Sains Malaysiana 54(9)(2025): 2301-2313
*http://doi.org/10.17576/jsm-2025-5409-16*

# A Novel Variant of Weighted Quadratic Mean Iterative Methods for Fredholm Integro-Differential Equations

(Varian Novel Kaedah Lelaran Min Kuadratik Berwajaran untuk Persamaan Integro-Differential Fredholm)

NG WEI LI[1] , ELAYARAJA ARUCHUNAN[2,*] & ZAILAN SIRI[1]

*[1]Institute of Mathematical Sciences, Universiti Malaya, 50603 Kuala Lumpur, Malaysia*
*[2]Department of Decision Science, Universiti Malaya, 50603 Kuala Lumpur, Malaysia*

*Received: 24 February 2025/Accepted: 10 July 2025*

## ABSTRACT

Integro-differential equations are critical for modelling real-world phenomena in physics, engineering, and biology. This paper introduces a Quadratic Mean iterative method to solve dense linear systems derived from the discretization of second- and fourth-order Fredholm integro-differential equations (FIDEs). The solution of the FIDEs is approximated using finite difference, composite trapezoidal, and composite Simpson's 1/3 and 3/8 schemes. The quadratic mean iterative method then solves the discretized system with different mesh sizes. As the resulting systems are large, a complexity reduction approach is implemented on the quadratic mean method to develop the half-sweep quadratic mean iterative method. The newly proposed iterative method includes a novel theorem, comprehensive proofs, and a detailed convergence analysis. The numerical results indicate that the quadratic mean method significantly outperforms the Gauss-Seidel iterative method in terms of efficiency, making it a promising solution for FIDEs.

Keywords: Fredholm integro-differential equations; quadratic mean; half-sweep iteration; finite difference; composite trapezoidal; Composite Simpson's rules

## ABSTRAK

Persamaan pembezaan-kamiran adalah penting untuk memodelkan fenomena dunia sebenar dalam fizik, kejuruteraan dan biologi. Dalam jurnal ini memperkenalkan kaedah lelaran Purata Kuadratik untuk menyelesaikan sistem linear tumpat yang diperoleh daripada membahagikan persamaan integro-pembezaan Fredholm tertib kedua dan keempat (FIDEs) kepada bahagian kecil. Penyelesaian FIDEs dianggarkan menggunakan perbezaan terhingga, trapezoid komposit dan skema 1/3 dan 3/8 komposit Simpson. Kemudian, kaedah lelaran purata kuadratik digunakan untuk menyelesaikan persamaan anggaran dengan saiz mesh yang berbeza. Memandangkan sistem yang akan diselesaikan adalah besar, pendekatan pengurangan kerumitan dilaksanakan pada kaedah purata kuadratik untuk membentuk kaedah lelaran purata kuadratik separuh sapuan. Kaedah lelaran yang baharu dicadangkan termasuk teorem novel, bukti komprehensif, dan analisis penumpuan terperinci. Keputusan berangka menunjukkan bahawa kaedah purata kuadratik dengan ketara mengatasi kaedah lelaran Gauss-Seidel dari segi kecekapan, menjadikannya penyelesaian terbaik untuk FIDEs.

Kata kunci: Persamaan pembezaan-kamiran; Fredholm; min kuadratik; lelaran separuh sapuan; beza terhingga; trapezoid komposit; Peraturan Simpson;

## INTRODUCTION

In the 21st century, mathematical models have emerged as indispensable tools for problem-solving across diverse fields. Among these, integro-differential equations (IDEs) play a pivotal role in formulating physical phenomena in various domains, including nano-hydrodynamics, potential theory, mechanics, fluid dynamics, glass-forming processes, biology, astronomy, and chemical kinetics (Benzi & Dayar 1995; Rathinasamy & Balachandran 2008; Salih et al. 2014; Yuhe et al. 1999). However, solving these equations analytically, especially for high-order cases (Zhao & Corless 2006), poses significant challenges due to their complexity and time-consuming nature (Aruchunan et al. 2015). As a result, obtaining an approximate solution through numerical methods becomes necessary. In this context, we focus on the general form of second and fourth order linear Fredholm integro-differential Equations (FIDEs) for investigation and resolution. In practical applications, the behaviors modelled by first, second and fourth-order IDEs often suffice to capture the essential dynamics of the system under study. Extending the analysis to higher-order equations was deemed unnecessary as no additional significant insights were anticipated. Many practical applications in physics, engineering, and other fields can be effectively modelled using first, second and fourth-order equations. For example, Newton's laws (which lead to second-order equations) are foundational in mechanics. Higher-order equations are often only necessary in more specialized contexts. The general equations of second and fourth order FIDEs are as follows, 2nd order FIDEs:

$$\frac{d^2G}{dx^2} + T(x)\frac{dG}{dx} + U(x)G(x) - \int_m^p K(x,t)G(t)\,dt = f(x), x \in (m,p) \quad (1)$$

with Dirichlet boundary conditions:

$$G(m) = \alpha, \qquad G(p) = b$$

4th order FIDEs:

$$\frac{d^4G}{dx^2} + T(x)\frac{d^2G}{dx^2} + U(x)G(x) - \int_m^p K(x,t)G(t)\,dt = f(x), x \in (m,p) \quad (2)$$

with the Dirichlet boundary conditions

$$G^{(m)} = \alpha_1, \quad G(p) = b_1$$
$$G''^{(m)} = \alpha_1, \quad G''(p) = b_2$$

where $S(x)$, $T(x)$, $U(x)$, $f(x)$ and the kernel $K(x,t)$ are the function that is already known while the $G(x)$ is the unknown function to be determined.

This paper focuses on the two-stage iterative methods with some variation. This method is widely used in solving matrix problems such as SOR (Cai, Xiao & Zhao-Hong 2010; Katuri & Maroju 2025), Iterative Alternating Decomposition Explicit (IADE) method (Sahimi, Ahmad & Bakar 1993), two-stage Initial-value Iterative Neural Network (IINN) method (Jin et al. 2024), Mixed-Precision Conjugate Gradient algorithm (Aihara, Ozaki & Mukunoki 2024) and Weighted Mean (WM) methods. The family of WM iterative methods are one of the best numerical algorithms to solve system of equations which converge quickly and smoothly. Under the WM methods family, there are several methods which has developed namely Arithmetic Mean (AM) (Galligani & Ruggiero 1990) and Geometric Mean (GM) (Sulaiman et al. 2006) iterative methods. In this paper, a newly proposed method, called the quadratic mean method is developed and implemented to solve the dense linear systems arising from the Finite Difference, Composite Trapezoidal, Composite Simpson's 1/3, and Composite Simpson's 3/8 schemes. The discretization schemes typically generate dense systems, which are often computationally intensive (Aruchunan et al. 2022). To mitigate this complexity, a reduction technique is applied to the quadratic mean method, leading to the development of the half-sweep quadratic mean method. To assess the effectiveness of the proposed methods, three key parameters are considered: the number of iterations, execution time, and maximum absolute error. The primary contribution of this paper is the introduction of the quadratic mean iterative method and its variants for solving FIDEs, supported by the development and proof of a corresponding theorem. Table 1 presents the abbreviations for the numerical schemes and methods utilized in this study.

## MATERIALS AND METHODS

### COMPUTATIONAL COMPLEXITY-REDUCTION TECHNIQUES

The proposed quadratic mean (QM) method, which can also be referred to as the Full-Sweep Quadratic Mean (FSQM), along with the discretization schemes, is optimized to reduce computational complexity. The core concept behind the application of half-sweep techniques is to minimize this complexity. Initially, the solution domain of the half-sweep is divided into $N$ equally spaced intervals, as illustrated in the figures below. This approach helps restructure the computation process, enhancing efficiency.
where the blue circles and red triangles refer to the nodal points and $h$ is the step size.

