

Numerical Evaluation of Quarter-Sweep KSOR Method to Solve Time-Fractional Parabolic Equations

Fatihah Anas Muhiddin¹, Jumat Sulaiman², Andang Sunarto³

¹Faculty Computer and Mathematical Sciences, Universiti Teknologi MARA Caw. Sabah, 88997 Kota Kinabalu, Malaysia

²Mathematics with Economics Programme, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Malaysia

³Faculty of Economic and Islamic Business, IAIN Bengkulu, Indonesia

ABSTRACT

We study the performance of the combination of quarter-sweep iteration concept with the Kauds Successive Over-Relaxation (KSOR) iterative method in solving the discretized one-dimensional time-fractional parabolic equation. We called the mixed of these two concepts as QSKSOR. The time-fractional derivative in Grünwald sense, together with the implicit finite difference scheme was used to discretized the tested problems to form the quarter-sweep implicit finite difference approximation equations in the sense of Grünwald type. This approximation equation of half-sweep will then generate a linear system. Next, we used the proposed QSKSOR iterative method to the generated linear systems before comparing the effectiveness between the other family of KSOR method, FSKSOR and HSKSOR with respect to the full- and half-sweep cases respectively. To do so, three examples are included. The results of this study show the superiority of the QSKSOR iterative method in terms of iteration numbers and execution time in comparison to the other two methods.

Keywords: Grünwald-type, Fractional derivative, Finite difference, Implicit scheme, QSKSOR iteration.

1. INTRODUCTION

The use of fractional partial differential equations (FPDEs) could be found in many areas of applications. Previous researches have mentioned it often. For instance, refer [1]-[3]. Generally, the one-dimensional inhomogeneous time-fractional parabolic equation (TFPE) can be defined as

$$\frac{\partial^\alpha U}{\partial t^\alpha} + p(x) \frac{\partial U}{\partial x} = q(x) \frac{\partial^2 U}{\partial x^2} + f(x, t), \quad (1)$$

$$a \leq x \leq b, \quad 0 \leq t \leq T, \quad 0 < \alpha \leq 1,$$

subject to the following initial and boundary conditions:

$$U(x, 0) = g_1(x), \quad a \leq x \leq b,$$

$$U(a, t) = g_2(t); \quad U(b, t) = g_3(t) \quad 0 < t \leq T,$$

where $p(x)$, $q(x)$ and the function $f(x, t)$ are the convection parameter, diffusion parameter and source term respectively.

Usually, solving FPDEs analytically is difficult, hence many researches have applied the mesh-based numerical techniques such as finite volume, finite element and finite difference methods [1]-[6] to obtain the approximate solution of problem (1). Then, this will lead to sparse linear systems.

Discretization of the time factor in equation (1) needed to be done using fractional operator. In this study, we applied the fractional derivative in the sense of Grünwald type fractional operator, which defined in [7]-[8] as the following

$$D_G^\alpha f(t) = \frac{1}{(\Delta t)^\alpha} \lim_{N \rightarrow \infty} \sum_{k=0}^j g_{\alpha,k} f(t - k\Delta t), \quad 0 < \alpha < 1 \quad (2)$$

$$\text{where } g_{\alpha,k} = \frac{\Gamma(k - \alpha)}{\Gamma(-\alpha)\Gamma(k + 1)}.$$

This study is motivated by the Kauds Successive Over-Relaxation (KSOR) iteration technique which has been developed by [9] in 2012. Apart from that, the application of quarter-sweep iteration concept in this study is inspired from the work done by [10], where they used the Modified Explicit Group (MEG) iterative method to find the numerical solution of two-dimensional Poisson equations. Originally, this work which they modified from their previous work [11], has extended the half-sweep iteration concept (see in [12]-[15]) to the quarter-sweep iteration concept in order to accelerate the iteration process by considering only one-fourth of the total node points located in the solution domain.

Due to the excellent performance of quarter-sweep

concept, it has been extensively discussed in many areas of problem solving (see in [6], [16]-[19]). Figure 1 illustrates the ideas of the iteration processes with respect to the full-, half- and quarter-sweep concepts.

