Gustave Robin (1855-1897). In an ordinary or a partial differential equation, it is a requirement where both values of function in the linear combination and also in derivative form on the domain boundaries. Usually, Robin boundary conditions often can be seen in electromagnetic problems. Assume that if Ω is the domain on which a given equation is to be determined and denotes its boundary, the Robin boundary condition is as follows

$$au + b\frac{\partial \psi}{\partial n} = g, \tag{2.19}$$

where a and b are non-zero constants and g is a given function which is defined on $\partial\Omega$, ψ is an unknown solution which is defined on Ω, while $\frac{\partial \psi}{\partial n}$ denotes the normal derivative on the boundary. In common, a and b may be given functions, rather than constants.

Basically, in one dimension, if $\Omega = [0, 1]$, then the Robin boundary conditions become as

$$a\psi(0) - b\psi'(0) = g(0) \tag{2.20}$$

$$a\psi(1) + b\psi'(1) = g(1). \tag{2.21}$$

Robin boundary conditions are usually applied in solving Sturm-Liouville problems which arise in many contexts in science and engineering applications. In addition, the Robin boundary condition is a common form of the insulating condition for convection-diffusion equation [5].

2.6 Iterative Methods for Solving Linear Systems

In general, the basic concept of solving IDEs by using numerical methods is to approximate the problems (1.3) in the form of system of linear algebraic equations. Fundamentally, the generated linear systems can be solved by using either direct or iterative methods. The direct methods or, also known as elimination methods, are executed via a finite number of arithmetic operations (without taking into account of round-off errors). Meanwhile, the iterative methods are a sequential execution of mathematical procedures by repeating the procedures at each step of iteration, to compute the approximation solution. In this research, further discussions are emphasised to the iterative methods.

Generally, iterative methods are devised by executing the splitting of nonsingular matrix A in Eq. (1.3) into $A = H - K$. Hence, the associated iterative scheme for solving linear system (1.3) is in the form of

$$\psi^{(k+1)} = H^{-1}K\psi^{(k)} + H^{-1}f, \ \ k = 0, 1, 2, \dots. \tag{2.22}$$

converge to $A^{-1}f$ for any ψ^0.

Definition 2.6.1. [113] *Let $A = [a_{i,j}]$ be an $N \times N$ complex matrix with eigenvalue λ_i, $1 \leq i \leq N$. Then,*

$$\rho(A) = \max_{1 \leq i \leq N} \mid \lambda_i \mid \tag{2.23}$$

is the spectral radius of the matrix A.

Theorem 2.6. [110] *Let Z be an $N \times N$ matrix. Then successive approximations*

$$\psi^{(k+1)} = Z\psi^{(k)} + c, \ \ k = 0, 1, 2, \dots \tag{2.24}$$

convergence for each $c \in C^n$ and each $\psi^{(0)}$ if and only if

$$\rho(Z) < 1 \tag{2.25}$$

for the spectral radius of Z.

Proof. The proof is given in [110]. □

By referring to Definition 2.6.1 and Theorem 2.6 , it is known that the iterative scheme (2.22) converges for all initial datum (or guess), $\psi^{(0)}$ to the solution of

Eq. (1.3) if and only if the spectral radius of the iteration matrix $H^{-1}K$ is less than $\rho(H^{-1}K) < 1$ [4, 44].

2.6.1 Stationary and Nonstationary Iterative Methods

According to Van Rienen [122], stationary iterative methods such as Jacobi, Gauss-Seidel (GS), Richardson, Successive Over-Relaxation (SOR) and Symmetric Successive Over-Relaxation (SSOR) methods are considered as one-stage iterative methods. Other modifications based on SOR named modified Successive Over-Relaxation (MSOR) iterative method for solving linear systems with red-black matrix is introduced by Young [25].

Meanwhile, nonstationary methods are distinct from stationary methods where the computation constants are calculated by taking the inner products of residuals or other vectors arising from the mathematical procedures. Basically, nonstationary iterative methods need high computational cost due to its complex mathematical operations, especially when the inner product of two matrices is required. The nonstationary iterative methods are namely Conjugate Gradient (CG), Conjugate Gradient Normal Equation (CGNR), variant of CG (Bi-CG), Biconjugate gradient stabilized (Bi-CGSTAB), Transpose-Free Quasi-Minimal Residual (TFQMR), Generalised Minimal Residual (GMRES) and etc [106].

2.6.2 Two-Stage Iterative Methods

Apart from the methods in Section 2.6.1, another iterative scheme called Two-Stage iterative method was introduced to solve the generated linear system effectively. The fundamental concept of the two-stage iterative method is to determine the solution from the outer iteration by using another iterative procedure called inner iteration. This concept is also referred as an inner/outer iteration scheme which is extensively studied by researchers for solving various matrix problems.

In the Two-Stage iterative procedures, the outer iteration (2.22) is solving iteratively by using inner iteration. In particular, let us consider the splitting of $H = F - G$ and perform $s(k)$ inner iterations in each outer step k. Hence, the resulting method is

$$\psi^{(k+1)} = (F^{-1}G)^{s(k)}\psi^{(k)} + \sum_{i=0}^{s(k)-1} (F^{-1}G)^i F^{-1}(K\psi^{(k)} + f), \quad k = 0, 1, 2, \ldots \tag{2.26}$$

where $s(k) = k$ for $s \geq 1$ is the number of inner iterations in each outer step k.

Therefore, the iterative scheme equation (2.26) is called as stationary Two-Stage iterative method. In the case of nonstationary Two-Stage method, the number of inner iterations may change with the outer iteration.

The iteration matrix of equation (2.26) is

$$Z_{s(k)} = (F^{-1}G)^{s(k)} + \left(I - (F^{-1}G)^{s(k)}\right)H^{-1}K \tag{2.27}$$

where I denotes the identity matrix. Since, the stationary case $Z_{s(k)} = Z_s$ for all $k = 0, 1, 2, \ldots$, then the convergence of equation (2.26) is equivalent to $\rho(Z_s) < 1$. Meanwhile, the nonstationary case of vector sequence generated by iteration equation (2.26), converges to the solution of equation (1.4) if and only if $\lim_{k\to\infty} Z_{s(k)}Z_{s(k-1)} \cdots Z_{s(0)} = O$, where O is the null matrix [124]. The further investigations on monotonicity and convergence of the Two-Stage methods have been studied with various types of matrices (refer to [125, 134, 136, 137]).

In the work of [34] and [35], applications of the Chebyshev or Richardson method as the outer iteration for Two-Stage iterative methods were considered. Consequently, the solutions of linear systems by using Two-Stage iterative methods with general inner iteration methods were investigated by Nicholas [99]. Then, Yun and Kim [42] investigated an application of the Two-Stage iterative methods using incomplete factorization as inner splitting for linear systems whose coefficient matrices are H-matrices (Hermitian matrices) and symmetric positive definite matrices.

Lanzkron [104] studied on a general class of iterative methods. Without loss of generality, both outer and inner iterations were depicted by splittings. The convergence conditions of the outer and inner splittings of both stationary and nonstationary iterative methods were also discussed. The conditions imply that the coefficient matrix A must be a monotone.

Based on the Two-Stage iterative procedures, various methods such as Alternating Group Explicit (AGE) [23], Modified Alternating Group Explicit (MAGE) [22], Iterative Alternating Decomposition Explicit (IADE) [90], Block SOR [33] and Weighted Mean (WM) [37, 58] have been developed for solving generated linear systems arising from various scientific and engineering problems. The further discussions are focused on the development of WM iterative methods and their applications.

The WM iterative methods are classified as a family of algorithms, which have been broadly applied for the solution of matrix equation problems. One of the methods under WM family is the Arithmetic Mean (AM) iterative method [37].

The effectiveness of the AM iterative method and its variants were examined on linear and nonlinear models of various scientific and engineering problems (refer to [31, 53, 55, 70, 78, 126, 129]). Ruggiero and Galligani observed that the AM method compare favourably with the existing methods such as the Bi-CG method, particularly when A is strongly asymmetric. The AM method is also used as a preconditioner to CG method and successfully solved the symmetric positive definite linear system (refer to [38, 128]). In addition, the AM procedure was used as first degree iterative method to determine the minimal eigenpair of generalized eigen problem [29].

Among the variants of AM methods, Half-Sweep Arithmetic Mean (HSAM) [55, 85] and Quarter-Sweep Arithmetic Mean (QSAM) [53, 87] iterative methods are applied extensively for sparse linear systems which arise from PDE problems. For example, water quality model [47, 78], fourth order parabolic equations [50], elliptic equations [51], one dimensional diffusion equations [54] and Poisson equations [52].

Besides the family of AM methods, another family based on WM method was developed and called as Geometric Mean (GM) method. The fundamental idea of the standard GM iterative method was derived from the concept of AM method. Two variants of GM method i.e. Half-Sweep Geometric Mean (HSGM) [58, 86] and Quarter-Sweep Geometric Mean (QSGM) [49, 88] methods were also suggested. Following that, several researches involving GM and its variants for solving various scientific problems have been conducted [49, 78, 88].

2.7 Computational Complexity Reduction Techniques

In recent decades, computational complexity of reduction technique has been studied for solving the generated systems which arise from various scientific problems. The basic idea of complexity reduction technique is to reduce the computational complexity of the solution methods. An example of complexity reduction technique is the Reduced Iterative Alternating Decomposition Explicit (RIADE) [89] iterative method which is the variant of IADE [90] method. The RIADE method succeeds in reducing the computational complexity and speed-up the existing IADE method in solving linear systems associated with the heat conduction equation.

In the meantime, other smart and remarkable techniques were introduced and came to be known as half- and quarter-sweep iteration techniques, which are categorised as computational complexity reduction technique. The half- and quarter-sweep iteration techniques were first envisioned via the Explicit Decoupled Group (EDG) by Abdullah [17]. Then, Yousif and Evans [130] extended the method to six and nine points groupings and demonstrated that they can be easily parallelised on an MIMD multiprocessor.

Besides, Othman and Abdullah [80] introduced the Modified Explicit Group (MEG) methods for solving Poisson equations. Since then, the efficiency of the standard EDG and MEG methods and their variants for solving linear systems generated from various problems have been conducted. Furthermore, the MEG iterative method has been extensively applied to solve finite difference systems of various scientific and engineering problems (refer to [58, 82]).

Apart from the EDG and MEG methods, another technique of complexity reduction was proposed via the Modified Explicit Decoupled Group (MEDG) method [95], which is an extension work of the existing EDG method. Some studies involving MEDG and its variants have also been conducted to assert the performance of the methods (refer to [11, 94, 95]).

The applications of the half- and quarter-sweep iteration techniques have also been extended for solving nonlinear systems such as Newton-Quarter Sweep Alternating Group Explicit (NQSAGE) [98] and Newton-Quarter Sweep Alternating Decomposition Explicit iterative (NQSADE) [97] methods. These methods are derived by combining the complexity reduction technique into the standard method to solve the two-point boundary value problems. The applications of half- and quarter-sweep iteration techniques were also extended to the direct solution for free space wave propagation problems (refer to [75, 76, 78]). A short review on the application of half- and quarter-sweep iteration techniques to some numerical methods for solving scientific problem have been presented in [74].

CHAPTER 3

Construction of Numerical Methods

3.1 Introduction

In this chapter, the general procedure and implementation of a new proposed MWM iterative methods, namely modified Arithmetic Mean (MAM), modified Geometric Mean (MGM) and modified Harmonic Mean (MHM), are presented together with their new convergence Theorems and Proofs. The general framework and application of the approximation schemes, i.e. Central Difference (CD) and Composite Closed Newton-Cotes (CCNC) as shown in Figure 1.1, are developed and applied to discretise the mth order FIDEs. Furthermore, the half- and quarter-sweep iteration techniques are implemented to all the proposed MWM iterative methods together with their new convergence Theorems and Proofs, corresponding to the generated Half- and Quarter-Sweep CD-CCNC dense systems.

3.2 Modified Weighted Mean Formulation

The main focus of this section is to introduce new methods based on WM iterative methods, namely the MAM, MGM and MHM iterative methods to solve the equation (1.1) subjecting to equation (1.2) and equation (1.3). The formulations of three new proposed methods are derived associated with some new Theorems and Proofs. Generally, the iteration cycle for the MWM methods has two stages, i.e. Stage 1 and Stage 2. Formulation in Stage 1 involves an iteration process for an independent system, $\widehat{\psi}^{(\mathscr{F})}$ and Stage 2 emphasises on the formulation of an iteration process for an independent system, $\widehat{\psi}^{(\mathscr{B})}$ of an even order IDE.

3.2.1 Optimal Parameter Value

In numerical methods, optimal parameter value or accelerated parameter is crucial to improve the rate of convergence of the iterative methods ([24] and [8]). For the Weighted Mean (WM) iterative methods, new parameter value, θ_2, is introduced for backward iteration to accelerate the convergence and improve the performances of the methods overall. The θ_2 is set of trial values by determining the best θ_1 which is selected based on the least number of iterations. These, two optimal parameter values have improved the convergence rate compared to the standard WM methods. In the case of $\theta_1 = \theta_2$, the MWM methods is equivalent to the conventional WM methods. The values of the optimal parameters θ_1 and θ_2 are in the range between 0 and 2. The next section is focused on construction of a system of linear equations in order to carry out the implementation of the proposed methods.

3.3 Construction of Dense Systems

There are various ways to construct the approximation equations. In this section, the construction of the dense system of linear equations based on the finite-difference and Composite Closed Newton-Cotes methods to approximate equation (1.1) is shown.

3.3.1 Finite Difference Schemes

The rudimentary of the Finite Difference (FD) schemes is to approximate differential terms by appropriate the difference qoutients to obtain a transformed system of algebraic equations. The following subsections present FD and quadrature schemes to approximate the mth derivatives for $m \geq 2$ and m is even.

The basic idea of FD schemes is to solve a differential equations numerically in a discretised domain. This will result in a number of algebraic equations that can be solved iteratively to obtain values of ψ_i. Let an interval (a, b) be in the form of

$$\sum_{i=0}^{N} h_i = b - a \tag{3.1}$$

and a discrete set of points be given by $x_i = a + ih$ $(i = 0, 1, 2, \cdots, N)$. A basic role to estimate the error involved in FD approximations of derivatives is determined by the Taylor's series expansion as follows:

$$f(\psi_{i+1}) = f(\psi_i) + f'(\psi_i)(h) + \frac{f''(\psi_i)}{2!}(h^2) + \frac{f'''(\psi_i)}{3!}(h^3) + O(h^4), \tag{3.2}$$

and

$$f(\psi_{i-1}) = f(\psi_i) - f'(\psi_i)(h) - \frac{f''(\psi_i)}{2!}(h^2) - \frac{f'''(\psi_i)}{3!}(h^3) - O(h^4), \tag{3.3}$$

and $O(h^4)$ is the error introduced by truncating the series. Therefore, subtracting (3.3) by (3.2) yields

$$f(\psi_{i+1}) - f(\psi_{i-1}) = 2hf'(\psi_i) + O(h^2). \tag{3.4}$$

Thus, the Eq. (3.4) can be expressed as

$$f'(\psi_i) = \frac{f(\psi_{i+1}) - f(\psi_{i-1})}{2h} + O(h^2), \tag{3.5}$$

which is the first derivative of centered finite divided difference approximation with truncation error $O(h^2)$. The forward and backward finite divided difference formulas (3.6) and (3.7) are obtained by rearranging (3.2) and (3.3), respectively.

$$f'(\psi_i) = \frac{f(\psi_{i+1}) - f(\psi_i)}{h} + O(h), \tag{3.6}$$

and

$$f'(\psi_i) = \frac{f(\psi_i) - f(\psi_{i-1})}{h} + O(h). \tag{3.7}$$

High-order approximations to the mth derivative for $m \geq 2$ are obtained by using more terms in Taylor series, and also by appropriately weighting the various expansions in a sum. Therefore, by adding equations (3.2) with (3.3) are has

$$f(\psi_{i+1}) + f(\psi_{i-1}) = 2f(\psi_i) + f''(\psi_i)h^2 + O(h^2). \tag{3.8}$$

Now, the second order centered finite divided difference scheme can be expressed as

$$f''(\psi_i) = \frac{f(\psi_{i+1}) - 2f(\psi_i) + f(\psi_{i-1})}{h^2} + O(h^2), \tag{3.9}$$

or

$$f''(\psi_i) = \frac{-f(\psi_{i+1}) - 2f(\psi_i) + f(\psi_{i-1})}{h^2} + O(h^2). \qquad (3.10)$$

Similarly, we can obtain the fourth order centered finite divided difference scheme as follows

$$f^{(iv)}(\psi_i) = \frac{f(\psi_{i+2}) - 4f(\psi_{i+1}) + 6f(\psi_i) - 4f(\psi_{i-1}) + f(\psi_{i-2})}{h^4} + O(h^2). \quad (3.11)$$

All the truncation errors can be made arbitrarily small with mesh refinement.

3.3.2 Central Difference Operator Formulae

In general, the Central Difference (CD) formulae also can be expressed in the operator form as follows

$$\delta\psi_i = (\delta^{\frac{1}{2}} - \delta^{-\frac{1}{2}})\psi_i = \delta^{\frac{1}{2}}\psi_i - \delta^{-\frac{1}{2}}\psi_i = \psi_{i+\frac{1}{2}} - \psi_{i-\frac{1}{2}}, \qquad (3.12)$$

Similarly, the m order Central Difference can be defined as follows:

$$\delta^m f(\psi) = \delta^{m-1} f(x + \frac{h}{2}) - \delta^{m-1} f(x - \frac{h}{2}) \qquad (3.13)$$

with $m = 0$

$$\delta^0 f(x) = f(x). \qquad (3.14)$$

3.3.3 Composite Closed Newton-Cotes Schemes

In this research, the quadrature formulae are discussed to approximate definite integral which is in the form of

$$Q(\psi) = \int_a^b \psi(\xi) d\xi + \varepsilon_N \qquad (3.15)$$

of a continuous function ψ over the interval $[a, b]$ with $a < b$ and a weighted sum

$$Q_n(\psi) = \sum_{j=0}^{N} B_j \psi(\xi_j) + \varepsilon_N(\psi) \qquad (3.16)$$

with $N + 1$ distinct quadrature points $\xi_0, \xi_1, \cdots, \xi_{N-1}, \xi_N \in [a, b]$, quadrature weights B_0, B_1, $\cdots$, B_{N-1}, $B_N \in \mathbb{R}$ and $\varepsilon_N(\psi)$, truncation error which can be minimised by mesh refinement.

Therefore, the Composite Closed Newton-Cotes (CCNC) quadrature formula of 2-, 3-, and 5-point Composite Closed Newton-Cotes schemes are investigated by solving the tested problems in Chapter 4. Truncation errors of the CCNC can be made arbitrarily small with mesh refinement. A list of three different orders of the CCNC quadrature formulae are given in the Table 3.1 [16, 110].

Table 3.1: Composite Closed Newton-Cotes Quadrature Schemes

(i) 2-point composite closed Newton-Cotes(2CCNC)

$$Q_1 = \frac{h}{2}\Big[\psi(a) + 2\sum_{j=1,2,3}^{N-1}\psi(a+jh) + \psi(b)\Big] - \frac{b-a}{12}h^2 f^{(2)}(\varepsilon) \quad (3.17)$$

(ii) 3-point composite closed Newton-Cotes(3CCNC)

$$\begin{aligned} Q_2 &= \frac{h}{3}\Big[\psi(a) + 4\sum_{j=1,3,5}^{N-1}\psi(a+jh) + 2\sum_{j=2,4,6}^{N-2}\psi(a+jh) + \psi(b)\Big] \\ &\quad -\frac{b-a}{180}h^4 f^{(4)}(\varepsilon) \end{aligned} \quad (3.18)$$

(iii) 5-point composite closed Newton-Cotes(5CCNC)

$$\begin{aligned} Q_3 &= \frac{h}{45}\Big[14\psi(a) + 64\sum_{j=1,3,5}^{N-1}\psi(a+jh) + 24\sum_{j=2,6,10}^{N-2}\psi(a+jh) \\ &\quad +28\sum_{j=4,8,12}^{N-4}\psi(a+jh) + 14\psi(b)\Big] - \frac{2(b-a)}{945}h^6 f^{(6)}(\varepsilon) \end{aligned} \quad (3.19)$$

3.3.4 Standard Approximation Equations

From subsections 3.3.1, 3.3.2 and 3.3.3, three different combination sets of discretisation schemes as described in Figure 1.1, i.e. CD-2CCNC, CD-3CCNC and CD-5CCNC are applied to discretise the problem (1.1). Firstly, the following notations are introduced for simplicity:

$$\begin{aligned}
&\widehat{\psi}_i \equiv \widehat{\psi}(x_i),\\
&f_i \equiv f(x_i),\\
&\xi_i \equiv \xi_j^{(k)},\\
&B_j \equiv B_j^{(k)},\\
&K_{i,j} \equiv K(x_i, \xi_j),\\
&Q_N \equiv Q_{n,N}.
\end{aligned}$$

In Figure 3.1, all the node points of type ● are used in iteration process. This standard process also can be referred as Full-Sweep iteration.

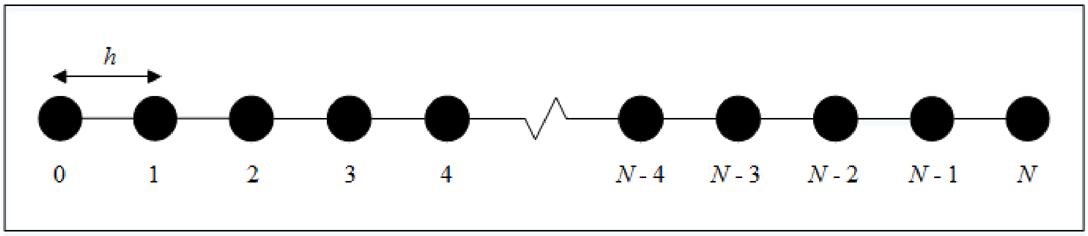

Figure 3.1: Standard or Full-Sweep iteration

In order to carry out the general approximation equations, let consider the finite grid network of the distribution of nodal points in Figure 3.1. Based on Figure 3.1, in showing the standard or Full-Sweep iteration, the unknown functions are computed for the nodal points of type ● until the convergence criterion is reached. Based on the differential and integral terms as in the equation (1.1), the discretisation formula of Central Difference and Composite Closed Newton-Cotes are applied, respectively. Therefore, the algebraic approximation equations of the mth derivative of FIDEs can be obtained as follows

$$\sum_{m=1}^{M-1} \frac{\delta^m \psi_i}{s h_i^m} - \sum_{j=1}^{N-1} B_j K_{i,j} \psi_j = f_i, \quad i = 0, 1, \ldots, N \tag{3.20}$$

where $s = 1$ when m is even, otherwise $s = 2$.

Let

$$\mathcal{D}\widehat{\psi} = \sum_{m=1}^{M-1} \frac{\delta^m \widehat{\psi}_i}{sh_i^m} \tag{3.21}$$

and

$$\mathcal{K}\widehat{\psi} = \sum_{j=1}^{N-1} B_j K_{i,j} \widehat{\psi}_j = \widehat{f}_i, \tag{3.22}$$

the solution of Eq. (3.20) is approximated by the equation

$$\mathcal{D}\widehat{\psi}_N - \mathcal{K}\widehat{\psi}_N = \widehat{f} \tag{3.23}$$

which can be reduced to a finite-dimensional linear systems

$$\sum_{m=1}^{M-1} \frac{\delta^m \psi_N(x_i)}{sh_i^m} - \sum_{j=1}^{N-1} B_j K(x_i, \xi_j) \psi_N(\xi_j) = f(x_i) \quad x_i \in (a, b) \tag{3.24}$$

for $\quad i = 1, 2, \ldots, N-1$. Based on Table 3.1, the 2CCNC, 3CCNC and 5CCNC quadrature weights, B_j can be expressed as follows:

$$B_j = \begin{cases} \frac{1}{2} h_j, & for \quad j = 0, N \\ h_j, & for \quad otherwise \end{cases}, \tag{3.25}$$

$$B_j = \begin{cases} \frac{1}{3} h_j, & for \quad j = 0, N \\ \frac{4}{3} h_j, & for \quad j = 1, 3, 5, \cdots, N-1 \\ \frac{2}{3} h_j, & for \quad otherwise \end{cases} \tag{3.26}$$

and

$$B_j = \begin{cases} \frac{14}{45} h_j, & for \quad j = 0, N \\ \frac{64}{45} h_j, & for \quad j = 1, 3, 5, \cdots, N-1 \\ \frac{28}{45} h_j, & for \quad j = 4, 8, 10, \cdots, N-4 \\ \frac{24}{45} h_j, & for \quad otherwise. \end{cases}. \tag{3.27}$$

3.3.5 Dense Linear Systems

In the standard or Full-Sweep approximation scheme (3.20), the system of linear equations can be expressed as

$$\mathcal{A}\widehat{\psi} = \widehat{f}, \tag{3.28}$$

where

$$\mathcal{A} = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \sigma_{1,3} & \cdots & \sigma_{1,N-3} & \sigma_{1,N-2} & \sigma_{1,N-1} \\ \sigma_{2,1} & \sigma_{2,2} & \sigma_{2,3} & \cdots & \sigma_{2,N-3} & \sigma_{2,N-2} & \sigma_{2,N-1} \\ \sigma_{3,1} & \sigma_{3,2} & \sigma_{3,3} & \cdots & \sigma_{3,N-3} & \sigma_{3,N-2} & \sigma_{3,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \sigma_{N-3,1} & \sigma_{N-3,2} & \sigma_{N-3,3} & \cdots & \sigma_{N-3,N-3} & \sigma_{N-3,N-2} & \sigma_{N-3,N-1} \\ \sigma_{N-2,1} & \sigma_{N-2,2} & \sigma_{N-2,3} & \cdots & \sigma_{N-2,N-3} & \sigma_{N-2,N-2} & \sigma_{N-2,N-1} \\ \sigma_{N-1,1} & \sigma_{N-1,2} & \sigma_{N-1,3} & \cdots & \sigma_{N-1,N-3} & \sigma_{N-1,N-2} & \sigma_{N-1,N-1} \end{bmatrix},$$

$$\widehat{\psi} = \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 & \cdots & \psi_{N-3} & \psi_{N-2} & \psi_{N-1} \end{bmatrix}^T$$

and

$$\widehat{f} = \begin{bmatrix} f_1 + (\sigma_{0,0})\psi_0 & f_2 & f_3 & \cdots & f_{N-3} & f_{N-2} & f_{N-1} + (\sigma_{N,N})\psi_N \end{bmatrix}^T.$$

$\mathcal{A}$ is a nonsymmetric dense square matrix of $(N-1) \times (N-1)$ order, $\widehat{\psi}$ is the unknown vector function and $\widehat{f}$ denotes a load vector function.

As afore-mentioned, the iteration process for MWM methods involves solving two independent systems. Therefore, to develop the formulation for all MWM methods, let the coefficient matrix $\mathcal{A}$ of equation (3.28) be decomposed as

$$\mathcal{A}^F = \mathcal{D}^F - \mathcal{L}^F - \mathcal{U}^F, \tag{3.29}$$

$$\mathcal{D}^F = \begin{bmatrix} \sigma_{1,1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \sigma_{2,2} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \sigma_{3,3} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{N-3,N-3} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{N-2,N-2} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{N-1,N-1} \end{bmatrix},$$

$$\mathcal{L}^F = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \sigma_{2,1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \sigma_{3,1} & \sigma_{3,2} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \sigma_{N-3,1} & \sigma_{N-3,2} & 0 & \cdots & 0 & 0 & 0 \\ \sigma_{N-2,1} & \sigma_{N-2,2} & 0 & \cdots & \sigma_{N-2,N-3} & 0 & 0 \\ \sigma_{N-1,1} & \sigma_{N-1,2} & 0 & \cdots & \sigma_{N-1,N-3} & \sigma_{N-1,N-2} & 0 \end{bmatrix},$$

and

$$\mathcal{U}^F = \begin{bmatrix} 0 & \sigma_{1,2} & \sigma_{1,3} & \cdots & \sigma_{1,N-3} & \sigma_{1,N-2} & \sigma_{1,N-1} \\ 0 & 0 & 0 & \cdots & \sigma_{2,N-3} & \sigma_{2,N-2} & \sigma_{2,N-1} \\ 0 & 0 & 0 & \cdots & \sigma_{3,N-3} & \sigma_{3,N-2} & \sigma_{3,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{N-2q,N-2q} & \sigma_{N-3,N-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{N-2,N-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix},$$

where $\mathcal{D}^F$, $\mathcal{L}^F$ and $\mathcal{U}^F$ are correspondingly the diagonal, strictly lower triangular and strictly upper triangular matrices.

3.3.6 Matrix Splitting

Fundamentally, the matrix splitting is crucial in order to carry out the systematic analysis of convergence conditions. Thus, the coefficient matrix of the generated CD-CCNC systems is analysed. Therefore, let matrix splitting of $\mathcal{A}$ be expressed as

$$\mathcal{A}^F = \mathcal{H}_r^F - \mathcal{K}_r^F, \quad r = 1, 2, \tag{3.30}$$

where $\mathcal{H}_1^F$ and $\mathcal{H}_2^F$ are defined by

$$\mathcal{H}_1^F = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & & & & & \\ \sigma_{2,1} & \sigma_{2,2} & & & & & \\ & & \sigma_{3,3} & \sigma_{3,4} & & & \\ & & \sigma_{4,3} & \sigma_{4,4} & & & \\ & & & & \ddots & & \\ & & & & & \sigma_{N-2,N-2} & \sigma_{N-2,N-1} \\ & & & & & \sigma_{N-1,N-2} & \sigma_{N-1,N-1} \end{bmatrix}$$

and

$$\mathcal{H}_2^F = \begin{bmatrix} \sigma_{1,1} & & & & & & \\ & \sigma_{2,2} & \sigma_{2,3} & & & & \\ & \sigma_{3,2} & \sigma_{3,3} & & & & \\ & & & \ddots & & & \\ & & & & \sigma_{N-3,N-3} & \sigma_{N-3,N-2} & \\ & & & & \sigma_{N-2,N-3} & \sigma_{N-2,N-2} & \\ & & & & & & \sigma_{N-1,N-1} \end{bmatrix},$$

respectively. Consequently, $\mathcal{H}_r^F$ is expressed as

$$\mathcal{K}_r^F = \mathcal{H}_r^F - \mathcal{A}^F, \quad r = 1, 2. \tag{3.31}$$

3.4 Standard or Full-Sweep Modified Arithmetic Mean iterative method

The general formulation for Full-Sweep Modified Arithmetic Mean (FSMAM) iterative method is defined as follows

$$\left.\begin{aligned} (\mathcal{D}^F - \theta_1\mathcal{L}^F)\widehat{\psi}^{(\mathscr{F})} &= ((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F)\widehat{\psi}^{(k)} + \theta_1\widehat{f} \\ (\mathcal{D}^F - \theta_2\mathcal{U}^F)\widehat{\psi}^{(\mathscr{B})} &= ((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)\widehat{\psi}^{(k)} + \theta_2\widehat{f} \\ \widehat{\psi}^{(k+1)} &= \frac{1}{2}(\widehat{\psi}^{(\mathscr{F})} + \widehat{\psi}^{(\mathscr{B})}) \end{aligned}\right\} \quad k = 0, 1, \ldots \tag{3.32}$$

where $(\mathcal{D}^F - \theta_1\mathcal{L}^F)$, $(\mathcal{D}^F - \theta_2\mathcal{U}^F)$, $((1-\theta_1)\mathcal{D}^F - \theta_1\mathcal{L}^F)$ and $((1-\theta_2)\mathcal{D}^F - \theta_2\mathcal{U}^F)$ are nonsingular triangular matrices, θ_1 and θ_2 with optimal parameter values, and $\widehat{\psi}^{(k)}$ is an initial vector at the kth iteration to the solutions of $\widehat{\psi}^{(\mathscr{F})}$ and $\widehat{\psi}^{(\mathscr{B})}$. In the case of $\theta_1 = \theta_2$, the FSMAM method is equivalent to the conventional FSAM method.

From the equation (3.32), the FSMAM iterative scheme for the solution of the linear system (3.28) is

$$\widehat{\psi}^{(k+1)} = \mathcal{Z}_{FSMAM}\widehat{\psi}^{(k)} + z_{FSMAM}\widehat{f}, \quad k = 0, 1, 2, \ldots \tag{3.33}$$

where

$$\mathcal{Z}_{FSMAM} = \frac{1}{2}\Big[(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1}((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F) + (\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)\Big]$$

is a square matrix. Meanwhile,

$$z_{FSMAM} = \frac{1}{2}\Big[\theta_1(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1} + \theta_2(\mathcal{D}^F - \theta_2\mathcal{U}^F)\Big]$$

is a load vector that is obtained from the coefficients of $\widehat{f}$.
The general condition which guarantees the convergence of the FSMAM iterative method in (3.32) for solving the linear system (3.28) is described in the following Theorems and Proofs.

Based on the matrix splitting of $\mathcal{A}^F$ in Eq. (3.30), the conditions that guarantees the convergence of the FSMAM iterative method are described in the following Theorems.

Theorem 3.1. *Given an $(N-1) \times (N-1)$ nonsingular diagonally dominant matrix $\mathcal{A}^F$ with its components $\sigma_{i,i} > 0$, $fori = 1, 2, \ldots N-1$, and*

$$\mathcal{A}^F = \mathcal{H}_1^F - \mathcal{K}_1^F = \mathcal{H}_2^F - \mathcal{K}_2^F$$

where matrices $(\mathcal{H}_1^F)^{-1}$ and $(\mathcal{H}_2^F)^{-1}$ are nonsingular with $\|(\mathcal{H}_1^F)^{-1}\| \geq 0$, $\|\mathcal{K}_1^F\| \geq 0$ and $\|(\mathcal{H}_2^F)^{-1}\| \geq 0$, $\|\mathcal{K}_2^F\| \geq 0$. The FSMAM iterative scheme (3.33) is convergent for $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$.

Proof. By hypothesis, $\mathcal{A}^F$ is an $(N-1) \times (N-1)$ nonsingular matrix.

Since $\mathcal{H}_1^F = \mathcal{D}^F - \theta_1\mathcal{L}^F$ and $\mathcal{H}_2^F = \mathcal{D}^F - \theta_2\mathcal{U}^F$ are strictly diagonally dominant matrices with positive entries on the diagonal for $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$.

The matrices $\mathcal{K}_1^F = (1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F$ and $\mathcal{K}_2^F = (1-\theta_2)\mathcal{D}^F + \theta_1\mathcal{L}^F$ are triangular and nonnegative.

Since

$$\mathcal{H}_1^F - \mathcal{K}_1^F = \mathcal{H}_2^F - \mathcal{K}_2^F = \mathcal{A}^F$$

then we have

$$\mathcal{Q}^F = \frac{1}{2}(\mathcal{H}_1^F)^{-1}\mathcal{K}_1^F + \frac{1}{2}(\mathcal{H}_2^F)^{-1}\mathcal{K}_2^F = \mathcal{I} - \left[\frac{1}{2}(\mathcal{H}_1^F)^{-1} + \frac{1}{2}(\mathcal{H}_2^F)^{-1}\right]\mathcal{A}^F \qquad (3.34)$$

or also can be written as

$$\frac{1}{2}(\mathcal{H}_1^F)^{-1} + \frac{1}{2}(\mathcal{H}_2^F)^{-1} = (\mathcal{I} - \mathcal{Q}^F)(\mathcal{A}^F)^{-1}. \qquad (3.35)$$

The proof of the theorem runs parallel to a standard proof given in [44]. Since $\mathcal{Q}^F = (\mathcal{H}_r^F)^{-1}\mathcal{K}_r^F$, then the spectral radius is

$$\rho_{FSMAM}(\mathcal{Q}^F) < 1. \qquad (3.36)$$

Therefore, the FSMAM iterative scheme (3.33) converges for any initial vector $\widehat{\psi}^{(0)}$ with conditions of $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$. Hence, the Theorem 3.1 is proved. □

3.5 Standard or Full-Sweep Modified Geometric Mean Iterative Method

Consider the Full-Sweep Modified Arithmetic Mean (FSMGM) iterative method which consists of two independent systems i.e $\widehat{\psi}^{(\mathscr{F})}$ and $\widehat{\psi}^{(\mathscr{B})}$ can be expressed as follows:

$$\left.\begin{aligned} (\mathcal{D}^F - \theta_1\mathcal{L}^F)\widehat{\psi}^{(\mathscr{F})} &= ((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F)\widehat{\psi}^{(k)} + \theta_1\widehat{f} \\ (\mathcal{D}^F - \theta_2\mathcal{U}^F)\widehat{\psi}^{(\mathscr{B})} &= ((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)\widehat{\psi}^{(k)} + \theta_2\widehat{f} \\ \widehat{\psi}^{(k+1)} &= (\widehat{\psi}^{(\mathscr{F})} \circ \widehat{\psi}^{(\mathscr{B})})^{\frac{1}{2}}, \quad \widehat{\psi}^{(\mathscr{F})} \circ \widehat{\psi}^{(\mathscr{B})} \geq 0 \end{aligned}\right\} \quad k = 0, 1, \ldots \qquad (3.37)$$

where θ_1 and θ_2 represent two optimal FSMGM parameters, $\widehat{\psi}^{(k)}$ is initial vector approximation at the kth iteration to the solutions of $\widehat{\psi}^{(\mathscr{F})}$ and $\widehat{\psi}^{(\mathscr{B})}$, '$\circ$' is the Hadamard product operator and $(\circ)^{\frac{1}{2}}$ is the Hadamard power. The Hadamard product or Schur product [21] '$\circ$', is a binary operation that takes two vectors from the same dimensions, and produces another vector. In the case of $\theta_1 = \theta_2$, this method is equivalent to the conventional FSGM method.

Based on the formulation (3.37), the iterative form of the FSMGM method for solving the linear system (3.28) is as follows:

$$\widehat{\psi}^{(k+1)} = \mathcal{Z}_{FSMGM}\widehat{\psi}^{(k)} + z_{FSMGM}\widehat{f}, \quad k = 0, 1, 2, \ldots \tag{3.38}$$

where $\mathcal{Z}_{FSMGM}$, is defined by

$$\left[(\mathcal{D}^{\mathcal{F}} - \theta_1\mathcal{L}^F)^{-1}((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F)(\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)\right]^{\frac{1}{2}},$$

is square iteration matrices for FSMGM. Meanwhile,

$$z_{FSMGM} = \left[\theta_1\theta_2(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1}(\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}\right]^{\frac{1}{2}}$$

is a load vector obtained from the right hand side $\widehat{f}$.
The general condition which guarantees the convergence of the FSMGM iterative method for solving linear systems (3.28) is described in the following Theorems.

Theorem 3.2. *Let $\mathcal{Z}_{FSMGM}$ be an $(N-1)\times(N-1)$ matrix. Then, the necessary conditions for the FSMGM method to be convergent are that $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$.*

Proof. Since the eigenvalues λ_j of $\mathcal{Z}_{FSMGM}$ is the zero of the characteristic polynomial, the determinant of the $\mathcal{Z}_{FSMGM}$ satisfies the following relation

$$\det(\mathcal{Z}_{FSMGM}) = \prod_{j=0}^{N} \lambda_j \tag{3.39}$$

where multiple eigenvalues are repeated according to their algebraic multiplicity. Since $(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{\frac{1}{2}}$, $(\mathcal{D}^F - \theta_2\mathcal{U}^F)^{\frac{1}{2}}$, $((1-\theta_1)\mathcal{D}^F - \theta_1\mathcal{U}^F)^{\frac{1}{2}}$ and $((1-\theta_2)\mathcal{D}^F - \theta_2\mathcal{L}^F)^{\frac{1}{2}}$ are nonsingular triangular matrices, hence the det $\mathcal{Z}_{FSMGM}$ is defined as follows:

$$\det\left[(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1}((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F)(\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)\right]^{\frac{1}{2}}. \tag{3.40}$$

Because $\mathcal{L}$ and $\mathcal{U}$ are strictly lower and upper triangular matrices respectively, $\det(\mathcal{D}^F)^{-1} = \det(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1} = \det(\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}$

$$\begin{aligned}
&= \det\left[(\mathcal{D}^F)^{-1}((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F)(\mathcal{D}^F)^{-1}((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)\right]^{\frac{1}{2}} \\
&= \det\{(\mathcal{D}^F)^{-1}\}^{\frac{1}{2}} \det\{(1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F\}^{\frac{1}{2}} \det\{(\mathcal{D}^F)^{-1}\}^{\frac{1}{2}} \det\{(1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F\}^{\frac{1}{2}} \\
&= \det\{(1-\theta_1)\mathcal{I} + \theta_1(\mathcal{D}^F)^{-1}\mathcal{U}^F\}^{\frac{1}{2}} \det\{(1-\theta_2)\mathcal{I} + \theta_2(\mathcal{D}^F)^{-1}\mathcal{L}^F\}^{\frac{1}{2}} \\
&= \left((1-\theta_1)^{\frac{1}{2}}\right)^{N-1}\left((1-\theta_2)^{\frac{1}{2}}\right)^{N-1} \\
&= (1-\theta_1)^{\frac{N-1}{2}}(1-\theta_2)^{\frac{N-1}{2}}.
\end{aligned}$$

This now implies

$$\rho(\mathcal{Z}_{FSMGM}) \geq \left(\mid 1-\theta_1 \mid^{\frac{N-1}{2}} \mid 1-\theta_2 \mid^{\frac{N-1}{2}} \right)^{\frac{2}{N-1}} = \mid (1-\theta_1)(1-\theta_2) \mid . \quad (3.41)$$

Therefore, based on the Theorems 2.6 and 3.2, the FSMGM iterative scheme in Eq. (3.38) converges for any initial vector $\widehat{\psi}^{(0)}$ with conditions $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$. Hence Theorem 3.2 is proved. □

Definition 3.5.1. *For the ith element of independent vectors $\widehat{\psi}^{(\mathscr{F})}$ and $\widehat{\psi}^{(\mathscr{B})}$, the unknown vector $\widehat{\psi}^{(k+1)}$ for $i = 1, 2, \ldots, N-1$ can be determined as follows:*

(i) $(\widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})})^{\frac{1}{2}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} > 0 \quad \& \quad \widehat{\psi}_i^{(\mathscr{B})} > 0 \quad (Case\ 1),$

(ii) $(\widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})})^{\frac{1}{2}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} < 0 \quad \& \quad \widehat{\psi}_i^{(\mathscr{B})} < 0 \quad (Case\ 2),$

(iii) $(\widehat{\psi}_i^{(\mathscr{F})} - \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid^{\frac{1}{2}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} > 0 \quad \& \quad \widehat{\psi}_i^{(\mathscr{B})} < 0 \quad (Case\ 3),$

(iv) $(\widehat{\psi}_i^{(\mathscr{B})} - \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid^{\frac{1}{2}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} < 0 \quad \& \quad \widehat{\psi}_i^{(\mathscr{B})} > 0 \quad (Case\ 4).$

3.6 Standard or Full-Sweep Modified Harmonic Mean Iterative Method

Basically, the standard or Full-Sweep Modified Harmonic Mean (FSMHM) iterative method is developed based on the relationship between FSMAM and FSMGM iterative methods. Typically, it is appropriate for situations when the average of rates is desired. The general form of FSMHM iterative method for approximating the two independent solutions $\widehat{\psi}^{(\mathscr{F})}$ and $\widehat{\psi}^{(\mathscr{B})}$ are as follows:

$$\left.\begin{aligned} (\mathcal{D}^F - \theta_1\mathcal{L}^F)\widehat{\psi}^{(\mathscr{F})} &= ((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F)\widehat{\psi}^{(k)} + \theta_1\widehat{f} \\ (\mathcal{D}^F - \theta_2\mathcal{U}^F)\widehat{\psi}^{(\mathscr{B})} &= ((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)\widehat{\psi}^{(k)} + \theta_2\widehat{f} \\ \widehat{\psi}^{k+1} &= \frac{2\widehat{\psi}^{(\mathscr{F})} \circ \widehat{\psi}^{(\mathscr{B})}}{\widehat{\psi}^{(\mathscr{F})} + \widehat{\psi}^{(\mathscr{B})}}, \quad \widehat{\psi}^{(\mathscr{F})} + \widehat{\psi}^{(\mathscr{B})} \neq 0 \end{aligned}\right\} \quad k = 0, 1, \ldots \quad (3.42)$$

where θ_1 and θ_2 represent two optimal FSMHM parameters, $\widehat{\psi}^{(k)}$ is an unknown vector at the kth iteration, '$\circ$' is the Hadamard product operator and $(\circ)^{\frac{1}{2}}$ is the Hadamard power [21]. The Hadamard product of '$\circ$' is a binary operation that takes two vectors of the same dimensions, and produces another vector. In the case of $\theta_1 = \theta_2$, this method is equivalent to the standard FSHM method.