TABLE 1. Descriptions of Abbreviated Numerical Codes

| Discretization Schemes | |
|---|---|
| FDCT | Finite Difference-Composite Trapezoidal |
| FDCS1 | Finite Difference-Composite Simpson's 1/3 |
| FDCS2 | Finite Difference-Composite Simpson's 3/8 |
| Iterative Methods | |
| FSGS | Full-Sweep Gauss-Seidel |
| HSGS | Half-Sweep Gauss-Seidel |
| FSAM | Full-Sweep Arithmetic Mean |
| HSAM | Half-Sweep Arithmetic Mean |
| FSQM | Full-Sweep Quadratic Mean |
| HSQM | Half-Sweep Quadratic Mean |
| Performance Metrics: | |
| $N$ | Number of iterations |
| $t$ | Execution time (seconds) |
| ${}^{\varepsilon}N$ | Maximum absolute error |

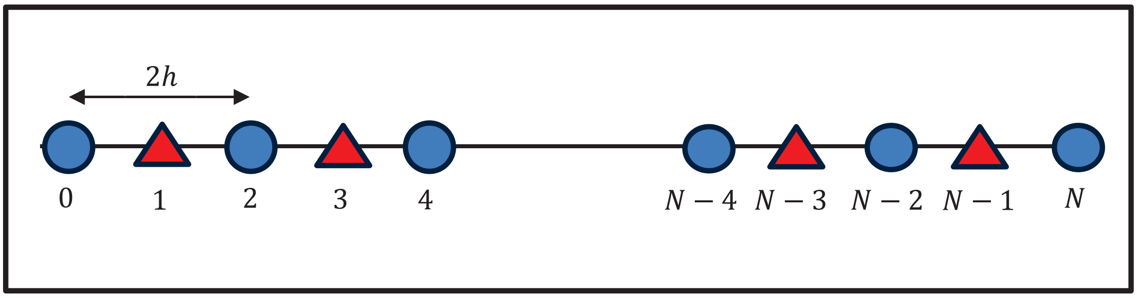

FIGURE 1. Node distribution for half-sweep iteration

Based on Figure 1, the half-sweep iterative method, only the ● nodal point is considered for the process of converging. The ▲ remaining points are calculated by using the average mean method. The half-sweep techniques reduce the complexity of the iterative method to $\frac{1}{2}$ . The GS and AM iterative methods are also refined using this iteration concept. Consequently, the proposed FSQM method will be further explored by incorporating half-sweep iteration techniques to assess its performance, focusing on the number of iterations, execution time, and maximum absolute error.

## FINITE DIFFERENCE-COMPOSITE CLOSED NEWTON-COTES SYSTEMS

In this paper, all the iterative methods mentioned are GS (Connolly, Burns & Weiss 1990) or also known as Full-sweep Gauss-Seidel (FSGS) (Aruchunan et al. 2022), Half-sweep Gauss-Seidel (HSGS) (Aruchunan & Sulaman 2012), Full-sweep Arithmetic Mean (FSAM) (Galligani & Ruggiero 1990), Half-sweep Arithmetic Mean (HSAM) (Muthuvalu & Sulaiman 2011). All the combinations of discretization schemes will be presented namely finite difference with composite Trapezoidal (FDCT), finite difference with composite Simpson's first rule (FDCS1) and finite difference with composite Simpson's second rule (FDCS2) schemes to form a linear system (Aruchunan & Sulaiman 2012). Now let an interval $(m, p)$ be divided uniformly into n subintervals with size of $\gamma = \frac{p-m}{n}$ and the discrete set of points of $x$ be given by $x_i = m + i\,\gamma$ $(i = 0,1,2, \ldots, N)$.Throughout the following sections, the value of $u$ corresponds to 1 and 2 represents the full- and half-sweep iteration concept (Aruchunan et al. 2014). The FDCT, FDCS1 and FDCS2 schemes are applied to discretize equations (1) and (2) as follows:

Second order FIDEs:

$$S_i\frac{G(x_{i+u})-2G(x_i)+G(x_{i-u})}{(u\gamma)^2}+T_i\frac{G(x_{i+u})-G(x_u)}{u\gamma}+U_iG(x_i)=f(x)+\sum_{j=0,u,2u}^{n}A_jK_{i,j}G_j \quad (3)$$

Fourth order FIDEs:

$$S_i\frac{G(x_{i+2u})-4G(x_{i+u})+6G(x_i)-4G(x_{i-u})-G(x_{i-2u})}{(u\gamma)^4}+T_i\frac{G(x_{i+1})-2G(x_i)+G(x_{i-1})}{(u\gamma)^2}+U_iG(x_i)$$
$$=f(x)+\sum_{j=0,2u,3u}^{n}A_jK_{i,j}G_j \quad (4)$$

for $i = 1,2,3, \ldots ,n$ and $A_j$ in equation (3) and (4) is given by

Composite Trapezoidal: $A_j=\begin{cases}\frac{uh}{2}, & j=0,n\\ h, & otherwise\end{cases}$

Composite Simpson's 1/3: $A_j=\begin{cases}\frac{uh}{3}, & j=0,n\\ \frac{4uh}{3}, & j=1,3,5,\ldots,n-1\\ \frac{2uh}{3}, & otherwise\end{cases}$

Composite Simpson's 3/8: $A_j=\begin{cases}\frac{3uh}{8}, & j=0,n\\ \frac{3uh}{4}, & j=3,6,0,\ldots,n-3\\ \frac{9uh}{8}, & otherwise\end{cases}$

Both linear equations can also be rearranged and written in the matrix form as

$$Y\tilde{X}=\tilde{V} \quad (5)$$

where

Second order IDE

$$Y=\begin{bmatrix}\kappa_{1,1} & \zeta_{1,2} & \omega_{3,1} & \cdots & \omega_{1,n-3} & \omega_{1,n-2} & \omega_{1,n-1}\\ \varrho_{2,1} & \kappa_{2,2} & \zeta_{2,3} & \cdots & \omega_{2,n-3} & \omega_{2,n-2} & \omega_{2,n-1}\\ \aleph_{3,1} & \varrho_{3,2} & \kappa_{3,3} & \cdots & \omega_{3,n-3} & \omega_{3,n-2} & \omega_{3,n-1}\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ \omega_{n-3,1} & \omega_{n-3,2} & \omega_{n-3,3} & \cdots & \kappa_{n-3,n-3} & \zeta_{n-3,n-2} & \omega_{n-3,n-1}\\ \omega_{n-2,1} & \omega_{n-2,2} & \omega_{n-2,3} & \cdots & \varrho_{n-2,n-3} & \kappa_{n-2,n-2} & \zeta_{n-2,n-1}\\ \omega_{n-1,1} & \omega_{n-1,2} & \omega_{n-1,3} & \cdots & \omega_{n-1,n-3} & \varrho_{n-1,n-2} & \kappa_{n-1,n-1}\end{bmatrix}_{(n-1)\times(n-1)}$$

in which

$\kappa_{i,i} = \gamma^2\, U_{\mathrm{i}} - 2S_i - \gamma T_i - \gamma^2 A_i k_{i,i}$, $\zeta_{i,j} = S_i + \gamma T_i - \gamma^2 A_j k_{i,j}$, $\varrho_{i,j} = S_i - \gamma^2 A_j k_{i,j}$ and $\omega_{i,j} = -\,\gamma^2\, A_{jki,j}$

$$\tilde{X}=\begin{bmatrix}\gamma^2f_1+(S_0+\gamma^2A_0k_{1,0})G_0+(\gamma^2A_nk_{1,n})G_n\\ \gamma^2f_2+(\gamma^2A_0k_{2,0})G_0+(\gamma^2A_nk_{2,n})G_n\\ \gamma^2f_3+(\gamma^2A_0k_{3,0})G_0+(\gamma^2A_nk_{3,n})G_n\\ \vdots\\ \gamma^2f_{n-3}+(\gamma^2A_0k_{n-3,0})G_0+(\gamma^2A_nk_{n-3,n})G_n\\ \gamma^2f_{n-2}+(\gamma^2A_0k_{n-2,0})G_0+(\gamma^2A_nk_{n-2,n})G_n\\ \gamma^2f_{n-1}+(\gamma^2A_0k_{n-1,0})G_0+(-S_i-\gamma T_{n-1}+\gamma^2A_nk_{n-1,n})G_n\end{bmatrix}$$

$$\text{and } \widetilde{V} = \begin{bmatrix} G(x_1) \\ G(x_2) \\ G(x_3) \\ \vdots \\ G(x_{n-3}) \\ G(x_{n-2}) \\ G(x_{n-1}) \end{bmatrix}.$$

Fourth order IDE

$$Y = \begin{bmatrix} -\frac{1}{\gamma h_4} + \rho_{1,1} & \theta_{1,2} & \beth_{1,3} & \aleph_{1,4} & \cdots & \aleph_{1,n-3} & \aleph_{1,n-2} & \aleph_{1,n-1} \\ \tau_{2,1} & \rho_{2,2} & \theta_{2,3} & \beth_{2,4} & \cdots & \aleph_{2,n-3} & \aleph_{2,n-2} & \aleph_{2,n-1} \\ \beth_{3,1} & \tau_{3,2} & \rho_{3,3} & \theta_{3,4} & \cdots & \aleph_{3,n-3} & \aleph_{3,n-2} & \aleph_{3,n-1} \\ \aleph_{3,1} & \beth_{4,2} & \tau_{4,3} & \rho_{4,4} & \cdots & \aleph_{4,n-3} & \aleph_{4,n-2} & \aleph_{4,n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \aleph_{n-3,1} & \aleph_{n-3,2} & \aleph_{n-3,3} & \aleph_{n-3,4} & \cdots & \rho_{n-3,n-3} & \theta_{n-3,n-2} & \beth_{n-3,n-1} \\ \aleph_{n-2,1} & \aleph_{n-2,2} & \aleph_{n-2,3} & \aleph_{n-2,4} & \cdots & \tau_{n-2,n-3} & \rho_{n-2,n-2} & \theta_{n-2,n-1} \\ \aleph_{n-1,1} & \aleph_{n-1,2} & \aleph_{n-1,3} & \aleph_{n-1,4} & \cdots & \beth_{n-1,n-3} & \tau_{n-1,n-2} & -\frac{1}{\gamma_4} + \rho_{n-1,n-1} \end{bmatrix}$$

in which $\rho_{i,i} = 6 - 2\gamma^2 S_i + \gamma^2 T_i - \gamma^4 A_i k_{i,j}$, $\rho_{1,1} = 1 + \rho_{1,1}$, $\theta_{i,j} = -4 + \gamma^2 S_i - \gamma^4 A^{jki,j}$, $\beth_{i,j} = 1 - \gamma^4 A_{jki,j}$ and $\aleph_{i,j} = -\gamma^4 A_j k_{i,j}$ with the order of the matrix is $(n-1) \times (n-1)$.