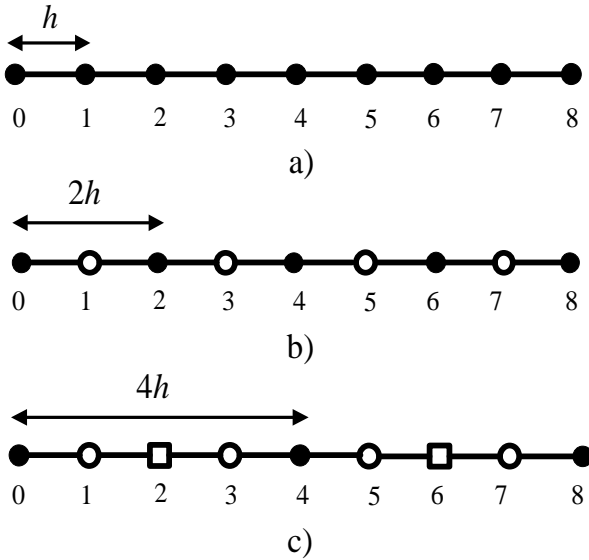


Figure 1 a), b) and c) depicted the full-, half- and quarter-sweep cases of uniform node points distribution in the given solution domain.

Firstly, finite grid network requires to be established to guide us in implementing the quarter-sweep concept in any iterative schemes. Based on Figure 1, formulation of iterative methods is performed based on the solid interior grid points (\bullet) until it met convergence test. Meanwhile, for the rest of the remaining node points of type (\circ) and (\square), the direct method will be used in the calculation.

More specifically, we examine the effectiveness of the quarter-sweep iteration concept with the Kaupp Successive Over-Relaxation (KSOR) iterative method based on the Grünwald implicit difference approximation equations for solving problem (1). We also employed the half-sweep HSKSOR and full-sweep FSKSOR method as control techniques to demonstrate the efficiency of the QSKSOR method.

The remainder of this paper is organized as follows: Section 2 demonstrate the formulation of the quarter-sweep Grünwald implicit difference approximation equations. Then, Section 3 elaborates the derivation of the family of Kaupp Successive Over-Relaxation (KSOR) iterative method. Next, Section 4 illustrates some numerical experiments. Finally, Section 5 stated the conclusion of this study

II. QUARTER-SWEEP GRÜNWALD IMPLICIT DIFFERENCE APPROXIMATION EQUATIONS

In this section, we are discussing how to derive the quarter-sweep Grünwald implicit approximation equations from discretization of problem (1) using the time derivative of the Grünwald operator and implicit finite difference scheme. This step will lead to sparse and large linear systems.

By using quarter-sweep Grünwald implicit finite difference scheme, the approximation equations at $i=4, 8, 12, \dots, N-8, N-4$ can be stated as

$$\frac{1}{(\Delta t)^\alpha} \sum_{k=0}^j g_{\alpha,k} U_{i,j-k} + \frac{p(x)}{8\Delta x} (U_{i+4,j} - U_{i-4,j}) + \frac{q(x)}{(4\Delta x)^2} (U_{i+4,j} - 2U_{i,j} + U_{i-4,j}) = f_{i,j} \quad (3)$$

Then, by letting

$$G_k = \frac{g_{\alpha,k}}{(\Delta t)^\alpha}, \quad \rho_i = \frac{p(x)}{8\Delta x}, \quad \text{and} \quad \lambda_i = \frac{q(x)}{(4\Delta x)^2},$$

the quarter-sweep Grünwald implicit approximation equation (3) can be simplified and expressed as

$$\alpha_i U_{i-4,j} + \beta_i U_{i,j} + \gamma_i U_{i+4,j} = F_{i,j} \quad (4)$$

where

$$\begin{aligned} \alpha_i &= \lambda_i - \rho_i, \\ \beta_i &= G_0 - 2\lambda_i, \\ \gamma_i &= \lambda_i + \rho_i, \end{aligned}$$

and where

$$F_{i,j} = \begin{cases} f_{i,1} - G_1 U_{i,0} & j=1 \\ f_{i,j} - \sum_{k=1}^j G_k U_{i,j-k} & j=2, 3, \dots, M \end{cases}$$