Based on Eq. (3.42), the iterative form of the FSMHM method for solving linear system (3.28) is as follows:

$$\widehat{\psi}^{(k+1)} = \mathcal{Z}_{FSMHM}\widehat{\psi}^{(k)} + z_{FSMHM}\widehat{f}, \quad k = 0, 1, 2, \ldots \tag{3.43}$$

where $\mathcal{Z}_{FSMHM}$, defined by

$$\frac{2(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1}((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F)(\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)}{(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1}((1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F) + (\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}((1-\theta_2)\mathcal{D}^F + \theta_2\mathcal{L}^F)},$$

is a square iteration matrix for FSMHM. Meanwhile,

$$z_{FSMHM} = \frac{2\theta_1\theta_2(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1}(\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}}{(\mathcal{D}^F - \theta_1\mathcal{L}^F)^{-1} + (\mathcal{D}^F - \theta_2\mathcal{U}^F)^{-1}}$$

is a load vector that is obtained from the right hand side $\widehat{f}$.

Based on the matrix splitting of $\mathcal{A}^{\mathcal{F}}$ in (3.30), the conditions which guarantees the convergence of the FSMHM method are described in the following Theorem:

Theorem 3.3. *Given an $(N-1)\times(N-1)$ nonsingular diagonally dominant matrix $\mathcal{A}^F$ with its components $\sigma_{i,i} > 0$, $fori = 1, 2, \ldots N-1$, and*

$$\mathcal{A}^F = \mathcal{H}_1^F - \mathcal{K}_1^F = \mathcal{H}_2^F - \mathcal{K}_2^F$$

($\mathcal{H}_1^F$ and $\mathcal{H}_2^F$ are nonsingular) with $\|(\mathcal{H}_1^F)^{-1}\| \geq 0$, $\|\mathcal{K}_1^F\| \geq 0$ and $\|(\mathcal{H}_2^F)^{-1}\| \geq 0$, $\|\mathcal{K}_2^F\| \geq 0$. The FSMHM iterative scheme (3.43) is convergent for $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$.

Proof. By hypothesis, let $\mathcal{A}^F$ be an $(N-1)\times(N-1)$ nonsingular matrix.

Since $\mathcal{H}_1^F = \mathcal{D}^F - \theta_1\mathcal{L}^F$ and $\mathcal{H}_2^F = \mathcal{D}^F - \theta_2\mathcal{U}^F$ are strictly diagonally dominant matrices with positive entries on the diagonal for of $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$.

The matrices $\mathcal{K}_1^F = (1-\theta_1)\mathcal{D}^F + \theta_1\mathcal{U}^F$ and $\mathcal{K}_2^F = (1-\theta_2)\mathcal{D}^F + \theta_1\mathcal{L}^F$ are triangular and nonnegative.

Since

$$\mathcal{H}_1^F - \mathcal{K}_1^F = \mathcal{H}_2^F - \mathcal{K}_2^F = \mathcal{A}^F$$

then it can be written as

$$\mathcal{Q}^F = \frac{2[(\mathcal{H}_1^F)^{-1}\mathcal{K}_1^F \circ (\mathcal{H}_2^F)^{-1}\mathcal{K}_2^F]}{(\mathcal{H}_1^F)^{-1}\mathcal{K}_1^F + (\mathcal{H}_2^F)^{-1}\mathcal{K}_2^F} = \mathcal{I} - \left[\frac{2[(\mathcal{H}_1^F)^{-1} \circ (\mathcal{H}_2^F)^{-1}]}{(\mathcal{H}_1^F)^{-1} + (\mathcal{H}_2^F)^{-1}}\right]\mathcal{A}^F \tag{3.44}$$

or also can be expressed as

$$\frac{2[(\mathcal{H}_1^F)^{-1} \circ (\mathcal{H}_2^F)^{-1}]}{(\mathcal{H}_1^F)^{-1} + (\mathcal{H}_2^F)^{-1}} = (\mathcal{I} - \mathcal{Q}^F)(\mathcal{A}^F)^{-1}. \tag{3.45}$$

The proof of the theorem runs parallel to a standard proof given in [44]. Since $\mathcal{Q}^F = (\mathcal{H}_r^F)^{-1}\mathcal{K}_r^F$, then the spectral radius is

$$\rho_{FSMHM}(\mathcal{Q}^F) < 1. \tag{3.46}$$

Therefore, the FSMHM iterative scheme in (3.43) converges for any initial vector $\widehat{\psi}^{(0)}$ with conditions $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$. Hence, Theorem 3.3 is proved. □

Definition 3.6.1. *For the ith element of independent vectors $\widehat{\psi}_i^{(\mathscr{F})}$ and $\widehat{\psi}_i^{\mathscr{B}}$, the unknown vector $\widehat{\psi}_i^{(k+1)}$ for $i = 1, 2, \ldots, N-1$ can be determined as follows:*

(i) $0, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})} = 0 \qquad (Case\ 1),$

(ii) $\dfrac{2\widehat{\psi}_i^{(\mathscr{F})} \circ \widehat{\psi}_i^{(\mathscr{B})}}{\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} > 0\ \&\ \widehat{\psi}_i^{\mathscr{B}} > 0 \qquad (Case\ 2),$

(iii) $\dfrac{2\widehat{\psi}_i^{(\mathscr{F})} \circ \widehat{\psi}_i^{(\mathscr{B})}}{\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})}}, \quad if\ \widehat{\psi}_i^{(\mathscr{F})} < 0\ \&\ \widehat{\psi}_i^{(\mathscr{B})} < 0 \qquad (Case\ 3),$

(iv) $\dfrac{2[(\widehat{\psi}_i^{(\mathscr{F})})^2 - 2\widehat{\psi}_i^{(\mathscr{F})}(\mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid)^{\frac{1}{2}} + \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid]}{\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})}},$
$if \quad \widehat{\psi}_i^{(\mathscr{F})} > 0\ \&\ \widehat{\psi}_i^{(\mathscr{B})} < 0 \qquad (Case\ 4),$

(v) $\dfrac{2[(\widehat{\psi}_i^{(\mathscr{B})})^2 - 2\widehat{\psi}_i^{(\mathscr{B})}(\mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid)^{\frac{1}{2}} + \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid]}{\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})}},$
$if \quad \widehat{\psi}_i^{(\mathscr{F})} < 0\ \&\ \widehat{\psi}_i^{\mathscr{B}} > 0 \qquad (Case\ 5).$

for Case 2 to Case 5, $\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})} \neq 0$ are hold.

3.7 Computational Complexity-Reduction Approaches

In this Section, the proposed MWM iterative methods with their corresponding approximation schemes are modified further to reduce the computational complexity as mentioned in Section 2.7. The general idea of the implementation of half- and quarter-sweep iteration techniques are also known as computational complexities-reduction approaches. Therefore, to execute these techniques, let the solution domain half- and quarter-sweep be divided into N equidistant subintervals.

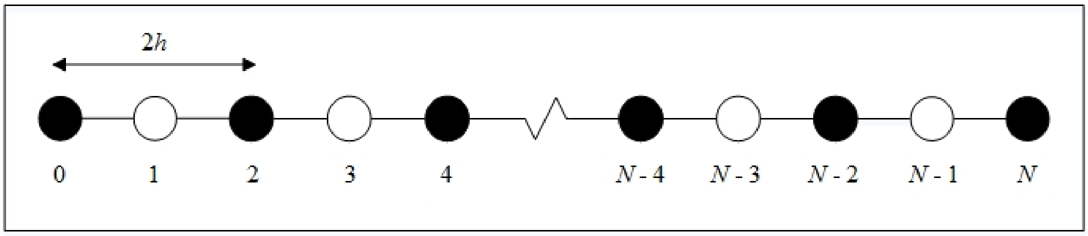

Figure 3.2: Half-Sweep iteration

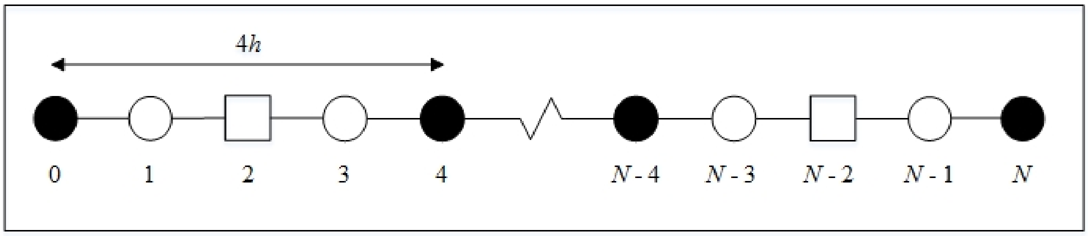

Figure 3.3: Quarter-Sweep iteration

Similar to the Full-Sweep iteration technique, implementations of the half- and quarter-sweep iteration techniques will only consider nodal points of type ● for the iteration process until the convergence criterion is reached. Then, computations for the remaining points i.e. types of ○ and □ can be carried out independently [17, 80]. Therefore, in this research, second order Lagrange interpolation techniques are implemented to compute the remaining nodal points. Further discussion on the remaining points calculations are given in Section 3.11.4. The applications of the half- and quarter-sweep iteration techniques reduce the computational complexity of the full-sweep methods approximately to half and quarter, respectively. These techniques are combined with Gauss-Seidel, Arithmetic Mean, Modified Arithmetic Mean, Geometric Mean, Modified Geometric Mean, Harmonic Mean and Modified Harmonic Mean iterative methods in order to solve the linear Fredholm IDEs.

3.8 Modified Finite Difference Schemes

In this section, the standard Finite Difference (FD) schemes is modified by combining the half- and quarter-sweep iteration technique as explained in Section 3.7. Throughout this chapter, The values of q are determined by

$$q = \begin{cases} 1 & \text{for Full-Sweep} \\ 2 & \text{for Half-Sweep} \\ 4 & \text{for Quarter-Sweep} \end{cases} . \tag{3.47}$$

The modified FD schemes can be expressed as follows:

(i) First order derivative forward finite divided difference approximation:

$$f'(\psi_i) = \frac{f(\psi_{i+q}) - f(\psi_i)}{qh} + O(h), \tag{3.48}$$

(ii) First order derivative backward finite divided difference approximation:

$$f'(\psi_i) = \frac{f(\psi_i) - f(\psi_{i-q})}{qh} + O(h), \tag{3.49}$$

(iii) First order derivative centered finite divided difference approximation:

$$f'(\psi_i) = \frac{f(\psi_{i+q}) - f(\psi_{i-q})}{2qh} + O(h^2), \tag{3.50}$$

(iv) Second order derivative centered finite divided difference approximation:

$$f''(\psi_i) = \frac{f(\psi_{i+q}) - 2f(\psi_i) + f(\psi_{i-q})}{(qh)^2} + O(h^2), \tag{3.51}$$

(v) Fourth order derivative centered finite divided difference approximation:

$$f^{(iv)}(\psi_i) = \frac{f(\psi_{i+2q}) - 4f(\psi_{i+q}) + 6f(\psi_i) - 4f(\psi_{i-q}) + f(\psi_{i-2q})}{(qh)^4} + O(h^2). \tag{3.52}$$

3.8.1 Modified Central Difference Operator Formula

Generally, the corresponding Half- and Quarter-Sweep Central Difference can also be written in the operator form. Let

$$\delta = H^{\frac{1}{2}q} - H^{-\frac{1}{2}q}, \tag{3.53}$$

we then have

$$\delta\psi_i = (H^{\frac{1}{2}q} - H^{-\frac{1}{2}q})\psi_i = H^{\frac{1}{2}q}\psi_i - H^{-\frac{1}{2}q}\psi_i = \psi_{i+\frac{1}{2}q} - \psi_{i-\frac{1}{2}q}. \tag{3.54}$$

Similarly, the mth order derivative of Half- and Quarter-Sweep Central Difference may be defined by

$$\delta^m \psi_i = \delta^{m-1}(\delta\psi_i) = \delta^{m-1}\psi_{i+\frac{1}{2}q} - \delta^{m-1}\psi_{i-\frac{1}{2}q}, \tag{3.55}$$

where the condition of q is as described in the equation (3.47).

3.9 Modified Composite Closed Newton-Cotes Schemes

The Composite Closed Newton-Cotes (CCNC) schemes namely 2CCNC, 3CCNC and 5CCNC schemes as shown in Table 3.1 are modified based on the half- and quarter-sweep iteration technique namely Half- and Quarter-Sweep CCNC. The Half- and Quarter-Sweep 2CCNC, 3CCNC and 5CCNC are given in Table 3.2. The condition of q for the Half- and Quarter-Sweep CCNC schemes is described in equation (3.47).

3.9.1 Generalised Approximation Equations

The generated standard approximation equation (3.20) may be modified by amalgamating the half- and quarter-sweep iteration technique as described in Section 3.7. Therefore, the generalised approximation equations can be expressed as follows:

$$\sum_{m=1}^{M-1} \frac{\delta^m \psi_N(x_i)}{s(qh_i)^m} - \sum_{j=q,2q,3q}^{N-q} B_j K(x_i,\xi_j)\psi_N(\xi_j) = f(x_i) \quad x \in (a,b) \tag{3.56}$$

for $i = q, 2q \ldots, N-1$. Based on the modified 2CCNC, 3CCNC and 5CCNC schemes for the integral term of problem (1.1), the quadrature weights, B_j, satisfy the following relations respectively

$$B_j = \begin{cases} \frac{1}{2}qh_j, & for \quad j = 0, N \\ qh_j, & for \quad otherwise \end{cases}, \tag{3.57}$$

$$B_j = \begin{cases} \frac{1}{3}qh_j, & for \quad j = 0, N \\ \frac{4}{3}qh_j, & for \quad j = 1, 3, 5, \cdots, N-1 \\ \frac{2}{3}qh_j, & for \quad otherwise \end{cases}, \tag{3.58}$$

and

$$B_j = \begin{cases} \frac{14}{45}qh_j, & for \quad j = 0, N \\ \frac{64}{45}qh_j, & for \quad j = 1, 3, 5, \cdots, N-1 \\ \frac{28}{45}qh_j, & for \quad j = 4, 8, 10, \cdots, N-4 \\ \frac{24}{45}qh_j, & for \quad otherwise \end{cases} \tag{3.59}$$

where the value q is described by equation (3.47).

Table 3.2: Modified Composite Closed Newton-Cotes Formulae

(i) Modified 2-point composite closed Newton-Cotes (2CCNC)

$$Q_1 = \frac{qh}{2}\Big[\psi(a) + 2\sum_{j=q,2q,3q}^{N-q}\psi(a+jh) + \psi(b)\Big] - \frac{b-a}{12}h^2 f^{(2)}(\varepsilon) \tag{3.60}$$

(ii) Modified 3-point composite closed Newton-Cotes (3CCNC)

$$\begin{aligned} Q_2 &= \frac{qh}{3}\Big[\psi(a) + 4\sum_{j=q,3q,5q}^{N-q}\psi(a+jh) + 2\sum_{j=2q,4q,6q}^{N-2q}\psi(a+jh) + \psi(b)\Big] \\ &\quad -\frac{b-a}{180}h^4 f^{(4)}(\varepsilon) \end{aligned} \tag{3.61}$$

(iii) Modified 5-point composite closed Newton-Cotes (5CCNC)

$$\begin{aligned} Q_3 &= \frac{qh}{45}\Big[14\psi(a) + 64\sum_{j=q,3q,5q}^{N-q}\psi(a+jh) + 24\sum_{j=2q,6q,10q}^{N-2q}\psi(a+jh) \\ &\quad +28\sum_{j=4q,8q,12q}^{N-4q}\psi(a+jh) + 14\psi(b)\Big] - \frac{2(b-a)}{945}h^6 f^{(6)}(\varepsilon) \end{aligned} \tag{3.62}$$

3.9.2 Generalised Dense Linear Systems

Based on the generalised approximation equation (3.56), the generalised Full-, Half- and Quarter-Sweep dense system of linear equations can be expressed as

$$\mathcal{A}_g\widehat{\psi} = \widehat{f}_g, \tag{3.63}$$

where $\mathcal{A}_g$ is generalised coefficient matrix, $\widehat{f}_g$ is generalised load vector function and $\widehat{\psi}$ is unknown vector function to be determined. Equation (3.63) may be expressed as follows

$$\mathcal{A}_g = \begin{bmatrix} \sigma_{q,q} & \sigma_{q,2q} & \sigma_{q,3q} & \cdots & \sigma_{q,N-3q} & \sigma_{q,N-2q} & \sigma_{q,N-q} \\ \sigma_{2q,q} & \sigma_{2q,2q} & \sigma_{2q,3q} & \cdots & \sigma_{2q,N-3q} & \sigma_{2q,N-2q} & \sigma_{2q,N-q} \\ \sigma_{3q,q} & \sigma_{3q,2q} & \sigma_{3q,3q} & \cdots & \sigma_{3q,N-3q} & \sigma_{3q,N-2q} & \sigma_{3q,N-q} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \sigma_{N-3q,q} & \sigma_{N-3q,2q} & \sigma_{N-3q,3q} & \cdots & \sigma_{N-3q,N-3q} & \sigma_{N-3q,N-2q} & \sigma_{N-3q,N-q} \\ \sigma_{N-2q,q} & \sigma_{N-2q,2q} & \sigma_{N-2q,3q} & \cdots & \sigma_{N-2q,N-3q} & \sigma_{N-2q,N-2q} & \sigma_{N-2q,N-q} \\ \sigma_{N-q,q} & \sigma_{N-q,2q} & \sigma_{N-q,3q} & \cdots & \sigma_{N-q,N-3q} & \sigma_{N-q,N-2q} & \sigma_{N-q,N-q} \end{bmatrix},$$

$$\widehat{f}_g = \begin{bmatrix} f_q + (\sigma_{0,0})\psi_0 & f_{2q} & f_{3q} & \cdots & f_{N-3q} & f_{N-2q} & f_{N-q} + (\sigma_{N,N})\psi_N \end{bmatrix}^T$$

and

$$\widehat{\psi} = \begin{bmatrix} \psi_q & \psi_{2q} & \psi_{3q} & \cdots & \psi_{N-3q} & \psi_{N-2q} & \psi_{N-q} \end{bmatrix}^T,$$

respectively.

$\mathcal{A}_g$ is a nonsymmetric dense square matrix of $(\frac{N}{q} - 1) \times (\frac{N}{q} - 1)$ order, where the value of q is determined by Half- and Quarter-Sweep cases as described in equation (3.47). The elements in a coefficient matrix $\mathcal{A}_g$ depend on the order of FIDEs and its corresponding generalised approximation equations. Therefore, the Half- and the quarter Sweep iteration techniques reduce the orders of the original or full matrix from $(N-1) \times (N-1)$ to $(\frac{N}{2} - 1) \times (\frac{N}{2} - 1)$ and from $(\frac{N}{2} - 1) \times (\frac{N}{2} - 1)$ to $(\frac{N}{4} - 1) \times (\frac{N}{4} - 1)$, respectively.

As afore-mentioned, the iteration process for the proposed MWM methods involves solving two independent systems. Therefore, to develop the formulation for all proposed MWM methods, let the generalised coefficient matrix $\mathcal{A}_g$ in equation (3.63) be decomposed as

$$\mathcal{A}_g^H = \mathcal{D}_g^H - \mathcal{L}_g^H - \mathcal{U}_g^H, \tag{3.64}$$

and

$$\mathcal{A}_g^Q = \mathcal{D}_g^Q - \mathcal{L}_g^Q - \mathcal{U}_g^Q, \tag{3.65}$$

where $\mathcal{A}_g^H$ and $\mathcal{A}_g^Q$ represent respectively the generalised Half- and Quarter-Sweep matrices with a superscript index H for $q = 2$, and an index Q for $q = 4$.

Therefore, the generalised matrix decomposition can be expressed as

$$\mathcal{D}_g = \begin{bmatrix} \sigma_{q,q} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \sigma_{2q,2q} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \sigma_{3q,3q} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{N-3q,N-3q} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{N-2q,N-2q} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{N-q,N-q} \end{bmatrix},$$

$$\mathcal{L}_g = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \sigma_{2q,q} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \sigma_{3q,q} & \sigma_{3q,2q} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \sigma_{N-3q,q} & \sigma_{N-3q,2q} & 0 & \cdots & 0 & 0 & 0 \\ \sigma_{N-2q,q} & \sigma_{N-2q,2q} & 0 & \cdots & \sigma_{N-2q,N-3q} & 0 & 0 \\ \sigma_{N-q,q} & \sigma_{N-q,2q} & 0 & \cdots & \sigma_{N-q,N-3q} & \sigma_{N-q,N-2q} & 0 \end{bmatrix}$$

and

$$\mathcal{U}_g = \begin{bmatrix} 0 & \sigma_{q,2q} & \sigma_{q,3q} & \cdots & \sigma_{q,N-3q} & \sigma_{q,N-2q} & \sigma_{q,N-q} \\ 0 & 0 & 0 & \cdots & \sigma_{2q,N-3q} & \sigma_{2q,N-2q} & \sigma_{2q,N-q} \\ 0 & 0 & 0 & \cdots & \sigma_{3q,N-3q} & \sigma_{3q,N-2q} & \sigma_{3q,N-q} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{N-2q,N-2q} & \sigma_{N-3q,N-q} \\ 0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{N-2q,N-q} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}$$

for the diagonal, the strictly lower triangular and strictly upper triangular matrices, respectively.

3.9.3 Generalised Matrix Splitting

The Half- and Quarter-Sweep matrix splitting of $\mathcal{A}_g$ are respectively

$$\mathcal{A}_g^H = \mathcal{H}_r^H - \mathcal{K}_r^H, \quad r = 1, 2, \tag{3.66}$$

and

$$\mathcal{A}_g^Q = \mathcal{H}_r^Q - \mathcal{K}_r^Q, \quad r = 1, 2, \tag{3.67}$$

with the generalised $\mathcal{H}_1$ and $\mathcal{H}_2$ defined by

$$\mathcal{H}_1 = \begin{bmatrix} \sigma_{q,q} & \sigma_{q,2q} & & & & & \\ \sigma_{2q,q} & \sigma_{2q,2q} & & & & & \\ & & \sigma_{3q,3q} & \sigma_{3q,4q} & & & \\ & & \sigma_{4q,3q} & \sigma_{4q,4q} & & & \\ & & & & \ddots & & \\ & & & & & \sigma_{N-2q,N-2q} & \sigma_{N-2q,N-q} \\ & & & & & \sigma_{N-q,N-2q} & \sigma_{N-q,N-q} \end{bmatrix}$$

and

$$\mathcal{H}_2 = \begin{bmatrix} \sigma_{q,q} & & & & & & \\ & \sigma_{2q,2q} & \sigma_{2q,3q} & & & & \\ & \sigma_{3q,2q} & \sigma_{3q,3q} & & & & \\ & & & \ddots & & & \\ & & & & \sigma_{N-3q,N-3q} & \sigma_{N-3q,N-2q} & \\ & & & & \sigma_{N-2q,N-3q} & \sigma_{N-2q,N-2q} & \\ & & & & & & \sigma_{N-q,N-q} \end{bmatrix},$$

respectively, where $\mathcal{H}_r^H$ and $\mathcal{H}_r^Q$ are for $q = 2$ and $q = 4$, respectively. Consequently, $\mathcal{K}_r^H$ and $\mathcal{K}_r^Q$ are expressed as

$$\mathcal{K}_r^H = \mathcal{H}_r^H - \mathcal{A}_g^H, \quad r = 1, 2 \tag{3.68}$$

and

$$\mathcal{K}_r^Q = \mathcal{H}_r^Q - \mathcal{A}_g^Q, \quad r = 1, 2, \tag{3.69}$$

respectively.

3.10 Half- and Quarter-Sweep Modified Weighted Mean methods

In this section, the implementation of half- and quarter-sweep iterations techniques are applied on the proposed MWM modified iterative methods to reduce the complexity of the computational elements. Consequently, the convergence Theorems and Proofs for the Half- and Quarter-Sweep proposed MWM iterative methods are presented.

3.10.1 Half- and Quarter-Sweep Modified Arithmetic Mean iterative methods

The formulation of the Half-Sweep Modified Arithmetic Mean (HSMAM) and the Quarter-Sweep Modified Arithmetic Mean (QSMAM) methods are presented together with systematic conditions of convergence. The HSMAM and QSMAM iterative methods for approximating the solution $\widehat{\psi}^{(k+1)}$ of generalised system (3.63) as described in Section 3.9.2 are as follows:

$$\left.\begin{aligned} (\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)\widehat{\psi}^{(\mathscr{F})} &= ((1-\theta_1)\mathcal{D}_g^H + \theta_1\mathcal{U}_g^H)\widehat{\psi}^{(k)} + \theta_1\widehat{f}_g \\ (\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)\widehat{\psi}^{(\mathscr{B})} &= ((1-\theta_2)\mathcal{D}_g^H + \theta_2\mathcal{L}_g^H)\widehat{\psi}^{(k)} + \theta_2\widehat{f}_g \\ \widehat{\psi}^{(k+1)} &= \frac{1}{2}(\widehat{\psi}^{(\mathscr{F})} + \widehat{\psi}^{(\mathscr{B})}) \end{aligned}\right\} \quad k = 0, 1, \ldots \tag{3.70}$$

and

$$\left.\begin{aligned} (\mathcal{D}_g^Q - \theta_1\mathcal{L}_g^Q)\widehat{\psi}^{(\mathscr{F})} &= ((1-\theta_1)\mathcal{D}_g^Q + \theta_1\mathcal{U}_g^Q)\widehat{\psi}^{(k)} + \theta_1\widehat{f}_g \\ (\mathcal{D}_g^Q - \theta_2\mathcal{U}_g^Q)\widehat{\psi}^{(\mathscr{B})} &= ((1-\theta_2)\mathcal{D}_g^Q + \theta_2\mathcal{L}_g^Q)\widehat{\psi}^{(k)} + \theta_2\widehat{f}_g \\ \widehat{\psi}^{(k+1)} &= \frac{1}{2}(\widehat{\psi}^{(\mathscr{F})} + \widehat{\psi}^{(\mathscr{B})}) \end{aligned}\right\} \quad k = 0, 1, \ldots \tag{3.71}$$

with $(k = 0, 1, \ldots)$, $(\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)$, $(\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)$, $(\mathcal{D}_g^Q - \theta_1\mathcal{L}_g^Q)$, $(\mathcal{D}_g^Q - \theta_2\mathcal{U}_g^Q)$, $((1-\theta_1)\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)$, $((1-\theta_2)\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)$, $((1-\theta_1)\mathcal{D}_g^Q - \theta_1\mathcal{L}_g^Q)$ and $((1-\theta_2)\mathcal{D}_g^Q - \theta_2\mathcal{U}_g^Q)$ are nonsingular triangular matrices with optimal parameter values θ_1 and θ_2 and $\widehat{\psi}^{(k)}$ is initial vector at the kth iteration to the solutions of $\widehat{\psi}^{(\mathscr{F})}$ and $\widehat{\psi}^{(\mathscr{B})}$. In the case of $\theta_1 = \theta_2$, the HSMAM and QSMAM methods are equivalent to the HSAM and QSAM methods respectively.

Based on the formulations in equations (3.70) and (3.71), the iterative forms of the HSMAM and QSMAM methods are

$$\widehat{\psi}^{(k+1)} = \mathcal{Z}_{HSMAM}\widehat{\psi}^{(k)} + z_{HSMAM}\widehat{f}_g, \quad k = 0, 1, 2, \ldots \tag{3.72}$$

and

$$\widehat{\psi}^{(k+1)} = \mathcal{Z}_{QSMAM}\widehat{\psi}^{(k)} + z_{QSMAM}\widehat{f}_g, \quad k = 0, 1, 2, \ldots \tag{3.73}$$

where $\mathcal{Z}_{HSMAM}$ defined by

$$\frac{1}{2}\Big[(\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)^{-1}((1-\theta_1)\mathcal{D}_g^H + \theta_1\mathcal{U}_g^H) + (\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)^{-1}((1-\theta_2)\mathcal{D}_g^H + \theta_2\mathcal{L}_g^H)\Big]$$

and $\mathcal{Z}_{QSMAM}$ defined by

$$\frac{1}{2}\Big[(\mathcal{D}_g^Q - \theta_1\mathcal{L}_g^Q)^{-1}((1-\theta_1)\mathcal{D}_g^Q + \theta_1\mathcal{U}_g^Q) + (\mathcal{D}_g^Q - \theta_2\mathcal{U}_g^Q)^{-1}((1-\theta_2)\mathcal{D}_g^Q + \theta_2\mathcal{L}_g^Q)\Big]$$

respectively. The $\mathcal{Z}_{HSMAM}$ and $\mathcal{Z}_{QSMAM}$ are square iteration matrices for HSMAM and QSMAM methods, respectively. Meanwhile,

$$z_{HSMAM} = \frac{1}{2}\Big[\theta_1(\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)^{-1} + \theta_2(\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)\Big]$$

and

$$z_{QSMAM} = \frac{1}{2}\Big[\theta_1(\mathcal{D}_g^Q - \theta_1\mathcal{L}_g^Q)^{-1} + \theta_2(\mathcal{D}_g^Q - \theta_2\mathcal{U}_g^Q)\Big]$$

are load vectors obtained from the right hand side $\widehat{f}_g$.
The general conditions which guarantee the convergence of the HSMAM and the QSMAM iterative schemes in equations (3.72) and (3.73) for solving generalised linear systems (3.63) are described in the following Theorem and Proofs.

Theorem 3.4. *Let $\mathcal{A}_g^H$ and $\mathcal{A}_g^Q$ be nonsingular diagonally dominant matrices with order respectively $(\frac{N}{2}-1)\times(\frac{N}{2}-1)$ and $(\frac{N}{4}-1)\times(\frac{N}{4}-1)$ with its components $\sigma_{i,i} > 0,\ for i = 1, 2, \ldots N-1$.*

Let the matrix splitting for Half- and Quarter-Sweep cases as in equations (3.66) and (3.67) are defined by

$$\mathcal{A}_g^H = \mathcal{H}_1^H - \mathcal{K}_1^H = \mathcal{H}_2^H - \mathcal{K}_2^H$$

and

$$\mathcal{A}_g^Q = \mathcal{H}_1^Q - \mathcal{K}_1^Q = \mathcal{H}_2^Q - \mathcal{K}_2^Q.$$

where $\mathcal{H}_1^H$ *and* $\mathcal{H}_2^H$, $\mathcal{H}_1^Q$ *and* $\mathcal{H}_2^Q$ *are nonsingular respectively with* $\|(\mathcal{H}_1^H)^{-1}\| \geq 0$, $\|\mathcal{K}_1^H\| \geq 0$, $\|(\mathcal{H}_2^H)^{-1}\| \geq 0$, *and* $\|\mathcal{K}_2^H\| \geq 0$ $\|(\mathcal{H}_1^Q)^{-1}\| \geq 0$, $\|\mathcal{K}_1^Q\| \geq 0$, $\|(\mathcal{H}_2^Q)^{-1}\| \geq 0$*and* $\|\mathcal{K}_2^Q\| \geq 0$.

Therefore, the iterative schemes of HSMAM (3.72) and the QSMAM (3.73) are convergent for $0 < \theta_1 < 2$ *and* $0 < \theta_2 < 2$.

Proof. By hypothesis, Let $\mathcal{A}_g^H$ and $\mathcal{A}_g^Q$ be nonsingular matrices with order of $(\frac{N}{2}-1) \times (\frac{N}{2}-1)$ and $(\frac{N}{4}-1) \times (\frac{N}{4}-1)$ respectively.

Since $\mathcal{H}_1^H = \mathcal{D}_g^H - \theta_1\mathcal{U}_g^H$, $\mathcal{H}_2^H = \mathcal{D}_g^H - \theta_2\mathcal{L}_g^H$, $\mathcal{H}_1^Q = \mathcal{D}_g^Q - \theta_1\mathcal{U}_g^Q$ and $\mathcal{H}_2^Q = \mathcal{D}_g^Q - \theta_2\mathcal{L}_g^Q$ are strictly diagonally dominant matrices with positive entries on the diagonal for $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$.

The matrices $\mathcal{K}_1^H = (1-\theta_1)\mathcal{D}_g^H + \theta_1\mathcal{U}_g^H$, $\mathcal{K}_2^H = (1-\theta_2)\mathcal{D}_g^H + \theta_1\mathcal{L}_g^H$, $\mathcal{K}_1^Q = (1-\theta_1)\mathcal{D}_g^Q + \theta_1\mathcal{U}_g^Q$ and $\mathcal{K}_2^Q = (1-\theta_2)\mathcal{D}_g^Q + \theta_1\mathcal{L}_g^Q$ are triangular.

Since

$$\mathcal{H}_1^H - \mathcal{K}_1^H = \mathcal{H}_2^H - \mathcal{K}_2^H = \mathcal{A}_g^H$$

and

$$\mathcal{H}_1^Q - \mathcal{K}_1^Q = \mathcal{H}_2^Q - \mathcal{K}_2^Q = \mathcal{A}_g^Q$$

then it can be written as

$$\mathcal{Q}^H = \frac{1}{2}(\mathcal{H}_1^H)^{-1}\mathcal{K}_1^H + \frac{1}{2}(\mathcal{H}_2^H)^{-1}\mathcal{K}_2^H = \mathcal{I} - \left[\frac{1}{2}(\mathcal{H}_1^H)^{-1} + \frac{1}{2}(\mathcal{H}_2^H)^{-1}\right]\mathcal{A}_g \tag{3.74}$$

or also can be written as

$$\frac{1}{2}(\mathcal{H}_1^H)^{-1} + \frac{1}{2}(\mathcal{H}_2^H)^{-1} = (\mathcal{I} - \mathcal{Q}^H)\mathcal{A}_g^{-1} \tag{3.75}$$

and similarly,

$$\mathcal{Q}^Q = \frac{1}{2}(\mathcal{H}_1^Q)^{-1}\mathcal{K}_1^Q + \frac{1}{2}(\mathcal{H}_2^Q)^{-1}\mathcal{K}_2^Q = \mathcal{I} - \left[\frac{1}{2}(\mathcal{H}_1^Q)^{-1} + \frac{1}{2}(\mathcal{H}_2^Q)^{-1}\right]\mathcal{A}_g \tag{3.76}$$

or also can be written as

$$\frac{1}{2}(\mathcal{H}_1^Q)^{-1}\mathcal{K}_1^Q + \frac{1}{2}(\mathcal{H}_2^Q)^{-1}\mathcal{K}_2^Q = (\mathcal{I} - \mathcal{Q}^Q)\mathcal{A}_g^{-1}. \tag{3.77}$$

The proof of the theorem runs parallel to a standard proof given in [44]. Since $\mathcal{Q}^H = (\mathcal{H}_r^H)^{-1}\mathcal{K}_r^H$ and $\mathcal{Q}^Q = (\mathcal{H}_r^Q)^{-1}\mathcal{K}_r^Q$, hence the spectral radii are

$$\rho_{HSMAM}(\mathcal{Q}^H) < 1 \tag{3.78}$$

and

$$\rho_{QSMAM}(\mathcal{Q}^Q) < 1. \tag{3.79}$$

Therefore, the HSMAM and QSMAM iterative schemes in equations (3.72) and (3.73) are converge for any initial vector $\widehat{\psi}^{(0)}$ with the conditions $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$. Hence, Theorem 3.4 is proved. □

3.10.2 Half- and Quarter-Sweep Modified Geometric Mean Iterative Methods

The formulation of the Half-Sweep Modified Geometric Mean (HSMGM) and the Quarter-Sweep Modified Geometric Mean (QSMGM) methods are presented together with conditions of convergence. The HSMGM and QSMGM iterative methods for approximating the solution $\widehat{\psi}^{(k+1)}$ of generalised linear system (3.63) are as follows:

$$\left.\begin{aligned} (\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)\widehat{\psi}^{(\mathscr{F})} &= ((1-\theta_1)\mathcal{D}_g^H + \theta_1\mathcal{U}_g^H)\widehat{\psi}^{(k)} + \theta_1\widehat{f}_g \\ (\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)\widehat{\psi}^{(\mathscr{B})} &= ((1-\theta_2)\mathcal{D}_g^H + \theta_2\mathcal{L}_g^H)\widehat{\psi}^{(k)} + \theta_2\widehat{f}_g \\ \widehat{\psi}^{(k+1)} &= (\widehat{\psi}^{(\mathscr{F})} \circ \widehat{\psi}^{(\mathscr{B})})^{\frac{1}{2}}, \quad \widehat{\psi}^{(\mathscr{F})} \circ \widehat{\psi}^{(\mathscr{B})} \geq 0 \end{aligned}\right\} \quad k = 0, 1, \ldots \tag{3.80}$$

and

$$\left.\begin{aligned} (\mathcal{D}_g^Q - \theta_1\mathcal{L}_g^Q)\widehat{\psi}^{(\mathscr{F})} &= ((1-\theta_1)\mathcal{D}_g^Q + \theta_1\mathcal{U}_g^Q)\widehat{\psi}^{(k)} + \theta_1\widehat{f}_g \\ (\mathcal{D}_g^Q - \theta_2\mathcal{U}_g^Q)\widehat{\psi}^{(\mathscr{B})} &= ((1-\theta_2)\mathcal{D}_g^Q + \theta_2\mathcal{L}_g^Q)\widehat{\psi}^{(k)} + \theta_2\widehat{f}_g \\ \widehat{\psi}^{(k+1)} &= (\widehat{\psi}^{(\mathscr{F})} \circ \widehat{\psi}^{(\mathscr{B})})^{\frac{1}{2}}, \quad \widehat{\psi}^{(\mathscr{F})} \circ \widehat{\psi}^{(\mathscr{B})} \geq 0 \end{aligned}\right\} \quad k = 0, 1, \ldots \tag{3.81}$$

Meanwhile, θ_1, θ_2, $\circ$ and $(\circ)^{\frac{1}{2}}$ and $\widehat{\psi}^{(k)}$ are optimal HSMGM and QSMGM parameters, Hadamard product, Hadamard power [21] and unknown vector at the kth iteration, respectively. In the case of $\theta_1 = \theta_2$, these methods are equivalent to the conventional HSGM and QSGM iterative methods, respectively.

Based on the formulae (3.80) and (3.81), the iterative forms of the HSMGM and the QSMGM methods are as follows:

$$\widehat{\psi}^{(k+1)} = \mathcal{Z}_{HSMGM}\widehat{\psi}^{(k)} + z_{HSMGM}\widehat{f}_g, \quad k = 0, 1, 2, \ldots \tag{3.82}$$

and

$$\widehat{\psi}^{(k+1)} = \mathcal{Z}_{QSMGM}\widehat{\psi}^{(k)} + z_{QSMGM}\widehat{f}_g, \quad k = 0, 1, 2, \ldots \tag{3.83}$$

where $\mathcal{Z}_{HSMGM}$ is defined by

$$\left[(\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)^{-1}((1-\theta_1)\mathcal{D}_g^H + \theta_1\mathcal{U}_g^H)(\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)^{-1}((1-\theta_2)\mathcal{D}_g^H + \theta_1\mathcal{L}_g^H)\right]^{\frac{1}{2}}$$

and $\mathcal{Z}_{QSMGM}$ is

$$\left[(\mathcal{D}_g^Q - \theta_1\mathcal{L}_g^Q)^{-1}((1-\theta_1)\mathcal{D}_g^Q + \theta_1\mathcal{U}_g^Q)(\mathcal{D}_g^Q - \theta_2\mathcal{U}_g^Q)^{-1}((1-\theta_2)\mathcal{D}_g^Q + \theta_1\mathcal{L}_g^Q)\right]^{\frac{1}{2}}$$

which are both square iteration matrices for HSMGM and QSMGM methods respectively. Meanwhile,

$$z_{HSMGM} = \left[\theta_1\theta_2(\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)^{-1}(\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)^{-1}\right]^{\frac{1}{2}}$$

and

$$z_{QSMGM} = \left[\theta_1\theta_2(\mathcal{D}_g^Q - \theta_1\mathcal{L}_g^Q)^{-1}(\mathcal{D}_g^Q - \theta_2\mathcal{U}_g^Q)^{-1}\right]^{\frac{1}{2}}$$

are load vectors that obtained from the right hand side $\widehat{f}_g$.
The general conditions which guarantee the convergence of the HSMGM and the QSMGM methods for solving generalised linear systems (3.63) are described in the following Theorem.