$$\tilde{X} = \begin{bmatrix} \gamma^4 f_1 + \gamma^2 a_2 + (6 - \gamma^2 S_0 - \aleph_{1,0})G_0 - (\aleph_{1,n})G_n \\ \gamma^4 f_2 + (-1 - \aleph_{2,0})G_0 - (\aleph_{2,n})G_n \\ \gamma^4 f_3 + (\aleph_{3,0})G_0 - (\aleph_{3,n})G_n \\ \vdots \\ \gamma^4 f_{n-3} + (\aleph_{n-3,0})G_0 - (\aleph_{n-3,n})G_n \\ \gamma^4 f_{n-2} + (\aleph_{n-2,0})G_0 + (-1 - \aleph_{n-2,n})G_n \\ \gamma^4 f_{n-1} + \gamma^2 b_2 - (\aleph_{n-1,0})G_0 + (6 - \gamma^2 S_{n-1} - \aleph_{n-1,n})G_n \end{bmatrix} \text{ and}$$

$$\widetilde{V} = \begin{bmatrix} G(x_1) \\ G(x_2) \\ G(x_3) \\ \vdots \\ G(x_{n-3}) \\ G(x_{n-2}) \\ G(x_{n-1}) \end{bmatrix}$$

The linear system of equations is then solved by FSGS and FSQM iterative method.

## FORMULATION OF FULL-SWEEP QUADRATIC MEAN ITERATIVE METHOD

Basically, FSQM iterative method is similar to AM and GM methods which involved two levels of calculation which if forward iteration, $\emptyset_f$ and backward iteration, $\emptyset_b$. These two independent systems are created by rewriting the coefficient matrix $Y$ in the form of

$$Y = D - L - U \qquad (6)$$

where $D$, $L$ and $U$ are:

Second order FIDEs:

$$D = \begin{bmatrix} \kappa & 0 & \cdots & 0 & 0 \\ 0 & \kappa & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \kappa & 0 \\ 0 & 0 & \cdots & 0 & \kappa \end{bmatrix}$$

where $\kappa = \kappa_{i,i}$.

$$L = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \varrho_{2,1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \aleph_{3,1} & \varrho_{3,2} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \omega_{n-3,1} & \omega_{n-3,2} & \omega_{n-3,3} & \cdots & 0 & 0 & 0 \\ \omega_{n-2,1} & \omega_{n-2,2} & \omega_{n-2,3} & \cdots & \varrho_{n-2,n-3} & 0 & 0 \\ \omega_{n-1,1} & \omega_{n-1,2} & \omega_{n-1,3} & \cdots & \omega_{n-1,n-3} & \varrho_{n-1,n-2} & 0 \end{bmatrix}$$

and

$$U = \begin{bmatrix} 0 & \zeta_{1,2} & \omega_{3,1} & \cdots & \omega_{1,n-3} & \omega_{1,n-2} & \omega_{1,n-1} \\ \varrho_{2,1} & 0 & \zeta_{2,3} & \cdots & \omega_{2,n-3} & \omega_{2,n-2} & \omega_{2,n-1} \\ \aleph_{3,1} & \varrho_{3,2} & 0 & \cdots & \omega_{3,n-3} & \omega_{3,n-2} & \omega_{3,n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \zeta_{n-3,n-2} & \omega_{n-3,n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & \zeta_{n-2,n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}$$

Fourth order FIDEs:

$$D = \begin{bmatrix} -\frac{1}{h_4} + \rho & 0 & \cdots & 0 & 0 \\ 0 & \rho & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \rho & 0 \\ 0 & 0 & \cdots & 0 & -\frac{1}{h_4} + \rho \end{bmatrix}$$

where $\rho = \rho_{i,i}$.

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \tau_{2,1} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \beth_{3,1} & \tau_{3,2} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \aleph_{3,1} & \beth_{4,2} & \tau_{4,3} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \aleph_{n-3,1} & \aleph_{n-3,2} & \aleph_{n-3,3} & \aleph_{n-3,4} & \cdots & 0 & 0 & 0 \\ \aleph_{n-2,1} & \aleph_{n-2,2} & \aleph_{n-2,3} & \aleph_{n-2,4} & \cdots & \tau_{n-2,n-3} & 0 & 0 \\ \aleph_{n-1,1} & \aleph_{n-1,2} & \aleph_{n-1,3} & \aleph_{n-1,4} & \cdots & \beth_{n-1,n-3} & \tau_{n-1,n-2} & 0 \end{bmatrix}$$

and

$$U = \begin{bmatrix} 0 & \theta_{1,2} & \beth_{1,3} & \aleph_{1,4} & \cdots & \aleph_{1,n-3} & \aleph_{1,n-2} & \aleph_{1,n-1} \\ 0 & 0 & \theta_{2,3} & \beth_{2,4} & \cdots & \aleph_{2,n-3} & \aleph_{2,n-2} & \aleph_{2,n-1} \\ 0 & 0 & 0 & \theta_{3,4} & \cdots & \aleph_{3,n-3} & \aleph_{3,n-2} & \aleph_{3,n-1} \\ 0 & 0 & 0 & 0 & \cdots & \aleph_{4,n-3} & \aleph_{4,n-2} & \aleph_{4,n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \theta_{n-3,n-2} & \beth_{n-3,n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \theta_{n-2,n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}$$

The general formulation of the FSQM iterative method is displayed herewith:

$$\left.\begin{aligned} (D-\alpha L)\emptyset_f &= \big((1-\alpha)D-\alpha U\big)\emptyset_f + \alpha f \\ (D-\alpha U)\emptyset_b &= \big((1-\alpha)D-\alpha L\big)\emptyset_b + \alpha f \\ \emptyset &= \left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}} \end{aligned}\right\} \tag{7}$$

where $\alpha$ represents the optimal parameters. The $i$th element of vector $\emptyset$ is given as follows:

i) $\left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}}$, if $\emptyset_f > 0$ and $\emptyset_b > 0$

ii) $-1 \times \left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}}$, if $\emptyset_f < 0$ and $\emptyset_b < 0$

iii) $\emptyset_f - \left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}}$, if $\emptyset_f > 0$ and $\emptyset_b < 0$

iv) $\emptyset_b - \left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}}$, if $\emptyset_f < 0$ and $\emptyset_b > 0$

Based on these formulation from above, iterative form of the FSQM method for solving linear system of equations is

$$\emptyset^{k+1} = Y\emptyset^k + z, \qquad k = 0,1,2,\ldots \tag{8}$$

with

$$Y = \left[\frac{L_1^2+L_2^2}{2}\right]^{\frac{1}{2}}$$

and

$$z = cf$$

where

$$L_1 = (D-\alpha L)^{-1}\big((1\text{-}\alpha)D-\alpha U\big)$$

$$L_2 = (D-\alpha U)^{-1}\big((1\text{-}\alpha)D-\alpha L\big)$$

$$c = \left[\frac{\alpha(D-\alpha L)^{-2}+\alpha(D-\alpha U)^{-2}}{2}\right]^{\frac{1}{2}}$$

The general condition which assures the FSQM iterative method to converge in solving linear systems is proved below with some theorems.

*Theorem 1* Let $Y$ be an $n \times n$ nonsingular diagonally dominant matrix, the components $\kappa_{i,i} > 0$. For $i = 1,2,\ldots, n$, and by using matrix splitting,

$$Y = H_1 - K_1 = H_2 - K_2$$

$$Q = (H_r)^{-1} K_r, r = 1,2$$

where matrices $(H_1)^{-1}$ and $(H_2)^{-1}$ are nonsingular with $\|(H_1)^{-1}\| \geq 0$, $\|(K_1)^{-1}\| \geq 0$, and $\|(H_2)^{-1}\| \geq 0$, $\|(K_2)^{-1}\| \geq 0$. The FSQM iterative scheme in Equation (8) converges for the optimal relaxation parameter α in the interval $0 < \alpha < 2$.

*Proof.* By hypothesis, $Y_n$ is an $n \times n$ nonsingular matrix. Since $H_1 = D - \alpha L$ and $H_2 = D - \alpha U$ are strictly diagonally dominant for $0 < \alpha < 2$ .