Next, the approximation equations (4) can be easily shown in matrix form as follows

$$A \underline{U}_j = \underline{F}_j \quad (5)$$

where

$$A = \begin{bmatrix} \beta_4 & \gamma_4 & & & & \\ \alpha_8 & \beta_8 & \gamma_8 & & & \\ & \alpha_{12} & \beta_{12} & \gamma_{12} & & \\ & & \alpha_{16} & \beta_{16} & \gamma_{16} & \\ & & & O & O & O \\ & & & & \alpha_{N-8} & \beta_{N-8} & \gamma_{N-8} \\ & & & & & \alpha_{N-4} & \beta_{N-4} \end{bmatrix}_{(N-4) \times (N-4)},$$

$$\underline{U}_j = [U_{4,j} \quad U_{8,j} \quad U_{12,j} \quad K \quad U_{N-8,j} \quad U_{N-4,j}]^T,$$

$$\underline{F}_j = [F_{4,j} - \alpha_4 U_{0,j} \quad F_{8,j} \quad F_{12,j} \quad K \quad F_{N-8,j} \quad F_{N-4,j} - \gamma_{N-4} U_{N,j}]^T.$$

III. FAMILY OF KSOR ITERATIVE METHOD

In this paper, we applied the FSKSOR, HSKSOR and QSKSOR iterative methods to solve linear systems generated from discretization of problem (1). To construct these methods, firstly we decompose the coefficient matrix, A as

$$A = D + L + V \quad (6)$$

where D , L and V are the diagonals, lower triangulation and upper triangulation matrices respectively.

Then, by referring to the decomposition matrix in equation (6), the general scheme of QSKSOR iterative method can be stated as follows [9],[19]

$$\underline{U}_j^{(k+1)} = \left(((1+\omega)D - \omega L)^{-1} (D + \omega V) \right) \underline{U}_j^{(k)} + ((1+\omega)D - \omega L)^{-1} (\omega \underline{F}_j) \quad (7)$$

where the relaxation parameter ω is extended to $R - [-2, 0]$ for KSOR iterative method.

Therefore, by determining the values of matrices D , L and V , the proposed algorithm for QSKSOR iterative method to solve equation (7) could be described as in Algorithm 1.

Algorithm 1: QSKSOR Scheme

- Initialize $\underline{U}_j^{(k+1)} \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$.
- For $j=1, 2, \dots, M$, and $i=4, 8, 12, \dots, N-4$, calculate:

$$\underline{U}_j^{(k+1)} \leftarrow \left(((1+\omega)D - \omega L)^{-1} (D + \omega V) \right) \underline{U}_j^{(k)} + ((1+\omega)D - \omega L)^{-1} (\omega \underline{F}_j)$$
- Perform the convergence test,

$$\left| \underline{U}_i^{(k+1)} - \underline{U}_i^{(k)} \right| \leq \varepsilon = 10^{-10}.$$
 If yes, go to step (iv). Otherwise, repeat step (ii).
- Compute the remaining points (i.e. \circ and \square) using direct method.
- Display approximate solutions.

IV. NUMERICAL EXAMPLES

In order to verify the effectiveness of the QSKSOR method over the HSKSOR and FSKSOR iterative methods, three numerical examples are tested. Three parameters namely the iteration numbers (k), execution time (t) and maximum absolute error (MAE) are considered at the different order of α -values (i.e. α as 0.333, 0.666, 0.999). Meanwhile, in the implementation of the iterative methods, the convergence test considered the tolerance error $\varepsilon = 10^{-10}$.

Example 1 [20] As the first example, we consider the following one-dimensional linear inhomogeneous fractional Burger's equation:

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} + \frac{\partial U(x,t)}{\partial x} - \frac{\partial^2 U(x,t)}{\partial x^2} = f(x,t),$$

with $p(x)=1$ and $q(x)=-1$,

$$\text{while } f(x,t) = \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} + 2x - 2,$$

subject to the initial condition $U(x,0) = x^2$.