Theorem 3.5. *Let $\mathcal{Z}_{HSMGM}$ and $\mathcal{Z}_{QSMGM}$ be $(\frac{N}{2}-1)\times(\frac{N}{2}-1)$ and $(\frac{N}{4}-1)\times(\frac{N}{4}-1)$ nonsingular square matrices, respectively. Then, the necessary conditions for the HSMGM and QSMGM methods to be convergent are that $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$.*

Proof. Since the eigenvalues λ_j of $\mathcal{Z}_H$ and $\mathcal{Z}_Q$ are the zeroes of the characteristic polynomial, the determinants of $\mathcal{Z}_{HSMGM}$ and $\mathcal{Z}_{QSMGM}$ satisfy the following relations

$$det(\mathcal{Z}_{HSMGM}) = \prod_{j=0,2,4}^{N} \lambda_j \tag{3.84}$$

and

$$det(\mathcal{Z}_{QSMGM}) = \prod_{j=0,4,8}^{N} \lambda_j, \tag{3.85}$$

respectively where multiple eigenvalues are repeated according to their algebraic multiplicity. Since $(\mathcal{D}_g^H - \theta_1 \mathcal{L}_g^H)^{\frac{1}{2}}$, $(\mathcal{D}_g^H - \theta_2 \mathcal{U}_g^H)^{\frac{1}{2}}$, $((1-\theta_1)\mathcal{D}_g^H - \theta_1 \mathcal{U}_g^H)^{\frac{1}{2}}$, $((1-\theta_2)\mathcal{D}_g^H - \theta_2 \mathcal{L}_g^H)^{\frac{1}{2}}$ and $(\mathcal{D}_g^Q - \theta_1 \mathcal{L}_g^Q)^{\frac{1}{2}}$, $(\mathcal{D}_g^Q - \theta_2 \mathcal{U}_g^Q)^{\frac{1}{2}}$, $((1-\theta_1)\mathcal{D}_g^Q - \theta_1 \mathcal{U}_g^Q)^{\frac{1}{2}}$, $((1-\theta_2)\mathcal{D}_g^Q - \theta_2 \mathcal{L}_g^Q)^{\frac{1}{2}}$ are nonsingular triangular matrices, hence,
$\mathcal{Z}_{HSMGM}$

$$= \det\left[(\mathcal{D}_g^H - \theta_1 \mathcal{L}_g^H)^{-1}((1-\theta_1)\mathcal{D}_g^H + \theta_1 \mathcal{U}_g^H)(\mathcal{D}_g^H - \theta_2 \mathcal{U}_g^H)^{-1}((1-\theta_2)\mathcal{D}_g^H + \theta_2 \mathcal{L}_g^H)\right]^{\frac{1}{2}} \tag{3.86}$$

and $\mathcal{Z}_{QSMGM}$

$$= \det\left[(\mathcal{D}_g^Q - \theta_1 \mathcal{L}_g^Q)^{-1}((1-\theta_1)\mathcal{D}_g^Q + \theta_1 \mathcal{U}_g^Q)(\mathcal{D}_g^Q - \theta_2 \mathcal{U}_g^Q)^{-1}((1-\theta_2)\mathcal{D}_g^Q + \theta_2 \mathcal{L}_g^Q)\right]^{\frac{1}{2}}, \tag{3.87}$$

respectively.

Because of $\mathcal{L}_g^H$ and $\mathcal{U}_g^H$ are strictly lower and upper triangular matrices, respectively, then $\det(\mathcal{D}_g^H - \theta_1 \mathcal{L}_g^H)^{-1} = \det(\mathcal{D}_g^H - \theta_2 \mathcal{U}_g^H)^{-1} = \det(\mathcal{D}_g^H)^{-1}$.

Therefore,
$\det(\mathcal{Z}_{HSMGM})$=$\det\left[(\mathcal{D}_g^H)^{-1}((1-\theta_1)\mathcal{D}_g^H + \theta_1 \mathcal{U}_g^H)(\mathcal{D}_g^H)^{-1}((1-\theta_2)\mathcal{D}_g^H + \theta_2 \mathcal{L}_g^H)\right]^{\frac{1}{2}}$
$=\det\{(\mathcal{D}_g^H)^{-1}\}^{\frac{1}{2}} \det\{(1-\theta_1)\mathcal{D}_g^H + \theta_1 \mathcal{U}_g^H\}^{\frac{1}{2}} \det\{(\mathcal{D}_g^H)^{-1}\}^{\frac{1}{2}} \det\{(1-\theta_2)\mathcal{D}_g^H + \theta_2 \mathcal{L}_g^H\}^{\frac{1}{2}}$
$=\det\{(1-\theta_1)\mathcal{I} + \theta_1(\mathcal{D}_g^H)^{-1}\mathcal{U}_g^H\}^{\frac{1}{2}} \det\{(1-\theta_2)\mathcal{I} + \theta_2(\mathcal{D}_g^H)^{-1}\mathcal{L}_g^H\}^{\frac{1}{2}}$
$=\left((1-\theta_1)^{\frac{1}{2}}\right)^{\frac{N}{2}-1}\left((1-\theta_2)^{\frac{1}{2}}\right)^{\frac{N}{2}-1}$
$=(1-\theta_1)^{\frac{N-2}{4}}(1-\theta_2)^{\frac{N-2}{4}}$.

This now implies

$$\rho(\mathcal{Z}_{HSMGM}) \geq \left(\mid 1-\theta_1 \mid^{\frac{N-2}{4}} \mid 1-\theta_2 \mid^{\frac{N-2}{4}}\right)^{\frac{4}{N-2}} = \mid (1-\theta_1)(1-\theta_2) \mid . \tag{3.88}$$

Similarly, because of $\mathcal{L}_g^Q$ and $\mathcal{U}_g^Q$ are strictly lower and upper triangular matrices, respectively, then $\det(\mathcal{D}_g^Q - \theta_1 \mathcal{L}_g^Q)^{-1} = \det(\mathcal{D}_g^Q - \theta_2 \mathcal{U}_g^Q)^{-1} = \det(\mathcal{D}_g^Q)^{-1}$.

Therefore,
$\det(\mathcal{Z}_{QSMGM})$=$\det\left[(\mathcal{D}_g^Q)^{-1}((1-\theta_1)\mathcal{D}_g^Q + \theta_1 \mathcal{U}_g^Q)(\mathcal{D}_g^Q)^{-1}((1-\theta_2)\mathcal{D}_g^Q + \theta_2 \mathcal{L}_g^Q)\right]^{\frac{1}{2}}$
$=\det\{(\mathcal{D}_g^Q)^{-1}\}^{\frac{1}{2}} \det\{(1-\theta_1)\mathcal{D}_g^Q + \theta_1 \mathcal{U}_g^Q\}^{\frac{1}{2}} \det\{(\mathcal{D}_g^Q)^{-1}\}^{\frac{1}{2}} det\{(1-\theta_2)\mathcal{D}_g^Q + \theta_2 \mathcal{L}_g^Q\}^{\frac{1}{2}}$
$=\det\{(1-\theta_1)\mathcal{I} + \theta_1(\mathcal{D}_g^Q)^{-1}\mathcal{U}_g^Q\}^{\frac{1}{2}} \det\{(1-\theta_2)\mathcal{I} + \theta_2(\mathcal{D}_g^Q)^{-1}\mathcal{L}_g^Q\}^{\frac{1}{2}}$

$$
\begin{aligned}
&=\left((1-\theta_1)^{\frac{1}{2}}\right)^{\frac{N}{4}-1}\left((1-\theta_2)^{\frac{1}{2}}\right)^{\frac{N}{4}-1}\\
&=(1-\theta_1)^{\frac{N-4}{8}}(1-\theta_2)^{\frac{N-4}{8}}.
\end{aligned}
$$

This now implies

$$
\rho(\mathcal{Z}_{QSMGM}) \geq \left(\mid 1-\theta_1 \mid^{\frac{N-4}{8}} \mid 1-\theta_2 \mid^{\frac{N-4}{8}} \right)^{\frac{8}{N-4}} = \mid (1-\theta_1)(1-\theta_2) \mid . \quad (3.89)
$$

Therefore, based on Theorem 2.6 and 3.5, the HSMGM and QSMGM iteration schemes in equations (3.82) and (3.83) are converge for any initial vector $\widehat{\psi}^{(0)}$ with conditions $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$. Hence, Theorem 3.5 is proved. □

Definition 3.10.1. *For the ith element of independent vectors $\widehat{\psi}^{(\mathscr{F})}$ and $\widehat{\psi}^{(\mathscr{B})}$ the unknown vector $\widehat{\psi}^{(k+1)}$ for $i = q, 2q \ldots, N-q$ can be determined as follows:*

(i) $(\widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})})^{\frac{1}{2}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} > 0 \quad \& \quad \widehat{\psi}_i^{(\mathscr{B})} > 0 \quad (Case\ 1),$

(ii) $(\widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})})^{\frac{1}{2}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} < 0 \quad \& \quad \widehat{\psi}_i^{(\mathscr{B})} < 0 \quad (Case\ 2),$

(iii) $(\widehat{\psi}_i^{(\mathscr{F})} - \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid^{\frac{1}{2}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} > 0 \quad \& \quad \widehat{\psi}_i^{(\mathscr{B})} < 0 \quad (Case\ 3),$

(iv) $(\widehat{\psi}_i^{(\mathscr{B})} - \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid^{\frac{1}{2}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} < 0 \quad \& \quad \widehat{\psi}_i^{(\mathscr{B})} > 0 \quad (Case\ 4).$

3.10.3 Half- and Quarter-Sweep Modified Harmonic Mean Iterative Methods

Similar to FSMHM in Section 3.6, the formulation of the Half-Sweep Modified Harmonic Mean (HSMHM) and the Quarter-Sweep Modified Arithmetic Mean (QSMHM) methods are presented together with the systematic conditions for convergence. The HSMHM and QSMHM methods for approximating the solution $\widehat{\psi}^{(k+1)}$ of generalised linear system (3.63) are respectively

$$
\left.
\begin{aligned}
(\mathcal{D}_g^H - \theta_1\mathcal{L}_g^H)\widehat{\psi}^{(\mathscr{F})} &= ((1-\theta_1)\mathcal{D}_g^H + \theta_1\mathcal{U}_g^H)\widehat{\psi}^{(k)} + \theta_1\widehat{f}_g\\
(\mathcal{D}_g^H - \theta_2\mathcal{U}_g^H)\widehat{\psi}^{(\mathscr{B})} &= ((1-\theta_2)\mathcal{D}_g^H + \theta_2\mathcal{L}_g^H)\widehat{\psi}^{(k)} + \theta_2\widehat{f}_g\\
\widehat{\psi}^{(k+1)} &= \frac{2\widehat{\psi}^{\mathscr{F}} \circ \widehat{\psi}^{\mathscr{B}}}{\widehat{\psi}^{\mathscr{F}} + \widehat{\psi}^{\mathscr{B}}}, \quad \widehat{\psi}^{\mathscr{F}} + \widehat{\psi}^{\mathscr{B}} \neq 0
\end{aligned}
\right\} \quad k = 0, 1, \ldots \quad (3.90)
$$

and

$$\left.\begin{aligned}(\mathcal{D}_g^Q-\theta_1\mathcal{L}_g^Q)\widehat{\psi}^{(\mathscr{F})}&=((1-\theta_1)\mathcal{D}_g^Q+\theta_1\mathcal{U}_g^Q)\widehat{\psi}^{(k)}+\theta_1\widehat{f}_g\\(\mathcal{D}_g^Q-\theta_2\mathcal{U}_g^Q)\widehat{\psi}^{(\mathscr{B})}&=((1-\theta_2)\mathcal{D}_g^Q+\theta_2\mathcal{L}_g^Q)\widehat{\psi}^{(k)}+\theta_2\widehat{f}_g\\\widehat{\psi}^{(k+1)}&=\frac{2\widehat{\psi}^{\mathscr{F}}\circ\widehat{\psi}^{\mathscr{B}}}{\widehat{\psi}^{\mathscr{F}}+\widehat{\psi}^{\mathscr{B}}},\quad\widehat{\psi}^{\mathscr{F}}+\widehat{\psi}^{\mathscr{B}}\neq 0\end{aligned}\right\}\quad k=0,1,\ldots \tag{3.91}$$

where $(\mathcal{D}_g^H-\theta_1\mathcal{L}_g^H)$, $(\mathcal{D}_g^H-\theta_2\mathcal{U}_g^H)$, $(\mathcal{D}_g^Q-\theta_1\mathcal{L}_g^Q)$ $(\mathcal{D}_g^Q-\theta_2\mathcal{U}_g^Q)$, $((1-\theta_1)\mathcal{D}_g^H-\theta_1\mathcal{L}_g^H)$, $((1-\theta_2)\mathcal{D}_g^H-\theta_2\mathcal{U}_g^H)$, $((1-\theta_1)\mathcal{D}_g^Q-\theta_1\mathcal{L}_g^Q)$ and $((1-\theta_2)\mathcal{D}_g^Q-\theta_2\mathcal{U}_g^Q)$ are nonsingular triangular matrices with optimal parameter values θ_1 and θ_2, $\widehat{\psi}^{(k)}$ is an unknown vector at the kth iteration, '$\circ$' is the Hadamard product operator and $(\circ)^{\frac{1}{2}}$ is the Hadamard power [21]. In the case of $\theta_1=\theta_2$, the HSMHM and QSMHM method are equivalent to the HSHM and QSHM methods, respectively.

From equations (3.90) and (3.91), the iterative forms of the HSMHM and the QSMHM methods are as follows:

$$\widehat{\psi}^{(k+1)}=\mathcal{Z}_{HSMHM}\widehat{\psi}^{(k)}+z_{HSMHM}\widehat{f}_g,\quad k=0,1,2,\ldots \tag{3.92}$$

and

$$\widehat{\psi}^{(k+1)}=\mathcal{Z}_{QSMHM}\widehat{\psi}^{(k)}+z_{QSMHM}\widehat{f}_g,\quad k=0,1,2,\ldots \tag{3.93}$$

where $\mathcal{Z}_{HSMHM}$ is defined as

$$\frac{2(\mathcal{D}_g^H-\theta_1\mathcal{L}_g^H)^{-1}((1-\theta_1)\mathcal{D}_g^H+\theta_1\mathcal{U}_g^H)(\mathcal{D}_g^H-\theta_2\mathcal{U}_g^H)^{-1}((1-\theta_2)\mathcal{D}_g^H+\theta_2\mathcal{L}_g^H)}{(\mathcal{D}_g^H-\theta_1\mathcal{L}_g^H)^{-1}((1-\theta_1)\mathcal{D}_g^H+\theta_1\mathcal{U}_g^H)+(\mathcal{D}_g^H-\theta_2\mathcal{U}_g^H)^{-1}((1-\theta_2)\mathcal{D}_g^H+\theta_2\mathcal{L}_g^H)}$$

and $\mathcal{Z}_{QSMHM}$ is

$$\frac{2(\mathcal{D}_g^Q-\theta_1\mathcal{L}_g^Q)^{-1}((1-\theta_1)\mathcal{D}_g^Q+\theta_1\mathcal{U}_g^Q)(\mathcal{D}_g^Q-\theta_2\mathcal{U}_g^Q)^{-1}((1-\theta_2)\mathcal{D}_g^Q+\theta_2\mathcal{L}_g^Q)}{(\mathcal{D}_g^Q-\theta_1\mathcal{L}_g^Q)^{-1}((1-\theta_1)\mathcal{D}_g^Q+\theta_1\mathcal{U}_g^Q)+(\mathcal{D}_g^Q-\theta_2\mathcal{U}_g^Q)^{-1}((1-\theta_2)\mathcal{D}_g^Q+\theta_2\mathcal{L}_g^Q)}$$

which are both square iteration matrices for the HSMHM and QSMHM methods respectively. Meanwhile,

$$z_{HSMHM}=\frac{2\theta_1\theta_2(\mathcal{D}_g^H-\theta_1\mathcal{L}_g^H)^{-1}(\mathcal{D}_g^H-\theta_2\mathcal{U}_g^H)^{-1}}{(\mathcal{D}_g^H-\theta_1\mathcal{L}_g^H)^{-1}+(\mathcal{D}_g^H-\theta_2\mathcal{U}_g^H)^{-1}}$$

and

$$z_{QSMHM}=\frac{2\theta_1\theta_2(\mathcal{D}_g^Q-\theta_1\mathcal{L}_g^Q)^{-1}(\mathcal{D}_g^Q-\theta_2\mathcal{U}_g^Q)^{-1}}{(\mathcal{D}_g^Q-\theta_1\mathcal{L}_g^Q)^{-1}+(\mathcal{D}_g^Q-\theta_2\mathcal{U}_g^Q)^{-1}}$$

are load vectors that are obtained from the right hand side $\widehat{f}_g$.

Based on the matrix splitting of $\mathcal{A}_g$ in equations(3.66) and (3.67), the conditions which the convergence of the HSMHM and QSMHM methods are described in the following Theorems and Proofs.

Theorem 3.6. *Let $\mathcal{A}_g^H$ and $\mathcal{A}_g^Q$ be nonsingular diagonally dominant matrices with order respectively $(\frac{N}{2}-1)\times(\frac{N}{2}-1)$ and $(\frac{N}{4}-1)\times(\frac{N}{4}-1)$ with its components $\sigma_{i,i}>0,\ for i=1,2,\dots N-1$.*

Let the matrix splitting for Half- and Quarter-Sweep cases as in equations (3.66) and (3.67) are defined by

$$\mathcal{A}_g^H = \mathcal{H}_1^H - \mathcal{K}_1^H = \mathcal{H}_2^H - \mathcal{K}_2^H$$

and

$$\mathcal{A}_g^Q = \mathcal{H}_1^Q - \mathcal{K}_1^Q = \mathcal{H}_2^Q - \mathcal{K}_2^Q.$$

where $\mathcal{H}_1^H$ and $\mathcal{H}_2^H$, $\mathcal{H}_1^Q$ and $\mathcal{H}_2^Q$ are nonsingular respectively with $\|(\mathcal{H}_1^H)^{-1}\|\geq 0$, $\|\mathcal{K}_1^H\|\geq 0$, $\|(\mathcal{H}_2^H)^{-1}\|\geq 0$, and $\|\mathcal{K}_2^H\|\geq 0$, $\|(\mathcal{H}_1^Q)^{-1}\|\geq 0$, $\|\mathcal{K}_1^Q\|\geq 0$, $\|(\mathcal{H}_2^Q)^{-1}\|\geq 0$, and $\|\mathcal{K}_2^Q\|\geq 0$.

Therefore, the iterative schemes of HSMHM (3.92) and the QSMHM (3.93) are convergent for $0<\theta_1<2$ and $0<\theta_2<2$.

Proof. By hypothesis, Let $\mathcal{A}_g^H$ and $\mathcal{A}_g^Q$ be nonsingular matrices having order of $(\frac{N}{2}-1)\times(\frac{N}{2}-1)$ and $(\frac{N}{4}-1)\times(\frac{N}{4}-1)$ respectively.

Since $\mathcal{H}_1^H=\mathcal{D}_g^H-\theta_1\mathcal{U}_g^H$, $\mathcal{H}_2^H=\mathcal{D}_g^H-\theta_2\mathcal{L}_g^H$ and $\mathcal{H}_1^Q=\mathcal{D}_g^Q-\theta_1\mathcal{U}_g^Q$, $\mathcal{H}_2^Q=\mathcal{D}_g^Q-\theta_2\mathcal{L}_g^Q$ are strictly diagonally dominant matrices with positive entries on the diagonal for $0<\theta_1<2$ and $0<\theta_2<2$.

The nonsingular matrices $\mathcal{K}_1^H=(1-\theta_1)\mathcal{D}^H+\theta_1\mathcal{U}^H$, $\mathcal{K}_2^H=(1-\theta_2)\mathcal{D}^H+\theta_1\mathcal{L}^H$, $\mathcal{K}_1^Q=(1-\theta_1)\mathcal{D}^Q+\theta_1\mathcal{U}^Q$ and $\mathcal{K}_2^Q=(1-\theta_2)\mathcal{D}^Q+\theta_1\mathcal{L}^Q$ are triangular.

Since

$$\mathcal{H}_1^H - \mathcal{K}_1^H = \mathcal{H}_2^H - \mathcal{K}_2^H = \mathcal{A}_g^H$$

and

$$\mathcal{H}_1^Q - \mathcal{K}_1^Q = \mathcal{H}_2^Q - \mathcal{K}_2^Q = \mathcal{A}_g^Q$$

then it can be written as

$$\mathcal{Q}^H = \frac{2[(\mathcal{H}_1^H)^{-1}\mathcal{K}_1^H \circ (\mathcal{H}_2^H)^{-1}\mathcal{K}_2^H]}{(\mathcal{H}_1^H)^{-1}\mathcal{K}_1^H + (\mathcal{H}_2^H)^{-1}\mathcal{K}_2^H} = \mathcal{I} - \left[\frac{2[(\mathcal{H}_1^H)^{-1} \circ (\mathcal{H}_2^H)^{-1}]}{(\mathcal{H}_1^H)^{-1} + (\mathcal{H}_2^H)^{-1}}\right]\mathcal{A}_g^H \tag{3.94}$$

or also can be written as

$$\frac{2[(\mathcal{H}_1^H)^{-1} \circ (\mathcal{H}_2^H)^{-1}]}{(\mathcal{H}_1^H)^{-1} + (\mathcal{H}_2^H)^{-1}} = (\mathcal{I} - \mathcal{Q}^H)(\mathcal{A}_g^H)^{-1}. \tag{3.95}$$

Similarly,

$$\mathcal{Q}^Q = \frac{2[(\mathcal{H}_1^Q)^{-1}\mathcal{K}_1^Q \circ (\mathcal{H}_2^Q)^{-1}\mathcal{K}_2^Q]}{(\mathcal{H}_1^Q)^{-1}\mathcal{K}_1^Q + (\mathcal{H}_2^Q)^{-1}\mathcal{K}_2^Q} = \mathcal{I} - \left[\frac{2[(\mathcal{H}_1^Q)^{-1} \circ (\mathcal{H}_2^Q)^{-1}]}{(\mathcal{H}_1^Q)^{-1} + (\mathcal{H}_2^Q)^{-1}}\right]\mathcal{A}_g^Q \tag{3.96}$$

or also can be written as

$$\frac{2[(\mathcal{H}_1^Q)^{-1} \circ (\mathcal{H}_2^Q)^{-1}]}{(\mathcal{H}_1^Q)^{-1} + (\mathcal{H}_2^Q)^{-1}} = (\mathcal{I} - \mathcal{Q}^Q)(\mathcal{A}_g^Q)^{-1}. \tag{3.97}$$

The proof of the theorem runs parallel to a standard proof given in [44]. Since $\mathcal{Q}^H = (\mathcal{H}_r^H)^{-1}\mathcal{K}_r^H$ and $\mathcal{Q}^Q = (\mathcal{H}_r^Q)^{-1}\mathcal{K}_r^Q$, hence, the spectral radii are

$$\rho_{HSMHM}(\mathcal{Q}^H) < 1 \tag{3.98}$$

and

$$\rho_{QSMHM}(\mathcal{Q}^Q) < 1. \tag{3.99}$$

Therefore, the HSMHM and QSMHM iterative schemes are converge for any initial vector $\widehat{\psi}^{(0)}$ with the conditions $0 < \theta_1 < 2$ and $0 < \theta_2 < 2$. Hence, Theorem 3.6 is proved. □

Definition 3.10.2. *For the ith element of independent vectors $\widehat{\psi}_i^{(\mathscr{F})}$ and $\widehat{\psi}_i^{\mathscr{B}}$, the unknown vector $\widehat{\psi}_i^{(k+1)}$ for $i = q, 2q, \ldots, N-q$ can be determined as follows:*

(i) $0,$ $\quad if \quad \widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})} = 0 \qquad (Case\ 1),$

(ii) $\dfrac{2\widehat{\psi}_i^{(\mathscr{F})} \circ \widehat{\psi}_i^{(\mathscr{B})}}{\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})}}, \quad if \quad \widehat{\psi}_i^{(\mathscr{F})} > 0 \ \&\ \widehat{\psi}_i^{\mathscr{B}} > 0 \qquad (Case\ 2),$

(iii) $\dfrac{2\widehat{\psi}_i^{(\mathscr{F})} \circ \widehat{\psi}_i^{(\mathscr{B})}}{\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})}}, \quad if\ \widehat{\psi}_i^{(\mathscr{F})} < 0 \ \&\ \widehat{\psi}_i^{(\mathscr{B})} < 0 \qquad (Case\ 3),$

(iv) $\dfrac{2[(\widehat{\psi}_i^{(\mathscr{F})})^2 - 2\widehat{\psi}_i^{(\mathscr{F})}(\mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid)^{\frac{1}{2}} + \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid]}{\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})}},$
$if \quad \widehat{\psi}_i^{(\mathscr{F})} > 0 \ \&\ \widehat{\psi}_i^{(\mathscr{B})} < 0 \qquad (Case\ 4),$

(v) $\dfrac{2[(\widehat{\psi}_i^{(\mathscr{B})})^2 - 2\widehat{\psi}_i^{(\mathscr{B})}(\mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid)^{\frac{1}{2}} + \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid]}{\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})}}$,
if $\widehat{\psi}_i^{(\mathscr{F})} < 0$ & $\widehat{\psi}_i^{\mathscr{B}} > 0$ $\qquad (Case\ 5).$

for Case 2 *to Case* 5, $\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})} \neq 0$ *hold.*

3.11 Algorithms

The algorithms for the family of MWM methods for solving the generalised linear system (3.63) as described in Section 3.9.2 arisen from problem (1.1) is described in the following part. The algorithm is explicitly performed by $\widehat{\psi}_i^{(\mathscr{F})}$ and $\widehat{\psi}_i^{(\mathscr{B})}$ alternately until the convergence criterion is satisfied, i.e. the maximum norm $\parallel \widehat{\psi}^{(k+1)} - \widehat{\psi}^{(k)} \parallel_\infty \leq \epsilon$ where ϵ is the convergence criterion. Therefore the Full-, Half- and Quarter-Sweep iterative methods of MAM, MGM and MHM are shown in the Algorithms 3.11.1, 3.11.2 and 3.11.3 correspondingly associated with the generated CD-CCNC systems.

3.11.1 Standard or Full-, Half- and Quarter-Sweep MAM Algorithms

Step i. Set $\widehat{\psi}^{(\mathscr{F})} = \widehat{\psi}^{(\mathscr{B})} = 0$ and MaxK=10000

Step ii. Iteration cycle

for $k = 0, 1, 2, \ldots,$MaxK

for $i = q, 2q, \ldots, N-q$; $q = 1, 2, 4$

Compute

$\widehat{\psi}_i^{(\mathscr{F})} \leftarrow (1-\theta_1)\widehat{\psi}_i^{(k)} + \frac{\theta_1}{\sigma_{i,i}}\left[\widehat{f}_i - \sum_{j=q,2q}^{i-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(\mathscr{F})} - \sum_{j=i+q,i+2q,i+3q}^{N-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(k)}\right]$

for $i = N-q, N-2q, \ldots, q$

Compute

$\widehat{\psi}_i^{(\mathscr{B})} \leftarrow (1-\theta_2)\widehat{\psi}_i^{(k)} + \frac{\theta_2}{\sigma_{i,i}}\left[\widehat{f}_i - \sum_{j=q,2q}^{i-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{k} - \sum_{j=i+q,i+2q,i+3q}^{N-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(\mathscr{B})}\right]$

for $i = q, 2q, \cdots, N-q$

Compute

$$\widehat{\psi}_i^{(k+1)} \leftarrow \left\{\tfrac{1}{2}(\widehat{\psi}_i^{(\mathscr{F})} + \widehat{\psi}_i^{(\mathscr{B})})\right.$$

Step iii. Check the convergence. If convergence is achieved, go to Step (iv); otherwise, repeat the iteration cycle (i.e., go to Step (ii))

Step iv. Stop

3.11.2 Standard or Full-, Half- and Quarter-Sweep MGM Algorithms

Step i. Set $\widehat{\psi}^{(\mathscr{F})} = \widehat{\psi}^{(\mathscr{B})} = 0$ and MaxK=10000

Step ii. Iteration cycle

for $k = 0, 1, 2, \ldots,$MaxK

for $i = q, 2q, \ldots, N-q\ ;\ q = 1, 2, 4$

Compute

$\widehat{\psi}_i^{(\mathscr{F})} \leftarrow (1-\theta_1)\widehat{\psi}_i^{(k)} + \frac{\theta_1}{\sigma_{i,i}}\left[\widehat{f}_i - \sum_{j=q,2q}^{i-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(\mathscr{F})} - \sum_{j=i+q,i+2q,i+3q}^{N-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(k)}\right]$

for $i = N-q, N-2q, \ldots, q$

Compute

$\widehat{\psi}_i^{(\mathscr{B})} \leftarrow (1-\theta_2)\widehat{\psi}_i^{(k)} + \frac{\theta_2}{\sigma_{i,i}}\left[\widehat{f}_i - \sum_{j=q,2q}^{i-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{k} - \sum_{j=i+q,i+2q,i+3q}^{N-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(\mathscr{B})}\right]$

for $i = q, 2q, \cdots, N-q$

Compute

$$\widehat{\psi}_i^{(k+1)} \leftarrow \begin{cases} (\widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})})^{\frac{1}{2}} & (Case\ \ 1) \\ \\ -(\widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})})^{\frac{1}{2}} & (Case\ \ 2) \\ \\ (\widehat{\psi}_i^{(\mathscr{F})} - \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid^{\frac{1}{2}} & (Case\ \ 3) \\ \\ (\widehat{\psi}_i^{(\mathscr{B})} - \mid \widehat{\psi}_i^{(\mathscr{F})}\widehat{\psi}_i^{(\mathscr{B})} \mid^{\frac{1}{2}} & (Case\ \ 4) \end{cases}$$

Step iii. Check the convergence. If convergence is achieved, go to Step (iv); otherwise, repeat the iteration cycle (i.e., go to Step (ii))

Step iv. Stop

3.11.3 Standard or Full-, Half- and Quarter-Sweep MHM Algorithms

Step i. Set $\widehat{\psi}^{(\mathscr{F})} = \widehat{\psi}^{(\mathscr{B})} = 0$ and MaxK=10000

Step ii. Iteration cycle

for $k = 0, 1, 2, \ldots,$MaxK

for $i = q, 2q, \ldots, N-q\ ;\ q = 1, 2, 4$

Compute

$\widehat{\psi}_i^{(\mathscr{F})} \leftarrow (1-\theta_1)\widehat{\psi}_i^{(k)} + \frac{\theta_1}{\sigma_{i,i}}\left[\widehat{f}_i - \sum_{j=q,2q}^{i-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(\mathscr{F})} - \sum_{j=i+q,i+2q,i+3q}^{N-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(k)}\right]$

for $i = N-q, N-2q, \ldots, q$

Compute

$\widehat{\psi}_i^{(\mathscr{B})} \leftarrow (1-\theta_2)\widehat{\psi}_i^{(k)} + \frac{\theta_2}{\sigma_{i,i}}\left[\widehat{f}_i - \sum_{j=q,2q}^{i-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{k} - \sum_{j=i+q,i+2q,i+3q}^{N-q} \mathcal{A}_{i,j}\widehat{\psi}_j^{(\mathscr{B})}\right]$

for $i = q, 2q, \cdots, N-q$

Compute

$$\widehat{\psi}_i^{(k+1)} \leftarrow \begin{cases} 0, & (Case\ \ 1) \\[2ex] \dfrac{2\widehat{\psi}_i^{\mathscr{F}} \circ \widehat{\psi}_i^{\mathscr{B}}}{\widehat{\psi}_i^{\mathscr{F}} + \widehat{\psi}_i^{\mathscr{B}}} & (Case\ \ 2) \\[2ex] \dfrac{2\widehat{\psi}_i^{\mathscr{F}} \circ \widehat{\psi}_i^{\mathscr{B}}}{\widehat{\psi}_i^{\mathscr{F}} + \widehat{\psi}_i^{\mathscr{B}}} & (Case\ \ 3) \\[2ex] \dfrac{2\left[(\widehat{\psi}_i^{\mathscr{F}})^2 - 2\widehat{\psi}_i^{\mathscr{F}}(\mid \widehat{\psi}_i^{\mathscr{F}}\widehat{\psi}_i^{\mathscr{B}} \mid)^{\frac{1}{2}} + \mid \widehat{\psi}_i^{\mathscr{F}}\widehat{\psi}_i^{\mathscr{B}} \mid\right]}{\widehat{\psi}_i^{\mathscr{F}} + \widehat{\psi}_i^{\mathscr{B}}} & (Case\ \ 4) \\[2ex] \dfrac{2\left[(\widehat{\psi}_i^{\mathscr{B}})^2 - 2\widehat{\psi}_i^{\mathscr{B}}(\mid \widehat{\psi}_i^{\mathscr{F}}\widehat{\psi}_i^{\mathscr{B}} \mid)^{\frac{1}{2}} + \mid \widehat{\psi}_i^{\mathscr{F}}\widehat{\psi}_i^{\mathscr{B}} \mid\right]}{\widehat{\psi}_i^{\mathscr{F}} + \widehat{\psi}_i^{\mathscr{B}}} & (Case\ \ 5) \end{cases}$$

Step iii. Check the convergence. If convergence is achieved, go to Step (iv); otherwise, repeat the iteration cycle (i.e., go to Step (ii))

Step iv. Stop

3.11.4 Remaining Point Calculations for Half- and Quarter-Sweep Iterative Methods

In this research, additional calculations are required for all Half- and Quarter-Sweep iterative methods, i.e. HSMAM, HSMGM, HSMHM, QSMAM, QSMGM and QSMHM. Therefore, second order Lagrange interpolation techniques are applied to determine ψ_i at the remaining points after the convergence criterion is satisfied. Basically, the formulations of second order Lagrange techniques for all Half- and Quarter-Sweep MWM methods are defined by

$$\widehat{\psi}_i = \begin{cases} \frac{3}{8}\widehat{\psi}_{i-1} + \frac{3}{4}\widehat{\psi}_{i+1} - \frac{1}{8}\widehat{\psi}_{i+3}, & i = 1, 3, 5, \cdots, N-3 \\ \frac{3}{4}\widehat{\psi}_{i-1} + \frac{3}{8}\widehat{\psi}_{i+1} - \frac{1}{8}\widehat{\psi}_{i-3}, & i = N-1 \end{cases} \tag{3.100}$$

and

$$\widehat{\psi}_i = \begin{cases} \frac{3}{8}\widehat{\psi}_{i-2} + \frac{3}{4}\widehat{\psi}_{i+2} - \frac{1}{8}\widehat{\psi}_{i+6}, & i = 2, 6, 10, \cdots, N-6 \\ \frac{3}{4}\widehat{\psi}_{i-2} + \frac{3}{8}\widehat{\psi}_{i+2} - \frac{1}{8}\widehat{\psi}_{i-6}, & i = N-2 \\ \frac{3}{8}\widehat{\psi}_{i-1} + \frac{3}{4}\widehat{\psi}_{i+1} - \frac{1}{8}\widehat{\psi}_{i+3}, & i = 1, 3, 5, \cdots, N-3 \\ \frac{3}{4}\widehat{\psi}_{i-1} + \frac{3}{8}\widehat{\psi}_{i+1} - \frac{1}{8}\widehat{\psi}_{i-3}, & i = N-1 \end{cases} \tag{3.101}$$

respectively. The derivations of equations (3.100) and (3.101) are shown in Appendix A.

3.12 Percentage Reduction Calculation

In this research, analysis of percentage reduction is carried out for the tested problems to evaluate the performance of the proposed MWM iterative methods. There are two criteria namely a number of iterations and CPU time (or computational time) are considered to analyse the percentage reduction of the proposed methods relative to the standard or Full-Sweep Guess-Seidel (FSGS) and the standard or Full-Sweep Weighted Mean (FSWM) methods for six different mesh sizes. The percentage reduction is determined by

$$\mathcal{R} = \left(\frac{\varphi_{old} - \varphi_{new}}{\varphi_{old}} \right) \times 100\%, \tag{3.102}$$

where φ_{old} and φ_{new} denote a number of iterations or CPU time of some existing iterative methods and all nine new MWM iterative methods. In Chapter 4, a vigorous analysis of the percentage reduction for all proposed MWM methods are presented.

CHAPTER 4

Numerical Analysis

4.1 Numerical Simulation

In this research, two linear Fredholm IDEs problems are investigated by using the developed iterative methods in Chapter 3. These IDEs are approximated by using Central Difference-Composite Closed Newton-Cotes schemes to obtain the corresponding approximation equations in order to generate the system of linear equations. Following that, the proposed methods namely the FSMAM, HSMAM, QSMAM, FSMGM, HSMGM, QSMGM, FSMHM, HSMHM and QSMHM are applied to solve the generated linear systems to evaluate the effectiveness and efficiency. All the numerical results are shown in the tables for the performance analysis and discussed in detail. In the last section, computational complexities of the proposed and existing methods are analysed.

4.1.1 Numerical Setup

In this section, the second and fourth order linear Fredholm IDEs are solved by using the proposed iterative methods and the results are compared and analysed with some existing methods. The second and fourth order linear Fredholm IDEs problems are chosen from different levels of difficulty in order to claim that the methods have smooth convergence for all types of problems. Therefore, three comparative criteria are considered for performance analysis as shown in Table 4.1.

These criteria are used to measure the effectiveness of the proposed methods associated with several mesh sizes. The CPU time for each computational result was recorded by executing computer programs several times and taking the average values. On the other hand, the new optimal parameter value θ_2 is determined along with the best optimal parameter value θ_1 obtained within

Table 4.1: Measurement criteria

Criteria	Description
Iteration steps	Number of iterations taken to yield the solutions
CPU time	Computational time taken for solutions (in seconds)
RMSE (ε_N)	Root mean square error

± 0.01 from the family of standard WM, where its number of iterations is smallest. For convergence test, the tolerance error is determined with the range $\varepsilon = 10^{-10}$. All the tested methods are implemented by using Intel(R)Core(TM)i5 with CPU@3.20GHz and the codes are written in C programming language. The classic iterative methods such as Gauss-Seidel (GS) and the standard WM methods were used for vigorous mathematical analysis and comparing the performance of all the proposed methods. Therefore, variants of GS methods which are Half-Sweep Gauss-Seidel (HSGS) and Quarter-Sweep Gauss-Seidel (QSGS), and the variants of WM methods namely FSAM, HSAM, QSAM, FSGM, HSGM, QSGM, FSHM, HSHM and QSHM iterative methods were used with the corresponding Full-, Half- and Quarter-Sweep CD-CCNC approximation equations to solve problems 1 and 2.

4.1.2 CD-CCNC Approximation Equations of Second Order Fredholm IDEs

Generally, the second order Fredholm IDEs of second kind can be defined as follows

$$\frac{d^2\psi}{dx^2}(x) - \gamma \int_a^b K(x,\xi)\psi(\xi)d\xi = f(x), x \in (a,b) \tag{4.1}$$

with boundary conditions

$$\psi(a) = a_0 \quad \text{and} \quad \psi(b) = b_0,$$

where $K(x,\xi)$ and $f(x)$ are known functions, α, γ, a_0 and b_0 are assumed constants, and $\psi(x)$ is the unknown function to be determined.

As discussed in the Section 3.9.1, the generalised approximation equations of by CD-CCNC methods of equation (4.1) can be reduced as follows as described in Section 3.9.2

$$\frac{\psi_{i+q} - 2\psi_i + \psi_{i-q}}{(qh)^2} - \sum_{j=q,2q,3q}^{N-q} B_j K_{i,j} \psi_j = f_i \tag{4.2}$$

for $i = q, 2q \cdots, N-q$.

Based on the linear system in equation (3.63) as described in Section 3.9.2, the nonsymmetric dense coefficient matrix, $\mathcal{A}_g$ for equation (4.1) is given by

$$\mathcal{A}_g = \begin{bmatrix} \sigma_{q,q} & \zeta_{q,2q} & \tau_{q,3q} & \tau_{q,4q} & \cdots & \tau_{q,N-q} \\ \zeta_{2q,q} & \sigma_{2q,2q} & \zeta_{2q,3q} & \tau_{2q,4q} & \cdots & \tau_{2q,N-q} \\ \tau_{3q,q} & \zeta_{3q,2q} & \sigma_{3q,3q} & \zeta_{3q,4q} & \cdots & \tau_{3q,N-q} \\ \tau_{4q,q} & \tau_{4q,2q} & \zeta_{4q,3q} & \sigma_{4q,4q} & \cdots & \tau_{4q,N-q} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \tau_{N-q,q} & \tau_{N-q,2q} & \tau_{N-q,3q} & \tau_{N-q,4q} & \cdots & \sigma_{N-q,N-q} \end{bmatrix}$$

with

$$\sigma_{i,i} = 1 - \tfrac{2}{h^2} - B_i K_{i,i}, \quad \zeta_{i,j} = \tfrac{1}{h^2} - B_j K_{i,j} \quad \text{and} \quad \tau_{i,j} = -B_j K_{i,j}$$

and the load vector $\widehat{f}$ in the right hand side is given by

$$\widehat{f}_g = \begin{bmatrix} f_q + (\frac{2}{h^4} - \frac{1}{h^2} - \tau_{1,0})\psi_0 - (\tau_{1,N})\psi_N \\ f_{2q} + (-\frac{1}{h^4} - \tau_{2,0})\psi_0 - (\tau_{2,N})\psi_N \\ f_{3q} - (\tau_{3,0})\psi_0 - (\tau_{3,N})\psi_N \\ f_{4q} - (\tau_{4,0})\psi_0 - (\tau_{4,N})\psi_N \\ \vdots \\ f_{N-q} - (\tau_{N-q,0})\psi_0 + (\frac{2}{h^4} - \frac{1}{h^2} - \tau_{N-q,N})\psi_N \end{bmatrix}.$$

Each element in the coefficient matrix $\mathcal{A}_g$, depends on the kernel functions and the quadrature weights of the 2CCNC, 3CCNC and 5CCNC schemes that have been applied to discretise equation (4.1). Subsequently, the generated linear systems (3.63) are solved by using the proposed MWM iterative methods as discussed in Algorithms 3.11.1, 3.11.2 and 3.11.3.

4.1.3 Numerical Examinations

Problem – 1 [68]

Considering the second order linear FIDE

$$\frac{d^2\psi}{dx^2}(x) - x + 2 - 60\int_0^1 (x-\xi)\psi(\xi)d\xi = 0 \qquad x \in (0,1) \tag{4.3}$$

with boundary conditions

$$\psi(0) = 0 \quad \text{and} \quad \psi(1) = 0,$$

the exact solution is

$$\psi(x) = x.$$

For numerical investigation of Problem-1, the interval $(0,1)$ is divided into even number of abscissae, i.e. 120, 240, 480, 960, 1920 and 3840.