The matrices $K_1 = (1 - \alpha)\, D + \alpha U$ and $K_2 = (1 - \alpha)\, D + \alpha L$ are triangular and nonnegative.

Since

$$H_1 - K_1 = H_2 - K_2 = Y$$

Then we have

$$Q = \left\{\frac{1}{2}[(H_1)^{-1}K_1]^2 + \frac{1}{2}[(H_2)^{-1}K_2]^2\right\}^{\frac{1}{2}} = I - \left[\frac{1}{2}(H_1)^{-2} + \frac{1}{2}(H_2)^{-2}\right]^{\frac{1}{2}} Y$$

or also can be written as

$$\left[\frac{1}{2}(H_1)^{-2} + \frac{1}{2}(H_2)^{-2}\right]^{\frac{1}{2}} = (I-Q)(Y)^{-1}$$

The proof of the theorem runs parallel to the standard proof given in (Ortega 1973). Since $Q = (H_r)^{-1} K_r$, then the spectral radius is

$$\rho_{QM}(Q) < 1.$$

Therefore, the FSQM iterative method converges for any initial vector $\widetilde{\psi}^{(0)}$ with conditions of $0 < \alpha < 2$. Hence, Theorem 1 is proved.

The algorithm for solving FIDEs by using FDCT, FDCS1 and FDCS2 schemes and FSQM iterative method is stated below:

Step 1: Set $\emptyset_f = \emptyset_b = 0$ and $max = 10000$

Step 2: Iteration cycle

```
    for $k = 0,1,2, \ldots, max$
    for $i = u, 2u, \ldots, n - u$, compute
```

$$\emptyset_{f_i} = (1-\alpha)\emptyset_i^{(k)} + \frac{\alpha}{\omega_{i,i}}\left[f_i - \sum_{j=u,2u}^{i-u} A_{i,j}\emptyset_{f_j} - \sum_{j=i+u,i+2u,i+3u}^{N-p} A_{i,j}\,\emptyset_j^{(k)}\right]$$

for $i = u,\ 2u,\ \ldots, n - u$, compute

$$\emptyset_{b_i} = (1-\alpha)\emptyset_i^{(k)} + \frac{\alpha}{\omega_{i,i}}$$

$$\left[\hat{f}_i - \sum_{j=u,2u}^{i-u} A_{i,j}\emptyset_j^k - \sum_{j=i+u,i+2u,i+3u}^{n-u} A_{i,j}\,\emptyset_{b_j}\right]$$

for $i = u,\ 2u,,\ \ldots, n - u$, compute

$$\emptyset_i^{(k+1)} = \begin{cases} \left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}}, \text{if } \emptyset_f > 0 \text{ and } \emptyset_b > 0 \\ -1 * \left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}}, \text{if } \emptyset_f < 0 \text{ and } \emptyset_b < 0 \\ \emptyset_f - \left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}}, \text{if } \emptyset_f > 0 \text{ and } \emptyset_b < 0 \\ \emptyset_b - \left(\frac{\emptyset_f^2+\emptyset_b^2}{2}\right)^{\frac{1}{2}}, \text{if } \emptyset_f < 0 \text{ and } \emptyset_b > 0 \end{cases}$$

Step 3: If the convergence meets the requirement, proceed to Step 4.

Step 4: Stop.

## NUMERICAL SIMULATIONS

In this section, each example of the questions of second and fourth order FIDEs are selected for experimental calculations in order to analyze the efficiency of the proposed full- and half-sweep QM iterative method with FDCT, FDCS1, and FDCS2 discretization schemes. For studying the effectiveness of the proposed methods, three parameters are measured i.e., number of iterations, $N$, execution time, $t$, and maximum absolute error, $\varepsilon_N$.

**Problem 1** (Filiz 2000)

Given the second order FIDE

$$G''(x) = x - 2 + \int_0^1 60(x-t)G(t)dt$$

with the boundary conditions

$$G(0) = 0, \quad G(1) = 0$$

The exact solution for the problem 1 is

$$G(x) = x(x-1)^2.$$

**Problem 2** (Ullah 2015)

Given the fourth order FIDE

$$G''''(x) - G(x) + x(1+e^x) - 3e^x + \int_0^1 G(t)dt = 0$$

with the boundary conditions

$$G(0) = 1, \quad G(1) = 1 + e$$

$$G''(0) = 2, \quad G''(1) = 3e$$

The exact solution for the problem 2 is

$$G(x) = 1 + xe^x.$$

Throughout the simulations, the maximum tolerance absolute error is set to the range of $\varepsilon = 10^{-10}$. All the numerical simulations have been calculated in a computer with processor AMD Ryzen 5 5600H CPU @ 3.30GHz and all the algorithm codes will be written in Borland C++ programming language. The mesh sizes for the second order FIDE are 120, 240, 480, 960, 1200 while for the fourth order FIDE are 12, 24, 48, 96, 120.

## RESULTS AND DISCUSSIONS

Based on the numerical results in Table 2 and 3, it shows that the proposed FSQM and HSQM methods have a better result for solving second- and fourth-order FIDEs (Problem 1&2) compared to the FSGS method across all three discretization schemes (i.e., FDCT, FDCS1, and FDCS2). This is proven by the percentage reduction calculation as shown in Table 4. The FSQM method showed a significant reduction in the number of iterations and execution time, with decreases ranging from approximately 84.84% to 86.51% and 64.80% to 69.03%, respectively, across all three discretization schemes compared to the FSGS method for solving the second-order FIDE (Problem 1). Similarly, the HSQM method demonstrated even greater improvements, with the number of iterations and execution time reduced by approximately 95.42% to 97.61% and 83.33% to 99.53%, respectively, across the same discretization schemes, compared to the FSGS method. For Problem 2, which involves solving the fourth-order FIDE, the FSQM method achieved a decrease in the number of iterations and execution time by approximately 86.48% to 88.76% and 71.43% to 87.50%, respectively, across all discretization schemes, when compared to the FSGS method. In contrast, the HSQM method yielded even more impressive reductions, with the number of iterations and execution time decreasing by approximately 94.34% to 99.15% and 83.33% to 99.53%, respectively, across all three discretization schemes, relative to the FSGS method.

Although the FSQM and HSQM methods achieve the same level of accuracy as the FSAM and HSAM methods, they require significantly less execution time, especially for larger mesh sizes. This trend is also observed in the performance of the HSQM, HSGS, and HSAM iterative methods, as shown in the tables. Specifically, both the FSQM and HSQM methods are more efficient, as they achieve the same accuracy as the FSAM method but with reduced computation time, making them particularly advantageous when dealing with larger mesh sizes.

TABLE 2. Numerical results for the FSGS, HSGS, FSQM, HSQM, FSAM and HSAM iterative methods (Problem 1)