Table 4.2: Numerical results of the FSGS, HSGS and QSGS iterative methods by using the CD-2CCNC schemes (Problem-1)

Methods	**Iterations steps**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	12278	45129	162727	576449	2002098	6870649
HSGS	3251	12278	45129	162727	576449	2002098
QSGS	813	3251	12278	45129	162727	576449
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	1.72	15.38	196.32	2802.84	36179.10	352023.63
HSGS	0.41	3.95	49.58	681.92	8411.67	81228.04
QSGS	0.15	0.87	9.71	156.66	1921.69	13405.09
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	1.854E-5	4.332E-6	4.910E-6	6.414E-6	5.364E-7	1.201E-7
HSGS	1.338E-4	1.854E-5	4.332E-6	4.910E-6	6.414E-6	5.364E-7
QSGS	5.158E-4	1.338E-4	1.854E-5	4.332E-6	4.910E-6	6.414E-6

Table 4.3: Numerical results of the FSGS, HSGS and QSGS iterative methods by using the CD-3CCNC schemes (Problem-1)

Methods	**Iterations steps**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	12276	45127	162725	576448	2006809	6859550
HSGS	3249	12276	45127	162725	576448	1970378
QSGS	809	3249	12276	45127	162725	576448
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	1.75	18.12	224.49	3125.10	38641.24	370225.97
HSGS	0.43	4.05	51.08	692.03	8436.80	81366.01
QSGS	0.16	0.90	10.08	160.18	1953.63	13512.89
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	7.966E-8	3.407E-8	1.376E-8	5.532E-9	2.366E-9	6.269E-10
HSGS	1.371E-7	7.966E-8	3.407E-8	1.376E-8	5.532E-9	2.366E-9
QSGS	5.415E-6	1.371E-7	7.966E-8	3.407E-8	1.376E-8	5.532E-9

Table 4.4: Numerical results of the FSGS, HSGS and QSGS iterative methods by using the CD-5CCNC schemes (Problem-1)

Methods	**Iterations steps**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	12276	45127	162725	576448	2006809	6859550
HSGS	3249	12276	45127	162725	576448	1970378
QSGS	809	3249	12276	45127	162725	576448
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	1.77	20.14	246.29	3260.84	39842.05	271105.30
HSGS	0.45	4.11	54.33	704.36	8496.25	81402.77
QSGS	0.17	1.16	12.24	182.08	1996.20	13606.46
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGS	7.484E-8	3.194E-8	1.249E-8	5.315E-9	2.225E-10	6.036E-10
HSGS	1.268E-6	7.484E-8	3.194E-8	1.249E-8	5.315E-9	2.225E-10
QSGS	5.414E-4	1.268E-6	7.484E-8	3.194E-8	1.249E-8	5.315E-9

Table 4.5: Numerical results of the FSAM, HSAM, QSAM, FSMAM, HSMAM and QSMAM iterative methods by using the CD-2CCNC schemes (Problem-1)

Methods	**Iterations steps**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	1523	5630	20492	73866	263097	927384
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSAM	428	1523	5630	20492	73866	263097
θ_1	(1.76)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSAM	138	428	1523	5630	20492	73866
θ_1	(1.76)	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)
FSMAM	752	2423	7428	26412	89302	318277
θ_2	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)	(1.98)
HSMAM	232	752	2423	7428	26412	89802
θ_2	(1.94)	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)
QSMAM	124	232	752	2423	7428	26412
θ_2	(1.91)	(1.94)	(1.97)	(1.97)	(1.97)	(1.98)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	0.68	5.64	72.59	1036.01	13649.9	199816.1
HSAM	0.26	1.78	22.72	310.19	3645.84	45042.56
QSAM	0.11	0.52	5.25	80.21	765.38	6840.37
FSMAM	0.38	2.96	31.64	353.45	4852.41	64160.17
HSMAM	0.14	0.75	9.22	119.09	1190.28	14068.54
QSMAM	0.05	0.20	1.92	28.59	264.28	2278.65
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	1.862E-5	4.622E-6	1.026E-6	3.020E-7	1.262E-7	5.201E-8
HSAM	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7	1.262E-7
QSAM	5.158E-4	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7
FSMAM	1.860E-5	4.618E-6	1.020E-6	3.014E-7	1.253E-7	5.189E-8
HSMAM	1.338E-4	1.862E-5	4.642E-6	1.114E-6	3.020E-7	1.262E-7
QSMAM	5.158E-4	1.338E-4	1.862E-5	4.642E-6	1.114E-6	3.020E-7

Table 4.6: Numerical results of the FSAM, HSAM, QSAM, FSMAM, HSMAM and QSMAM iterative methods by using the CD-3CCNC schemes (Problem-1)

Methods	Iterations steps					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	1523	5630	20492	73866	263097	927384
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSAM	428	1523	5630	20492	73866	263097
θ_1	(1.76)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSAM	138	428	1523	5630	20492	73866
θ_1	(1.76)	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)
FSMAM	752	2423	7428	26412	89302	318277
θ_2	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)	(1.98)
HSMAM	232	752	2423	7428	26412	89802
θ_2	(1.94)	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)
QSMAM	124	232	752	2423	7428	26412
θ_2	(1.91)	(1.94)	(1.97)	(1.97)	(1.97)	(1.98)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	0.73	6.01	75.64	1040.30	13661.81	199891.44
HSAM	0.27	1.84	23.36	324.78	3685.39	46122.77
QSAM	0.13	0.61	5.98	85.19	778.30	6914.81
FSMAM	0.42	3.12	33.02	368.54	4904.90	64240.46
HSMAM	0.19	0.88	10.09	139.39	1226.25	14120.27
QSMAM	0.08	0.27	2.21	31.88	278.32	2302.31
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	4.906E-9	3.596E-9	1.455E-9	5.831E-10	2.022E-10	8.133E-11
HSAM	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10	2.022E-10
QSAM	5.415E-8	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10
FSMAM	4.901E-9	3.589E-9	1.448E-9	5.824E-10	2.013E-10	8.124E-11
HSMAM	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10	2.022E-10
QSMAM	5.415E-8	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10

Table 4.7: Numerical results of the FSAM, HSAM, QSAM, FSMAM, HSMAM and QSMAM iterative methods by using the CD-5CCNC schemes (Problem-1)

Methods	Iterations steps					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	1523	5630	20492	73866	263097	927384
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSAM	428	1523	5630	20492	73866	263097
θ_1	(1.76)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSAM	138	428	1523	5630	20492	73866
θ_1	(1.76)	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)
FSMAM	752	2423	7428	26412	89302	318277
θ_2	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)	(1.98)
HSMAM	232	752	2423	7428	26412	89802
θ_2	(1.94)	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)
QSMAM	124	232	752	2423	7428	26412
θ_2	(1.91)	(1.94)	(1.97)	(1.97)	(1.97)	(1.98)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	0.78	6.78	82.28	1108.05	13754.31	199985.06
HSAM	0.29	2.78	25.03	388.12	3779.04	46198.28
QSAM	0.14	0.70	6.42	88.47	789.25	6992.26
FSMAM	0.45	3.29	37.21	379.82	4986.06	64293.10
HSMAM	0.22	1.08	8.96	107.10	1381.03	14173.18
QSMAM	0.09	0.28	3.02	30.46	281.58	2355.39
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSAM	4.821E-9	3.419E-9	1.354E-9	5.729E-10	1.987E-10	8.021E-11
HSAM	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10	1.987E-10
QSAM	5.325E-8	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10
FSMAM	4.818E-9	3.412E-9	1.347E-9	5.721E-10	1.978E-10	8.011E-11
HSMAM	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10	1.987E-10
QSMAM	5.325E-8	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10

Table 4.8: Numerical results the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-2CCNC schemes (Problem-1)

Methods	Iterations steps					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	1158	4259	15425	55208	195670	688915
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSGM	348	1158	4259	15425	55208	195670
θ_1	(1.77)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSGM	115	348	1158	4259	15425	55208
θ_1	(1.74)	(1.77)	(1.81)	(1.81)	(1.82)	(1.82)
FSMGM	528	1628	4960	17507	58761	203982
θ_2	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)	(1.98)
HSMGM	170	528	628	4960	17507	58761
θ_2	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)
QSMGM	105	170	528	1628	4960	17507
θ_2	(1.90)	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	0.52	4.03	48.85	691.48	9406.82	134913.1
HSGM	0.20	1.32	16.81	221.52	2558.61	30626.77
QSGM	0.09	0.41	3.92	58.97	521.64	4582.02
FSMGM	0.30	2.29	23.29	254.12	3388.69	41245.87
HSMGM	0.11	0.56	6.71	82.92	810.23	9108.14
QSMGM	0.04	0.15	1.42	20.15	175.31	1470.55
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	1.860E-5	4.618E-6	1.021E-6	3.016E-7	1.256E-7	5.192E-8
HSGM	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7	1.262E-7
QSGM	5.158E-4	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7
FSMGM	1.858E-5	4.613E-6	1.015E-6	3.009E-7	1.248E-7	5.183E-8
HSMGM	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7	1.262E-7
QSMGM	5.158E-4	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7

Table 4.9: Numerical results the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-3CCNC schemes (Problem-1)

Methods	Iterations steps					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	1158	4259	15425	55208	195670	688915
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSGM	348	1158	4259	15425	55208	195670
θ_1	(1.77)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSGM	115	348	1158	4259	15425	55208
θ_1	(1.74)	(1.77)	(1.81)	(1.81)	(1.82)	(1.82)
FSMGM	528	1628	4960	17507	58761	203982
θ_2	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)	(1.98)
HSMGM	170	528	628	4960	17507	58761
θ_2	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)
QSMGM	105	170	528	1628	4960	17507
θ_2	(1.90)	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	0.55	4.21	53.01	706.16	9492.05	134992.72
HSGM	0.24	1.78	19.98	257.16	2604.98	30688.93
QSGM	0.11	0.52	4.54	66.11	558.45	4641.08
FSMGM	0.32	2.45	25.05	261.23	3428.12	41300.81
HSMGM	0.13	0.76	8.46	93.44	855.18	9182.27
QSMGM	0.06	0.23	2.05	28.34	192.05	1501.36
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	4.902E-9	3.593E-9	1.450E-9	5.825E-10	2.017E-10	8.127E-11
HSGM	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10	2.022E-10
QSGM	5.415E-8	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10
FSMGM	4.897E-9	3.586E-9	1.443E-9	5.816E-10	2.009E-10	8.118E-11
HSMGM	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10	2.022E-10
QSMGM	5.415E-8	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10

Table 4.10: Numerical results the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-5CCNC schemes (Problem-1)

Methods	Iterations steps					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	1158	4259	15425	55208	195670	688915
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSGM	348	1158	4259	15425	55208	195670
θ_1	(1.77)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSGM	115	348	1158	4259	15425	55208
θ_1	(1.74)	(1.77)	(1.81)	(1.81)	(1.82)	(1.82)
FSMGM	528	1628	4960	17507	58761	203982
θ_2	(1.97)	(1.97)	(1.97)	(1.98)	(1.98)	(1.98)
HSMGM	170	528	628	4960	17507	58761
θ_2	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)
QSMGM	105	170	528	1628	4960	17507
θ_2	(1.90)	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	0.58	4.59	55.32	720.04	9529.57	134039.32
HSGM	0.28	2.55	26.82	271.40	2678.34	30721.07
QSGM	0.13	0.82	6.54	77.13	572.46	4708.93
FSMGM	0.34	2.53	27.02	279.34	3498.43	41386.43
HSMGM	0.14	0.84	9.04	104.10	874.09	9248.75
QSMGM	0.07	0.45	2.65	38.69	216.64	1576.39
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSGM	4.818E-9	3.414E-9	1.348E-9	5.723E-10	1.980E-10	8.012E-11
HSGM	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10	1.987E-10
QSGM	5.325E-8	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10
FSMGM	4.815E-9	3.408E-9	1.341E-9	5.714E-10	1.971E-10	8.003E-11
HSMGM	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10	1.987E-10
QSMGM	5.325E-8	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10

Table 4.11: Numerical results of the FSHM, HSHM, QSHM, FSMHM, HSMHM and QSMHM iterative methods by using the CD-2CCNC schemes (Problem-1)

Methods	Iterations steps					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	1046	3845	13906	49582	174618	613697
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSHM	317	1046	3845	13906	49582	174618
θ_1	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSHM	106	317	1046	3845	13906	49582
θ_1	(1.75)	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)
FSMHM	472	1455	4417	15501	51819	178712
θ_2	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)	(1.98)
HSMHM	156	472	1455	4417	15501	51819
θ_2	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)
QSMHM	94	156	472	1455	4417	15501
θ_2	(1.90)	(1.93)	(1.96)	(1.96)	(1.97)	(1.98)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	0.45	3.45	41.68	585.78	7958.03	114078.5
HSHM	0.15	0.88	10.51	149.48	1871.81	26360.13
QSHM	0.08	0.35	3.34	50.08	442.16	3832.04
FSMHM	0.25	1.84	18.61	202.4	2668.31	32242.05
HSMHM	0.09	0.44	5.12	62.86	596.32	7003.28
QSMHM	0.03	0.11	1.04	14.67	127.18	1061.96
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	1.859E-5	4.616E-6	1.017E-6	3.014E-7	1.247E-7	5.187E-8
HSHM	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7	1.262E-7
QSHM	5.158E-4	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7
FSMHM	1.858E-5	4.614E-6	1.013E-6	3.007E-7	1.236E-7	5.168E-8
HSMHM	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7	1.262E-7
QSMHM	5.158E-4	1.338E-4	1.862E-5	4.622E-6	1.026E-6	3.020E-7

Table 4.12: Numerical results of the FSHM, HSHM, QSHM, FSMHM, HSMHM and QSMHM iterative methods by using the CD-3CCNC schemes (Problem-1)

Methods	Iterations steps					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	1046	3845	13906	49582	174618	613697
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSHM	317	1046	3845	13906	49582	174618
θ_1	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSHM	106	317	1046	3845	13906	49582
θ_1	(1.75)	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)
FSMHM	472	1455	4417	15501	51819	178712
θ_2	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)	(1.98)
HSMHM	156	472	1455	4417	15501	51819
θ_2	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)
QSMHM	94	156	472	1455	4417	15501
θ_2	(1.90)	(1.93)	(1.96)	(1.96)	(1.97)	(1.98)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	0.49	3.96	53.25	638.31	8021.28	114202.63
HSHM	0.13	1.01	11.34	152.60	1904.95	26453.58
QSHM	0.09	0.42	4.18	61.80	487.90	3924.06
FSMHM	0.29	2.26	19.04	212.30	2745.94	32303.47
HSMHM	0.18	0.52	6.82	80.86	647.21	7110.29
QSMHM	0.04	0.19	2.78	18.92	154.68	1115.12
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	4.900E-9	3.589E-9	1.445E-9	5.818E-10	2.012E-10	8.119E-11
HSHM	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10	2.022E-10
QSHM	5.415E-8	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10
FSMHM	4.995E-9	3.581E-9	1.438E-9	5.807E-10	2.001E-10	8.110E-11
HSMHM	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10	2.022E-10
QSMHM	5.415E-8	1.736E-8	4.906E-9	3.596E-9	1.455E-9	5.831E-10

Table 4.13: Numerical results of the FSHM, HSHM, QSHM, FSMHM, HSMHM and QSMHM iterative methods by using the CD-5CCNC schemes (Problem-1)

Methods	**Iterations steps**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	1046	3845	13906	49582	174618	613697
θ_1	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)	(1.83)
HSHM	317	1046	3845	13906	49582	174618
θ_1	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
QSHM	106	317	1046	3845	13906	49582
θ_1	(1.75)	(1.78)	(1.81)	(1.81)	(1.82)	(1.82)
FSMHM	472	1455	4417	15501	51819	178712
θ_2	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)	(1.98)
HSMHM	156	472	1455	4417	15501	51819
θ_2	(1.93)	(1.96)	(1.96)	(1.97)	(1.97)	(1.98)
QSMHM	94	156	472	1455	4417	15501
θ_2	(1.90)	(1.93)	(1.96)	(1.96)	(1.97)	(1.98)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	0.52	4.24	65.96	654.02	8085.95	114288.77
HSHM	0.14	1.29	15.60	165.27	1942.14	26516.10
QSHM	0.12	0.61	6.07	74.02	520.19	4032.46
FSMHM	0.29	2.02	24.54	225.79	2805.88	32381.38
HSMHM	0.20	0.68	8.14	102.43	682.47	7192.45
QSMHM	0.05	0.31	3.39	24.22	195.84	1148.06
Methods	**RMSE**					
	Mesh Sizes, N					
	120	**240**	**480**	**960**	**1920**	**3840**
FSHM	4.816E-9	3.411E-9	1.342E-9	5.717E-10	1.974E-10	8.003E-11
HSHM	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10	1.987E-10
QSHM	5.325E-8	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10
FSMHM	4.812E-9	3.404E-9	1.335E-9	5.707E-10	1.962E-10	7.991E-11
HSMHM	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10	1.987E-10
QSMHM	5.325E-8	1.632E-8	4.821E-9	3.419E-9	1.354E-9	5.729E-10

4.1.4 Computational Results and Discussions

The numerical results of the studied iterative methods for solving second order Fredholm IDE of Problem-1 are shown in Tables 4.2 to 4.13. Based on the numerical results, the reduction percentages in terms of number of iterations and CPU time for all the proposed MWM iterative methods relative to the existing FSGS and FSWM (i.e. FSAM, FSGM and FSHM) methods have been summarised in Tables 4.14 to 4.16.

Based on the numerical results in the Tables 4.5 to 4.7, it is noticeable that the proposed FSMAM, HSMAM and QSMAM methods are superior compared to the FSGS, HSGS, QSGS, FSAM, HSAM and QSAM iterative methods for the corresponding discretisation schemes namely CD-2CCNC, CD-3CCNC and CD-5CCNC. Tables 4.14 to 4.16 show that there is largest percentage reduction in terms of number of iterations and CPU time compared to the existing FSGS and FSWM methods. Therefore, it clearly shows that the proposed MAM methods performed well in terms of number of iterations and CPU time in solving Problem-1. Meanwhile, the accuracy of the proposed FSMAM method is slightly improved as compared to the existing families of the GS and WM methods.

Tables 4.8 to 4.10 show that the numerical results of proposed FSMGM, HSMGM and QSMGM methods are remarkably improve compared to FSGS, HSGS, QSGS, FSGM, HSGM and QSGM iterative methods for corresponding discretisation schemes i.e. CD-2CCNC, CD-3CCNC and CD-5CCNC. Tables 4.14 to 4.16 show that there is substantial percentage reduction in terms of number of iterations and CPU time compared to the FSGS and FSWM methods. Hence, it evidently show that the proposed MGM methods executed well in terms of number of iterations and CPU time in solving Problem-1. Meanwhile, the RMSE of the proposed FSMGM method is also moderately improved as compared to the existing families of the GS and WM methods.

Based on the proposed MHM methods, Tables 4.11 to 4.13 show that the Full-, Half- and Quarter-Sweep MHM methods are better than the FSGS, HSGS, QSGS, FSGM, HSGM and QSGM iterative methods corresponding to CD-2CCNC, CD-3CCNC and CD-5CCNC approximation schemes, respectively. Tables 4.14 to 4.16 show that there is significant percentage reduction in terms of number of iterations and CPU time compared to the FSGS and FSWM methods. Thus, the numerical treatments obviously show that the proposed MHM methods have improved the number of iterations and CPU time in solving Problem-1. Besides, the RMSE of the proposed FSMHM method is also improved as compared to the existing families of the GS and WM methods for all three combination sets of discretisation schemes.

Therefore, among the discussed methods, the QSMWM iterative methods i.e. QSMAM, QSMGM and QSMHM, are the best methods in terms of number of iterations and CPU time compared to the FSMWM and HSMWM methods for any approximation schemes. This is mainly because of a smaller amount of computational complexity involved for the Quarter-Sweep iterative methods than the Full- and Half-Sweep iterative methods. This statement is affirmed by the analysis of computational complexity study in Section 4.2. Among the methods of QSMAM, QSMGM and QSMHM methods, QSMHM methods are superior than QSMGM and QSMAM methods. Although, the accuracy of the QSMWM are slightly decreased, it can be improved by the mesh refinement. Also, the findings disclose that the iterative methods with CD-3CCNC and CD-5CCNC schemes require more execution time. It is mainly because of the high complexity of the numerical scheme.

Overall, the numerical experiments in solving Problem-1 have shown that the performance of the proposed MWM methods are arguably improved in all the three comparative criteria as mentioned in Section 4.1.1, as compared to the family of GS and the family of the standard WM iterative methods corresponding to CD-2CCNC, CD-3CCNC and CD-5CCNC schemes, respectively. The tables 4.14 to 4.16 show that there is significant percentage reduction in terms of number of iterations and CPU time.

Based on CD-2CCNC approximation schemes, the RMSE for the proposed FSMWM methods are marginally improved as the mesh size N is larger compared to its standard FSWM. The accuracy of the FSMHM is better than the FSMAM and FSMGM. However, CD-3CCNC and CD-5CCNC are exceptionally best approximation schemes compared to CD-2CCNC as they produce more accurate solution to the Problem-1 among all the proposed methods.

Figures 4.1, 4.3, 4.5 and 4.2, 4.4, 4.6 illustrate the analogy of number of iterations and the CPU time versus matrix size N, respectively for the Full-, Half- and Quarter-Sweep WM and MWM methods, by using the CD-2CCNC approximation schemes. Meanwhile, for CD-3CCNC and CD-5CCNC approximation combination schemes are illustrated in Appendix B.

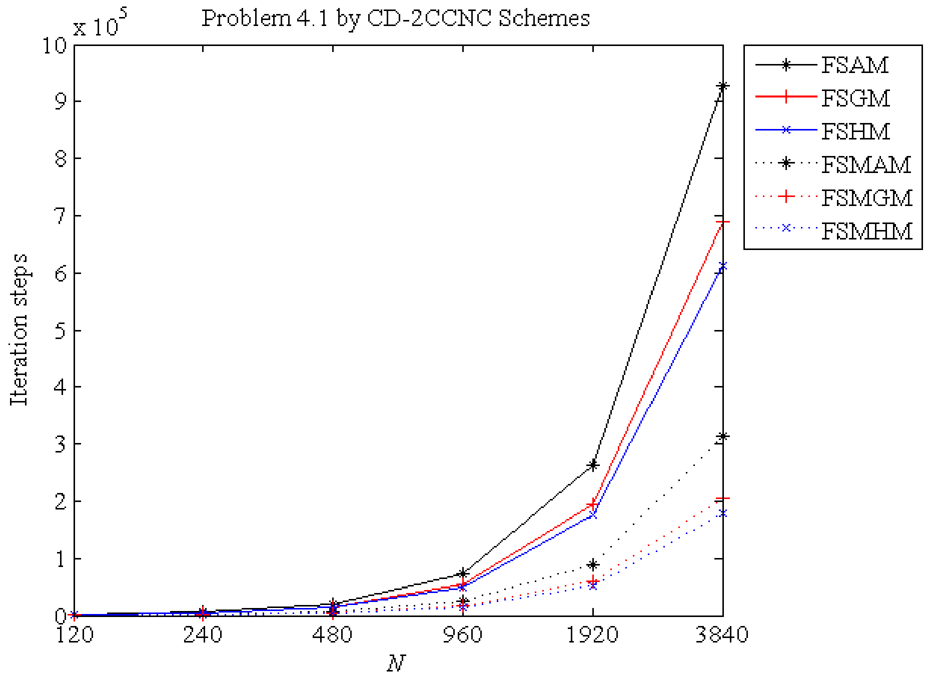

Figure 4.1: Comparison of the families of Full-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-2CCNC schemes.

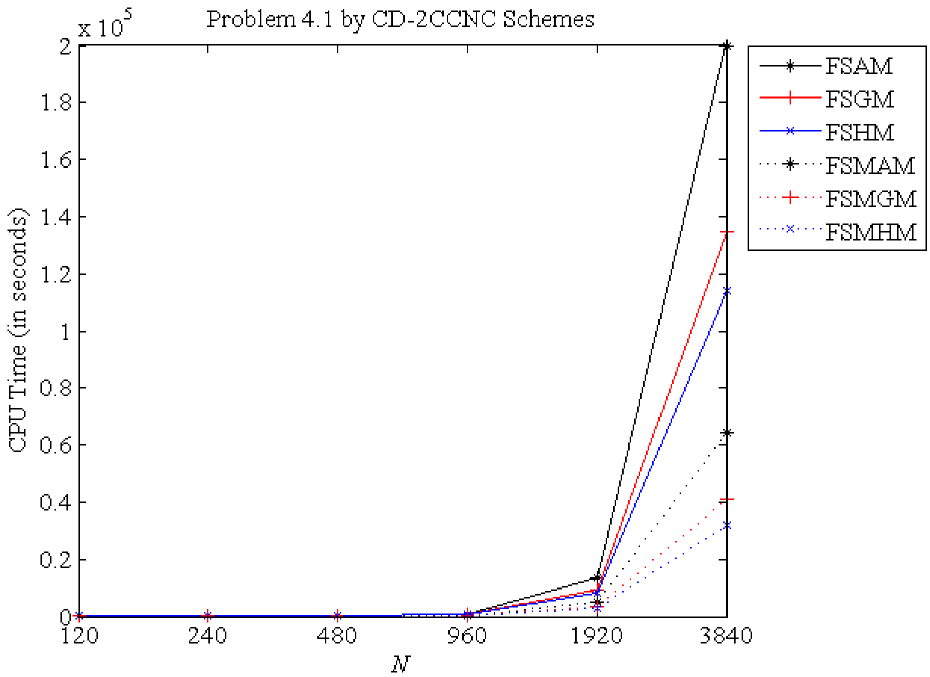

Figure 4.2: Comparison of the families of Full-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-2CCNC schemes.

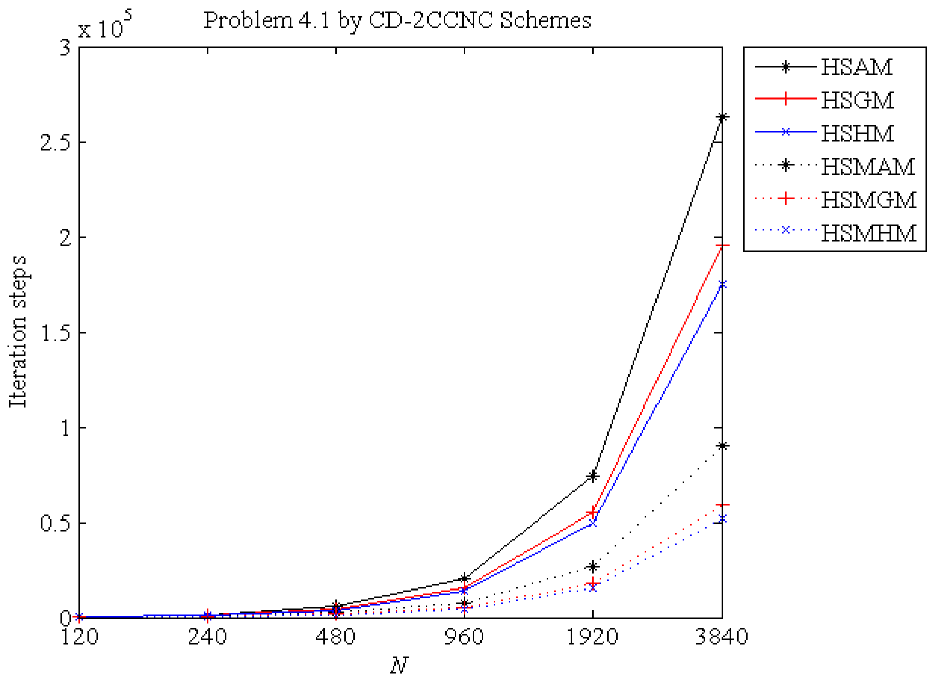

Figure 4.3: Comparison of the families of Half-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-2CCNC schemes.

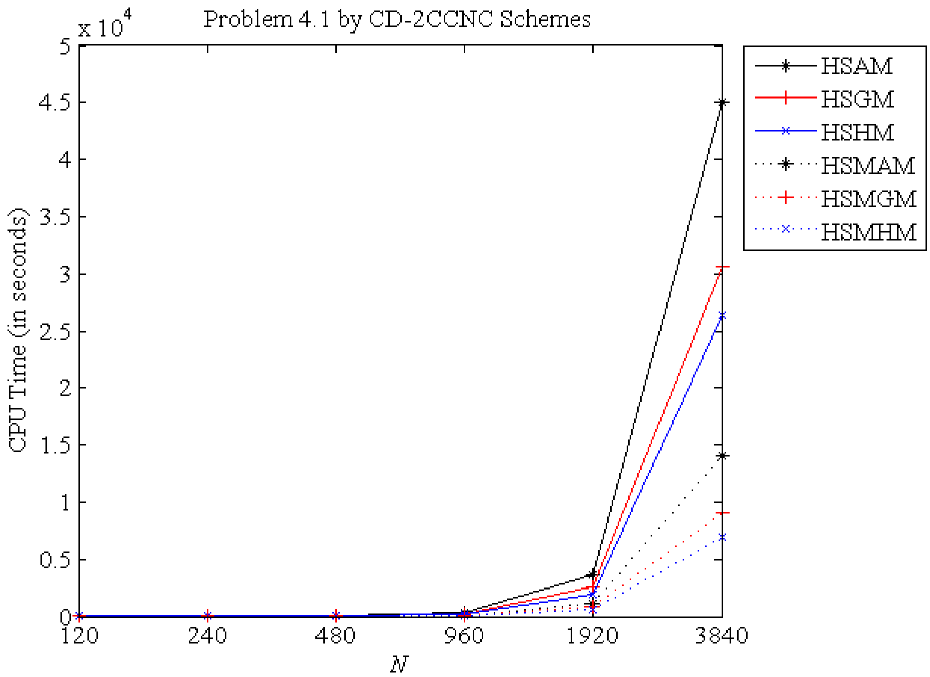

Figure 4.4: Comparison of the families of Half-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-2CCNC schemes.

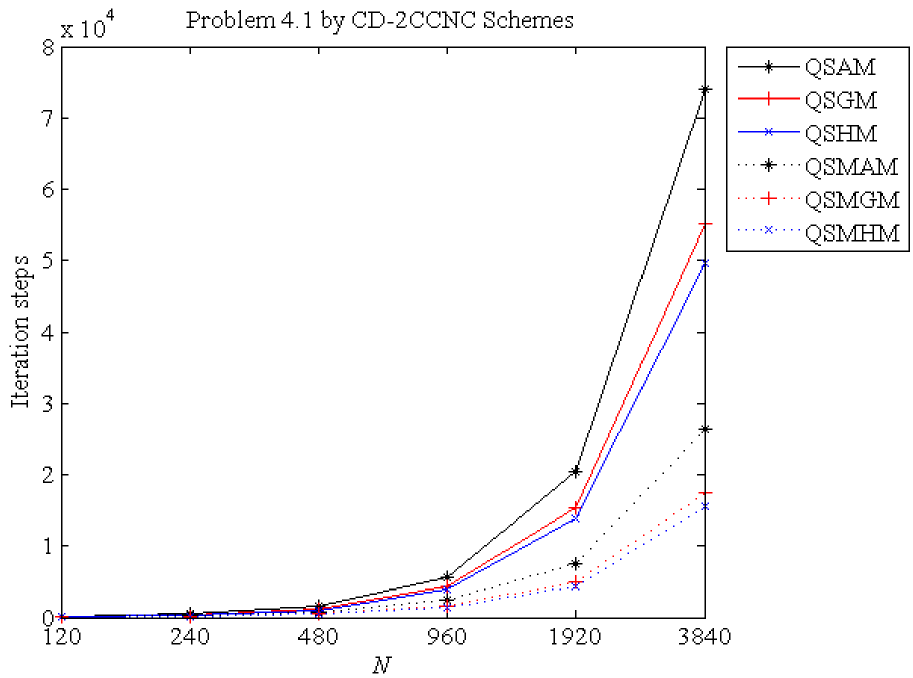

Figure 4.5: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-2CCNC schemes.

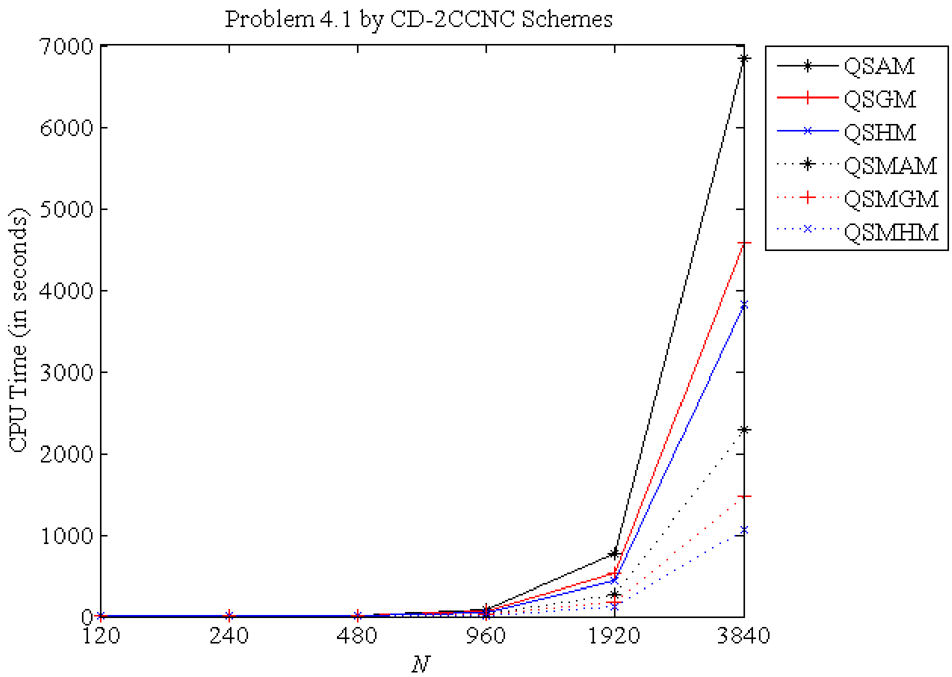

Figure 4.6: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-2CCNC schemes.

Table 4.14: The percentage reduction of Iteration steps and CPU time of proposed MWM relative to FSWM and FSGS methods using the CD-2CCNC schemes (Problem-1)

		Iterations Steps(%)					
Proposed	**Comparison**	**Mesh Sizes,** N					
Methods	**Methods**	**120**	**240**	**480**	**960**	**1920**	**3840**
FSMAM	**FSAM**	50.62	56.96	63.75	64.24	65.87	66.11
	FSGS	93.88	94.63	95.44	95.42	95.51	95.43
FSMGM	**FSGM**	54.40	61.78	67.84	68.29	69.97	70.39
	FSGS	95.70	96.39	96.95	96.96	97.07	97.03
FSMHM	**FSHM**	54.88	62.16	68.24	68.74	70.32	70.88
	FSGS	96.16	96.78	97.29	97.31	97.41	97.40
HSMAM	**FSAM**	84.77	86.64	88.18	89.94	89.96	90.32
	FSGS	98.11	98.33	98.51	98.71	98.68	98.69
HSMGM	**FSGM**	85.32	87.60	89.45	91.02	91.05	91.47
	FSGS	98.62	98.83	99.00	99.14	99.13	99.14
HSMHM	**FSHM**	85.09	87.72	89.54	91.09	91.12	91.56
	FSGS	98.73	98.95	99.11	99.23	99.23	99.25
QSMAM	**FSAM**	91.86	95.88	96.33	96.72	97.18	97.15
	FSGS	98.99	99.49	99.54	99.58	99.63	99.62
QSMGM	**FSGM**	90.93	96.01	96.58	97.05	97.47	97.46
	FSGS	99.14	99.62	99.68	99.72	99.75	99.75
QSMHM	**FSHM**	91.01	95.94	96.61	97.07	97.47	97.47
	FSGS	99.23	99.65	99.71	99.75	99.78	99.77
		CPU Time(%)					
Proposed	**Comparison**	**Mesh Sizes,** N					
Methods	**Methods**	**120**	**240**	**480**	**960**	**1920**	**3840**
FSMAM	**FSAM**	44.12	47.52	56.41	65.88	64.45	67.89
	FSGS	77.91	80.75	83.88	87.39	86.59	81.77
FSMGM	**FSGM**	42.31	43.18	52.32	63.25	63.98	69.43
	FSGS	82.56	85.11	88.14	90.93	90.63	88.28
FSMHM	**FSHM**	44.44	46.67	55.35	65.45	66.47	71.74
	FSGS	85.47	88.04	90.52	92.78	92.62	90.84
HSMAM	**FSAM**	79.41	86.70	87.30	88.50	91.28	92.96
	FSGS	91.86	95.12	95.30	95.75	96.71	96.00
HSMGM	**FSGM**	78.85	86.10	86.26	88.01	91.39	93.25
	FSGS	93.60	96.36	96.58	97.04	97.76	97.41
HSMHM	**FSHM**	80.00	87.25	87.72	89.27	92.51	93.86
	FSGS	94.77	97.14	97.39	97.76	98.35	98.01
QSMAM	**FSAM**	92.65	96.45	97.36	97.24	98.06	98.86
	FSGS	97.09	98.70	99.02	98.98	99.27	99.35
QSMGM	**FSGM**	92.31	96.28	97.09	97.09	98.14	98.91
	FSGS	97.67	99.02	99.28	99.28	99.52	99.58
QSMHM	**FSHM**	93.33	96.81	97.50	97.50	98.40	99.07
	FSGS	98.26	99.28	99.47	99.48	99.65	99.70

Table 4.15: The percentage reduction of Iteration steps and CPU time of proposed MWM relative to FSWM and FSGS methods using the CD-3CCNC schemes (Problem-1)

Proposed Methods	Comparison Methods	Iterations Steps(%) Mesh Sizes, N					
		120	**240**	**480**	**960**	**1920**	**3840**
FSMAM	**FSAM**	50.62	56.96	63.75	64.24	65.87	66.11
	FSGS	93.88	94.63	95.44	95.42	95.51	95.43
FSMGM	**FSGM**	54.40	61.78	67.84	68.29	69.97	70.39
	FSGS	95.70	96.39	96.95	96.96	97.07	97.03
FSMHM	**FSHM**	54.88	62.16	68.24	68.74	70.32	70.88
	FSGS	96.16	96.78	97.29	97.31	97.41	97.40
HSMAM	**FSAM**	84.77	86.64	88.18	89.94	89.96	90.32
	FSGS	98.11	98.33	98.51	98.71	98.68	98.69
HSMGM	**FSGM**	85.32	87.60	89.45	91.02	91.05	91.47
	FSGS	98.62	98.83	99.00	99.14	99.13	99.14
HSMHM	**FSHM**	85.09	87.72	89.54	91.09	91.12	91.56
	FSGS	98.73	98.95	99.11	99.23	99.23	99.25
QSMAM	**FSAM**	91.86	95.88	96.33	96.72	97.18	97.15
	FSGS	98.99	99.49	99.54	99.58	99.63	99.62
QSMGM	**FSGM**	90.93	96.01	96.58	97.05	97.47	97.46
	FSGS	99.14	99.62	99.68	99.72	99.75	99.75
QSMHM	**FSHM**	91.01	95.94	96.61	97.07	97.47	97.47
	FSGS	99.23	99.65	99.71	99.75	99.78	99.77

Proposed Methods	Comparison Methods	CPU Time(%) Mesh Sizes, N					
		120	**240**	**480**	**960**	**1920**	**3840**
FSMAM	**FSAM**	42.47	48.09	56.35	64.57	64.10	67.86
	FSGS	76.00	82.78	85.29	88.21	87.31	82.65
FSMGM	**FSGM**	41.82	41.81	52.74	63.01	63.88	69.41
	FSGS	81.71	86.48	88.84	91.64	91.13	88.84
FSMHM	**FSHM**	40.82	42.93	64.24	66.74	65.77	71.71
	FSGS	83.43	87.53	91.52	93.21	92.89	91.27
HSMAM	**FSAM**	73.97	85.36	86.66	86.60	91.02	92.94
	FSGS	89.14	95.14	95.51	95.54	96.83	96.19
HSMGM	**FSGM**	76.36	81.95	84.04	86.77	90.99	93.20
	FSGS	92.57	95.81	96.23	97.01	97.79	97.52
HSMHM	**FSHM**	63.27	86.87	87.19	87.33	91.93	93.77
	FSGS	89.71	97.13	96.96	97.41	98.33	98.08
QSMAM	**FSAM**	89.04	95.51	97.08	96.94	97.96	98.85
	FSGS	95.43	98.51	99.02	98.98	99.28	99.38
QSMGM	**FSGM**	89.09	94.54	96.13	95.99	97.98	98.89
	FSGS	96.57	98.73	99.09	99.09	99.50	99.59
QSMHM	**FSHM**	91.84	95.20	94.78	97.04	98.07	99.02
	FSGS	97.71	98.95	98.76	99.39	99.60	99.70

Table 4.16: The percentage reduction of Iteration steps and CPU time of proposed MWM relative to FSWM and FSGS methods using the CD-5CCNC schemes (Problem-1)

Proposed Methods	Comparison Methods	Iterations Steps(%) Mesh Sizes, N 120	240	480	960	1920	3840
FSMAM	**FSAM**	50.62	56.96	63.75	64.24	65.87	66.11
	FSGS	93.88	94.63	95.44	95.42	95.51	95.43
FSMGM	**FSGM**	54.40	61.78	67.84	68.29	69.97	70.39
	FSGS	95.70	96.39	96.95	96.96	97.07	97.03
FSMHM	**FSHM**	54.88	62.16	68.24	68.74	70.32	70.88
	FSGS	96.16	96.78	97.29	97.31	97.41	97.40
HSMAM	**FSAM**	84.77	86.64	88.18	89.94	89.96	90.32
	FSGS	98.11	98.33	98.51	98.71	98.68	98.69
HSMGM	**FSGM**	85.32	87.60	89.45	91.02	91.05	91.47
	FSGS	98.62	98.83	99.00	99.14	99.13	99.14
HSMHM	**FSHM**	85.09	87.72	89.54	91.09	91.12	91.56
	FSGS	98.73	98.95	99.11	99.23	99.23	99.25
QSMAM	**FSAM**	91.86	95.88	96.33	96.72	97.18	97.15
	FSGS	98.99	99.49	99.54	99.58	99.63	99.62
QSMGM	**FSGM**	90.93	96.01	96.58	97.05	97.47	97.46
	FSGS	99.14	99.62	99.68	99.72	99.75	99.75
QSMHM	**FSHM**	91.01	95.94	96.61	97.07	97.47	97.47
	FSGS	99.23	99.65	99.71	99.75	99.78	99.77

Proposed Methods	Comparison Methods	CPU Time(%) Mesh Sizes, N 120	240	480	960	1920	3840
FSMAM	**FSAM**	42.31	51.47	54.78	65.72	63.75	67.85
	FSGS	74.58	77.71	84.89	88.35	87.49	82.73
FSMGM	**FSGM**	41.38	44.88	51.16	61.20	63.29	69.12
	FSGS	80.79	87.44	89.03	91.43	91.22	88.89
FSMHM	**FSHM**	44.23	52.36	62.80	65.48	65.30	71.67
	FSGS	83.62	89.97	90.04	93.08	92.96	91.30
HSMAM	**FSAM**	71.79	84.07	89.11	90.33	89.96	92.91
	FSGS	87.57	94.64	96.36	96.72	96.53	96.19
HSMGM	**FSGM**	75.86	81.70	83.66	85.54	90.83	93.10
	FSGS	92.09	95.83	96.33	96.81	97.81	97.52
HSMHM	**FSHM**	61.54	83.96	87.66	84.34	91.56	93.71
	FSGS	88.70	96.62	96.69	96.86	98.29	98.07
QSMAM	**FSAM**	88.46	95.87	96.33	97.25	97.95	98.82
	FSGS	94.92	98.61	98.77	99.07	99.29	99.37
QSMGM	**FSGM**	87.93	90.20	95.21	94.63	97.73	98.82
	FSGS	96.05	97.77	98.92	98.81	99.46	99.58
QSMHM	**FSHM**	90.38	92.69	94.86	96.30	97.58	99.00
	FSGS	97.18	98.46	98.62	99.26	99.51	99.69

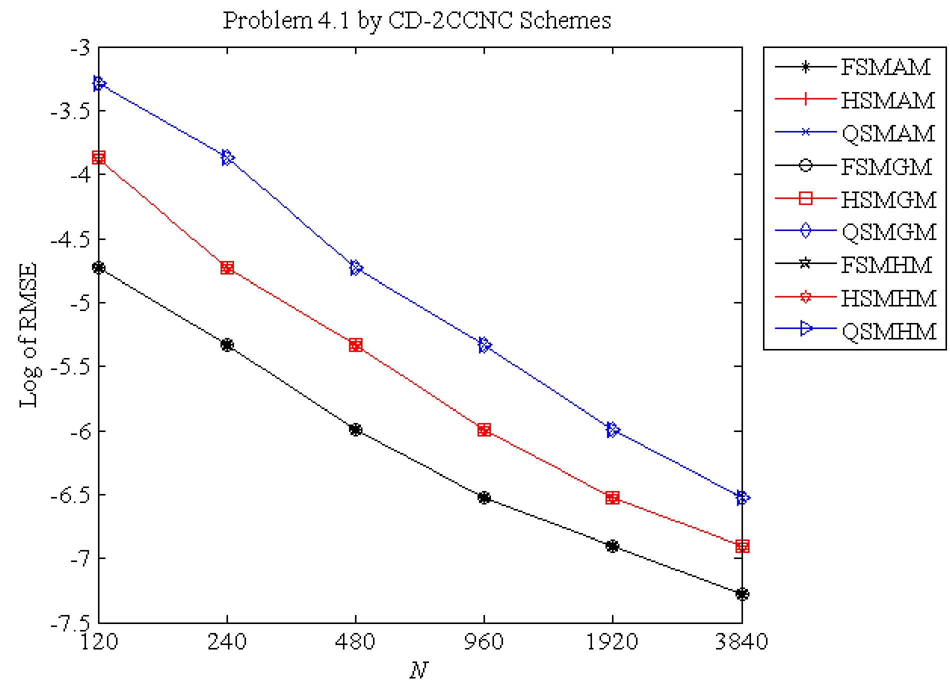

Figure 4.7: Log plot of RMSE for all proposed MWM iterative methods for solving Problem-1 using CD-2CCNC schemes.

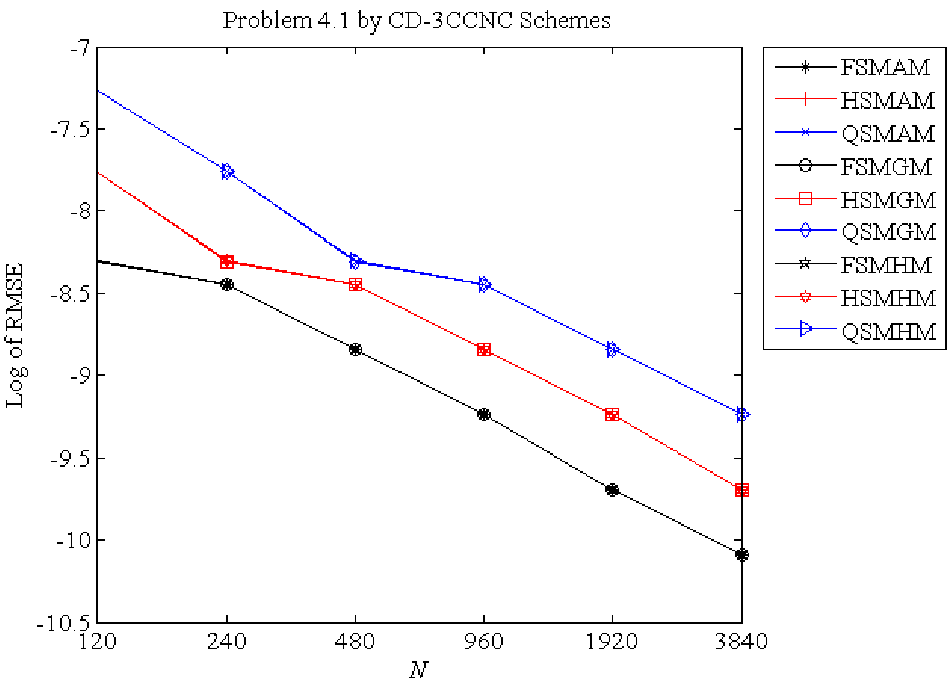

Figure 4.8: Log plot of RMSE for all proposed MWM iterative methods for solving Problem-1 using CD-3CCNC schemes.

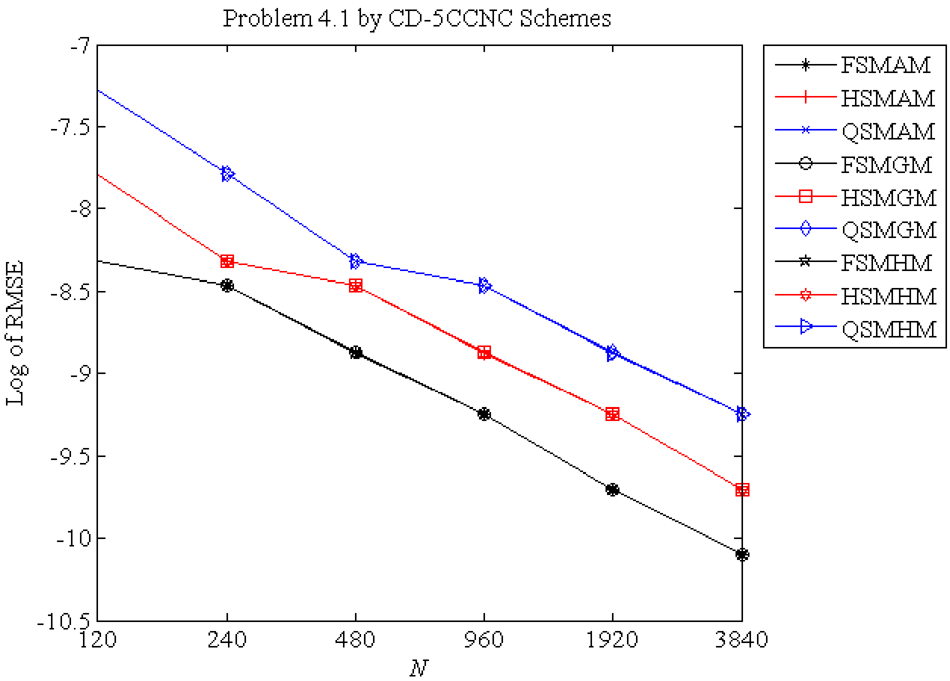

Figure 4.9: Log plot of RMSE for all proposed MWM iterative methods for solving Problem-1 using CD-5CCNC schemes.

4.1.5 CD-CNCC Approximation Equations of Fourth Order Fredholm IDEs

In general, the fourth order linear Fredholm IDEs of second kind can be expressed as follows

$$\frac{d^4\psi}{dx^4}(x) + \alpha\frac{d^2\psi}{dx^2}(x) + \beta\psi(x) - \gamma\int_a^b K(x,\xi)\psi(\xi)d\xi = f(x), x \in (a,b) \qquad (4.4)$$

with boundary conditions

$$\psi(a) = a_0, \quad \psi(b) = b_0, \quad \psi''(a) = a_2 \quad \text{and} \quad \psi''(b) = b_2$$

where $K(x,\xi)$ and $f(x)$ are known functions, α, β, γ, a_0, a_2, b_0 and b_2 are assumed constant, and $\psi(x)$ is the unknown function to be determined.

As discussed in the Section 3.9.1, the generalised Full-, Half- and Quarter-Sweep approximation equations by CD-CCNC methods of equation (4.4) can be reduced as follows

$$\Phi_1 + \alpha_i \Phi_2 + \beta_i \psi_i - \sum_{j=q,2q,3q}^{N-q} B_j K_{i,j} \psi_j = f_i, \qquad i = q, 2q \cdots, N-q \tag{4.5}$$

where,

$$\Phi_1 = \frac{\psi_{i+2q} - 4\psi_{i+q} + 6\psi_i - 4\psi_{i-q} + \psi_{i-2q}}{(qh)^4}$$

and

$$\Phi_2 = \frac{\psi_{i+q} - 2\psi_i + \psi_{i-q}}{(qh)^2}.$$

Based on the generalised linear system in equation (3.63) as described in Section 3.9.2, the dense coefficient matrix $\mathcal{A}_g$ is for equation (4.4) is given by

$$\mathcal{A}_g = \begin{bmatrix} -\frac{1}{h^4} + \sigma_{q,q} & \zeta_{q,2q} & \chi_{q,3q} & \tau_{q,4q} & \cdots & \tau_{q,N-q} \\ \zeta_{2q,q} & \sigma_{2q,2q} & \zeta_{2q,3q} & \chi_{2q,4q} & \cdots & \tau_{2q,N-q} \\ \chi_{3q,q} & \zeta_{3q,2q} & \sigma_{3q,3q} & \zeta_{3q,4q} & \cdots & \tau_{3q,N-q} \\ \tau_{4q,q} & \chi_{4q,2q} & \zeta_{4q,3q} & \sigma_{4q,4q} & \cdots & \tau_{4q,N-q} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \tau_{N-q,q} & \tau_{N-q,2q} & \tau_{N-q,3q} & \tau_{N-q,4q} & \cdots & -\frac{1}{h^4} + \sigma_{N-q,N-q} \end{bmatrix}$$

with
$\sigma_{i,i} = 1 + \frac{6}{h^4} - \frac{2}{h^2} - B_i K_{i,i}$, $\zeta_{i,j} = -\frac{4}{h^4} + \frac{1}{h^2} - B_j K_{i,j}$, $\chi_{i,j} = \frac{1}{h^4} - B_j K_{i,j}$ and $\tau_{i,j} = -B_j K_{i,j}$

and the load vector $\widehat{f}$ in the right hand side is given by

$$\widehat{f_g} = \begin{bmatrix} f_q + (\frac{2}{h^4} - \frac{1}{h^2} - \tau_{q,0})\psi_0 - (\tau_{q,N})\psi_N - \frac{a_2}{h^2} \\ f_{2q} + (-\frac{1}{h^4} - \tau_{2q,0})\psi_0 - (\tau_{2q,N})\psi_N \\ f_{3q} - (\tau_{3q,0})\psi_0 - (\tau_{3q,N})\psi_N \\ f_{4q} - (\tau_{4q,0})\psi_0 - (\tau_{4q,N})\psi_N \\ \vdots \\ f_{N-q} - (\tau_{N-q,0})\psi_0 + (\frac{2}{h^4} - \frac{1}{h^2} - \tau_{N-q,N})\psi_N - \frac{b_2}{h^2} \end{bmatrix}.$$

Each element in the coefficient matrix, $\mathcal{A}_g$, depends on the kernel functions and the quadrature weights of the 2CCNC, 3CCNC and 5CCNC schemes that have been applied to discretise equation (4.4). Subsequently, the generated linear system (3.63) as described in Section 3.9.2 have been solved by using the proposed MWM iterative methods as discussed in Algorithms 3.11.1, 3.11.2 and 3.11.3.

Problems – 2 [59]

Considering the fourth order linear FIDE

$$\frac{d^4\psi}{dx^4}(x) - 2\frac{d^2\psi}{dx^2}(x) - 3x - \int_0^1 x\psi(\xi)d\xi = 0 \tag{4.6}$$

with boundary conditions

$$\psi(0) = 0, \quad \psi(1) = 1, \quad \psi''(0) = 0, \quad \text{and} \quad \psi''(1) = -\frac{87}{47},$$

the exact solution is

$$\psi(x) = -\frac{14}{47}x^3 + \frac{61}{47}x.$$

For numerical examination of Problems-2, the interval $(0, 1)$ is divided into even number of abscissae, i.e. 24, 48, 72, 96, 120 and 144.

Table 4.17: Numerical results of the FSGS, HSGS and QSGS iterative methods by using the CD-2CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	29387	380320	1670949	4716312	10449275	19866526
HSGS	2263	29387	131845	380320	860748	1670949
QSGS	190	2263	10112	29387	67187	131845
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	1.35	22.37	169.88	643.08	2223.63	6052.82
HSGS	0.32	4.94	39.11	153.06	512.72	1374.63
QSGS	0.11	1.67	10.66	39.47	113.59	227.12
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	4.890E-5	1.361E-5	1.677E-5	9.872E-6	6.486E-6	4.059E-6
HSGS	1.877E-4	4.890E-5	2.213E-5	1.361E-5	1.192E-5	1.677E-5
QSGS	7.097E-4	1.877E-4	8.634E-5	4.890E-5	3.142E-5	2.213E-5

Table 4.18: Numerical results of the FSGS, HSGS and QSGS iterative methods by using the CD-3CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	29387	380320	1670949	4716312	10449275	19866526
HSGS	2263	29387	131845	380320	860748	670949
QSGS	190	2263	10112	29387	67187	131845
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	1.46	24.82	170.84	655.47	2306.41	6158.10
HSGS	0.34	5.44	42.36	161.30	521.77	1401.21
QSGS	0.13	1.71	12.97	42.13	115.26	230.56
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	4.882E-5	1.359E-5	1.675E-5	9.871E-6	6.482E-6	4.055E-6
HSGS	1.877E-4	4.882E-5	2.213E-5	1.359E-5	1.192E-5	1.675E-5
QSGS	7.097E-4	1.877E-4	8.634E-5	4.882E-5	3.142E-5	2.213E-5

Table 4.19: Numerical results of the FSGS, HSGS and QSGS iterative methods by using the CD-5CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	29387	380320	1670949	4716312	10449275	19866526
HSGS	2263	29387	131845	380320	860748	1670949
QSGS	190	2263	10112	29387	67187	131845
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	1.48	25.02	171.10	647.06	2240.36	6202.82
HSGS	0.39	5.84	44.68	165.12	531.06	1415.34
QSGS	0.14	1.83	14.03	45.08	117.16	232.69
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGS	4.882E-5	1.359E-5	1.675E-5	9.871E-6	6.482E-6	4.055E-6
HSGS	1.877E-4	4.882E-5	2.213E-5	1.359E-5	1.192E-5	1.675E-5
QSGS	7.097E-4	1.877E-4	8.634E-5	4.882E-5	3.142E-5	2.213E-5

Table 4.20: Numerical results of the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-2CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	7845	58579	249034	695078	1581209	2869243
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSAM	1097	7845	22561	62579	127695	249034
θ_1	(1.76)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSAM	160	1097	3775	7845	13566	22561
θ_1	(1.72)	(1.76)	(1.78)	(1.80)	(1.81)	(1.81)
FSMAM	4012	26910	103370	260144	611759	1035283
θ_2	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)	(1.98)
HSMAM	584	4012	11248	30910	61593	120170
θ_2	(1.85)	(1.96)	(1.96)	(1.96)	(1.96)	(1.97)
QSMAM	88	584	1989	4012	7044	11248
θ_2	(1.82)	(1.85)	(1.90)	(1.96)	(1.96)	(1.96)

Methods	CPU time (in seconds)					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	0.61	4.98	34.84	152.74	480.90	1269.34
HSAM	0.16	1.31	9.28	39.12	124.43	302.68
QSAM	0.08	0.45	2.46	7.82	17.24	33.39
FSMAM	0.32	2.26	15.69	69.08	234.21	606.34
HSMAM	0.09	0.72	4.86	20.42	64.8	156.35
QSMAM	0.05	0.25	1.32	4.01	8.99	16.39

Methods	RMSE					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	4.881E-5	1.244E-5	6.553E-6	5.092E-6	3.743E-6	2.022E-6
HSAM	1.876E-4	4.881E-5	2.177E-5	1.244E-5	8.336E-6	6.553E-6
QSAM	7.098E-4	1.876E-4	8.631E-5	4.881E-5	3.127E-5	2.177E-5
FSMAM	4.881E-5	1.240E-5	6.546E-6	5.069E-6	3.725E-6	1.981E-6
HSMAM	1.876E-4	4.881E-5	2.177E-5	1.244E-5	8.336E-6	6.553E-6
QSMAM	7.098E-4	1.876E-4	8.631E-5	4.881E-5	3.127E-5	2.177E-5

Table 4.21: Numerical results of the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-3CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	7845	58579	249034	695078	1581209	2869243
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSAM	1097	7845	22561	62579	127695	249034
θ_1	(1.76)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSAM	160	1097	3775	7845	13566	22561
θ_1	(1.72)	(1.76)	(1.78)	(1.80)	(1.81)	(1.81)
FSMAM	4012	26910	103370	260144	611759	1035283
θ_2	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)	(1.98)
HSMAM	584	4012	11248	30910	61593	120170
θ_2	(1.85)	(1.96)	(1.96)	(1.96)	(1.96)	(1.97)
QSMAM	88	584	1989	4012	7044	11248
θ_2	(1.82)	(1.85)	(1.90)	(1.96)	(1.96)	(1.96)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	0.62	5.02	36.02	154.21	482.23	1272.28
HSAM	0.18	1.51	10.68	39.98	126.50	305.18
QSAM	0.09	0.38	2.77	9.61	20.05	37.24
FSMAM	0.34	2.29	17.77	75.88	236.3	609.34
HSMAM	0.10	0.83	5.87	21.59	67.64	162.16
QSMAM	0.06	0.23	1.5	5.28	10.69	19.05
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	4.678E-5	1.019E-5	6.307E-6	4.878E-6	3.516E-6	1.898E-6
HSAM	1.782E-4	4.678E-5	2.002E-5	1.019E-5	8.244E-6	6.307E-6
QSAM	6.924E-4	1.782E-4	8.527E-5	4.694E-5	3.013E-5	2.002E-5
FSMAM	4.678E-5	1.017E-5	6.291E-6	4.852E-6	3.485E-6	1.854E-6
HSMAM	1.782E-4	4.678E-5	2.002E-5	1.019E-5	8.244E-6	6.307E-6
QSMAM	6.924E-4	1.782E-4	8.527E-5	4.694E-5	3.013E-5	2.002E-5

Table 4.22: Numerical results of the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-5CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	7845	58579	249034	695078	1581209	2869243
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSAM	1097	7845	22561	62579	127695	249034
θ_1	(1.76)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSAM	160	1097	3775	7845	13566	22561
θ_1	(1.72)	(1.76)	(1.78)	(1.80)	(1.81)	(1.81)
FSMAM	4012	26910	103370	260144	611759	1035283
θ_2	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)	(1.98)
HSMAM	584	4012	11248	30910	61593	120170
θ_2	(1.85)	(1.96)	(1.96)	(1.96)	(1.96)	(1.97)
QSMAM	88	584	1989	4012	7044	11248
θ_2	(1.82)	(1.85)	(1.90)	(1.96)	(1.96)	(1.96)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	0.63	5.08	37.71	159.85	488.21	1278.92
HSAM	0.20	1.66	10.94	41.13	128.67	308.64
QSAM	0.11	0.45	3.13	9.75	21.34	38.48
FSMAM	0.34	2.38	17.35	72.65	218.95	611.06
HSMAM	0.11	0.91	6.01	21.36	67.11	159.36
QSMAM	0.07	0.28	1.84	5.56	12.01	21.14
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSAM	4.678E-5	1.019E-5	6.307E-6	4.878E-6	3.516E-6	1.898E-6
HSAM	1.782E-4	4.678E-5	2.002E-5	1.019E-5	8.244E-6	6.307E-6
QSAM	6.924E-4	1.782E-4	8.527E-5	4.694E-5	3.013E-5	2.002E-5
FSMAM	4.678E-5	1.017E-5	6.291E-6	4.852E-6	3.485E-6	1.854E-6
HSMAM	1.782E-4	4.678E-5	2.002E-5	1.019E-5	8.244E-6	6.307E-6
QSMAM	6.924E-4	1.782E-4	8.527E-5	4.694E-5	3.013E-5	2.002E-5

Table 4.23: Numerical results of the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-2CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	6281	46562	197025	545289	1236083	2217532
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSGM	894	6281	17998	50062	101718	197025
θ_1	(1.75)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSGM	131	894	3051	6281	10828	17998
θ_1	(1.71)	(1.75)	(1.79)	(1.80)	(1.81)	(1.81)
FSMGM	3103	21535	86810	176366	389195	756617
θ_2	(1.95)	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)
HSMGM	450	3103	8796	23535	46821	89810
θ_2	(1.87)	(1.95)	(1.96)	(1.96)	(1.96)	(1.96)
QSMGM	72	450	1524	3103	5297	8796
θ_2	(1.83)	(1.87)	(1.93)	(1.96)	(1.96)	(1.96)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	0.52	4.12	28.82	125.93	394.13	1036.22
HSGM	0.13	1.06	7.49	31.38	99.56	242.09
QSGM	0.07	0.39	2.02	6.41	14.01	26.87
FSMGM	0.28	1.87	13.41	62.51	191.73	492.66
HSMGM	0.07	0.55	3.69	15.36	48.05	112.08
QSMGM	0.04	0.2	1.03	3.12	6.96	12.68
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	4.869E-5	1.221E-5	6.429E-6	4.986E-6	3.713E-6	1.973E-6
HSGM	1.867E-4	4.869E-5	2.162E-5	1.221E-5	8.316E-6	6.429E-6
QSGM	7.088E-4	1.867E-4	8.619E-5	4.869E-5	3.114E-5	2.162E-5
FSMGM	4.868E-5	1.215E-5	6.412E-6	4.944E-6	3.701E-6	1.945E-6
HSMGM	1.867E-4	4.869E-5	2.162E-5	1.221E-5	8.316E-6	6.429E-6
QSMGM	7.088E-4	1.867E-4	8.619E-5	4.869E-5	3.114E-5	2.162E-5

Table 4.24: Numerical results of the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-3CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	6281	46562	197025	545289	1236083	2217532
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSGM	894	6281	17998	50062	101718	197025
θ_1	(1.75)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSGM	131	894	3051	6281	10828	17998
θ_1	(1.71)	(1.75)	(1.79)	(1.80)	(1.81)	(1.81)
FSMGM	3103	21535	86810	176366	389195	756617
θ_2	(1.95)	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)
HSMGM	450	3103	8796	23535	46821	89810
θ_2	(1.87)	(1.95)	(1.96)	(1.96)	(1.96)	(1.96)
QSMGM	72	450	1524	3103	5297	8796
θ_2	(1.83)	(1.87)	(1.93)	(1.96)	(1.96)	(1.96)

Methods	CPU time (in seconds)					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	0.54	4.22	30.48	127.36	399.62	1045.08
HSGM	0.15	1.25	8.83	32.87	103.59	248.95
QSGM	0.08	0.32	2.32	7.88	16.29	30.04
FSMGM	0.30	1.97	15.16	64.21	194.35	495.98
HSMGM	0.08	0.62	4.28	15.61	48.88	115.30
QSMGM	0.05	0.19	1.23	4.27	8.61	15.32

Methods	RMSE					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	4.676E-5	1.009E-5	6.198E-6	4.766E-6	3.312E-6	1.794E-6
HSGM	1.694E-4	4.676E-5	1.986E-5	1.009E-5	8.232E-6	6.198E-6
QSGM	6.916E-4	1.694E-4	8.513E-5	4.676E-5	2.993E-5	1.986E-5
FSMGM	4.676E-5	1.003E-5	6.187E-6	4.717E-6	3.266E-6	1.742E-6
HSMGM	1.694E-4	4.676E-5	1.986E-5	1.009E-5	8.232E-6	6.198E-6
QSMGM	6.916E-4	1.694E-4	8.513E-5	4.676E-5	2.993E-5	1.986E-5

Table 4.25: Numerical results of the FSGM, HSGM, QSGM, FSMGM, HSMGM and QSMGM iterative methods by using the CD-5CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	6281	46562	197025	545289	1236083	2217532
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSGM	894	6281	17998	50062	101718	197025
θ_1	(1.75)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSGM	131	894	3051	6281	10828	17998
θ_1	(1.71)	(1.75)	(1.79)	(1.80)	(1.81)	(1.81)
FSMGM	3103	21535	86810	176366	389195	756617
θ_2	(1.95)	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)
HSMGM	450	3103	8796	23535	46821	89810
θ_2	(1.87)	(1.95)	(1.96)	(1.96)	(1.96)	(1.96)
QSMGM	72	450	1524	3103	5297	8796
θ_2	(1.83)	(1.87)	(1.93)	(1.96)	(1.96)	(1.96)

Methods	CPU time (in seconds)					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	0.54	4.24	31.36	132.09	402.9	1050.47
HSGM	0.17	1.35	9.09	32.60	101.93	243.29
QSGM	0.09	0.37	2.54	7.92	17.18	31.09
FSMGM	0.29	1.98	14.32	59.82	179.57	500.98
HSMGM	0.09	0.74	4.56	16.85	52.63	123.12
QSMGM	0.06	0.23	1.51	4.34	9.25	16.32

Methods	RMSE					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSGM	4.676E-5	1.009E-5	6.198E-6	4.766E-6	3.312E-6	1.794E-6
HSGM	1.694E-4	4.676E-5	1.986E-5	1.009E-5	8.232E-6	6.198E-6
QSGM	6.916E-4	1.694E-4	8.513E-5	4.676E-5	2.993E-5	1.986E-5
FSMGM	4.676E-5	1.003E-5	6.187E-6	4.717E-6	3.266E-6	1.742E-6
HSMGM	1.694E-4	4.676E-5	1.986E-5	1.009E-5	8.232E-6	6.198E-6
QSMGM	6.916E-4	1.694E-4	8.513E-5	4.676E-5	2.993E-5	1.986E-5

Table 4.26: Numerical results of the FSHM, HSHM, QSHM, FSMHM, HSMHM and QSMHM iterative methods by using the CD-2CCNC schemes (Problems-2)

Methods	**Iterations steps**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	5814	43081	181943	500890	1132491	2021864
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSHM	826	5794	16612	46443	93453	181943
θ_1	(1.75)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSHM	122	826	2818	5794	9992	16612
θ_1	(1.71)	(1.75)	(1.79)	(1.80)	(1.81)	(1.81)
FSMHM	2798	19385	78078	158527	338986	661713
θ_2	(1.95)	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)
HSMHM	408	2798	7872	20789	41022	78078
θ_2	(1.87)	(1.95)	(1.96)	(1.96)	(1.96)	(1.96)
QSMHM	66	408	1379	2798	4749	7872
θ_2	(1.83)	(1.87)	(1.93)	(1.95)	(1.96)	(1.96)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	0.49	3.80	26.4	115.19	360.05	938.12
HSHM	0.12	0.97	6.84	28.57	90.59	220.02
QSHM	0.06	0.34	1.76	5.60	12.29	23.64
FSMHM	0.26	1.70	12.14	56.45	172.59	440.81
HSMHM	0.06	0.47	3.14	13.05	40.52	95.19
QSMHM	0.03	0.16	0.91	2.74	6.07	11.01
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	4.867E-5	1.212E-5	6.408E-6	4.938E-6	3.691E-6	1.939E-6
HSHM	1.864E-4	4.867E-5	2.150E-5	1.212E-5	8.303E-6	6.408E-6
QSHM	7.086E-4	1.864E-4	8.611E-5	4.867E-5	3.099E-5	2.150E-5
FSMHM	4.866E-5	1.210E-5	6.397E-6	4.925E-6	3.682E-6	1.916E-6
HSMHM	1.864E-4	4.867E-5	2.150E-5	1.212E-5	8.303E-6	6.408E-6
QSMHM	7.086E-4	1.864E-4	8.611E-5	4.867E-5	3.099E-5	2.150E-5

Table 4.27: Numerical results of the FSHM, HSHM, QSHM, FSMHM, HSMHM and QSMHM iterative methods by using the CD-3CCNC schemes (Problems-2)

Methods	**Iterations steps**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	5814	43081	181943	500890	1132491	2021864
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSHM	826	5794	16612	46443	93453	181943
θ_1	(1.75)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSHM	122	826	2818	5794	9992	16612
θ_1	(1.71)	(1.75)	(1.79)	(1.80)	(1.81)	(1.81)
FSMHM	2798	19385	78078	158527	338986	661713
θ_2	(1.95)	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)
HSMHM	408	2798	7872	20789	41022	78078
θ_2	(1.87)	(1.95)	(1.96)	(1.96)	(1.96)	(1.96)
QSMHM	66	408	1379	2798	4749	7872
θ_2	(1.83)	(1.87)	(1.93)	(1.95)	(1.96)	(1.96)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	0.51	3.98	28.21	116.02	365.12	947.02
HSHM	0.14	1.15	8.18	30.26	95.18	227.08
QSHM	0.07	0.29	2.09	7.07	14.61	27.09
FSMHM	0.28	1.82	13.98	58.92	177.05	448.33
HSMHM	0.07	0.56	3.71	13.38	42.01	98.95
QSMHM	0.04	0.16	1.05	3.75	7.72	13.61
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	4.672E-5	1.004E-5	6.177E-6	4.701E-6	3.241E-6	1.763E-6
HSHM	1.690E-4	4.672E-5	1.972E-5	1.004E-5	8.215E-6	6.177E-6
QSHM	6.911E-4	1.690E-4	8.508E-5	4.672E-5	2.988E-5	1.972E-5
FSMHM	4.672E-5	1.001E-5	6.169E-6	4.681E-6	3.224E-6	1.730E-6
HSMHM	1.690E-4	4.672E-5	1.972E-5	1.004E-5	8.215E-6	6.177E-6
QSMHM	6.911E-4	1.690E-4	8.508E-5	4.672E-5	2.988E-5	1.972E-5

Table 4.28: Numerical results of the FSHM, HSHM, QSHM, FSMHM, HSMHM and QSMHM iterative methods by using the CD-5CCNC schemes (Problems-2)

Methods	Iterations steps					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	5814	43081	181943	500890	1132491	2021864
θ_1	(1.80)	(1.81)	(1.81)	(1.82)	(1.82)	(1.83)
HSHM	826	5794	16612	46443	93453	181943
θ_1	(1.75)	(1.80)	(1.81)	(1.81)	(1.81)	(1.81)
QSHM	122	826	2818	5794	9992	16612
θ_1	(1.71)	(1.75)	(1.79)	(1.80)	(1.81)	(1.81)
FSMHM	2798	19385	78078	158527	338986	661713
θ_2	(1.95)	(1.96)	(1.96)	(1.97)	(1.97)	(1.97)
HSMHM	408	2798	7872	20789	41022	78078
θ_2	(1.87)	(1.95)	(1.96)	(1.96)	(1.96)	(1.96)
QSMHM	66	408	1379	2798	4749	7872
θ_2	(1.83)	(1.87)	(1.93)	(1.95)	(1.96)	(1.96)
Methods	**CPU time (in seconds)**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	0.51	3.95	29.15	122.16	370.75	962.13
HSHM	0.16	1.28	8.27	29.82	93.28	221.64
QSHM	0.08	0.34	2.29	7.14	15.71	28.46
FSMHM	0.27	1.81	13.07	54.49	163.21	454.56
HSMHM	0.08	0.65	4.05	14.72	45.95	107.03
QSMHM	0.05	0.2	1.32	3.82	8.18	14.56
Methods	**RMSE**					
	Mesh Sizes, N					
	24	**48**	**72**	**96**	**120**	**144**
FSHM	4.672E-5	1.004E-5	6.177E-6	4.701E-6	3.241E-6	1.763E-6
HSHM	1.690E-4	4.672E-5	1.972E-5	1.004E-5	8.215E-6	6.177E-6
QSHM	6.911E-4	1.690E-4	8.508E-5	4.672E-5	2.988E-5	1.972E-5
FSMHM	4.672E-5	1.001E-5	6.169E-6	4.681E-6	3.224E-6	1.730E-6
HSMHM	1.690E-4	4.672E-5	1.972E-5	1.004E-5	8.215E-6	6.177E-6
QSMHM	6.911E-4	1.690E-4	8.508E-5	4.672E-5	2.988E-5	1.972E-5

4.1.6 Computational Results and Discussions

Numerical experiments are carried out for solving fourth order Fredholm IDE (i.e. Problems-2) by using all the considered methods, and the results are presented in the Tables 4.17 to 4.28. Based on the numerical results, the reduction percentages in terms of a number of iterations and CPU time for all proposed MWM iterative methods relative to FSGS and FSWM (i.e. FSAM, FSGM and FSHM) methods have been summarised in Tables 4.29 to 4.31.

Based on the numerical results in Tables 4.20 to 4.22, it clearly shows that the proposed FSMAM, HSMAM and QSMAM methods have generated a better results compared to the FSGS, HSGS, QSGS, FSAM, HSAM and QSAM iterative methods, for all discretisation schemes (i.e. CD-2CCNC, CD-3CCNC and CD-5CCNC). This statement is evidently supported by the Tables 4.29 to 4.31, where the percentage reductions in terms of a number of iterations and CPU time relative to the FSGS and FSWM methods. Meanwhile, the accuracy of the proposed FSMAM method is slightly improved as compared to those obtained by the existing families of GS and WM iterative methods.

Tables 4.23 to 4.25 show that the proposed FSMGM, HSMGM and QSMGM methods are better than FSGS, HSGS, QSGS, FSGM, HSGM and QSGM iterative methods for corresponding discretisation schemes (i.e. CD-2CCNC, CD-3CCNC and CD-5CCNC). Tables 4.29 to 4.31 also support the statement above as there is substantial percentage reduction in terms of a number of iterations and CPU time relative to the FSGS and FSWM methods. Meanwhile, the RMSE of the proposed FSMGM method is also slightly improved at large mesh sizes as compared to those obtained by the existing families of GS and WM iterative methods.

Furthermore, the numerical results of the proposed MHM methods are presented in the Tables 4.26 to 4.28. The results show that the Full-, Half- and Quarter-Sweep MHM methods are better than the FSGS, HSGS, QSGS, FSGM, HSGM and QSGM iterative methods for the CD-2CCNC, CD-3CCNC and CD-5CCNC approximation schemes. This is proven by the percentage reduction calculation as shown in Tables 4.29 to 4.31, where the number of iterations and CPU time decrease relative to the FSGS and FSWM methods. In addition, the RMSE of the proposed FSMHM method is also in good agreement as compared to the existing families of GS and WM iterative methods for the considered discretisation schemes.

In general, the numerical experiments in solving Problems-2 have shown that the performance of the proposed MWM methods are arguably improved in all three comparative criteria as mentioned in Section 4.1.1, compared to those obtained by the existing families of GS and WM iterative methods for all the CD-2CCNC, CD-3CCNC and CD-5CCNC schemes. Tables 4.14 to 4.16 also show that there are continuous improvement for all three comparative criteria, where the results are improved from family of MAM to the family of MGM and from the family of MGM to the family of MHM methods compared to the existing FSGS and FSWM methods for all three combination set of discretisation schemes. Among the Full-, Half- and Quarter-Sweep MWM methods, the FSMWM methods i.e. FSMAM, FSMGM and FSMHM methods have generated more accurate solutions than the HSMWM and QSMWM methods. The reason is that the Full-Sweep methods uses smaller step size h, while the Half- and the Quarter-Sweep iterative methods use $2h$ and $4h$ step sizes respectively. However, the accuracy of the approximation solutions is improved by the refinement of the mesh size, N.

Based on the CD-CCNC approximation schemes, the RMSE for the proposed MWM methods is marginally improved as the mesh size N increases compared to the standard WM. For the Full-Sweep, the accuracy of the FSMHM is better than those by FSMAM and FSMGM for each discretisation schemes. The CD-3CCNC and CD-5CCNC are exceptionally best approximation schemes compared to CD-2CCNC as they produce more accurate solution to the Problems-2 for all proposed methods though the CPU time is slightly higher than that in the CD-2CCNC scheme. It is mainly because of the high complexity of the numerical scheme.

Figures 4.10, 4.12, 4.14 and 4.11, 4.13, 4.15 illustrate the analogy of number of iterations and the CPU time versus matrix size N, respectively for the Full-, Half- and Quarter-Sweep of WM and the MWM methods by using the CD-2CCNC approximation equation. Meanwhile, the performance comparison of the studied iterative methods for the CD-3CCNC and the CD-5CCNC schemes are illustrated in Appendix B.

Overall, among the proposed MWM iterative methods, QSMAM, QSMGM and QSMHM are the best methods in terms of number of iterations and CPU time compared to the FSMWM and HSMWM methods for all approximation schemes. This is mainly because of a smaller amount of computational complexity involved for the Quarter-Sweep iterative methods than Full- and Half-Sweep iterative methods. This statement is affirmed by the analysis of computational complexity study in Section 4.2. Among the QSMWM methods, QSMHM method is better than the QSMGM and QSMAM methods.

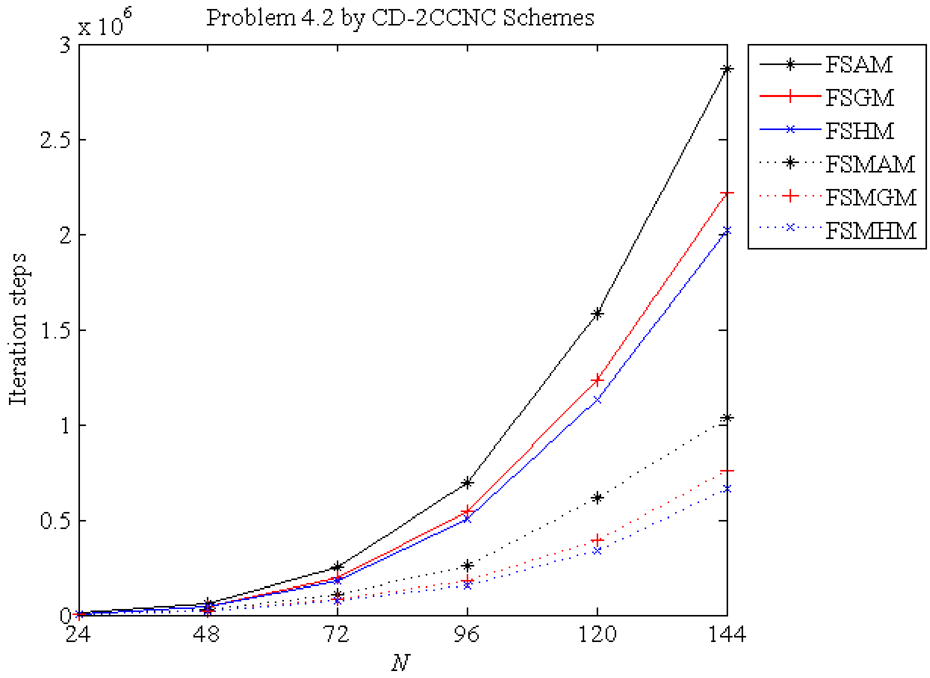

Figure 4.10: Comparison of the families of Full-Sweep WM and MWM methods in terms of Iteration steps for solving Problems-2 by CD-2CCNC schemes.

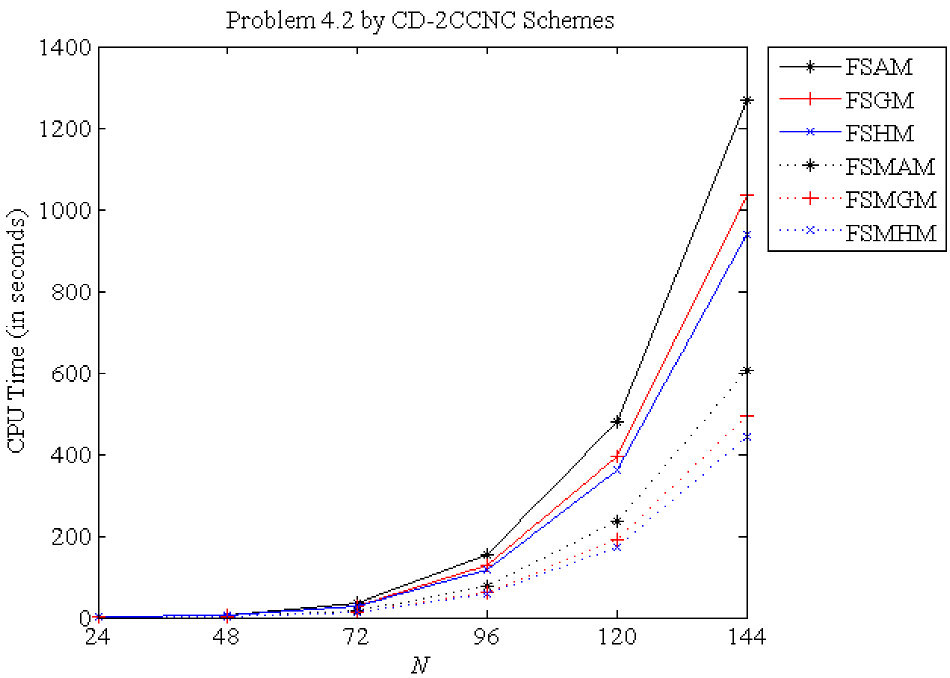

Figure 4.11: Comparison of the families of Full-Sweep WM and MWM methods in terms of CPU time for solving Problems-2 by CD-2CCNC schemes.

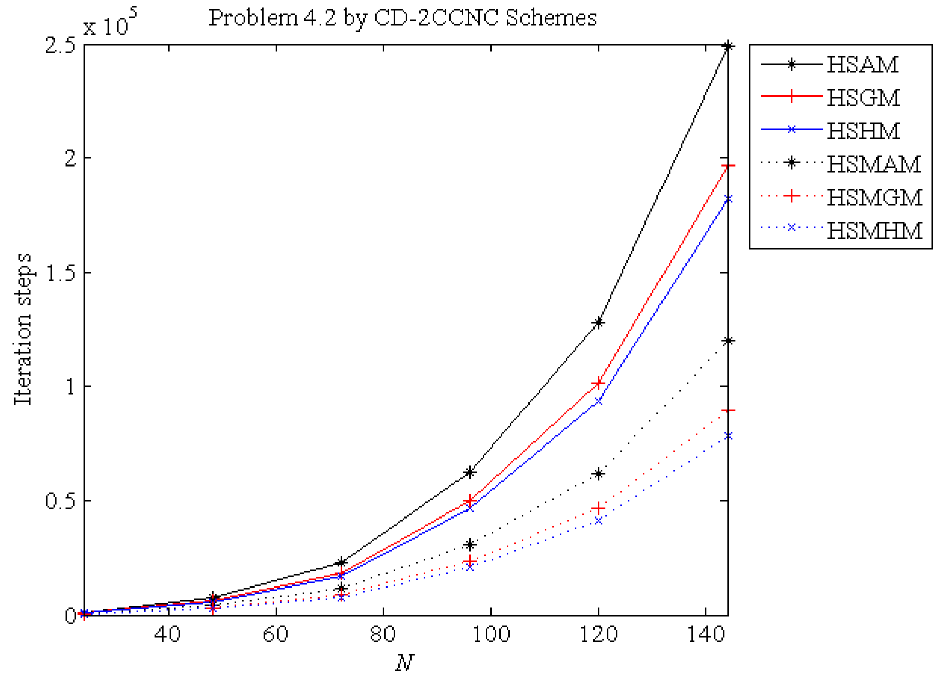

Figure 4.12: Comparison of the families of Half-Sweep WM and MWM methods in terms of Iteration steps for solving Problems-2 by CD-2CCNC schemes.