| Methods | FDCT | | | | | FDCS1 | | | | | FDCS2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Number of iterations | | | | | Number of iterations | | | | | Number of iterations | | | | |
| | Mesh sizes | | | | | Mesh sizes | | | | | Mesh sizes | | | | |
| | 120 | 240 | 480 | 960 | 1200 | 120 | 240 | 480 | 960 | 1200 | 120 | 240 | 480 | 960 | 1200 |
| FSGS | 12278 | 45129 | 162727 | 576449 | 862679 | 12276 | 45127 | 162725 | 576448 | 862677 | 12276 | 45127 | 162725 | 576448 | 862677 |
| HSGS | 3251 | 12278 | 45129 | 162727 | 244992 | 3249 | 12276 | 45127 | 162725 | 244990 | 3249 | 12276 | 45127 | 162725 | 244990 |
| FSAM | 1861 | 6296 | 21951 | 78095 | 117538 | 1860 | 6296 | 21951 | 78095 | 117538 | 1860 | 6296 | 21951 | 78095 | 117538 |
| HSAM | 562 | 1861 | 6296 | 21951 | 32998 | 562 | 1860 | 6296 | 21951 | 32998 | 562 | 1860 | 6296 | 21951 | 32998 |
| FSQM | 1861 | 6296 | 21951 | 78095 | 117538 | 1860 | 6296 | 21951 | 78095 | 117538 | 1860 | 6296 | 21951 | 78095 | 117538 |
| HSQM | 562 | 1861 | 6296 | 21951 | 32998 | 562 | 1860 | 6296 | 21951 | 32998 | 562 | 1860 | 6296 | 21951 | 32998 |
| Methods | Execution time (s) | | | | | Execution time (s) | | | | | Execution time (s) | | | | |
| | Mesh sizes | | | | | Mesh sizes | | | | | Mesh sizes | | | | |
| | 120 | 240 | 480 | 960 | 1200 | 120 | 240 | 480 | 960 | 1200 | 120 | 240 | 480 | 960 | 1200 |
| FSGS | 1.71 | 21.62 | 293.30 | 4037.54 | 9372.67 | 1.96 | 24.44 | 327.85 | 4460.34 | 10328.91 | 1.80 | 21.46 | 314.99 | 3939.65 | 9099.41 |
| HSGS | 0.19 | 1.72 | 21.99 | 299.83 | 695.46 | 0.20 | 1.95 | 25.02 | 330.29 | 761.59 | 0.19 | 1.75 | 21.82 | 323.99 | 702.25 |
| FSAM | 0.62 | 7.69 | 99.69 | 1401.99 | 3225.44 | 0.69 | 8.38 | 115.36 | 1824.83 | 3578.93 | 0.66 | 8.10 | 106.39 | 1479.35 | 3503.72 |
| HSAM | 0.07 | 0.62 | 7.54 | 98.87 | 231.19 | 0.08 | 0.68 | 8.32 | 114.68 | 254.22 | 0.07 | 0.60 | 7.40 | 100.66 | 226.17 |
| FSQM | 0.58 | 6.74 | 98.62 | 1284.31 | 3018.68 | 0.69 | 8.60 | 105.42 | 1470.90 | 3441.03 | 0.59 | 6.99 | 97.54 | 1321.59 | 3055.51 |
| HSQM | 0.06 | 0.59 | 6.91 | 93.20 | 216.64 | 0.06 | 0.67 | 8.07 | 110.57 | 246.59 | 0.07 | 0.58 | 7.12 | 94.88 | 218.73 |
| Methods | Maximum absolute error | | | | | Maximum absolute error | | | | | Maximum absolute error | | | | |
| | Mesh sizes | | | | | Mesh sizes | | | | | Mesh sizes | | | | |
| | 120 | 240 | 480 | 960 | 1200 | 120 | 240 | 480 | 960 | 1200 | 120 | 240 | 480 | 960 | 1200 |
| FSGS | 1.855E-5 | 4.333E-6 | 4.910E-7 | 5.249E-6 | 8.471E-6 | 7.967E-8 | 3.408E-7 | 1.377E-6 | 5.533E-6 | 8.653E-6 | 7.484E-8 | 3.405E-7 | 1.377E-6 | 5.533E-6 | 8.065E-6 |
| HSGS | 1.338E-4 | 3.409E-5 | 8.605E-6 | 2.165E-6 | 2.233E-6 | 1.372E-4 | 3.451E-5 | 8.657E-6 | 2.669E-6 | 2.904E-6 | 1.372E-4 | 3.451E-5 | 8.657E-6 | 2.669E-6 | 2.904E-6 |
| FSAM | 1.862E-5 | 4.615E-6 | 1.008E-6 | 3.434E-7 | 7.909E-7 | 7.985E-9 | 4.331E-8 | 1.621E-7 | 6.253E-7 | 9.724E-7 | 7.985E-9 | 4.331E-8 | 1.621E-7 | 6.253E-7 | 9.724E-7 |
| HSAM | 1.338E-4 | 3.4086E-5 | 8.6011E-6 | 2.1610E-6 | 1.385E-6 | 1.372E-4 | 3.451E-5 | 8.654E-6 | 2.18E-6 | 1.388E-6 | 1.372E-4 | 3.451E-5 | 8.654E-6 | 2.168E-6 | 1.388E-6 |
| FSQM | 1.862E-5 | 4.615E-6 | 1.008E-6 | 3.434E-7 | 7.909E-7 | 7.985E-9 | 4.331E-8 | 1.621E-7 | 6.253E-7 | 9.724E-7 | 3.202E-9 | 4.301E-8 | 1.620E-7 | 6.253E-7 | 9.724E-7 |
| HSQM | 1.338E-4 | 3.409E-5 | 8.601E-6 | 2.161E-6 | 1.385E-6 | 1.372E-4 | 3.451E-5 | 8.654E-6 | 2.168E-6 | 1.388E-6 | 1.372E-4 | 3.451E-5 | 8.654E-6 | 2.168E-6 | 1.388E-6 |

TABLE 3. Numerical results for the FSGS, HSGS, FSQM, HSQM, FSAM and HSAM iterative methods (Problem 2)

| Methods | FDCT | | | | | FDCS1 | | | | | FDCS2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Number of iterations | | | | | Number of iterations | | | | | Number of iterations | | | | |
| | Mesh sizes | | | | | Mesh sizes | | | | | Mesh sizes | | | | |
| | 12 | 24 | 48 | 96 | 120 | 12 | 24 | 48 | 96 | 120 | 12 | 24 | 48 | 96 | 120 |
| FSGS | 2510 | 33036 | 434422 | 5515068 | 12346958 | 2510 | 33036 | 434422 | 5515068 | 12346962 | 2510 | 33036 | 434422 | 5515068 | 12346962 |
| HSGS | 207 | 2510 | 33036 | 434422 | 989538 | 207 | 2510 | 33036 | 434422 | 989538 | 207 | 2510 | 33036 | 434422 | 989538 |
| FSAM | 336 | 3712 | 54779 | 733074 | 1669521 | 336 | 3712 | 54779 | 733075 | 1669520 | 336 | 3712 | 54779 | 733075 | 1669520 |
| HSAM | 142 | 336 | 3713 | 54800 | 127141 | 142 | 336 | 3712 | 54779 | 127095 | 142 | 336 | 3712 | 54779 | 127095 |
| FSQM | 336 | 3712 | 54779 | 733074 | 1669520 | 336 | 3712 | 54779 | 733075 | 1669520 | 336 | 3712 | 54779 | 733075 | 1669519 |
| HSQM | 142 | 336 | 3712 | 54779 | 127095 | 142 | 336 | 3712 | 54779 | 127095 | 142 | 336 | 3712 | 54779 | 127095 |
| Methods | Execution time (s) | | | | | Execution time (s) | | | | | Execution time (s) | | | | |
| | Mesh sizes | | | | | Mesh sizes | | | | | Mesh sizes | | | | |
| | 12 | 24 | 48 | 96 | 120 | 12 | 24 | 48 | 96 | 120 | 12 | 24 | 48 | 96 | 120 |
| FSGS | 0.06 | 0.91 | 20.99 | 725.74 | 2438.67 | 0.08 | 1.09 | 25.40 | 842.17 | 2740.19 | 0.07 | 1.13 | 25.41 | 793.23 | 2589.59 |
| HSGS | 0.01 | 0.08 | 1.15 | 26.79 | 79.46 | 0.01 | 0.08 | 1.11 | 25.67 | 76.13 | 0.01 | 0.08 | 1.09 | 25.16 | 72.24 |
| FSAM | 0.01 | 0.15 | 3.96 | 150.33 | 516.87 | 0.02 | 0.17 | 4.40 | 169.76 | 570.84 | 0.01 | 0.15 | 4.12 | 162.91 | 561.80 |
| HSAM | 0.01 | 0.01 | 0.16 | 3.90 | 12.39 | 0.01 | 0.02 | 0.16 | 4.26 | 13.81 | 0.07 | 1.13 | 25.41 | 793.23 | 2589.59 |
| FSQM | 0.01 | 0.12 | 3.59 | 137.21 | 467.53 | 0.01 | 0.15 | 4.10 | 160.02 | 543.61 | 0.02 | 0.15 | 3.75 | 144.82 | 483.30 |
| HSQM | 0.01 | 0.01 | 0.16 | 4.16 | 12.65 | 0.01 | 0.01 | 0.16 | 4.26 | 13.70 | 0.01 | 0.01 | 0.16 | 25.16 | 12.24 |
| Methods | Maximum absolute error | | | | | Maximum absolute error | | | | | Maximum absolute error | | | | |
| | Mesh sizes | | | | | Mesh sizes | | | | | Mesh sizes | | | | |
| | 12 | 24 | 48 | 96 | 120 | 12 | 24 | 48 | 96 | 120 | 12 | 24 | 48 | 96 | 120 |
| FSGS | 1.130E-3 | 5.117E-4 | 6.746E-4 | 7.710E-4 | 8.505E-4 | 1.132E-3 | 5.109E-4 | 6.744E-4 | 7.710E-4 | 8.505E-4 | 1.132E-3 | 5.109E-4 | 6.744E-4 | 7.710E-4 | 8.505E-4 |
| HSGS | 2.784E-2 | 7.082E-3 | 1.771E-3 | 6.746E-4 | 7.018E-4 | 2.785E-2 | 7.082E-3 | 1.771E-3 | 6.715E-4 | 6.948E-4 | 2.785E-2 | 7.082E-3 | 1.771E-3 | 6.744E-4 | 7.017E-4 |
| FSAM | 1.130E-3 | 5.115E-4 | 6.717E-4 | 7.256E-4 | 7.398E-4 | 1.132E-3 | 5.107E-4 | 6.715E-4 | 7.256E-4 | 7.397E-4 | 1.132E-3 | 5.107E-4 | 6.715E-4 | 7.256E-4 | 7.397E-4 |
| HSAM | 2.817E-2 | 7.266E-3 | 2.061E-3 | 9.153E-4 | 7.905E-4 | 2.785E-2 | 7.082E-3 | 1.771E-3 | 6.715E-4 | 6.948E-4 | 2.785E-2 | 7.082E-3 | 1.771E-3 | 6.715E-4 | 6.948E-4 |
| FSQM | 1.130E-3 | 5.115E-4 | 6.717E-4 | 7.256E-4 | 7.398E-4 | 1.132E-3 | 5.107E-4 | 6.715E-4 | 7.256E-4 | 7.397E-4 | 1.132E-3 | 5.107E-4 | 6.715E-4 | 7.256E-4 | 7.397E-4 |
| HSQM | 2.784E-2 | 7.082E-3 | 1.771E-3 | 6.717E-4 | 6.949E-4 | 2.785E-2 | 7.082E-3 | 1.771E-3 | 6.715E-4 | 6.948E-4 | 2.788E-2 | 7.082E-3 | 1.771E-3 | 6.715E-4 | 6.948E-4 |