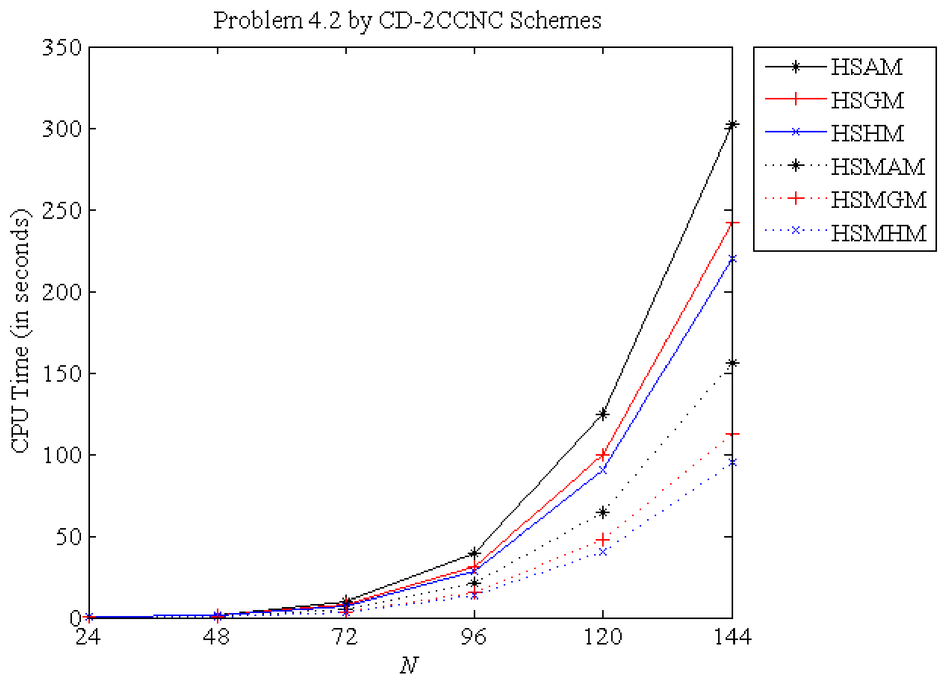

Figure 4.13: Comparison of the families of Half-Sweep WM and MWM methods in terms of CPU time for solving Problems-2 by CD-2CCNC schemes.

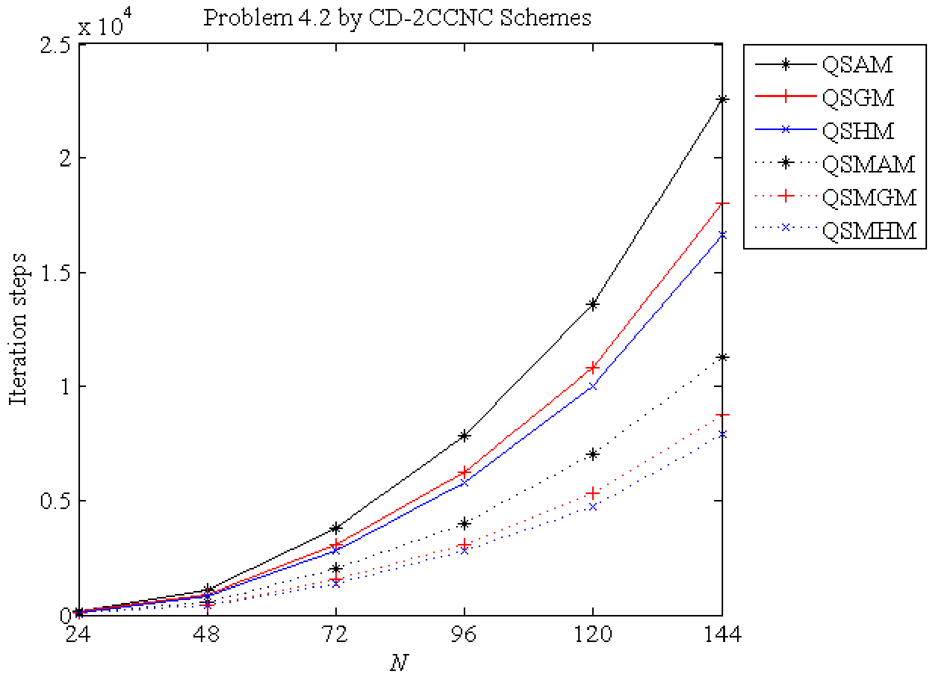

Figure 4.14: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of Iteration steps for solving Problems-2 by CD-2CCNC schemes.

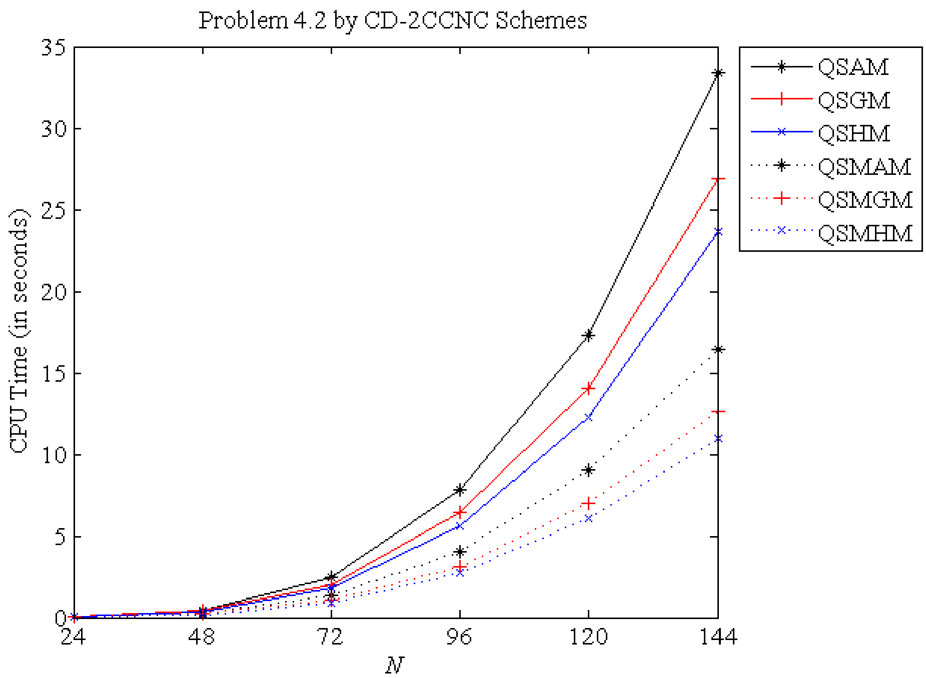

Figure 4.15: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of CPU time for solving Problems-2 by CD-2CCNC schemes.

Table 4.29: The reduction percentage in iteration steps and CPU time of the proposed iterative methods by using the CD-2CCNC scheme (Problems-2)

Proposed Methods	Comparison Methods	Iterations Steps(%) Mesh Sizes, N 24	48	72	96	120	144
FSMAM	**FSAM**	48.86	54.06	58.49	62.57	61.31	63.92
	FSGS	86.35	92.92	93.81	94.48	94.15	94.79
FSMGM	**FSGM**	50.60	53.75	55.94	67.66	68.51	65.88
	FSGS	89.44	94.34	94.80	96.26	96.28	96.19
FSMHM	**FSHM**	51.87	55.00	57.09	68.35	70.07	67.27
	FSGS	90.48	94.90	95.33	96.64	96.76	96.67
HSMAM	**FSAM**	92.56	93.15	95.48	95.55	96.10	95.81
	FSGS	98.01	98.95	99.33	99.34	99.41	99.40
HSMGM	**FSGM**	92.84	93.34	95.54	95.68	96.21	95.95
	FSGS	98.47	99.18	99.47	99.50	99.55	99.55
HSMHM	**FSHM**	92.98	93.51	95.67	95.85	96.38	96.14
	FSGS	98.61	99.26	99.53	99.56	99.61	99.61
QSMAM	**FSAM**	98.88	99.00	99.20	99.42	99.55	99.61
	FSGS	99.70	99.85	99.88	99.91	99.93	99.94
QSMGM	**FSGM**	98.85	99.03	99.23	99.43	99.57	99.60
	FSGS	99.75	99.88	99.91	99.93	99.95	99.96
QSMHM	**FSHM**	98.86	99.05	99.24	99.44	99.58	99.61
	FSGS	99.78	99.89	99.92	99.94	99.95	99.96
Proposed Methods	**Comparison Methods**	**CPU Time(%) Mesh Sizes, N 24**	**48**	**72**	**96**	**120**	**144**
FSMAM	**FSAM**	47.54	54.62	54.97	54.77	51.30	52.23
	FSGS	76.30	89.90	90.76	89.26	89.47	89.98
FSMGM	**FSGM**	46.15	54.61	53.47	50.36	51.35	52.46
	FSGS	79.26	91.64	92.11	90.28	91.38	91.86
FSMHM	**FSHM**	46.94	55.26	54.02	50.99	52.06	53.01
	FSGS	80.74	92.40	92.85	91.22	92.24	92.72
HSMAM	**FSAM**	85.25	85.54	86.05	86.63	86.53	87.68
	FSGS	93.33	96.78	97.14	96.82	97.09	97.42
HSMGM	**FSGM**	86.54	86.65	87.20	87.80	87.81	89.18
	FSGS	94.81	97.54	97.83	97.61	97.84	98.15
HSMHM	**FSHM**	87.76	87.63	88.11	88.67	88.75	89.85
	FSGS	95.56	97.90	98.15	97.97	98.18	98.43
QSMAM	**FSAM**	91.80	94.98	96.21	97.37	98.13	98.71
	FSGS	96.30	98.88	99.22	99.38	99.60	99.73
QSMGM	**FSGM**	92.31	95.15	96.43	97.52	98.23	98.78
	FSGS	97.04	99.11	99.39	99.51	99.69	99.79
QSMHM	**FSHM**	93.88	95.79	96.55	97.62	98.31	98.83
	FSGS	97.78	99.28	99.46	99.57	99.73	99.82

Table 4.30: The reduction percentage in iteration steps and CPU time of the proposed iterative methods by using the CD-3CCNC scheme (Problems-2)

Proposed Methods	Comparison Methods	Iterations Steps(%) Mesh Sizes, N					
		24	**48**	**72**	**96**	**120**	**144**
FSMGM	**FSGM**	50.60	53.75	55.94	67.66	68.51	65.88
	FSGS	89.44	94.34	94.80	96.26	96.28	96.19
FSMHM	**FSHM**	51.87	55.00	57.09	68.35	70.07	67.27
	FSGS	90.48	94.90	95.33	96.64	96.76	96.67
HSMAM	**FSAM**	92.56	93.15	95.48	95.55	96.10	95.81
	FSGS	98.01	98.95	99.33	99.34	99.41	99.40
HSMGM	**FSGM**	92.84	93.34	95.54	95.68	96.21	95.95
	FSGS	98.47	99.18	99.47	99.50	99.55	99.55
HSMHM	**FSHM**	92.98	93.51	95.67	95.85	96.38	96.14
	FSGS	98.61	99.26	99.53	99.56	99.61	99.61
QSMAM	**FSAM**	98.88	99.00	99.20	99.42	99.55	99.61
	FSGS	99.70	99.85	99.88	99.91	99.93	99.94
QSMGM	**FSGM**	98.85	99.03	99.23	99.43	99.57	99.60
	FSGS	99.75	99.88	99.91	99.93	99.95	99.96
QSMHM	**FSHM**	98.86	99.05	99.24	99.44	99.58	99.61
	FSGS	99.78	99.89	99.92	99.94	99.95	99.96

Proposed Methods	Comparison Methods	CPU Time(%) Mesh Sizes, N					
		24	**48**	**72**	**96**	**120**	**144**
FSMAM	**FSAM**	45.16	54.38	50.67	50.79	51.00	52.11
	FSGS	75.71	90.32	89.60	88.24	89.41	89.94
FSMGM	**FSGM**	44.44	53.32	50.26	49.58	51.37	52.54
	FSGS	78.57	91.67	91.13	90.05	91.29	91.81
FSMHM	**FSHM**	45.10	54.27	50.44	49.22	51.51	52.66
	FSGS	80.00	92.30	91.82	90.87	92.06	92.60
HSMAM	**FSAM**	83.87	83.47	83.70	86.00	85.97	87.25
	FSGS	92.86	96.49	96.56	96.66	96.97	97.32
HSMGM	**FSGM**	85.19	85.31	85.96	87.74	87.77	88.97
	FSGS	94.29	97.38	97.49	97.58	97.81	98.10
HSMHM	**FSHM**	86.27	85.93	86.85	88.47	88.49	89.55
	FSGS	95.00	97.63	97.83	97.93	98.12	98.37
QSMAM	**FSAM**	90.32	95.42	95.84	96.58	97.78	98.50
	FSGS	95.71	99.03	99.12	99.18	99.52	99.69
QSMGM	**FSGM**	90.74	95.50	95.96	96.65	97.85	98.53
	FSGS	96.43	99.20	99.28	99.34	99.61	99.75
QSMHM	**FSHM**	92.16	95.98	96.28	96.77	97.89	98.56
	FSGS	97.14	99.32	99.39	99.42	99.65	99.78

Table 4.31: The reduction percentage in iteration steps and CPU time of the proposed iterative methods by using the CD-5CCNC scheme (Problems-2)

Proposed Methods	**Comparison Methods**	**Iterations Steps(%)**					
		Mesh Sizes, N					
		24	**48**	**72**	**96**	**120**	**144**
FSMGM	**FSGM**	50.60	53.75	55.94	67.66	68.51	65.88
	FSGS	89.44	94.34	94.80	96.26	96.28	96.19
FSMHM	**FSHM**	51.87	55.00	57.09	68.35	70.07	67.27
	FSGS	90.48	94.90	95.33	96.64	96.76	96.67
HSMAM	**FSAM**	92.56	93.15	95.48	95.55	96.10	95.81
	FSGS	98.01	98.95	99.33	99.34	99.41	99.40
HSMGM	**FSGM**	92.84	93.34	95.54	95.68	96.21	95.95
	FSGS	98.47	99.18	99.47	99.50	99.55	99.55
HSMHM	**FSHM**	92.98	93.51	95.67	95.85	96.38	96.14
	FSGS	98.61	99.26	99.53	99.56	99.61	99.61
QSMAM	**FSAM**	98.88	99.00	99.20	99.42	99.55	99.61
	FSGS	99.70	99.85	99.88	99.91	99.93	99.94
QSMGM	**FSGM**	98.85	99.03	99.23	99.43	99.57	99.60
	FSGS	99.75	99.88	99.91	99.93	99.95	99.96
QSMHM	**FSHM**	98.86	99.05	99.24	99.44	99.58	99.61
	FSGS	99.78	99.89	99.92	99.94	99.95	99.96
Proposed Methods	**Comparison Methods**	**CPU Time(%)**					
		Mesh Sizes, N					
		24	**48**	**72**	**96**	**120**	**144**
FSMAM	**FSAM**	46.03	53.15	53.99	54.55	55.15	52.22
	FSGS	76.06	90.49	89.86	88.77	90.23	89.99
FSMGM	**FSGM**	46.30	53.30	54.34	54.71	55.43	52.31
	FSGS	79.58	92.09	91.63	90.76	91.98	91.79
FSMHM	**FSHM**	47.06	54.18	55.16	55.39	55.98	52.75
	FSGS	80.99	92.77	92.36	91.58	92.72	92.55
HSMAM	**FSAM**	82.54	82.09	84.06	86.64	86.25	87.54
	FSGS	92.25	96.36	96.49	96.70	97.00	97.39
HSMGM	**FSGM**	83.33	82.55	85.46	87.24	86.94	88.28
	FSGS	93.66	97.04	97.33	97.40	97.65	97.98
HSMHM	**FSHM**	84.31	83.54	86.11	87.95	87.61	88.88
	FSGS	94.37	97.40	97.63	97.73	97.95	98.25
QSMAM	**FSAM**	88.89	94.49	95.12	96.52	97.54	98.35
	FSGS	95.07	98.88	98.92	99.14	99.46	99.65
QSMGM	**FSGM**	88.89	94.58	95.18	96.71	97.70	98.45
	FSGS	95.77	99.08	99.12	99.33	99.59	99.73
QSMHM	**FSHM**	90.20	94.94	95.47	96.87	97.79	98.49
	FSGS	96.48	99.20	99.23	99.41	99.63	99.76

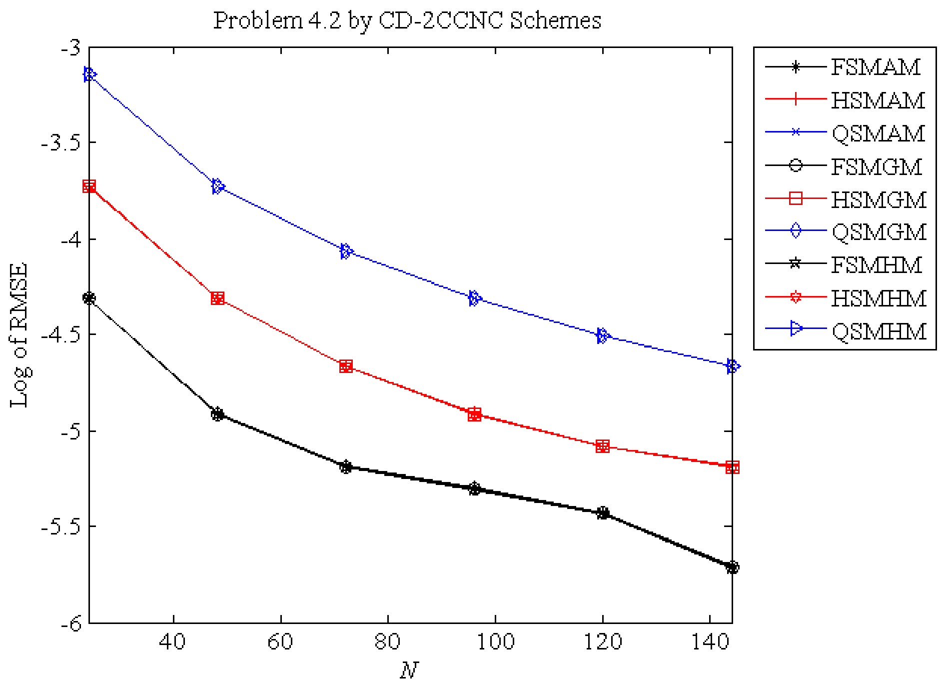

Figure 4.16: Log plot of RMSE for all proposed MWM iterative methods for solving Problems-2 using CD-2CCNC schemes.

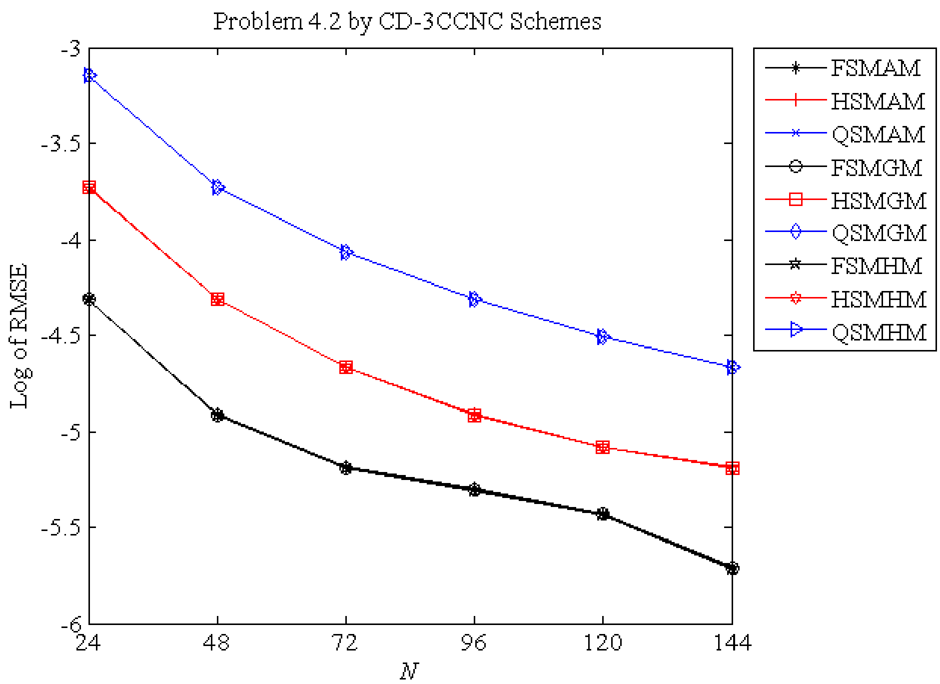

Figure 4.17: Log plot of RMSE for all proposed MWM iterative methods for solving Problems-2 using CD-3CCNC schemes.

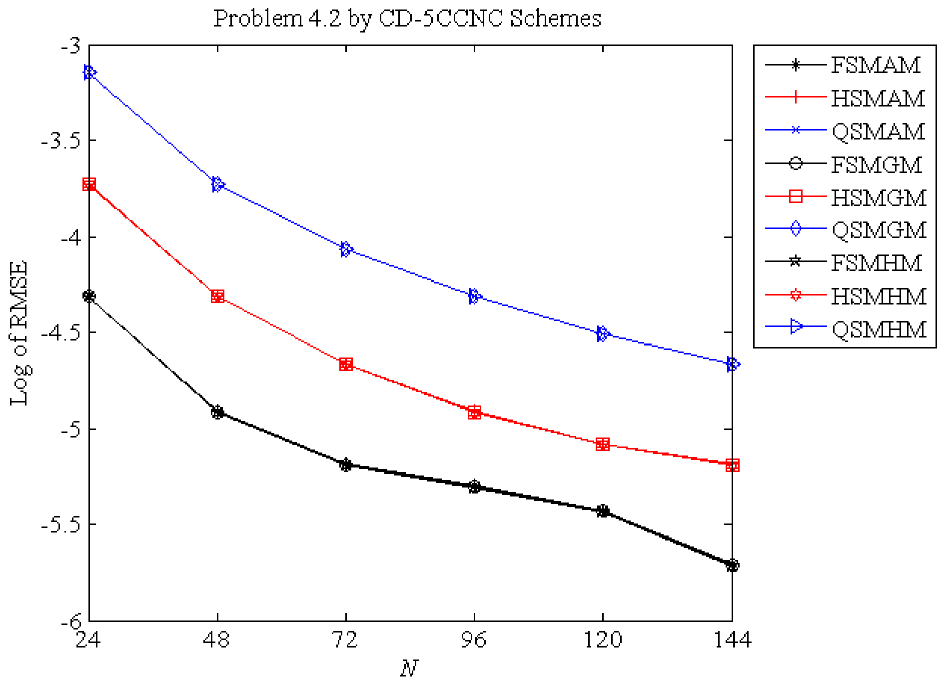

Figure 4.18: Log plot of RMSE for all proposed MWM iterative methods for solving Problems-2 using CD-5CCNC schemes.

4.2 Computational Complexity Analysis

In the field of numerical analysis, computational complexity is a crucial division to measure the performance speed of algorithms regardless the capacity of the computer features device. There are a few methods that can be used to analyse the complexity of the algorithms, such as the arithmetic complexity and the 'θ' notation (Khatim, 2008). In this research, the analysis of computational complexity based on arithmetic complexity is considered. Therefore, to measure the arithmetic complexity of the proposed MWM methods with the respective approximation equations for solving generated linear systems, an estimation of the amount of the computational work has been calculated. The computational cost is estimated based on the number of arithmetic operations performed per iteration. Hence, to proceed the estimation of computational cost for the proposed MWM iterative methods, the values of the elements such as kernels K, function f and weights B_j in the linear system are computed and stored beforehand. The Abbreviations of ADD/SUB, MUL/DIV and SQRT represent the operations of addition/subtraction, multiplication/division and square root respectively.

Based on Algorithms in subsection 3.11.1 incorporated with equations (4.2) and (4.5), the number of arithmetic operations ADD/SUB and MUL/DIV required in computing a value for each nodal point in the solution domain for the proposed FSMAM, HSMAM and QSMAM methods are summarised in Table 4.32. Hence, the total arithmetic operations involved per nodal point for FSMAM, HSMAM and QSMAM methods are $10N + 21$, $5N + 21$, $\frac{5}{2}N + 21$ and $14N+68$, $7N+68$, $\frac{7}{2}N+68$ for equations (4.2) and (4.5), respectively. Table 4.33 shows the summary of total arithmetic operations involved per iteration by the proposed methods for solving equations (4.2) and (4.5), respectively. Based on the analysis of computational complexity, the HSMAM and the QSMAM methods require only $\frac{N}{2} - 1$ and $\frac{N}{4} - 1$ mesh points, respectively.

Meanwhile, Table 4.34 shows the number of arithmetic operations ADD/SUB and MUL/DIV required in computing a value for each nodal point in the solution domain for the proposed FSMGM, HSMGM and QSMGM methods based on Algorithms in subsection 3.11.2 associated with equations (4.2) and (4.5). For Case 1 and Case 2, the total arithmetic operations involved per nodal point for the FSMGM, HSMGM and QSMGM are respectively $10N+31$, $5N+31$, $\frac{5}{2}N+31$ and $14N+67$, $7N+67$, $\frac{7}{2}N+67$ for equations (4.2) and (4.5). Meanwhile, for the Case 3 and the Case 4, the numbers are respectively $10N+32$, $5N+32$, $\frac{5}{2}N+32$ and $14N + 68$, $7N + 68$, $\frac{7}{2}N + 68$ for equations (4.2) and (4.5). Table 4.35 shows the summary of total arithmetic operations involved per iteration by the proposed family of MGM methods for solving equations (4.2) and (4.5), respectively.

Besides, Table 4.36 shows the number of arithmetic operations ADD/SUB and MUL/DIV required in computing a value for each nodal point in the solution domain for the proposed FSMHM, HSMHM and QSMHM methods based on Algorithms in subsections 3.11.3 associated with equations (4.2) and (4.5). Therefore, the total arithmetic operations involved per nodal point for the Case 1 are respectively $10N + 30$, $5N + 30$, $\frac{5}{2}N + 30$ and $14N + 66$, $7N + 66$, $\frac{7}{2}N + 66$ for equations (4.2) and (4.5). For the Case 2 and the Case 3 the numbers are respectively $10N + 34$, $5N + 34$, $\frac{5}{2}N + 34$ and $14N + 70$, $7N + 70$, $\frac{7}{2}N + 70$ for equations (4.2) and (4.5). Meanwhile, for the Case 4 and the Case 5, the numbers are respectively $10N + 41$, $5N + 41$, $\frac{5}{2}N + 41$ and $14N + 77$, $7N + 77$, $\frac{7}{2}N + 77$ for equations (4.2) and (4.5). Table 4.37 shows the summary of total arithmetic operations involved per iteration by the proposed family of MHM methods for solving equations (4.2) and (4.5), respectively.

Based on the analysis of computational complexity, the proposed HSMWM and the QSMWM methods require only $\frac{N}{2}-1$ and $\frac{N}{4}-1$ mesh points, respectively. However, these methods are required additional eight arithmetic operations to calculate a remaining nodal point after convergence by using second order Lagrange interpolation technique. Therefore, the total arithmetic operations per iteration after convergence are summarised in Tables 4.33, 4.35 and 4.37.

Table 4.32: Number of arithmetic operations required for computing the result in a nodal point for the FSMAM, HSMAM and QSMAM methods

Methods	Second Order IDE		
	Arithmetic Operations Per Node		
	ADD/SUB	MUL/DIV	SQRT
FSMAM	$2N+11$	$8N+21$	–
HSMAM	$N+11$	$4N+21$	–
QSMAM	$\frac{N}{2}$+11	$2N+21$	–
Methods	**Fourth Order IDE**		
	Arithmetic Operations Per Node		
	ADD/SUB	**MUL/DIV**	**SQRT**
FSMAM	$2N+21$	$12N+47$	–
HSMAM	$N+21$	$6N+47$	–
QSMAM	$\frac{N}{2}$+21	$3N+47$	–

Table 4.33: Total arithmetic operations required for computing the result for the FSMAM, HSMAM and QSMAM methods

Methods	Second Order IDE	
	Total Arithmetic Operations	
	Per Iteration	After Convergence
FSMAM	$10N^2+21N-32$	–
HSMAM	$\frac{5}{2}N^2+\frac{21}{2}N-32$	$4N$
QSMAM	$\frac{5}{8}N^2+\frac{21}{4}N-32$	$6N$
Methods	**Fourth Order IDE**	
	Total Arithmetic Operations	
	Per Iteration	**After Convergence**
FSMAM	$14N^2+54N-68$	–
HSMAM	$\frac{7}{2}N^2+27N-68$	$4N$
QSMAM	$\frac{7}{8}N^2+\frac{27}{2}N-68$	$6N$

Table 4.34: Number of arithmetic operations required for computing the result in a nodal point for the FSMGM, HSMGM and QSMGM methods

Methods	**Second Order IDE**		
	Arithmetic Operations Per Node		
	ADD/SUB	**MUL/DIV**	**SQRT**
FSMGM			
Case 1	$2N + 10$	$8N + 21$	1
Case 2	$2N + 10$	$8N + 21$	1
Case 3	$2N + 11$	$8N + 21$	1
Case 4	$2N + 11$	$8N + 21$	1
HSMGM			
Case 1	$N + 10$	$4N + 21$	1
Case 2	$N + 10$	$4N + 21$	1
Case 3	$N + 11$	$4N + 21$	1
Case 4	$N + 11$	$4N + 21$	1
QSMGM			
Case 1	$\frac{N}{2} + 10$	$2N + 21$	1
Case 2	$\frac{N}{2} + 10$	$2N + 21$	1
Case 3	$\frac{N}{2} + 11$	$2N + 21$	1
Case 4	$\frac{N}{2} + 11$	$2N + 21$	1
Methods	**Fourth Order IDE**		
	Arithmetic Operations Per Node		
	ADD/SUB	**MUL/DIV**	**SQRT**
FSMGM			
Case 1	$2N + 20$	$12N + 47$	1
Case 2	$2N + 20$	$12N + 47$	1
Case 3	$2N + 21$	$12N + 47$	1
Case 4	$2N + 21$	$12N + 47$	1
HSMGM			
Case 1	$N + 20$	$6N + 47$	1
Case 2	$N + 20$	$6N + 47$	1
Case 3	$N + 21$	$6N + 47$	1
Case 4	$N + 21$	$6N + 47$	1
QSMGM			
Case 1	$\frac{N}{2} + 20$	$3N + 47$	1
Case 2	$\frac{N}{2} + 20$	$3N + 47$	1
Case 3	$\frac{N}{2} + 21$	$3N + 47$	1
Case 4	$\frac{N}{2} + 21$	$3N + 47$	1

Table 4.35: Total arithmetic operations required for computing the result for the FSMGM, HSMGM and QSMGM methods

	Methods	Second order IDE	
		Total Arithmetic Operations	
		Per Iteration	**After Convergence**
	FSMGM	$10N^2 + 21N - 31$	$-$
Case 1	**HSMGM**	$\frac{5}{2}N^2 + \frac{21}{2}N - 31$	$4N$
	QSMGM	$\frac{5}{8}N^2 + \frac{21}{4}N - 31$	$6N$
	FSMGM	$10N^2 + 21N - 31$	$-$
Case 2	**HSMGM**	$\frac{5}{2}N^2 + \frac{21}{2}N - 31$	$4N$
	QSMGM	$\frac{5}{8}N^2 + \frac{21}{4}N - 31$	$6N$
	FSMGM	$10N^2 + 22N - 32$	$-$
Case 3	**HSMGM**	$\frac{5}{2}N^2 + 11N - 32$	$4N$
	QSMGM	$\frac{5}{8}N^2 + \frac{11}{2}N - 32$	$6N$
	FSMGM	$10N^2 + 22N - 32$	$-$
Case 4	**HSMGM**	$\frac{5}{2}N^2 + 11N - 32$	$4N$
	QSMGM	$\frac{5}{8}N^2 + \frac{11}{2}N - 32$	$6N$

	Methods	Fourth Order IDE	
		Total Arithmetic Operations	
		Per Iteration	**After Convergence**
	FSMGM	$14N^2 + 53N - 67$	$-$
Case 1	**HSMGM**	$\frac{7}{2}N^2 + \frac{53}{2}N - 67$	$4N$
	QSMGM	$\frac{7}{8}N^2 + \frac{53}{4}N - 67$	$6N$
	FSMGM	$14N^2 + 53N - 67$	$-$
Case 2	**HSMGM**	$\frac{7}{2}N^2 + \frac{53}{2}N - 67$	$4N$
	QSMGM	$\frac{7}{8}N^2 + \frac{53}{4}N - 67$	$6N$
	FSMGM	$14N^2 + 54N - 68$	$-$
Case 3	**HSMGM**	$\frac{7}{2}N^2 + 27N - 68$	$4N$
	QSMGM	$\frac{7}{8}N^2 + \frac{27}{2}N - 68$	$6N$
	FSMGM	$14N^2 + 54N - 68$	$-$
Case 4	**HSMGM**	$\frac{7}{2}N^2 + 27N - 68$	$4N$
	QSMGM	$\frac{7}{8}N^2 + \frac{27}{2}N - 68$	$6N$

Table 4.36: Number of arithmetic operations required for computing the result in a nodal point for the FSMHM, HSMHM and QSMHM methods

Methods	**Second Order IDE**		
	Arithmetic Operations Per Node		
	ADD/SUB	**MUL/DIV**	**SQRT**
FSMHM			
Case 1	$2N+10$	$8N+20$	–
Case 2	$2N+11$	$8N+23$	–
Case 3	$2N+11$	$8N+23$	–
Case 4	$2N+13$	$8N+27$	1
Case 5	$2N+13$	$8N+27$	1
HSMHM			
Case 1	$N+10$	$4N+20$	–
Case 2	$N+11$	$4N+23$	–
Case 3	$N+11$	$4N+23$	–
Case 4	$N+13$	$4N+27$	1
Case 5	$N+13$	$4N+27$	1
QSMHM			
Case 1	$\frac{N}{2}+10$	$2N+20$	–
Case 2	$\frac{N}{2}+11$	$2N+23$	–
Case 3	$\frac{N}{2}+11$	$2N+23$	–
Case 4	$\frac{N}{2}+13$	$2N+27$	1
Case 5	$\frac{N}{2}+13$	$2N+27$	1

Methods	Fourth Order IDE		
	Arithmetic Operations Per Node		
	ADD/SUB	MUL/DIV	SQRT
FSMHM			
Case 1	$2N + 20$	$12N + 46$	–
Case 2	$2N + 21$	$12N + 49$	–
Case 3	$2N + 21$	$12N + 49$	–
Case 4	$2N + 23$	$12N + 53$	1
Case 5	$2N + 23$	$12N + 53$	1
HSMHM			
Case 1	$N + 20$	$6N + 46$	–
Case 2	$N + 21$	$6N + 49$	–
Case 3	$N + 21$	$6N + 49$	–
Case 4	$N + 23$	$6N + 53$	1
Case 5	$N + 23$	$6N + 53$	1
QSMHM			
Case 1	$\frac{N}{2} + 20$	$3N + 46$	–
Case 2	$\frac{N}{2} + 21$	$3N + 49$	–
Case 3	$\frac{N}{2} + 21$	$3N + 53$	–
Case 4	$\frac{N}{2} + 23$	$3N + 53$	1
Case 5	$\frac{N}{2} + 23$	$8N + 53$	1

4.3 Concluding Remarks

This chapter, the effectiveness of the proposed MWM iterative methods associated with three different combination sets of CD-CCNC schemes for solving second and fourth order FIDEs have been investigated via Problem-1 and Problem-2. A vigorous analysis of the numerical results show that all the proposed MWM iterative methods are superior than the family of GS and WM methods for all the generated CD-2CCNC, CD-3CNCC and CD-5CNCC dense systems. The results of numerical simulations show that the families of MAM, MGM and MHM methods are highly effective iterative methods. Amongst, the families of MAM, MGM and MHM methods, the QSMAM, the QSMGM and the QSMHM are the best iterative methods for solving both problems with least number of iterations and CPU time. In the aspect of accuracy, the combination sets of CD-3CCNC and CD-5CCNC are more accurate and efficient than the CD-2CCNC approximation schemes.

Table 4.37: Total arithmetic operations required for computing the result for the FSMHM, HSMHM and QSMHM methods

	Methods	**Second order IDE**	
		Total Arithmetic Operations	
		Per Iteration	**After Convergence**
Case 1	**FSMHM**	$10N^2 + 20N - 30$	–
	HSMHM	$\frac{5}{2}N^2 + 10N - 30$	$4N$
	QSMHM	$\frac{5}{8}N^2 + 5N - 30$	$6N$
Case 2	**FSMHM**	$10N^2 + 24N - 34$	–
	HSMHM	$\frac{5}{2}N^2 + 12N - 34$	$4N$
	QSMHM	$\frac{5}{8}N^2 + 6N - 34$	$6N$
Case 3	**FSMHM**	$10N^2 + 24N - 34$	–
	HSMHM	$\frac{5}{2}N^2 + 12N - 34$	$4N$
	QSMHM	$\frac{5}{8}N^2 + 6N - 34$	$6N$
Case 4	**FSMHM**	$10N^2 + 30N - 40$	–
	HSMHM	$\frac{5}{2}N^2 + 15N - 40$	$4N$
	QSMHM	$\frac{5}{8}N^2 + \frac{15}{2}N - 40$	$6N$
Case 5	**FSMHM**	$10N^2 + 30N - 40$	–
	HSMHM	$\frac{5}{2}N^2 + 15N - 40$	$4N$
	QSMHM	$\frac{5}{8}N^2 + \frac{15}{2}N - 40$	$6N$
	Methods	**Fourth Order IDE**	
		Total Arithmetic Operations	
		Per Iteration	**After Convergence**
Case 1	**FSMHM**	$14N^2 + 52N - 66$	–
	HSMHM	$\frac{7}{2}N^2 + 26N - 66$	$4N$
	QSMHM	$\frac{7}{8}N^2 + 13N - 66$	$6N$
Case 2	**FSMHM**	$14N^2 + 56N - 70$	–
	HSMHM	$\frac{7}{2}N^2 + 28N - 70$	$4N$
	QSMHM	$\frac{7}{8}N^2 + 14N - 70$	$6N$
Case 3	**FSMHM**	$14N^2 + 56N - 70$	–
	HSMHM	$\frac{7}{2}N^2 + 28N - 70$	$4N$
	QSMHM	$\frac{7}{8}N^2 + 14N - 70$	$6N$
Case 4	**FSMHM**	$14N^2 + 62N - 76$	–
	HSMHM	$\frac{7}{2}N^2 + 31N - 76$	$4N$
	QSMHM	$\frac{7}{8}N^2 + \frac{31}{2}N - 76$	$6N$
Case 5	**FSMHM**	$14N^2 + 62N - 76$	–
	HSMHM	$\frac{7}{2}N^2 + 31N - 76$	$4N$
	QSMHM	$\frac{7}{8}N^2 + \frac{31}{2}N - 76$	$6N$

CHAPTER 5

Summary and Future Research Directions

5.1 Summary

The main objectives of this research are to introduce and analyse the performance of the proposed Modified Weighted Mean iterative methods in solving large dense linear systems arise from linear FIDEs. The existing conventional family of Gauss Seidel and the families of Weighted Mean iterative methods are also considered for the comparative analysis with the proposed methods. Therefore, the proposed MWM (i.e. FSMAM, HSMAM, QSMAM, FSMGM, HSMGM, QSMGM, FSMHM, HSMAM, QSMHM) methods are developed and investigated along with the strong convergence theorem and proofs. The effectiveness and efficiency of the proposed methods are demonstrated with some vigorous analysis such as analysis of percentage reduction in terms of number of iterations and computational time, and the analysis of computational complexity.

5.2 Achievement and Contribution of the Research

In this research, nine new variant of MWM methods are developed and analysed from the existing standard WM methods. The new proposed MWM methods have been successfully developed and implemented to solve second and fourth order linear FIDEs associated with three different combination sets of FD-CCNC schemes (i.e. CD-2CCNC, CD-3CCNC and CD-5CCNC schemes) which are applied to generate a system of linear equations. All the objectives of the thesis have successfully accomplished as per specified in Section 1.4. The outcome of the objectives on this thesis are as follows

(i) The Full-, Half- and Quarter-Sweep FD-CCNC (i.e. CD-2CCNC, CD-3CCNC and CD-5CCNC) approximation equations of second and fourth order FIDEs are successfully derived and implemented. Based on the investigations, the Full-, Half- and Quarter-Sweep FD-CCNC schemes on second and fourth order FIDEs lead to dense linear systems of normal equations (refer to Sections 3.3.4)

(ii) The Full-, Half- and Quarter-Sweep CD-3CCNC and CD-5CCNC approximation schemes yield high precision results compared to the Full-, Half- and Quarter-sweep CD-2CCNC scheme.(refer to Figures 4.7 to 4.9)

(iii) The application of the half- and quarter-sweep iteration reduce the order of dense coefficient matrix from $N-1$ to $\frac{N}{2}-1$ and from $\frac{N}{2}-1$ to $\frac{N}{4}-1$, respectively. Finally, it reduces the computational complexity of the proposed MWM iterative methods during the iteration processes for solving the second and fourth order linear FIDEs.

(iv) The considered iterative methods, the existing families of GS and WM, and the proposed families of MWM, are successfully formulated and implemented on the generated linear systems as derived in Chapter 3. In addition, the half- and quarter-sweep iteration techniques have been successfully applied to all the considered iterative methods. (Refer to Section 3.11).

(v) All the families of the proposed MWM (i.e. FSMAM, HSMAM, QSMAM, FSMGM, HSMGM, QSMGM, FSMHM, HSMAM, QSMHM) iterative methods have been presented together with a necessary convergence theorems with proofs. (refer Chapter 3).

(vi) The numerical performances of the proposed MWM iterative methods with the corresponding Full-, Half- and Quarter-Sweep FD-CCNC approximation schemes for solving the second and fourth order linear FIDEs show that the proposed MWM iterative methods perform exceptionally well for dense linear systems.

(vii) The analysis of computational complexity (refer to Tables 4.32 to 4.37) for the proposed MWM reveals that the implementation of the Quarter-Sweep iteration has reduced a number of iterations and CPU time as compared to the standard or Full- and Half-Sweep iterations.

(viii) The numerical results also disclose that the CD-3CCNC and CD-5CCNC discretisation schemes require more execution time for solving the generated linear systems compared to the CD-2CCNC scheme. The reason is that the CD-3CCNC and CD-5CCNC schemes consist of highly mathematical complexity than the CD-2CCNC scheme.

(ix) Overall, the findings reveal that the proposed MWM iterative methods (i.e. FSMAM, HSMAM, QSMAM, FSMGM, HSMGM, QSMGM, FSMHM, HSMAM, QSMHM) with their corresponding FD-CCNC approximation equations perform remarkably in terms of number of iterations and CPU time compared to the existing conventional family of GS and the families of the WM iterative methods for the second and fourth order FIDEs. Amongst the proposed families of MWM iterative methods, the QSMWM (i.e. QSMAM, QSMGM and QSMHM) iterative methods are superior to the other proposed MWM methods. Nevertheless, among the QSMWM methods, the QSMHM has performed outstandingly in terms of a number of iterations and CPU time compared to the QSMAM and QSMGM iterative methods.

5.3 Recommendation for Future Researches

A list of future work includes

(i) Investigate the performance of all the proposed MWM methods for solving other types of IDEs as mentioned in Section 1.2.1.

(ii) Investigate the performance of all the proposed MWM methods for solving 2D, 3D and n-D IDEs problems.

(iii) Broaden the current research to the nonlinear, fractional, impulsive, integro-partial derivative cases.

(iv) Analysis of high-order numerical schemes under finite elements methods.

(v) Investigate the application of the proposed MWM iterative methods to the real industrial mathematical models such as the PDE Oil reservoir simulation, mathematical models in epidemiology and population biology etc.

Bibliography

[1] A. A. Badr, "Integro-differential equation with Cauchy kernel," *Journal of Computational and Applied Mathematics*, vol. 134(1-2), pp. 191199, 2001.