Overall, the results clearly highlight the computational advantages of the FSQM and HSQM methods over the FSGS method. Both FSQM and HSQM methods consistently achieve significant reductions in execution time and iterations, while maintaining comparable numerical accuracy. This improvement is attributed to the QM method, which incorporates higher-order corrections into the iteration process, enhancing the stability and accuracy of the updates. As a result, errors are reduced more rapidly across the grid. The superiority of the QM method is evident across various discretization schemes and problem configurations, establishing it as an efficient solution for solving dense linear systems. These findings advocate for the use of the QM method, especially when combined with efficient schemes like FDCS2, as the preferred approach for solving second- and fourth-order FIDEs. Among the methods discussed, the HSQM iterative method emerges as the most efficient in terms of both the number of iterations and execution time. To provide better clarity, the performance of the proposed methods is illustrated in Figures 1 to 4. These figures present a detailed analysis of the number of iterations and execution time in relation to mesh size.

## INNOVATIONS AND PERFORMANCE OF THE FSQM/HSQM METHODS

The FSQM iterative method represents a significant departure from conventional iterative approaches through its sign-sensitive quadratic mean formulation. Unlike the FSGS method which relies on unidirectional updates without combining iterative sweeps or the FSAM method which has linear averaging $(\phi = (\phi_f + \phi_b)/2)$, the FSQM method dynamically weights forward $(\phi_f)$ and backward $(\phi_b)$ sweeps based on their sign alignment. This innovation employs a non-linear fusion: when both sweeps agree in sign, it computes their root-mean-square; when they conflict, it applies bias-correction terms. This approach enhances stability by amplifying consensus between sweeps and suppressing oscillatory errors, a limitation of simpler averaging techniques.

Further efficiency is achieved via the HSQM variant, which optimizes computational load by updating only every second grid point ($i = u, 2u, ..., n - u$). This strategy reduces operations per iteration by ≈ 50% while maintaining accuracy comparable to full-sweep methods (Tables $2 - 3$). The half-sweep paradigm also improves memory efficiency by minimizing data access in dense matrices, making it particularly advantageous for large-scale systems. Theoretically, the authors establish rigor through Theorem 1, which guarantees convergence for the relaxation parameter $\alpha \in (0,2)$, and a novel matrix decomposition ($Y = D - L - U$) that updates forward/backward sweeps in Equation (7). This foundation ensures robustness across diverse problem configurations.

Empirical results underscore the superiority of HSQM in practical applications. In speed and efficiency, the methods reduce iterations by 86-89% and execution time by 67-88% compared to FSGS across mesh sizes (Table 4, Figures 2, 4). They also outperform FSAM by

TABLE 4. Percentage reduction in number of iterations and execution time for FSQM and HSQM methods compared to the FSGS

| Methods | Problem 1 | | | | | |
|---|---|---|---|---|---|---|
| | FDCT | | FDCS1 | | FDCS2 | |
| | Iterations | Execution time | Iterations | Execution time | Iterations | Execution time |
| FSQM | 84.84 - 86.51% | 66.08 - 68.83% | 84.85 - 86.51% | 64.80 - 67.85% | 84.85 - 86.51% | 66.42 - 69.03% |
| HSQM | 95.42 - 96.19% | 96.49 - 97.69% | 95.42 - 96.19% | 96.49 - 97.61% | 95.42 - 97.61% | 96.11 - 97.74% |
| Methods | Problem 2 | | | | | |
| | FDCT | | FDCS1 | | FDCS2 | |
| | Iterations | Execution time | Iterations | Execution time | Iterations | Execution time |
| FSQM | 86.48 - 88.76% | 80.83 - 86.81% | 86.48 - 88.76% | 80.16 - 87.50% | 86.48 - 88.76% | 71.43 - 86.73% |
| HSQM | 94.34 - 99.15% | 83.33 - 99.24% | 94.34 - 99.15% | 87.50 - 99.50% | 94.34 - 99.15% | 85.71 - 99.53% |

TABLE 5. Summary of comparative advantages

| Method | Iterations | Speed | Memory use | Accuracy | Innovation |
|---|---|---|---|---|---|
| GS/HSGS | High | Slow | Moderate | Good | Baseline |
| AM/HSAM | Medium | Medium | Moderate | Good | Arithmetic mean |
| QM/HSQM (Proposed) | Lowest | Fastest | Efficient | Equal | Quadratic mean + half-sweep |

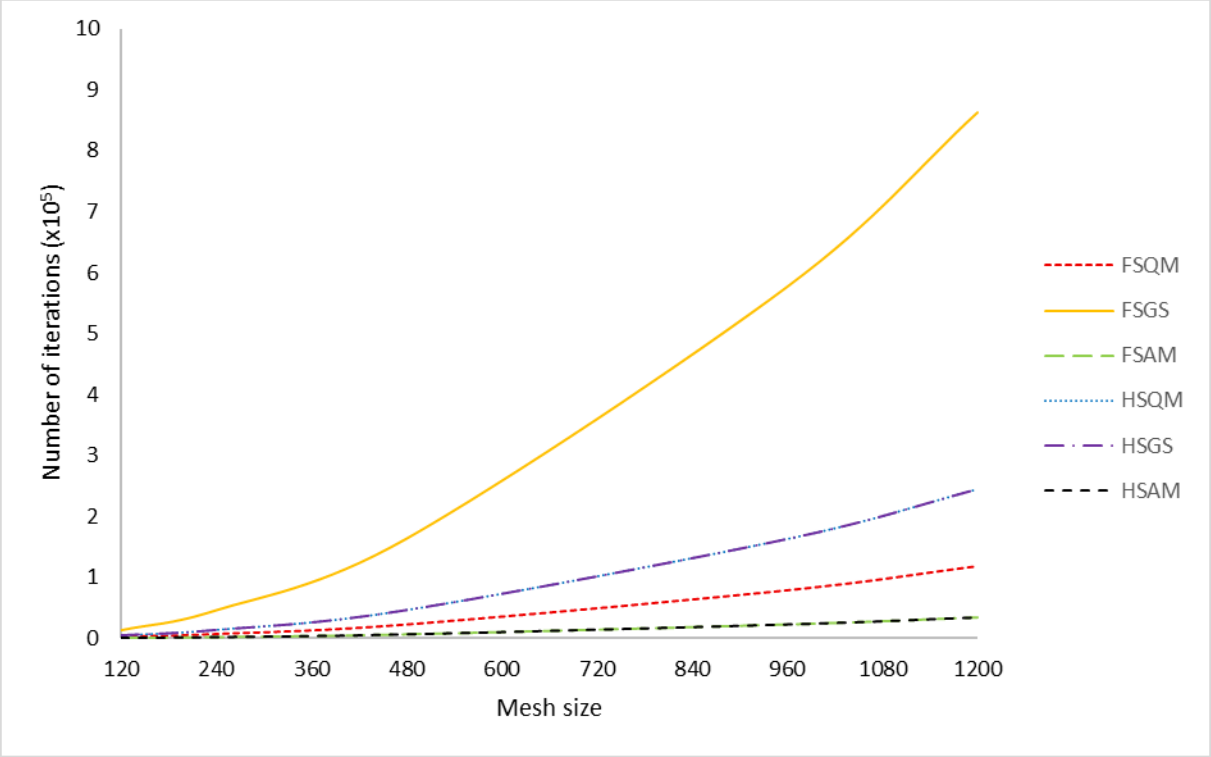

FIGURE 1. Number of iterations versus mesh size of the iterative methods used to solve Problem 1