[2] A. Arikoglu and I. Ozkol, "Solution of boundary value problems for integro-differential equations by using differential transform method", *Applied Mathematics and Computation*, vol. 168, pp.1145-1158, 2005.

[3] A. Avudainayagam, and C. Vani, "WaweletGalerkin method for integro-differential equations," *Applied Numerical Mathematics*, vol. 32, pp. 247254, 2000.

[4] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Philadelphia: Society for Industrial and Applied Mathematics, 1994.

[5] A. Cheng and D. T. Cheng, "Heritage and early history of the boundary element method," *Engineering Analysis with Boundary Elements*, vol. 29, vol. 268-302, 2005.

[6] A. D. Polyanin and A. V. Manzhirov, *Handbook of Integral Equations*, Florida: CRC Press LLC, 1998.

[7] A. F. Soliman, A. M. A. El-Asyed, M. S. El-Azab, "On The Numerical Solution of Partial integro-differential equations," *Math. Sci. Lett.*, vol. 1(1), pp. 71-80, 2012.

[8] A. Hadjidimos, "Successive overrelaxation (SOR) and related methods," *Journal of Computational and Applied Mathematics*, vol. 123, pp. 177-199, 2000.

[9] A. I. Fedotov, "Quadrature-difference methods for solving linear and nonlinear singular integro-differential equations," *Nonlinear Analysis*, vol. 71, pp. 303-308, 2009.

[10] A. Karamete and M. Sezer, "A Taylor Collocation Method for the Solution of Linear Integro-Differential Equations," *International Journal of Computer Mathematics*, vol. 79(9), pp. 987-1000, 2002.

[11] A. M. Saeed and N. H. M. Ali, "Preconditioned Modified Explicit Decoupled Group method in the solution of elliptic PDEs," *Applied Mathematical Sciences*, vol. 4(24), pp. 1165-1181, 2010.

[12] A. M. Wazwaz, "A reliable algorithm for solving boundary value problems for higher-order integro-differential equation," *Applied Mathematics and Computation*, vol. 118, pp. 327-342, 2001.

[13] A. M. Wazwaz, *Linear and Nonlinear Integral Equations*, New York: Springer Heidelberg, 2011.

[14] A. M. Wazwaz, S. M. El-Sayed, "A new modification of the Adomian decomposition method for linear and nonlinear operators," *Applied Mathematics and Computation*, vol. 122, pp. 393-404, 2001.

[15] A. Mohsen and M. El-Gamel, "A Sinc-Collocation method for the linear Fredholm integro-differential equations," *Zeitschrift fr Angewandte Mathematik und Physik (ZAMP)*, vol. 58(3), pp. 380-390, 2007.

[16] A. Quarteroni, R. Sacco and F. Saleri, *Numerical Mathematics*, New York: Springer-Verlag, 2000.

[17] A. R. Abdullah, "The four point Explicit Decoupled Group (EDG) method: A fast Poisson solver," *International Journal of Computer Mathematics*, vol. 38(1-2), pp. 61-70, 1991.

[18] A. Zerarka, A. Soukeur and H. Saidi, "On Some Volterra and Fredholm Problems via the Unified Integro-differential Quadrature Method," *ISRN Computational Mathematics*, vol.(2012), pp. 1-5, 2012.

[19] B. Q. Leyva, "On the numerical solution of the integro-differential equation of the point kinetics of nuclear reactors," *Annals of Nuclear Energy*, vol. 36(8), pp. 12801284, 2009.

[20] C. Brezinski & L. Wuytack, *Numerical Analysis: Historical Developments in the 20th Century*, Elsevier, 2012.

[21] D. Chandler, "The norm of the Schur product operation," *Numerische Mathematik*, vol. 4.1, pp. 343-344, 1962.

[22] D. J. Evans and W. S. Yousif, "The Modified Alternating Group Explicit (M.A.G.E.) method," *Applied Mathematical Modelling*, vol. 12(3), pp. 262-267, 1988.

[23] D. J. Evans, "The Alternating Group Explicit (AGE) matrix iterative method," *Applied Mathematical Modelling*, vol. 11(4), pp. 256-263, 1987.

[24] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, 1971.

[25] D. M. Young, *Iterative Solution of Large Linear Systems*, New York: Academic Press, 1971.

[26] E. Aruchunan, M. S. Muthuvalu and J. Sulaiman, "Quarter-Sweep Iteration Concept on Conjugate Gradient Normal Residual Method via Second Order Quadrature-Finite Difference Schemes for Solving Fredholm Integro-Differential Equation," *Sains Malaysiana*, vol. 44, pp. 139-146, 2015.

[27] E. Aruchunan, M. S. Muthuvalu, & J. Sulaiman, "Quarter-sweep iteration concept on conjugate gradient normal residual method via second order quadrature-finite difference schemes for solving fredholm integro-differential equations," *Sains Malaysiana*, vol. 44(1), pp. 139-146, 2015.

[28] E. Aruchunan, M. S. Muthuvalu, J. Sulaiman, W. S. Koh, & K. Md Akhir, "An iterative solution for second order linear Fredholm integro-differential equations," *Malaysian Journal of Mathematical Sciences*, vol. 8(2), pp. 157-170, 2014.

[29] E. Galligani and V. Ruggiero, "Computation of minimal eigenpair in the large sparse generalized eigen-problem using vector computers," *Advances in Parallel Computing*, vol. 4, pp. 193-201, 1992.

[30] E. Galligani and V. Ruggiero, "The two-stage Arithmetic Mean method," *Applied Mathematics and Computation*, vol. 85(2-3), pp. 245-264, 1997.

[31] E. Galligani, "The Arithmetic Mean method for solving systems of nonlinear equations in finite differences," *Applied Mathematics and Computation*, vol. 181(1), pp. 579-597, 2006.

[32] F. Brauer, "Asymptotic stability of a class of integro-differential equations," *Journal of Differential Equations*, vol. 28(2), pp. 180-188, 1978.

[33] F. Cai, J. Xiao and Z. H. Xiang, "Block SOR two-stage iterative methods for solution of symmetric positive definite linear systems," *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, August 20-22, 2010, Chengdu, China.*, 2010.

[34] G. H. Golub and M. L. Overton, "Convergence of a two-stage Richardson iterative procedure for solving systems of linear equations," In Watson, G. A. (ed.), Lecture Notes in Mathematics 912: Numerical Analysis, Berlin Heidelberg: Springer-Verlag. pp. 125-139, 1982.

[35] G. H. Golub and M. L. Overton, "The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems," *Numerische Mathematik*, vol. 53(5), pp. 571-593, 1988.

[36] G. M. Vainikko, "Convergence of quadrature-difference methods for linear integro-differential equations," *USSR Computational Mathematics and Mathematical Physics*, vol. 11(3), pp. 292-301, 1971.

[37] I. Galligani and V. Ruggiero, "The Arithmetic Mean method for solving essentially positive systems on a vector computer," *International Journal of Computer Mathematics*, vol. 32(1-2), pp. 113-121, 1990.

[38] I. Galligani, "A splitting method for large sparse linear systems on a multivector computer," *In Spedicato, E. (ed.). Computer Algorithms for Solving Linear Algebraic Equations - The State of the Art, Berlin: Springer-Verlag*, 1991.

[39] J. A. Sparenberg, T. C. Braakman and C. W. Benthem, "Discussion of a Wiener-Hopf type integro-differential equation," *Applied Scientific Research*, vol. 6(1), pp. 312322, 1957.

[40] J. Biazar, H. Ghazvini, and M. Eslami, "He's homotopy perturbation method for systems of integro-differential equations" *Chaos, Solitons & Fractals.*, vol. 39(3), pp. 1253-1258, 2009.

[41] J. H. He, "A coupling method of homotopy technique and perturbation technique for nonlinear problems," *International Journal Non-Linear Mechanics*, vol. 35(1), pp. 37-43, 2000.

[42] J. H. Yun and S. W. Kim, "Convergence of two-stage iterative methods using incomplete factorization," *Journal of Computational and Applied Mathematics*, vol. 166(2), pp. 565-580, 2004.

[43] J. I. Fujii and M. Fujii, "On geometric and harmonic means of positive operators," *Math. Japonica*, vol. 24(2), pp. 203207, 1979.

[44] J. M. Ortega, (1990). *Numerical analysis*: a second course, vol. 3, Siam.

[45] J. Morchalo, "On two point boundary value problem for integro-differential equation of second order," *Fasciculi Mathematici*, vol. 9, pp. 51-56, 1975.

[46] J. P. Bouchaud and A. Georges, "Anomalous diffusion in disordered media: Statistical mechnics, models and physical applications," *Physics reports*, vol. 195(4-5), 1990.

[47] J. Sulaiman, A.Saudi, M. H. Abdullah, M. K. Hasan and M. Othman, "Quarter-Sweep Arithmetic Mean algorithm for water quality model," *Proceedings of the International Symposium on Information Technology, August 26-29, 2008, Kuala Lumpur, Malaysia*, 2008.

[48] J. Sulaiman, M. H. Abdullah, A. Saudi, M. K. Hasan and M. Othman, "A numerical simulation on water quality model using Quarter-Sweep Geometric Mean (QSGM) iterative method," *Proceedings of the 2nd International Conference on Science and Technology: Applications in Industry and Education, 12-13 December, Penang, Malaysia*, pp. 1853-1859, 2008.

[49] J. Sulaiman, M. H. Abdullah, A. Saudi, M. K. Hasan and M. Othman. "A numerical simulation on water quality model using Quarter-Sweep Geometric Mean (QSGM) iterative method," *Proceedings of the 2nd International Conference on Science and Technology Applications in Industry and Education, Penang, Malaysia 12-13 December*, 2008.

[50] J. Sulaiman, M. K. Hasan, M. Othman and Z. Yaacob, "Quarter-Sweep Arithmetic Mean iterative algorithm to solve fourth-order parabolic equations," *Proceedings of the 2009 International Conference on Electrical Engineering and Informatics, August 5-7, 2009, Selangor, Malaysia*, 2009.

[51] J. Sulaiman, M. Othman and M. K. Hasan, "*Quarter-Sweep Arithmetic Mean iterative method to solve elliptic partial differential equations. Proceedings of the 4th International Conference on Mathematics and Statistics, August 13-15, 2009, Lampung, Indonesia*, 2009.

[52] J. Sulaiman, M. Othman and M. K. Hasan, "A fourth-order finite difference solver for Poisson's equations via the Half-Sweep Arithmetic Mean (HSAM) method," *Proceedings of the First IMT-GT Regional Conference on Mathematics, Statistics and Their Applications, June 13-16, 2005, Medan, Indonesia*, 2005.

[53] J. Sulaiman, M. Othman and M. K. Hasan, "A new Quarter Sweep Arithmetic Mean (QSAM) method to solve diffusion equations," *Chamchuri Journal of Mathematics*, vol. 1(2), pp. 93-103, 2009.

[54] J. Sulaiman, M. Othman and M. K. Hasan, "Quarter-Sweep Arithmetic Mean (QSAM) iterative method using linear element approximation for 1D diffusion equations," *Proceedings of the 5th International Conference on Numerical Analysis in Engineering, May 18-19, 2007, West Sumatera, Indonesia.*, 2007.

[55] J. Sulaiman, M. Othman, and M. K. Hasan, "A new Half-Sweep Arithmetic Mean (HSAM) algorithm for two-point boundary value problems," *Proceedings of the International Conference on Statistics and Mathematics and Its Application in the Development of Science and Technology, 4-6 October, Bandung, Indonesia* , 2004.

[56] J. Sulaiman, M. Othman, N. Yaacob and M. K. Hasan, "Half-sweep Geometric Mean (HSGM) method using fourth-order finite difference scheme for two-point boundary problems," *Proceedings of the First International Conference on Mathematics and Statistics, Bandung, Indonesia 19-21 June*, 2006.

[57] J. Sulaiman, M. Othman, N. Yaacob and M. K. Hasan, "Half-Sweep Geometric Mean (HSGM) method using fourth-order finite difference scheme for two-point boundary problems," *Proceedings of the First International Conference on Mathematics and Statistics, June 19-21, 2006, Bandung, Indonesia*, 2006.

[58] J. Sulaiman, M. Othman, N. Yaacob and M. K. Hasan, "Half-Sweep Geometric Mean (HSGM) method using fourth-order finite difference scheme for two-point boundary problems," *Proceedings of the First International Conference on Mathematics and Statistics, June 19-21, 2006, Bandung, Indonesia.* 25-33.2006.

[59] J. Zhao and R. M. Corless, "Compact finite difference method for integro-differential equations," *Applied Mathematics and Computation*, vol. 177, pp. 271-288, 2006.

[60] K. *Styś* and T. *Styś*, "A higher-order finite difference method for solving a system of integro-differential equations," *Journal of Computational and Applied Mathematics*, vol. 126, pp. 33-46, 2000.

[61] K. Atkinson and W. Han, *Theoretical Numerical Analysis: A Functional Analysis Framework (2nd edition)*", New York: Springer Science Business Media, 2005.

[62] K. Maleknejad and F. Mirzaee, "Numerical solution of integro-differential equations by using rationalized Haar functions method," *Kybernetes Int. J. Syst. Math.*, vol. 35, pp. 1735-1744, 2006.

[63] K. Maleknejad and Y. Mahmoudi, "Taylor polynomial solution of high-order nonlinear Volterra-Fredholm integro-differential equations," *Applied Mathematics and Computation*, vol. 145, pp. 641-653, 2003.

[64] K. Maleknejad, and Y. Mahmoudi, "Numerical solution of linear Fredholm integral equation by using hybrid Taylor and block-pulse functions," *Applied Mathematics and Computation*, vol. 149, pp. 799-806, 2004.

[65] K. Maleknejad, M. Rabbani, N. Aghazadeh, & M. Karami, "A wavelet PetrovGalerkin method for solving integro-differential equations," *International Journal of Computer Mathematics*, vol. 86(9), pp. 1572-1590, 2009.

[66] K. Maleknejad, N. Aghazadeh, and R. Mollapourasl, "Numerical solution of Fredholm integral equation of the first kind with collocation method and estimation of error bound," *Applied Mathematics and Computation*, vol. 179(1), pp. 352-359, 2006.

[67] L. M. Delves and J. L. Mohamed, *Computational methods for integral equations*, CUP Archive, 1988.

[68] L. M. Delves and J. L. Mohamed, *Computational Methods for Integral Equations*, London: Cambridge University Press, 1985.

[69] M. A. Abdou, M. A. Elsayed, "Fractional integro differential equation and spectral relationships," IJCST vol. 5, Issue 1, Jan-March, 2014.

[70] M. Benzi and T. Dayar, "The arithmetic mean method for finding the stationary vector of Markov chains," *Parallel Algorithms and Applications*, vol. 6(1), pp. 25-37, 1995.

[71] M. C. Mariani, M. Salas, and A. Vivas, "Solutions to integro-differential systems arising in Hydrodynamic models for charged transport in semiconductors"' *Nonlinear Analysis: Real World Applications*, vol. 11(5), pp. 3912-3922, 2010.

[72] M. C. Van Beurden, "Integro-differential equations for electromagnetic scattering : analysis and computation for objects with electric contrast," Ph.D. Thesis, Technische Universiteit Eindhoven, 2003.

[73] M. Dehghan and A. Saadatmandi, "Chebyshev finite difference method for Fredholm integro-differential equation," *International Journal of Computer Mathematics*, vol. 85(1), pp. 123-130, 2008.

[74] M. K. Hasan, J.Sulaiman, S. A. A. Karim and M. Othman, "Development of some numerical methods applying complexity reduction approach for solving scientific problem," *Journal of Applied Sciences*, vol. 11(7), pp. 1255-1260, 2011.

[75] M. K. Hasan, M.Othman, Z. Abbas, J. Sulaiman, and F. Ahmad, "Parallel solution of High Speed Low Order FDTD on 2D free space wave propagation," *In Gervasi, O. and Gavrilova, M. L. (eds.). Lecture Notes in Computer Science 4706: Computational Science and Its Applications, Berlin Heidelberg: Springer-Verlag*, 2007.

[76] M. K. Hasan, M.Othman, Z. Abbas, J. Sulaiman, and J. M. Desa, "New algorithms for time-domain Maxwell equations," *International Journal of Mathematics and Computer Science*, vol. 1(3), pp. 271-288, 2006.

[77] M. K. Hasan, Z. Yaacob, J. Sulaiman, A. Saudi, M. Othman and M. H. Abdullah, "Implementation of the quarter-sweep approach to arithmetic and geometric means for the solution of water quality model," *UniKL Research Journal*, 1(1), pp. 173-185, 2008.

[78] M. K. Hasan, Z.Yaacob, J. Sulaiman, A. Saudi, M. Othman and M. H. Abdullah. Implementation of the quarter-sweep approach to arithmetic and geometric means for the solution of water quality model. *UniKL Research Journal*, vol. 1(1), pp. 173-185, 2008.

[79] M. Khalid, M. Sultana and F. Zaidi, "Numerical Solution of Fourth Order Integro-differential Boundary Value Problems by Optimal Homotopy Asymptotic Method," *International Journal of Computer Applications*, vol. 106, pp. 38-44, 2014.

[80] M. Othman and A. R. Abdullah, "An efficient Four Points Modified Explicit Group Poisson solver," *International Journal of Computer Mathematics*, vol. 76(2), pp. 203-217, 2000.

[81] M. Othman and A. R. Abdullah, "An efficient parallel quarter-sweep point iterative algorithm for solving Poisson equation on SMP parallel computer," *Pertanika Journal of Science & Technology*, vol. 8(2), pp. 161-174, 2000.

[82] M. Othman, A. R. Abdullah D. J. Evans, "A parallel Four Points Modified Explicit Group algorithm on shared memory multiprocessors," *Parallel Algorithms and Applications*, vol. 19(1), pp. 1-9, 2004.

[83] M. Rahman, Z. Jackiewicz and B. D. Welfert, "Stochastic approximations of perturbed Fredholm Volterra Integro-differential equation arising in mathematical neurosciences," *Applied Mathematics and Computation*, vol. 186, pp. 11731182, 2007.

[84] M. Razzaghi and S. Yousefi, "Legendre wavelets method for the nonlinear Volterra-Fredholm integral equations," *Mathematics and Computers in Simulation*, vol. 70, pp. 1-8, 2005.

[85] M. S. Muthuvalu and J. Sulaiman, "Half-Sweep Arithmetic Mean method with composite trapezoidal scheme for solving linear Fredholm integral equations," *Applied Mathematics and Computation*, vol. 217(12), pp. 5442-5448, 2011.

[86] M. S. Muthuvalu and J. Sulaiman, "Half-Sweep Geometric Mean method for solution of linear Fredholm equations," *Matematika*, vol. 24, pp. 75-84, 2008.

[87] M. S. Muthuvalu and J. Sulaiman, "Quarter-Sweep Arithmetic Mean (QSAM) iterative method for second kind linear Fredholm integral equations," *Appl. Math. Sci*, vol. 4, pp. 2943-2953, 2010.

[88] M. S. Muthuvalu and J. Sulaiman, "The Quarter-Sweep Geometric Mean Method for Solving Second Kind Linear Fredholm Integral Equations," *Bulletin of The Malaysian Mathematical Sciences Society*, vol. 36, pp. 10091026, 2013.

[89] M. S. Sahimi and M. Khatim, "The Reduced Iterative Alternating Decomposition Explicit (RIADE) method for diffusion equation," *Pertanika Journal of Science & Technology*, vol. 9(1), pp. 13-20, 2001.

[90] M. S. Sahimi, A. Ahmad and A. A. Bakar, "The Iterative Alternating Decomposition Explicit (IADE) method to solve the heat conduction equation," *International Journal of Computer Mathematics*, vol. 47(3-4), pp. 219-229, 1993.

[91] M. T. Rashed, "Numerical solution of functional differential, integral and integro-differential equations," *Applied Numerical Mathematics*, vol. 156, pp. 485-492, 2004.

[92] M. T. Rashed, "Numerical solution of functional differential, integral and integro-differential equations," *Applied Numerical Mathematics*, vol. 156, pp. 485492, 2004.

[93] N. E. Humphries et al., "Environmental context explains Levy and Brownian movement patterns of marine predators," *Nature*, vol. 465, pp. 1066-1069, 2010.

[94] N. H. M. Ali and F. K. Pin, "Performance analysis of the Modified Group AOR algorithms in elliptic PDEs," *Proceedings of the International Conference on Computational and Information Sciences, 17-19 December, Chengdu, China.*, 2010.

[95] N. H. M. Ali and N. K.Fu, "Modified Explicit Decoupled Group method in the solution of 2-D elliptic PDEs," *Proceedings of the 12th WSEAS International Conference on Applied Mathematics, 29-31 December, Cairo, Egypt*, 2007.

[96] N. H. Sweilam, M. M. Khader and W. Y.Kota, "Numerical and Analytical Study for Fourth-Order Integro-Differential Equations Using a Pseudospectral Method," *Mathematical Problems in Engineering*, vol 2013, pp. 1-7, 2013.

[97] N. Jha, B. K. Sharma and R. C. Chaudhary, "Design and analysis of Quarter Sweep ADEI algorithm for linear and nonlinear two point boundary value problems containing singularity: Application to Burger's equation," *Asian Journal of Experimental Sciences*, vol. 23(1), pp. 329-340, 2009.

[98] N. Jha, "The application of sixth order accurate parallel Quarter Sweep Alternating Group Explicit algorithm for nonlinear boundary value problems with singularity," *Proceedings of the International Conference on Methods and Models in Computer Science, 13-14 December 2010, New Delhi, India*, 2010.

[99] N. K. Nichols, "On the convergence of two-stage iterative processes for solving linear equations," *SIAM Journal on Numerical Analysis*, vol. 10(3), pp. 460-469, 1973.

[100] N. Kurt and M. Sezer, "Polynomial solution of high-order linear Fredholm integro-differential equations with constant coefficients," *Journal of the Franklin Institute*, vol. 345, pp. 839-850, 2008.

[101] N. S. Gabbasov, "Order-optimal methods for integro-differential equations in the singular case," *Differential Equations*, vol. 52(9), pp. 1209-1218, 2016.

[102] O. Cormier, M. F. Singer, B. M. Trager and F. Ulmer, "Linear differential operators for polynomial equations," *J. Symbolic Computation*, vol. 34, pp. 335-398, 2002.

[103] O. Ozan, "Numerical implementation of differential transformations method for integro-differential equations," *International Journal of Computer Mathematics*, vol. 87(12), pp. 2786-2797, 2010.

[104] P. J. Lanzkron, D. J. Rose and D. B. Szyld, "Convergence of nested classical iterative methods for linear systems," *Numerische Mathematik*, vol. 58(1), pp. 685-702, 1990.

[105] P. Lintz, "A Method for the Approximation Solution of Linear Integro-differential Equation," *SIAM Journal on Numerical Analysis*, vol. 11(1), pp. 137-144, 1974.

[106] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, P. Pozo, C. Romine and H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Philadelphia: Society for Industrial and Applied Mathematics, 1993.

[107] R. C. Merton, "Option pricing when underlying stock returns are discontinuous," *Journal of financial economics*, vol. 3(1-2), pp. 125-144, 1976.

[108] R. Cont and E. Voltchkova, "Integro-differential equations for option prices in exponential Lévy models," *Finance Stochast*, vol. 9, pp. 299-325, 2005.

[109] R. J. Hangelbroek, G. H. Kaper and G. K. Leaf, "Collocation Methods for Integro-Differential Equations," *SIAM Journal on Numerical Analysis*, vol. 14(3), pp. 377-390, 1977.

[110] R. Kress, *Numerical Analysis*, New York: Springer-Verlag, 1998.

[111] R. P. Agarwal and D. O'Regan, *Integral and Integrodifferential Equationss Theory, Methods and Applications*, Amsterdam The Netherlands: Gordon and Breach Science Publishers, 2000.

[112] R. P. Agarwal, "Boundary value problems for higher order integro-differential equations," *Nonlinear Analysis, Theory, Methods and Applications*, vol. 7, pp. 259-270, 1983.

[113] R. S. Varga, *Matrix Iterative Analysis (2nd edition)*, Berlin Heidelberg: Springer-Verlag, 2000.

[114] R. Saadati, B. Raftari, H. Adibi, S. M. Vaezpour and A. Shakeri, "A Comparison Between the Variational Iteration Method and Trapezoidal Rule for Solving Linear Integro-Differential Equations," *World Applied Sciences Journal*, vol. 4(3), pp. 321-325, 2008.

[115] S. Behzadi, S. Abbasbandy and T. Allahviranloo, " A Study on singular integro-differential equations of Abel's type by iterative methods," *J. Appl. Math. &Informatics*, vol. 31(3-4), pp. 499-511, 2013.

[116] S. M. Hosseini, & S. Shahmorad, "Tau numerical solution of Fredholm integro-differential equations with arbitrary polynomial bases," *Applied Mathematical Modelling*, vol. 27(2), pp. 145-154, 2003.

[117] S. Yalcinbas and M. Sezer, "The approximate solution of high-order linear Volterra-Fredholm integro-differential equations in terms of Taylor polynomials," *Applied Mathematics and Computation*, vol. 112, pp. 291-308. 2000.

[118] S. Yzbasi, N. Sahi?n and M. Sezer, "Bessel polynomial solutions of high-order linear Volterra integro-differential equations," *Computers & Mathematics with Applications*, vol. 62(4), pp. 19401956, 2011.

[119] S. Yzbasi, N. Sahin and M. Sezer, "Bessel polynomial solutions of high-order linear Volterra integro-differential equations,"*Computers and Mathematics with Applications*, vol. 62(4), pp. 19401956, 2011.

[120] T. J. Chuang, K. I. Kagawa, J. R. Rice and L. B. Sills, "Non-equilibrium models for diffusive cavitation of grain interfaces," *Acta Metallurgica*, vol. 27, pp. 265-284, 1979.

[121] T. T. Kajani, M. Ghasemi and E. Babolian, "Numerical solution of linear integro-differential equation by using sine-cosine wavelets," *Appl. Math. Comput.*, vol. 180, pp. 569-574, 2006.

[122] U. Van Rienen, *Numerical Methods in Computational Electrodynamics: Linear Systems in Practical Applications*, Berlin Heidelberg: Springer-Verlag, 2001.

[123] V. Lakshmikantham and M. R. M. Rao, *Theory of Integro-differential equations*. Switzerland: Gordon and Breach Science Publishers, 1995.

[124] V. Migallon and J. Penades, "Convergence of two-stage iterative methods for Hermitian positive definite matrices," *Applied Mathematics Letters*, vol. 10(3), pp. 79-83, 1997.

[125] V. Migallon and J. Penades, "The monotonicity of two-stage iterative methods," *Applied Mathematics Letters*, vol. 12(8), pp. 73-76, 1999.

[126] V. Ruggiero and E. Galligani, "A parallel algorithm for solving block tridiagonal linear systems," *Computers & Mathematics with Applications*, vol. 24(4), pp. 15-21, 1992.

[127] V. Ruggiero and E. Galligani, "An iterative method for large sparse systems on a vector computer," *Computers & Mathematics with Applications*, vol. 20(1), pp. 25-28, 1990.

[128] V. Ruggiero and E. Galligani, "The Arithmetic Mean preconditioner for multivector computer," *International Journal of Computer Mathematics*, vol. 44(1-4), pp. 207-222, 1992.

[129] V. Ruggiero, and E. Galligani, "An iterative method for large sparse systems on a vector computer," *Computers & Mathematics with Applications*, vol. 20(1), pp. 25-28, 1990.

[130] W. S. Yousif and D. J. Evans, "Explicit De-Coupled Group iterative methods and their parallel implementations," *Parallel Algorithms and Applications*, 7(1-2), pp. 53-71, 1995.

[131] W. Wang and C. Lin. "A new algorithm for integral of trigonometric functions with mechanization," *Applied Mathematics and Computation*, vol. 164(1), pp. 71-82, 2005.

[132] X. R. Oton, "Integro-differential equations: Regularity Theory and Pohozaev Identities," Ph.D. Thesis, Universitat Politecnica de Catalunya, 2014.

[133] Y. Ren, B. Zhang and H. Qiao, "A simple Taylor-series expansion method for a class of second kind integral equations," *Journal of Computational and Applied Mathematics*, vol. 110, pp. 15-24, 1999.

[134] Z. H. Cao, "Convergence of block two-stage iterative methods for symmetric positive definite systems," *Numerische Mathematik*, vol. 90(1), pp. 47-63, 2001.

[135] Z. Jackiewicz, M. Rahman and B. D. Welfert, "Numerical solution of a Fredholm integro-differential equation modeling θ-neural networks," *Applied Mathematics and Computation*, vol. 195, pp. 523-536, 2008.

[136] Z. Y. Liu, H. B. Wu and L. Lin, "The two-stage iterative methods for symmetric positive definite matrices," *Applied Mathematics and Computation*, vol. 114(1), pp. 1-12, 2000.

[137] Z. Z. Bai, "The convergence of two-stage iterative method for Hermitian positive definite linear systems," *Applied Mathematics Letters*, vol. 11(2), pp. 1-5, 1998.

APPENDIX A

A.1 Second Order Lagrange Interpolation Technique for Half-Sweep Case

For $i = 1, 3, 5, \cdots, N-3$

$$\begin{aligned}
\psi_i &= \frac{[ih-(i+1)h][ih-(i+3)h]}{[(i-1)h-(i+1)h][(i-1)h-(i+3)h]}\psi_{i-1} \\
&\quad + \frac{[ih-(i-1)h][ih-(i+3)h]}{[(i+1)h-(i-1)h][(i+1)h-(i+3)h]}\psi_{i+1} \\
&\quad + \frac{[ih-(i-1)h][ih-(i+1)h]}{[(i+3)h-(i-1)h][(i+3)h-(i+1)h]}\psi_{i+3} \\
&= \frac{(i-i-1)(i-i-3)}{(i-1-i-1)(i-1-i-3)}\psi_{i-1} \\
&\quad + \frac{(i-i-1)(i-i-3)}{(i+1-i+1)(i+1-i-3)}\psi_{i+1} \\
&\quad + \frac{(i-i+1)(i-i-1)}{(i+3-i+1)(i+3-i-1)}\psi_{i+3} \\
&= \frac{(-1)(-3)}{(-2)(-4)}\psi_{i-1} + \frac{(1)(-3)}{(2)(-2)}\psi_{i+1} + \frac{(1)(-1)}{(4)(2)}\psi_{i+3} \\
&= \frac{3}{8}\psi_{i-1} + \frac{3}{4}\psi_{i+1} - \frac{1}{8}\psi_{i+3}
\end{aligned}$$

For $i = N - 1$

$$\begin{aligned}
\psi_i &= \frac{[ih-(i+1)h][ih-(i-3)h]}{[(i-1)h-(i+1)h][(i-1)h-(i-3)h]}\psi_{i-1} \\
&\quad + \frac{[ih-(i-1)h][ih-(i-3)h]}{[(i+1)h-(i-1)h][(i+1)h-(i-3)h]}\psi_{i+1} \\
&\quad + \frac{[ih-(i-1)h][ih-(i+1)h]}{[(i-3)h-(i-1)h][(i-3)h-(i+1)h]}\psi_{i-3} \\
&= \frac{(i-i-1)(i-i+3)}{(i-1-i-1)(i-1-i+3)}\psi_{i-1} \\
&\quad + \frac{(i-i+1)(i-i+3)}{(i+1-i+1)(i+1-i+3)}\psi_{i+1} \\
&\quad + \frac{(i-i+1)(i-i-1)}{(i-3-i+1)(i-3-i-1)}\psi_{i-3} \\
&= \frac{(-1)(3)}{(-2)(2)}\psi_{i-1} + \frac{(1)(3)}{(2)(4)}\psi_{i+1} + \frac{(1)(-1)}{(-2)(-4)}\psi_{i-3} \\
&= \frac{3}{4}\psi_{i-1} + \frac{3}{8}\psi_{i+1} - \frac{1}{8}\psi_{i-3}
\end{aligned}$$

A.2 Second Order Lagrange Interpolation Technique for Quarter-Sweep Case

For $i = 2, 6, 10, \cdots, N-6$

$$\begin{aligned}
\psi_i &= \frac{[ih-(i+2)h][ih-(i+6)h]}{[(i-2)h-(i+2)h][(i-2)h-(i+6)h]}\psi_{i-2} \\
&\quad + \frac{[ih-(i-2)h][ih-(i+6)h]}{[(i+2)h-(i-2)h][(i+2)h-(i+6)h]}\psi_{i+2} \\
&\quad + \frac{[ih-(i-2)h][ih-(i+2)h]}{[(i+6)h-(i-2)h][(i+6)h-(i+2)h]}\psi_{i+6} \\
&= \frac{(i-i-2)(i-i-6)}{(i-2-i-2)(i-2-i-6)}\psi_{i-2} \\
&\quad + \frac{(i-i-2)(i-i-6)}{(i+2-i+2)(i+2-i-6)}\psi_{i+2} \\
&\quad + \frac{(i-i+2)(i-i-2)}{(i+6-i+2)(i+6-i-2)}\psi_{i+6} \\
&= \frac{(-2)(-6)}{(-4)(-8)}\psi_{i-2} + \frac{(2)(-6)}{(4)(-4)}\psi_{i+2} + \frac{(2)(-2)}{(8)(4)}\psi_{i+6} \\
&= \frac{3}{8}\psi_{i-2} + \frac{3}{4}\psi_{i+2} - \frac{1}{8}\psi_{i+6}
\end{aligned}$$

For $i = N - 2$

$$
\begin{aligned}
\psi_i &= \frac{[ih-(i+2)h][ih-(i-6)h]}{[(i-2)h-(i+2)h][(i-2)h-(i-6)h]}\psi_{i-2} \\
&\quad + \frac{[ih-(i-2)h][ih-(i-6)h]}{[(i+2)h-(i-2)h][(i+2)h-(i-6)h]}\psi_{i+2} \\
&\quad + \frac{[ih-(i-2)h][ih-(i+2)h]}{[(i-6)h-(i-2)h][(i-6)h-(i+2)h]}\psi_{i-6} \\
&= \frac{(i-i-2)(i-i+6)}{(i-2-i-2)(i-2-i+6)}\psi_{i-2} \\
&\quad + \frac{(i-i+2)(i-i+6)}{(i+2-i+2)(i+2-i+6)}\psi_{i+2} \\
&\quad + \frac{(i-i+2)(i-i-2)}{(i-6-i+2)(i-6-i-2)}\psi_{i-6} \\
&= \frac{(-2)(6)}{(-4)(4)}\psi_{i-2} + \frac{(2)(6)}{(4)(8)}\psi_{i+2} + \frac{(2)(-2)}{(-4)(-8)}\psi_{i-6} \\
&= \frac{3}{4}\psi_{i-2} + \frac{3}{8}\psi_{i+2} - \frac{1}{8}\psi_{i-6}
\end{aligned}
$$

For $i = 1, 3, 5, \cdots, N-5, N-3, N-1$ refer to Appendix A.1.

APPENDIX B

B.1 Performance Comparison of the Methods by Using CD-3CCNC and CD-5CCNC Schemes for Problems 1 and 2

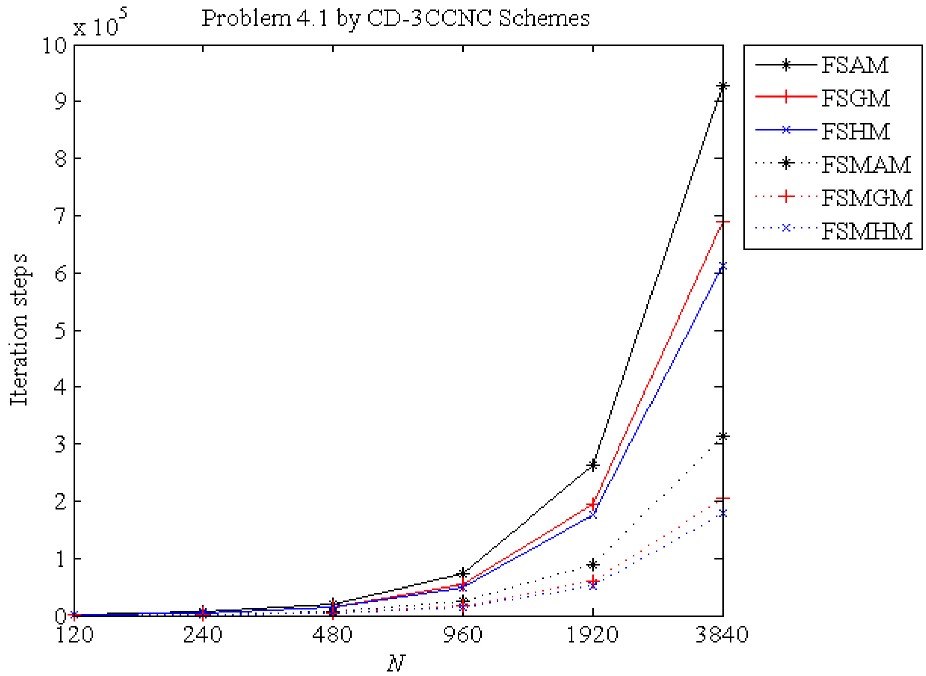

Figure B.1: Comparison of the families of Full-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-3CCNC schemes.

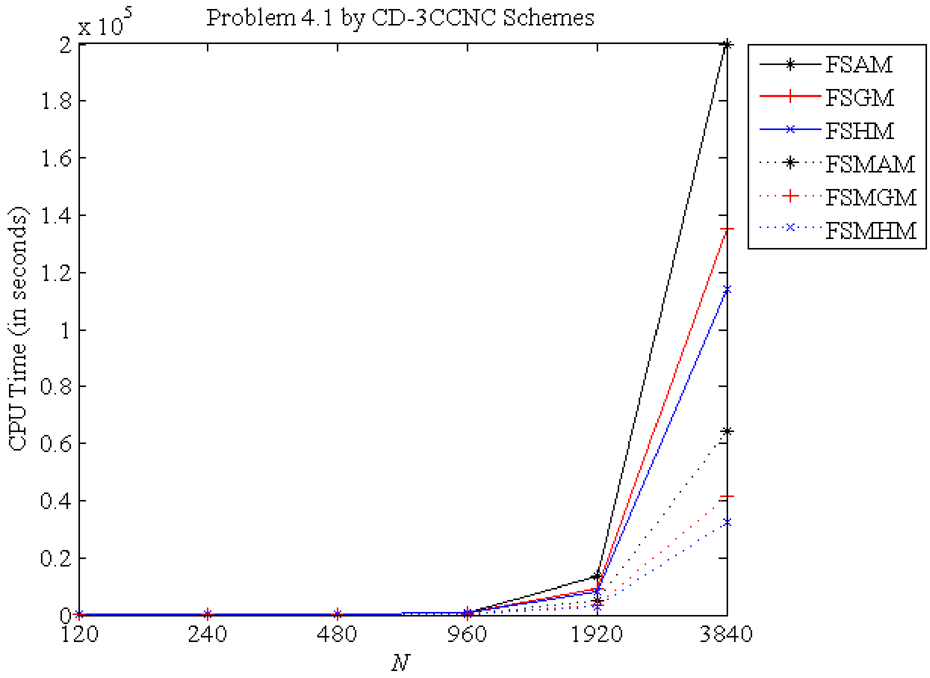

Figure B.2: Comparison of the families of Full-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-3CCNC schemes.

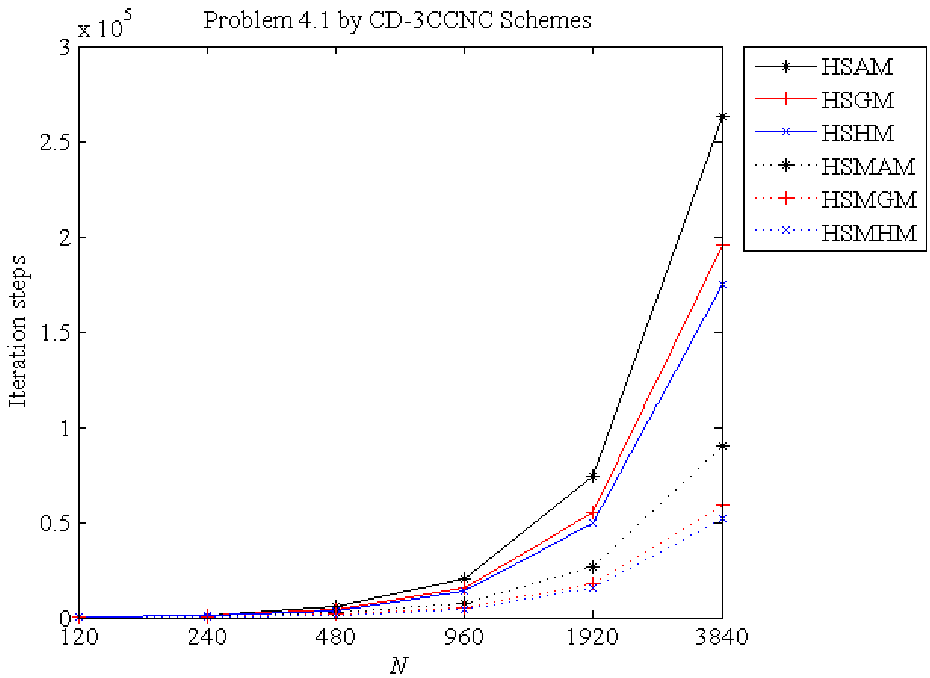

Figure B.3: Comparison of the families of Half-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-3CCNC schemes.

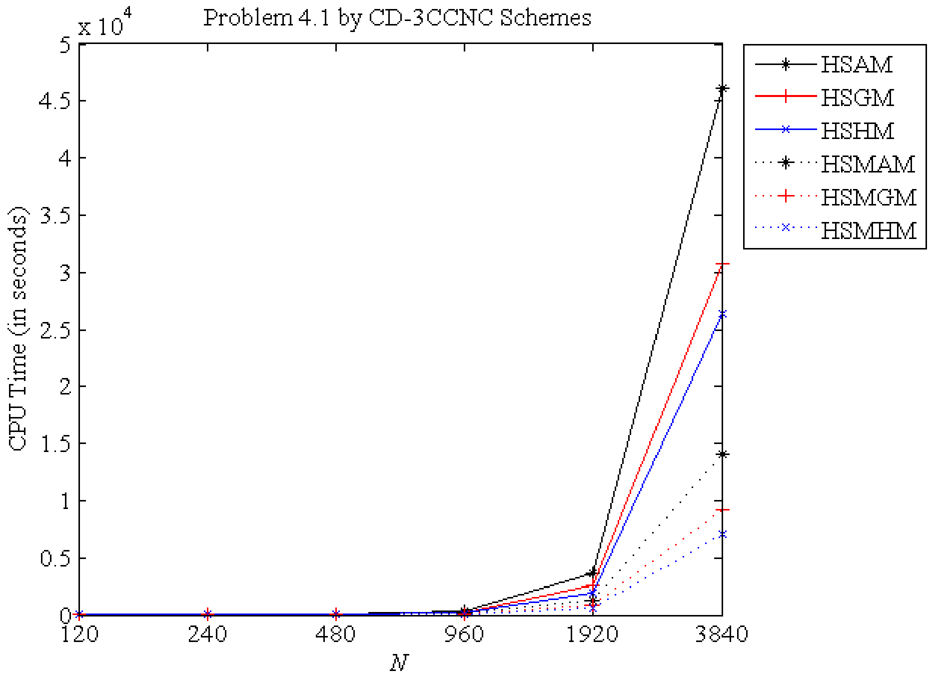

Figure B.4: Comparison of the families of Half-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-3CCNC schemes.

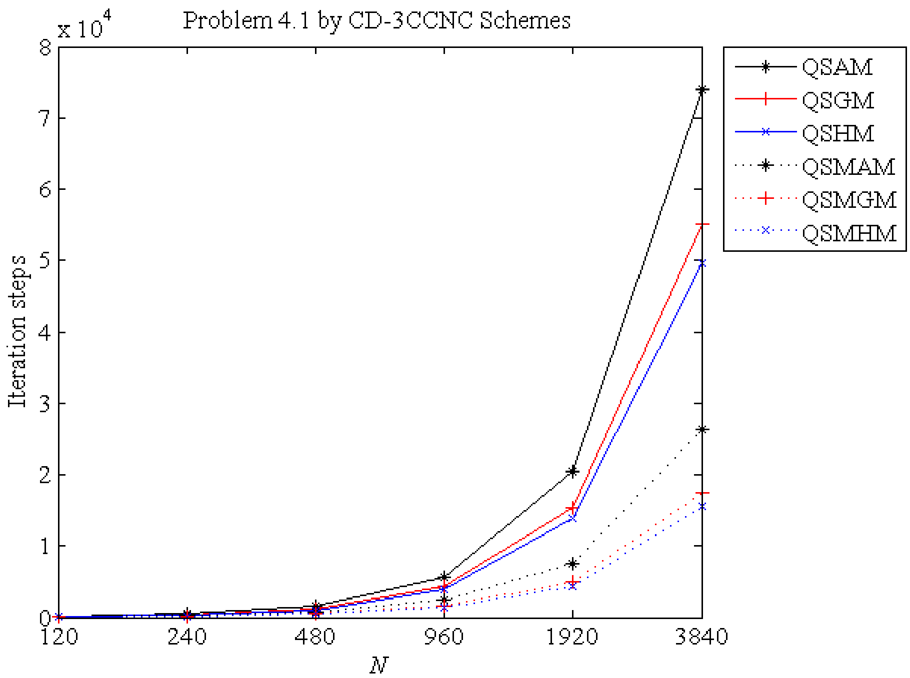

Figure B.5: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-3CCNC schemes.

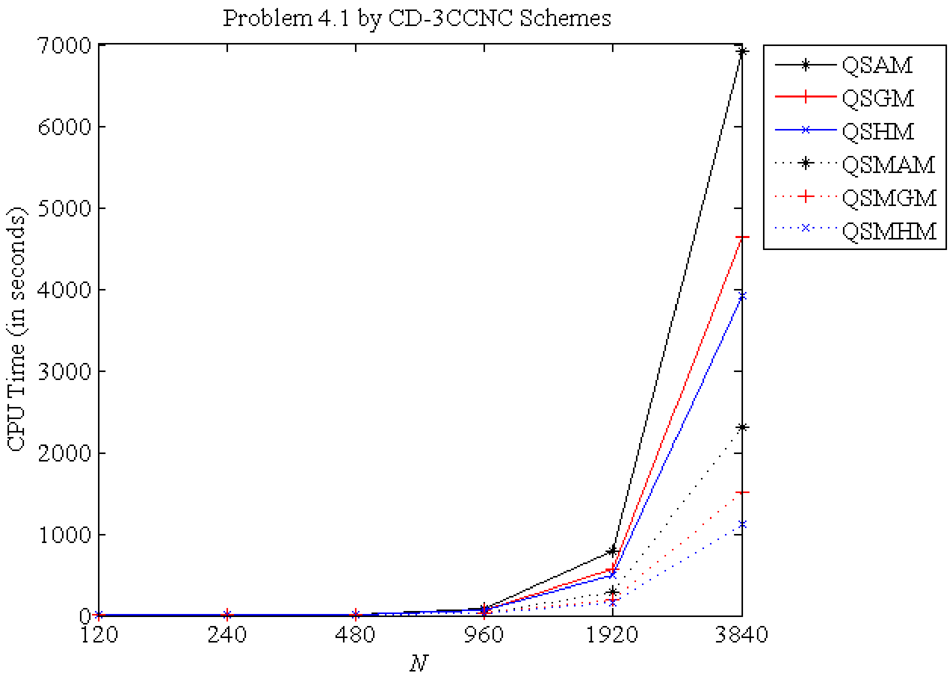

Figure B.6: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-3CCNC schemes.

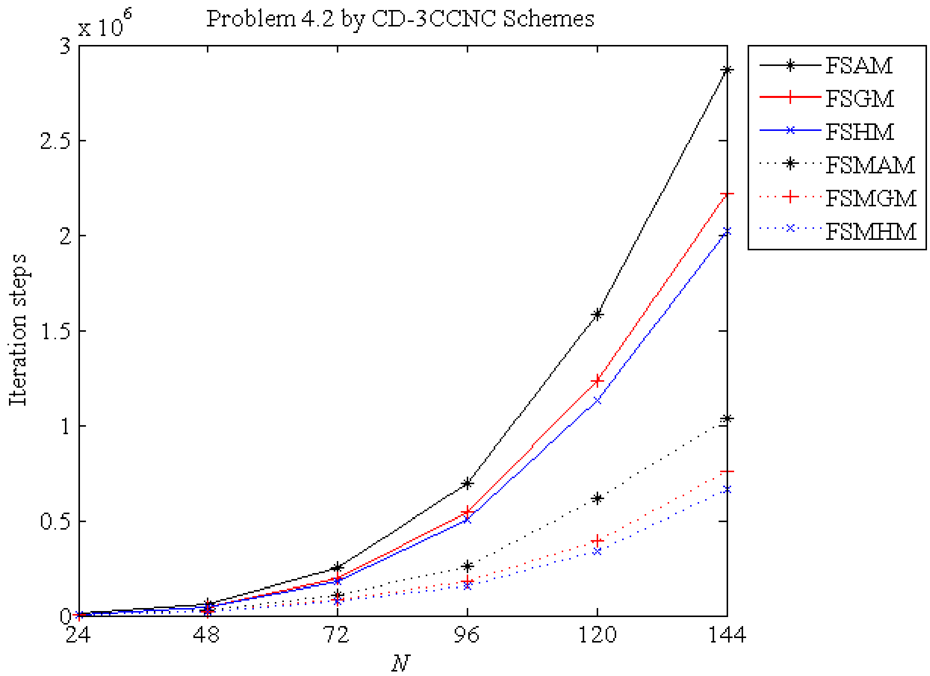

Figure B.7: Comparison of the families of Full-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-2 by CD-3CCNC schemes.

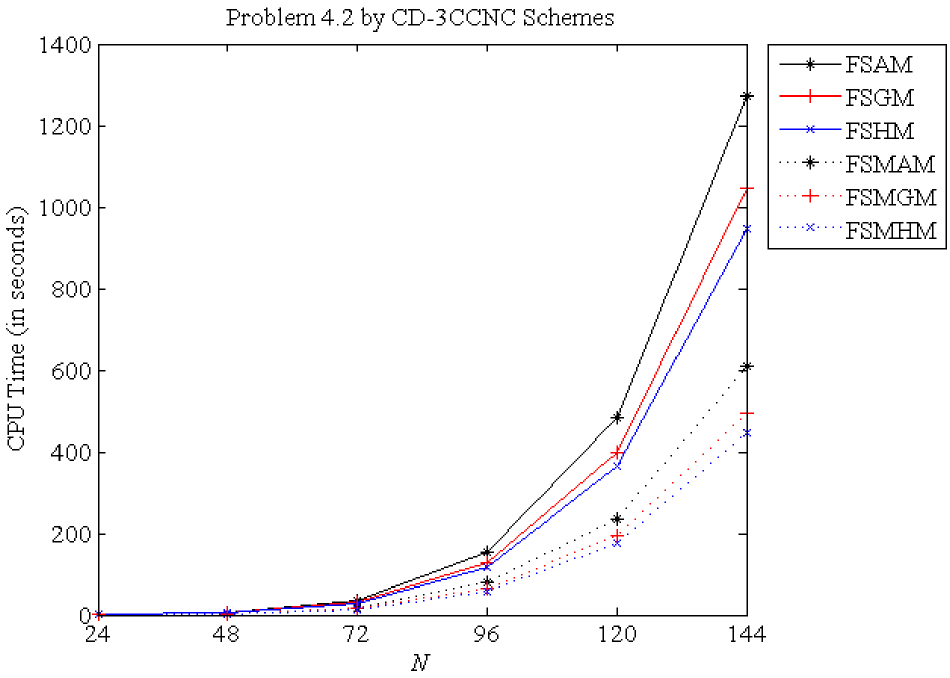

Figure B.8: Comparison of the families of Full-Sweep WM and MWM methods in terms of CPU time for solving Problem-2 by CD-3CCNC schemes.

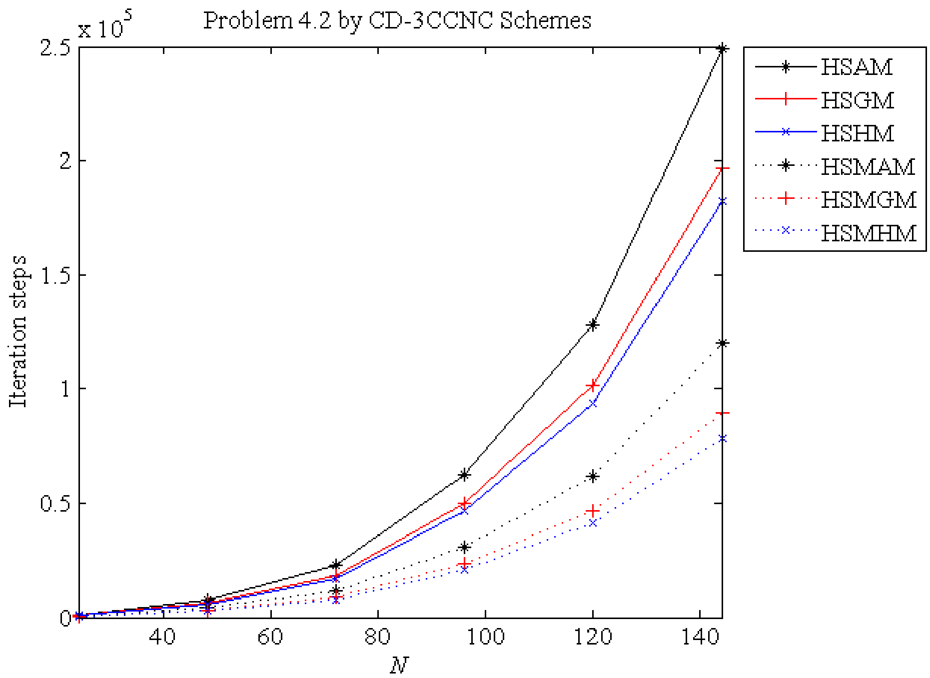

Figure B.9: Comparison of the families of Half-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-2 by CD-3CCNC schemes.

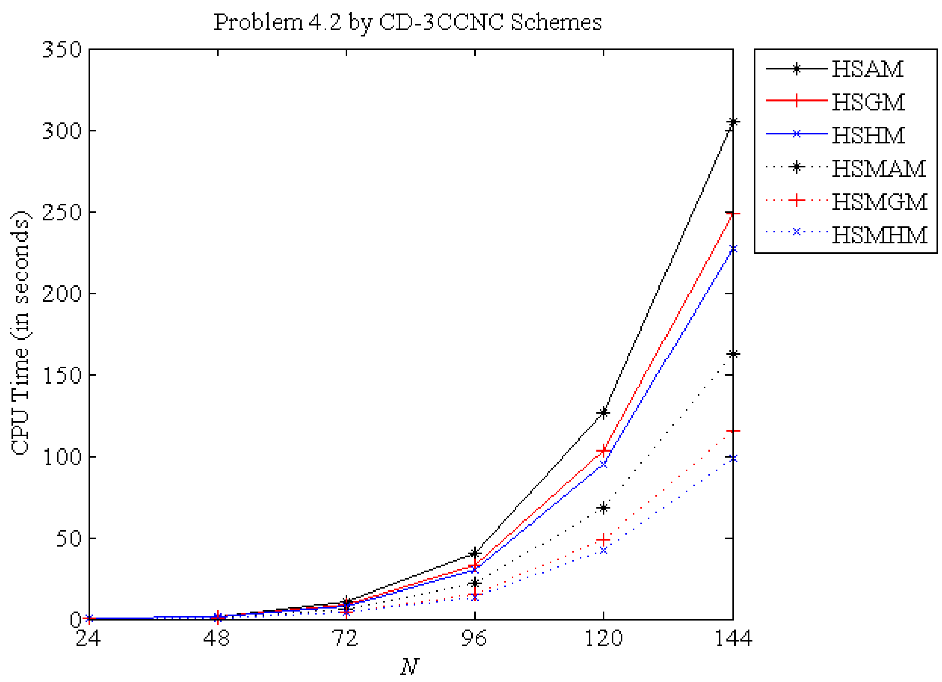

Figure B.10: Comparison of the families of Half-Sweep WM and MWM methods in terms of CPU time for solving Problem-2 by CD-3CCNC schemes.

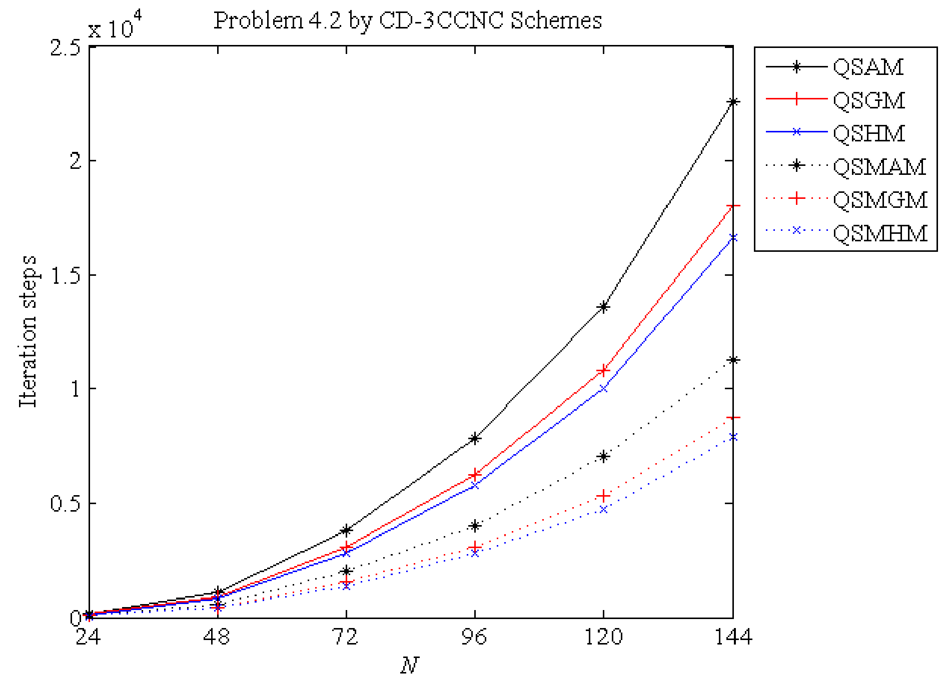

Figure B.11: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-2 by CD-3CCNC schemes.

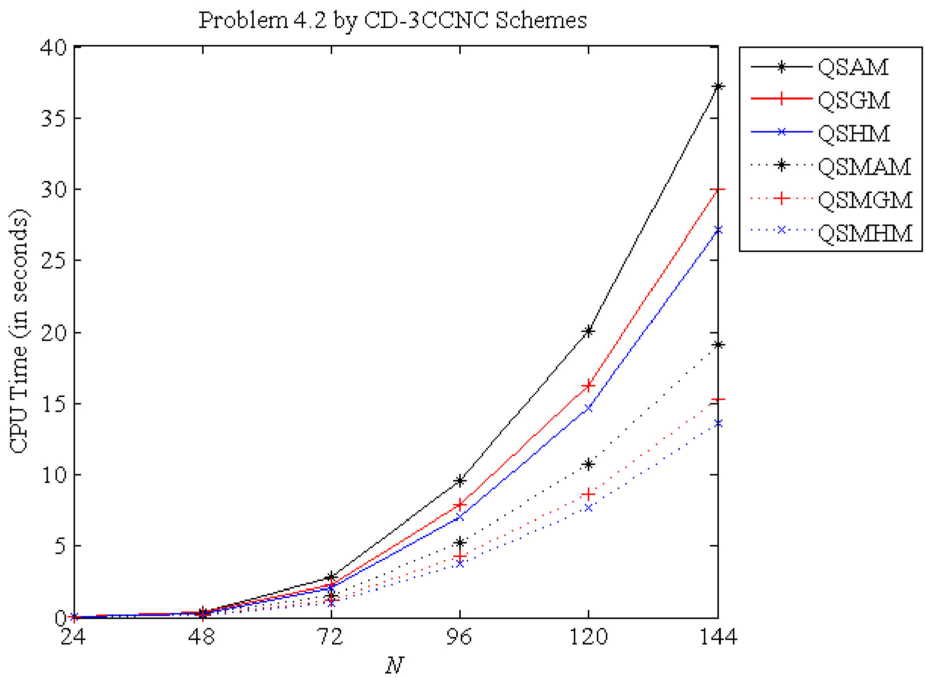

Figure B.12: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of CPU time for solving Problem-2 by CD-3CCNC schemes.

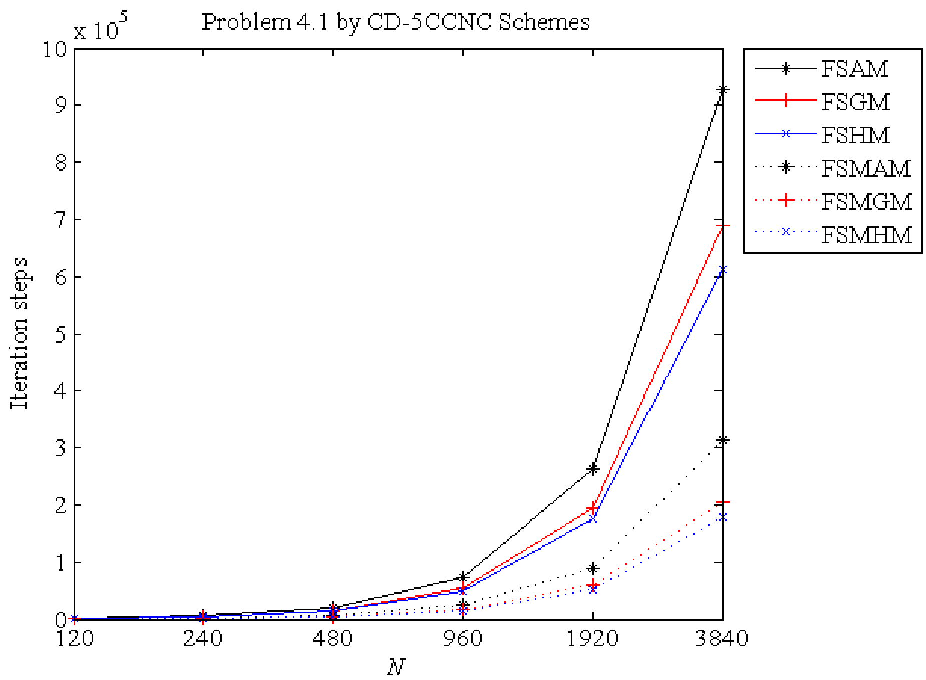

Figure B.13: Comparison of the families of Full-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-5CCNC schemes.

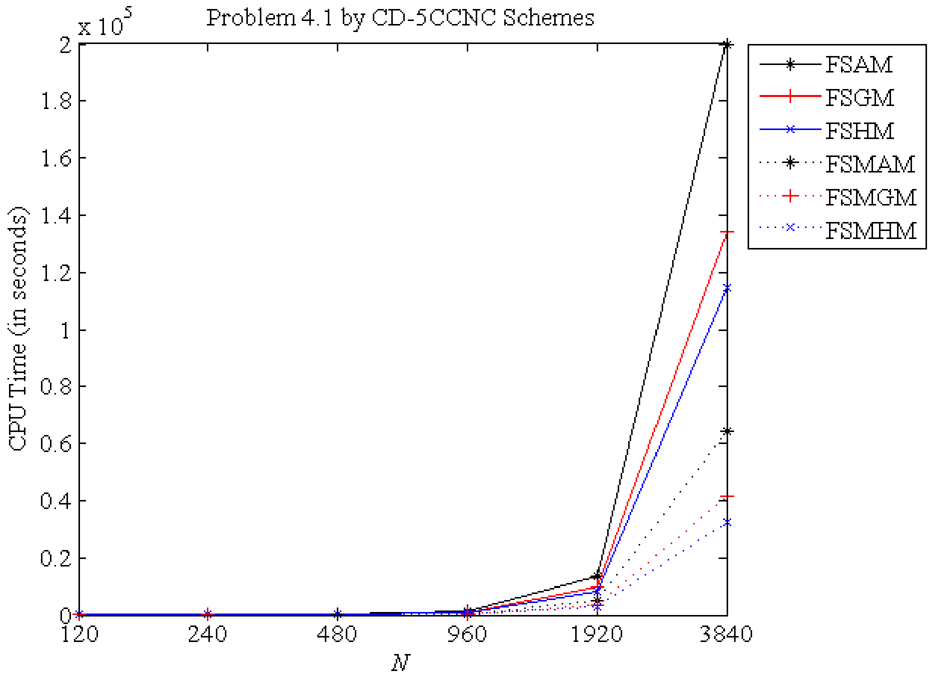

Figure B.14: Comparison of the families of Full-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-5CCNC schemes.

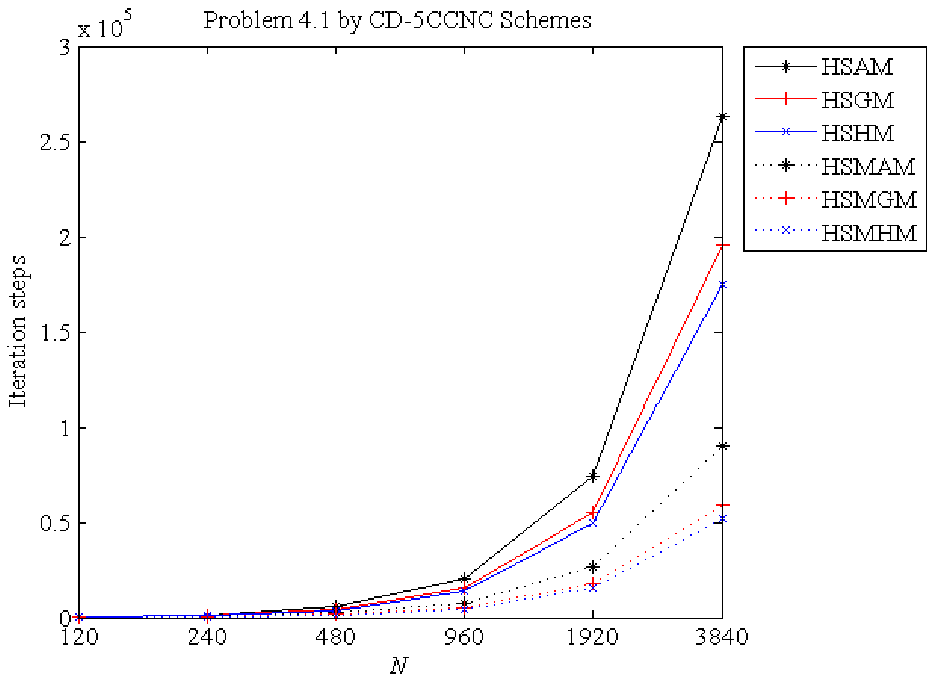

Figure B.15: Comparison of the families of Half-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-5CCNC schemes.

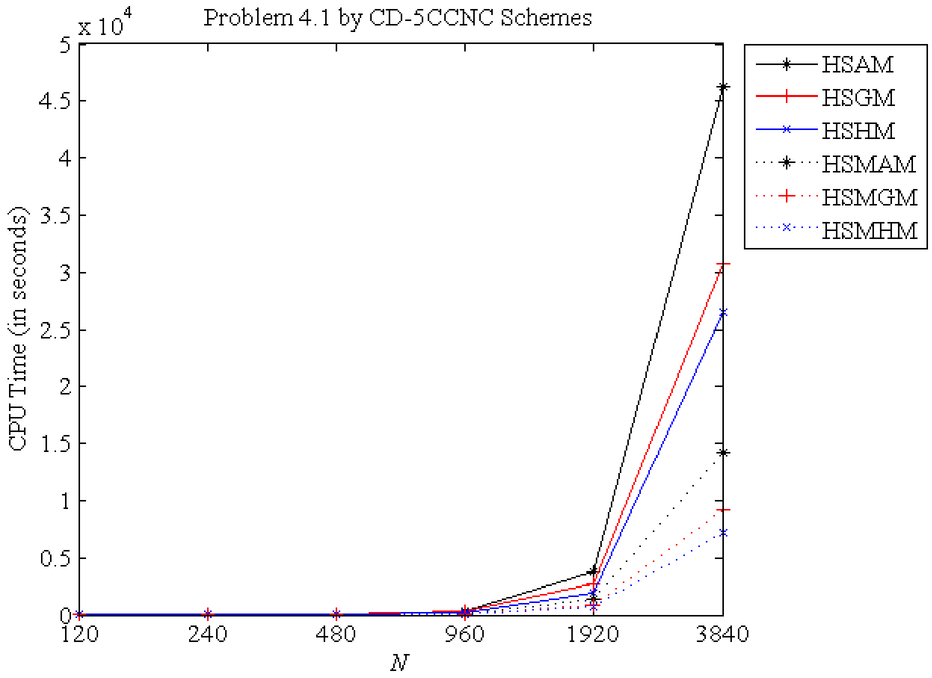

Figure B.16: Comparison of the families of Half-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-5CCNC schemes.

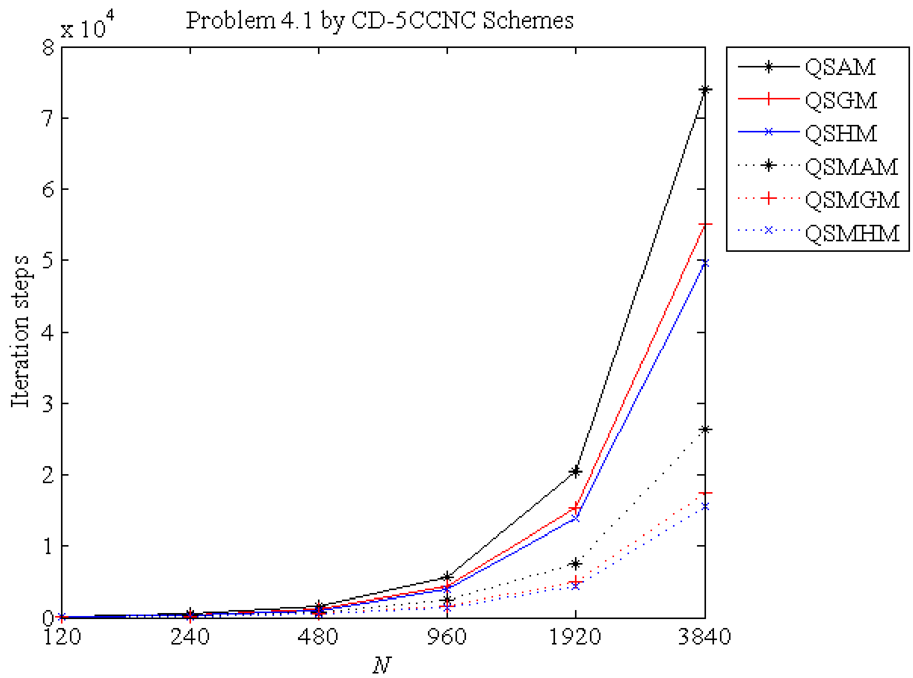

Figure B.17: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-1 by CD-5CCNC schemes.

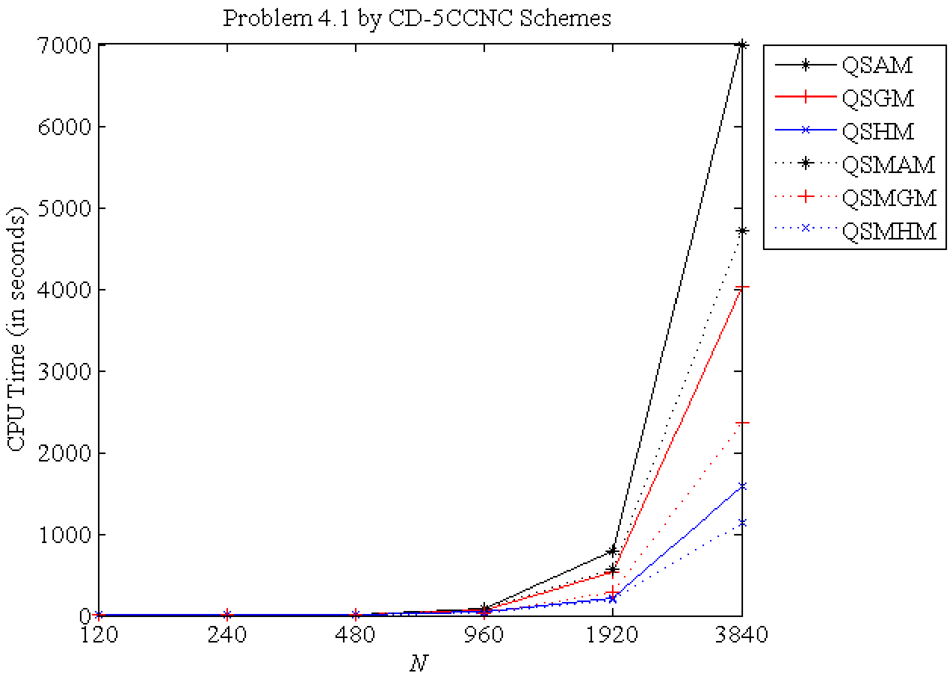

Figure B.18: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of CPU time for solving Problem-1 by CD-5CCNC schemes.

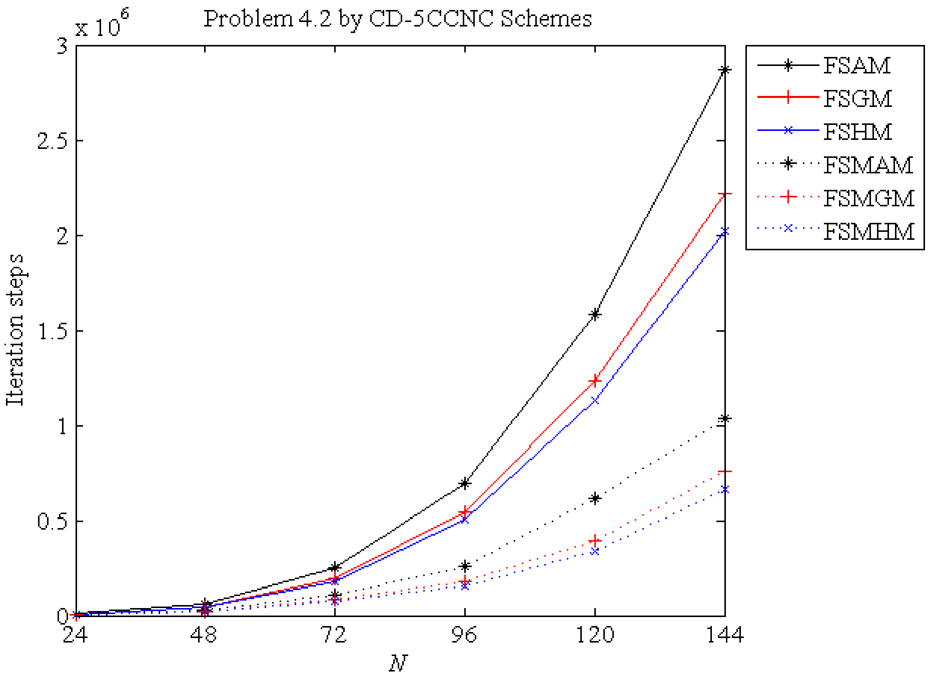

Figure B.19: Comparison of the families of Full-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-2 by CD-5CCNC schemes.

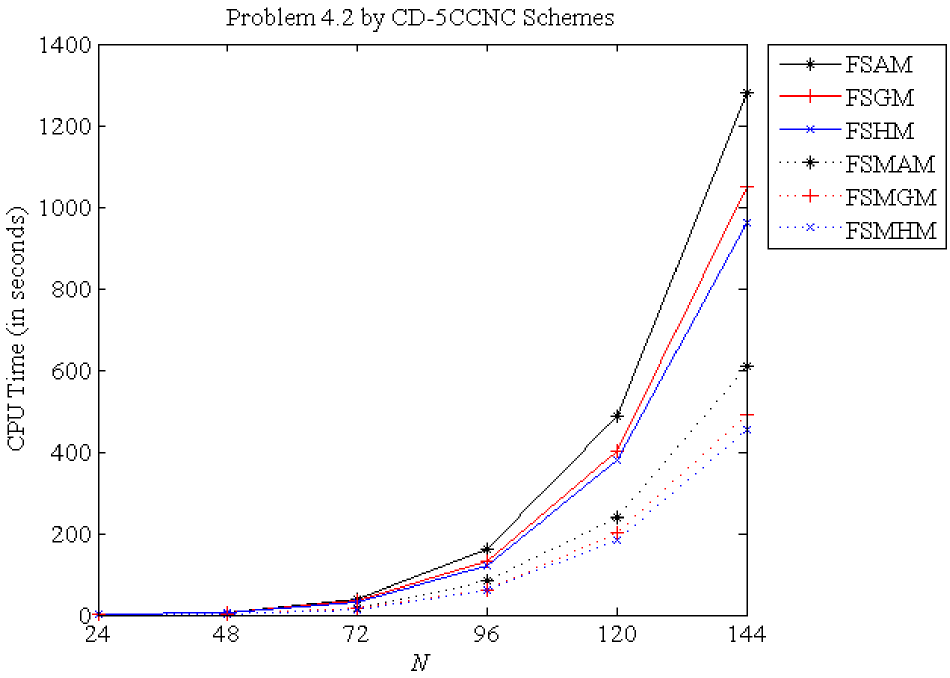

Figure B.20: Comparison of the families of Full-Sweep WM and MWM methods in terms of CPU time for solving Problem-2 by CD-5CCNC schemes.

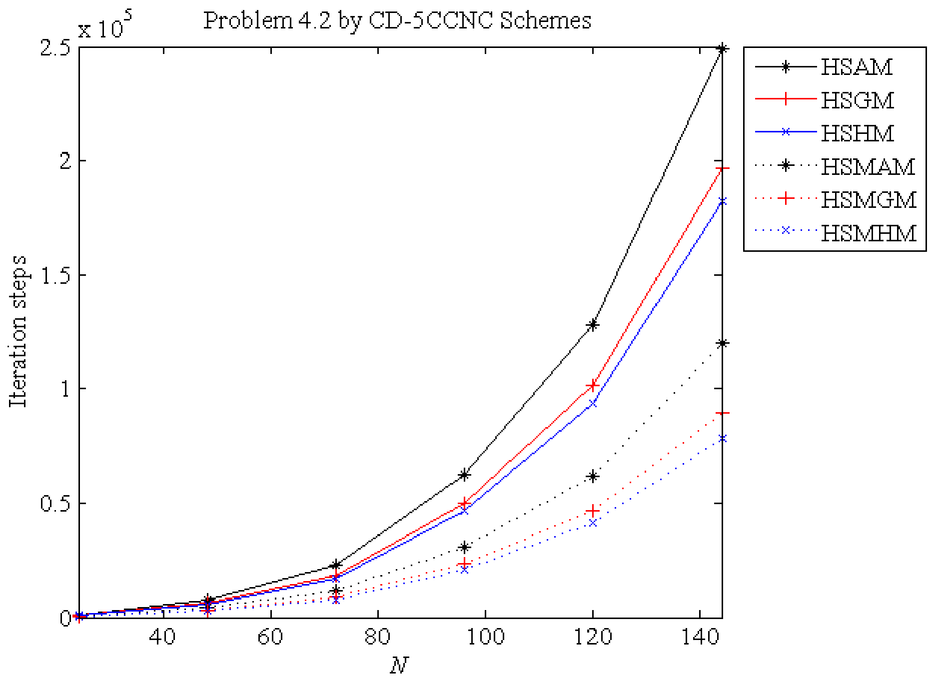

Figure B.21: Comparison of the families of Half-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-2 by CD-5CCNC schemes.

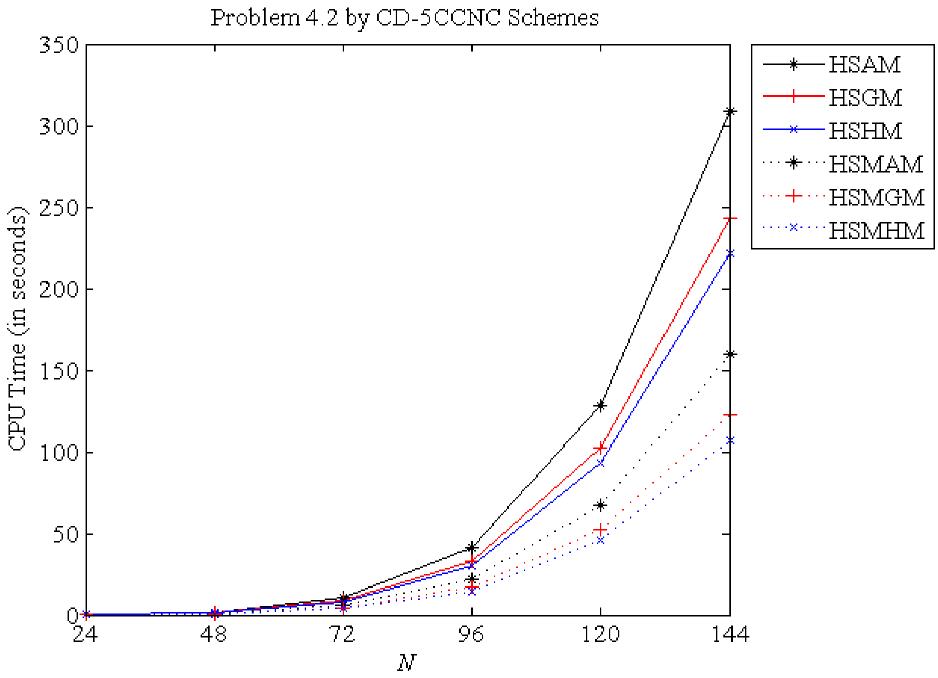

Figure B.22: Comparison of the families of Half-Sweep WM and MWM methods in terms of CPU time for solving Problem-2 by CD-5CCNC schemes.

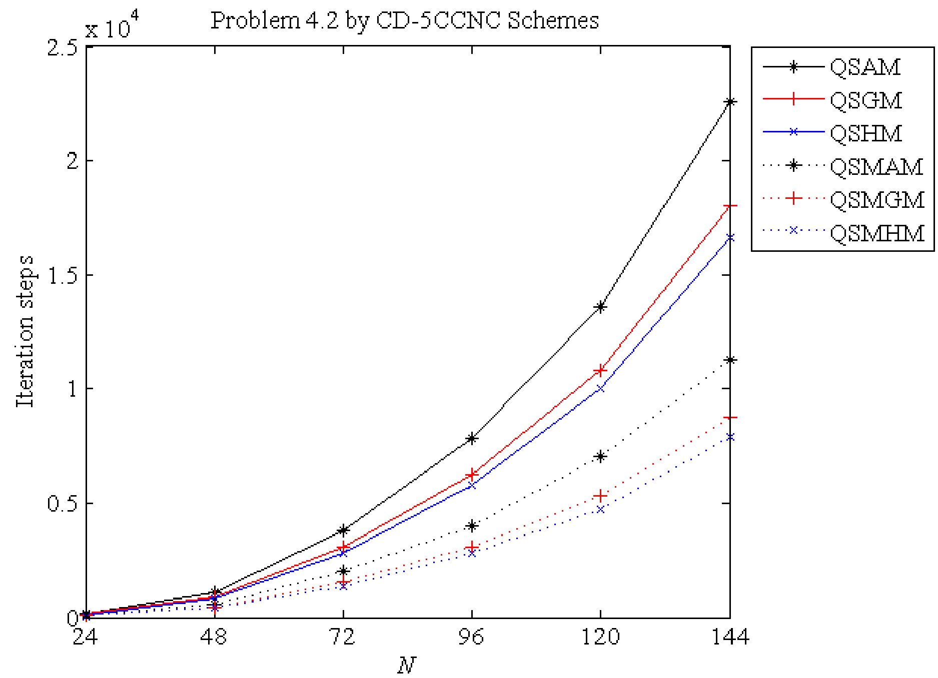

Figure B.23: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of Iteration steps for solving Problem-2 by CD-5CCNC schemes.

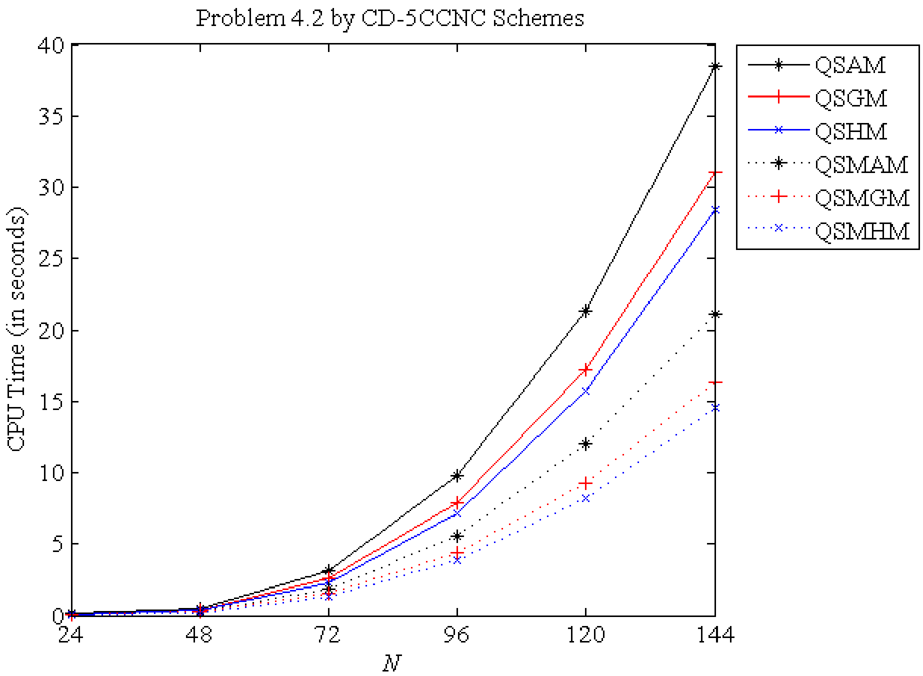

Figure B.24: Comparison of the families of Quarter-Sweep WM and MWM methods in terms of CPU time for solving Problem-2 by CD-5CCNC schemes.