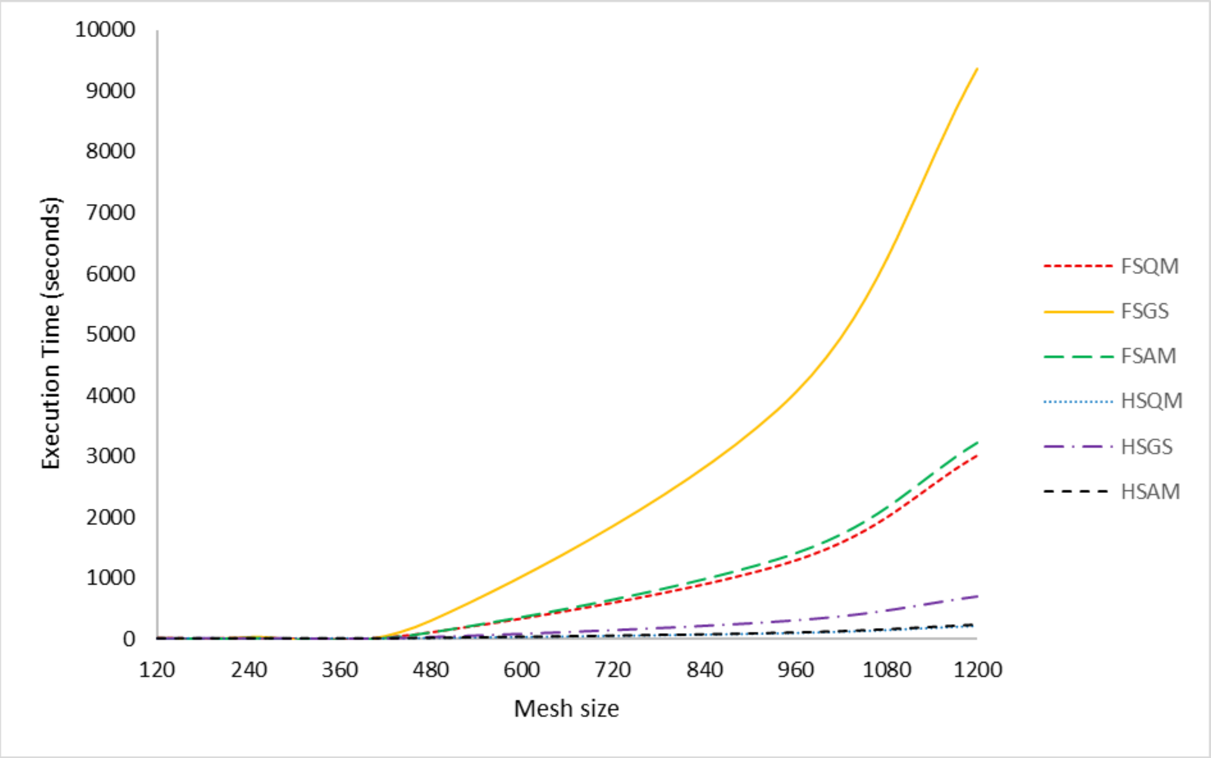

FIGURE 2. Execution time versus mesh size of the iterative methods used to solve Problem 1

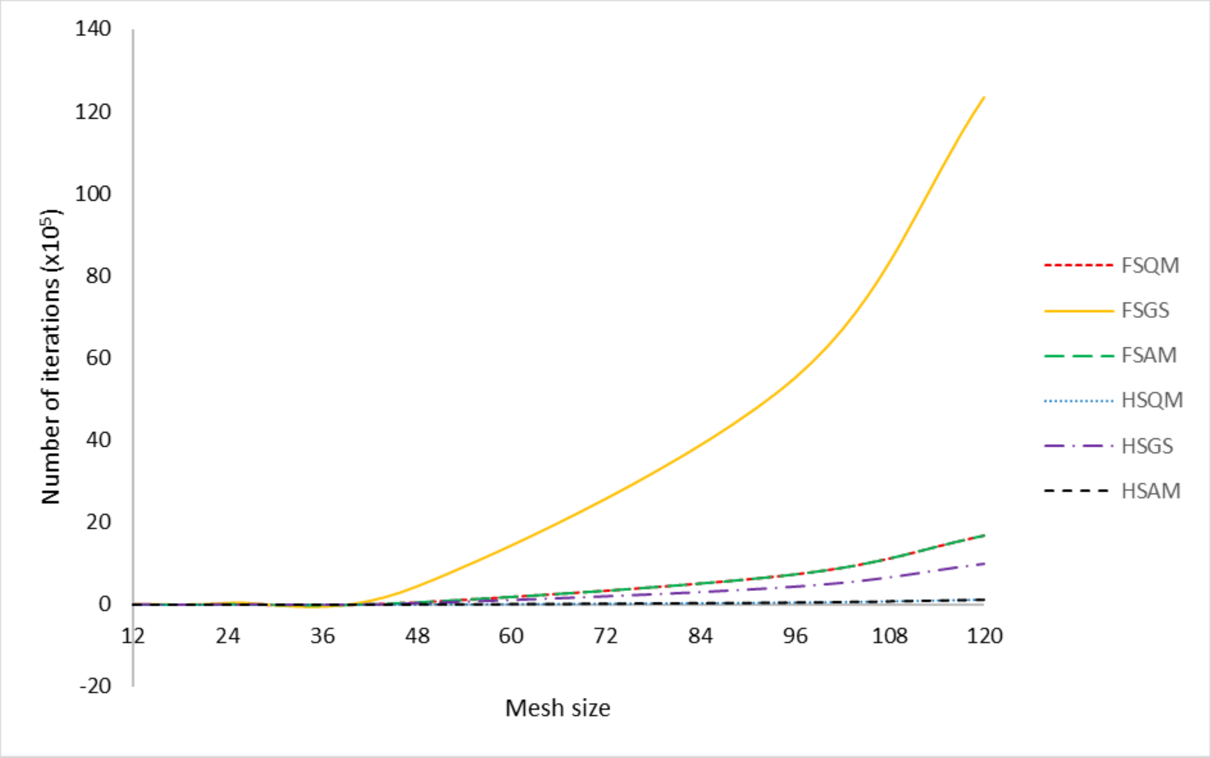

FIGURE 3. Number of iterations versus mesh size of the iterative methods used to solve Problem 2

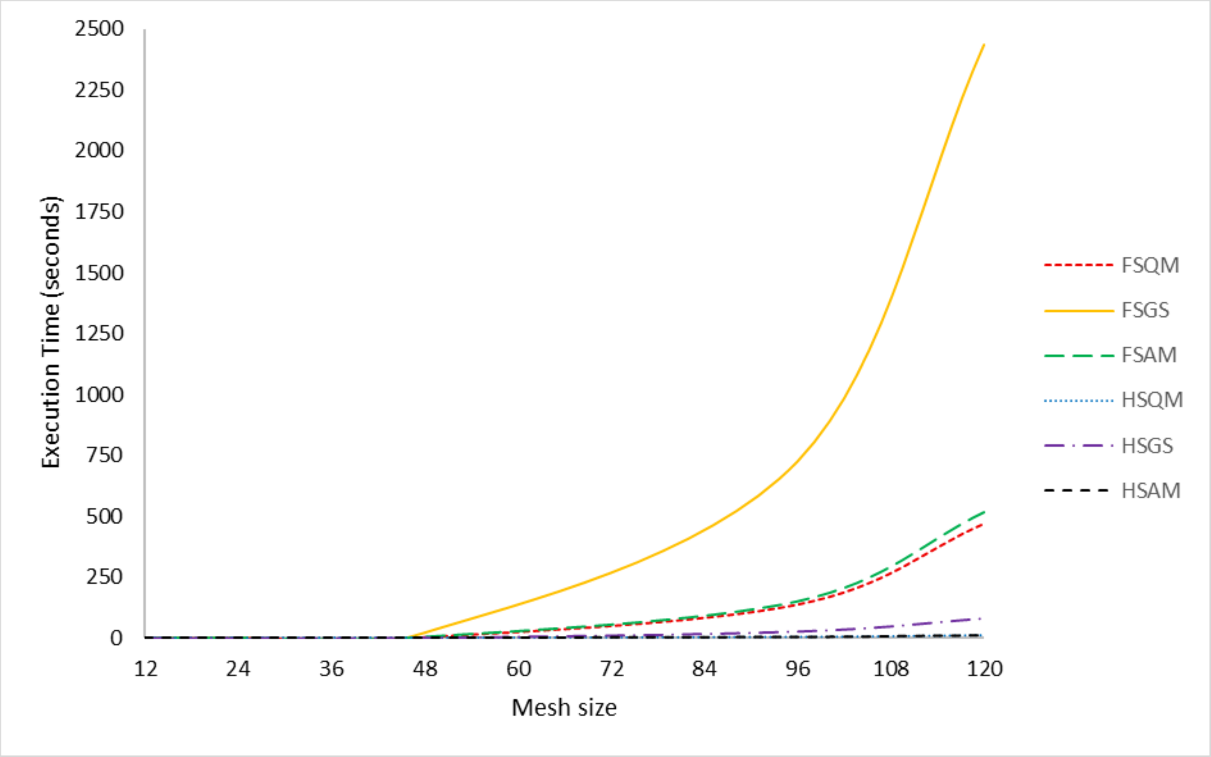

FIGURE 4. Execution time versus mesh size of the iterative methods used to solve Problem 2

5-10% in runtime while matching accuracy-attributed to the error-cancelling properties of quadratic fusion. Scalability is demonstrated in large-mesh scenarios: for Problem 1 (1,200 nodes), HSQM+FDCT solves in 216 seconds versus FSGS's 9,372 seconds (43× faster); for Problem 2, HSQM+FDCT finishes in 12.65 seconds versus FSGS's 2,438 seconds (193× faster). Crucially, these gains do not compromise accuracy preservation, as HSQM maintain error magnitudes ($\epsilon_N \sim 10^{-6}$ - $10^{-9}$) across FDCT, FDCS1, and FDCS2 discretization schemes. The synergy of HSQM with FDCS2 (Composite Simpson's 3/8) emerges as the optimal configuration for fourth-order FIDEs, achieving the lowest iterations and time in Table 3 by combining high-order integral approximation with half-sweep efficiency. The HSQM framework redefines computational efficiency for Fredholm integro-differential equations by merging mathematical innovation (sign-adaptive quadratic averaging) with strategic optimization (half-sweep updates). Its demonstrated advantages in speed, scalability, and accuracy position it as a transformative tool for large-scale scientific simulations. The effectiveness of the methods has been summarized in Table 5 at the Appendix. Future work could explore adaptive α tuning or parallelization to further amplify performance.

## CONCLUSIONS

This paper presents the novel FSQM and HSQM iterative methods for solving dense linear systems derived from the discretization of second- and fourth-order FIDEs. Through comprehensive numerical simulations, it is evident that the FSQM and HSQM methods outperform the traditional FSGS method in terms of computational efficiency. Both methods significantly reduce the number of iterations and execution time, with reductions ranging from 84.84% to 97.74% for the FSQM method and up to 99.53% for the HSQM method across various discretization schemes. The proposed methods, particularly the HSQM variant, offer substantial computational advantages, especially for larger mesh sizes, while maintaining comparable accuracy to existing methods. The efficiency of the FSQM and HSQM methods stems from their novel integration of the quadratic mean formulation and the half-sweep technique, which together enhance the stability, accuracy, and convergence speed of the iterative process.

These findings highlight the potential of the FSQM and HSQM methods as efficient tools for solving high-order integro-differential equations in scientific simulations, particularly in applications requiring large-scale computations. Future research could focus on further optimizing these methods, exploring adaptive tuning or parallelization strategies to extend their applicability to even larger and more complex systems. The success of these methods in improving computational efficiency without sacrificing accuracy marks a significant advancement in numerical solution techniques for Fredholm integro-differential equations.

## REFERENCES

Aihara, K., Ozaki, K. & Mukunoki, D. 2024. Mixed-precision conjugate gradient algorithm using the groupwise update strategy. *Japan Journal Industrial Applied Mathematics* 41: 837-855. https://doi.org/10.1007/s13160-024-00644-8

Aruchunan, E. & Sulaiman, J. 2011. Half-sweep conjugate gardient method for Solving first order linear fredholm integro-differential equations. *Australian Journal of Basic and Applied Sciences* 2: 38-43.

Aruchunan, E & Sulaiman, J. 2013. Half-sweep quadrature-difference schemes with iterative method in solving linear fredholm integro-differential equations. *Progress in Applied Mathematics*. 2: 11-21.

Aruchunan, E., Khajohnsaksumeth, N. & Wiwatanapataphee, B. 2016. A new algorithm of geometric mean for solving high-order fredholm integro-differential equations. 723-729. 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.128.

Aruchunan, E., Muthuvalu, M., Chew, J., Siri, Z. & Sulaiman, J. 2022. Examination of half-sweep closed newton–cotes quadrature schemes in solving dense system. 10.1007/978-3-030-79606-8_26.

Aruchunan, E. & Sulaiman, J. 2010. Numerical solution of second-order linear fredholm integro-differential equation using generalized minimal residual (GMRES) method. *American Journal of Applied Sciences, Science Publication* 7 (6): pp.780-783.

Aruchunan, E. & Sulaiman, J. 2010. Repeated Simpson 3/8 and backward difference schemes to solve first order integro-differential equation. *National Seminar of Science and Computer* (SKSKM 2010). ISBN: 978-967-363- 153-7.

Aruchunan, E. 2016. The New Variants of Modified Weighted Mean Iterative Methods for Fredholm Integro-Differential Equations (Doctoral dissertation, Curtin University).

Aruchunan, E., Wu, Y., Wiwatanapataphee, B. & Jitsangiam, P. 2015. A new variant of arithmetic mean iterative method for fourth order integro-differential equations solution. In 2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS). pp. 82-87. IEEE. doi: 10.1109/AIMS.2015.24.

Aruchunan, E. et al. 2022. Examination of half-sweep closed newton–cotes quadrature schemes in solving dense system. in: Abdul Karim, S.A., Shafie, A. (eds) Towards Intelligent Systems Modeling and Simulation. Studies in Systems, Decision and Control, vol 383. Springer, Cham. https://doi.org/10.1007/978-3-030-79606-8_26

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

Aruchunan, E. & Sulaiman, J. 2012. Comparison of closed repeated newton-cotes quadrature schemes with half-sweep iteration concept in solving linear fredholm integro-differential equations. *International Journal of Science and Engineering Investigations* 1 (9): pp. 90-96.

Avudainayagam, A. & Vani, C. 2000. Wavelet-Galerkin method for integro-differential equations. *Applied Mathematics and Computation*. 32: 247–254.

Benzi, M. & Dayar, T. 1995. The arithmetic mean method for finding the stationary vector of Markov chains. *International Journal of Parallel, Emergent and Distributed Systems* 6(1): pp. 25-37, 1995.

Cai, F., Xiao, J. & Xiang, Z.H. 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. 378-382.2010.

Connolly, C.I., Burns, J.B., Weiss, R. 1990. Path planning using Laplace's equation. Proceedings of the IEEE International Conference on Robotics and Automation: 2102–2106.

Darania, P. & Ebadian, A. 2007. A method for the numerical solution of the integro-differential equations. *Applied Mathematics and Computation*. 188(1): 657-668.

Filiz, A. 2000. Numerical solution of some Volterra integral equations, PhD Thesis, University of Manchester.

Jin S., Ming Z., George E.K. & Zhenya Y. 2024. Two-stage initial-value iterative physics-informed neural networks for simulating solitary waves of nonlinear wave equations. *Journal of Computational Physics* Volume 505 ISSN 0021-9991. https://doi.org/10.1016/j.jcp.2024.112917.

Katuri, S. & Maroju, P. 2025. A new approach for solving fuzzy non-linear equations using higher order iterative method. *Scientific Report* 15: 12972. https://doi.org/10.1038/s41598-025-97612-0

Muthuvalu, M.S., Aruchuan, E. & Sulaiman, J. 2013. Solving first kind linear Fredholm integral equations with semi-smooth kernel using 2-point half-sweep block arithmetic mean method. AIP Conference Proceedings vol. 1557: 350-354.

Muthuvalu, M.S. & Sulaiman, J., 2011. Half-sweep arithmetic mean method with composite trapezoidal scheme for solving linear fredholm integral equations. *Applied Mathematics and Computation* 217(12): 5442-5448..

Ortega, J. 1973. Numerical analysis: a Second Course. *Mathematics of Computation* 27(123): 669. https://doi.org/10.2307/2005671

Rathinasamy, A., & Balachandran, K. 2008. Mean-square stability of milstein method for linear hybrid stochastic delay integro-differential equations. *Nonlinear Analysis: Hybrid Systems*. 2(4): 1256-1263.

Ruggiero, V. & Galligani, E. 1990. An iterativemethod for large sparse systems on a vector computer. *Computers and Mathematics with Applications* 20:25-28.

Sahimi, M.S., Ahmad, A. & Bakar, A.A. 1993. The iterative alternating decomposition explicit (IADE) method to solve the heat conduction equation. *International Journal of Computer Mathematics* 47: 219-229.

Sezer, M. & Kaynak, M. 1977. Chebyshev polynomial solutions of linear differential equations. *International Journal of Mathematical Education in Science and Technology* 14: 607–618.

Sezer, M. 1996. A method for the approximate solution of the second order linear differential equations in terms of Taylor polynomials. *International Journal of Mathematical Education in Science and Technology* 27(4): 821–834.

Sulaiman, J., Othman, M., Yaacob, N. & Hasan. M.K. 2006. Half Sweep Geometric Mean (HSGM) method using fourth-order finite difference scheme for two-point boundary problems. In Proceedings of the First International Conference on Mathematics and Statistics. June 19-21. Bandung, Indonesia, pp. 25-33.

Ullah, M. A. 2015. Numerical integration and a proposed rule. *American Journal of Engineering Research (AJER)* 4(9): 120-123.

Yuhe, R. & Zhang, B. & Qiao, H. 1999. A simple Taylor-series expansion method for a class of second kind integral equations. *Journal of Computational and Applied Mathematics* 110: 15-24. doi: 10.1016/S0377-0427(99)00192-2.

Zhao, J. & Corless, R.M. 2006. Compact finite difference method for integro-differential equations. *Applied Mathematics and Computation* 177: 325-328. doi: 10.1016/j.amc.2005.11.007.

*Corresponding author; email: elayarajah@um.edu.my