The exact solution is $U(x,t) = x^2 + t^2$.

Example 2 [21] Consider the following one-dimensional inhomogeneous time-fractional parabolic equation

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} - \frac{\partial^2 U(x,t)}{\partial x^2} = f(x,t),$$

with $p(x)=0$ and $q(x)=-1$,

while

$$f(x,t) = \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} t^{2-\alpha} \sin(2\pi x) + 4\pi^2 t^2 \sin(2\pi x),$$

subject to the initial condition $U(x,0) = 0$.

The exact solution is given by $U(x,t) = t^2 \sin(2\pi x)$.

Example 3 [22] Consider the following one-dimensional linear inhomogeneous time-fractional parabolic equation

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} - \frac{\partial^2 U(x,t)}{\partial x^2} = f(x,t),$$

$$t > 0, \quad x \in R, \quad 0 < \alpha \leq 1,$$

with $p(x)=0$ and $q(x)=-1$,

$$\text{while } f(x,t) = \frac{2e^x t^{2-\alpha}}{\Gamma(3-\alpha)} - t^2 e^x,$$

subject to the initial condition $U(x,0) = 0$.

The exact solution is given by $U(x, t) = t^2 e^x$.

Then, the numerical results obtained from the implementation of QSKSOR, HSKSOR and FSKSOR iterative methods for examples 1 to 3 are recorded in

Tables 1 to 3 respectively. Whereas, Tables 4 and 5 show the decrement percentage of the iteration numbers and execution time in seconds.

Table 1. Comparison on Number of Iterations (k), execution time (seconds) and maximum absolute error (MAE) for the iterative methods using example 1 at $\alpha=0.333, 0.666, 0.999$

M	Method	$\alpha=0.333$			$\alpha=0.666$			$\alpha=0.999$		
		k	t	MAE	k	t	MAE	k	t	MAE
128	FSKSOR	404 ($\omega=-2.0613$)	3.20	2.5972e-02	283 ($\omega=-2.0909$)	3.11	1.3065e-02	166 ($\omega=-2.1743$)	3.14	1.2480e-03
	HSKSOR	199 ($\omega=-2.1261$)	1.88	2.5971e-02	140 ($\omega=-2.1899$)	1.87	1.3065e-02	86 ($\omega=-2.3794$)	1.87	1.2480e-03
	QSKSOR	102 ($\omega=-2.2687$)	1.86	2.5967e-02	75 ($\omega=-2.4017$)	1.87	1.3064e-02	45 ($\omega=-2.8629$)	1.86	1.2481e-03
256	FSKSOR	813 ($\omega=-2.0303$)	6.67	2.5972e-02	569 ($\omega=-2.0446$)	6.47	1.3065e-02	331 ($\omega=-2.0836$)	6.33	1.2480e-03
	HSKSOR	404 ($\omega=-2.0613$)	3.76	2.5972e-02	283 ($\omega=-2.0909$)	3.72	1.3065e-02	166 ($\omega=-2.1743$)	3.68	1.2480e-03
	QSKSOR	199 ($\omega=-2.1261$)	3.68	2.5971e-02	140 ($\omega=-2.1898$)	3.67	1.3065e-02	86 ($\omega=-2.3794$)	3.66	1.2480e-03
512	FSKSOR	1621 ($\omega=-2.0151$)	14.5 1	2.5972e-02	1136 ($\omega=-2.0221$)	13.6 3	1.3065e-02	659 ($\omega=-2.0411$)	13.0 8	1.2480e-03
	HSKSOR	813 ($\omega=-2.0303$)	7.71	2.5972e-02	569 ($\omega=-2.0446$)	7.59	1.3065e-02	331 ($\omega=-2.0836$)	7.45	1.2480e-03
	QSKSOR	404 ($\omega=-2.0613$)	7.37	2.5972e-02	283 ($\omega=-2.0909$)	7.34	1.3065e-02	166 ($\omega=-2.1743$)	7.33	1.2480e-03
1024	FSKSOR	3246 ($\omega=-2.0076$)	34.0 7	2.5972e-02	2269 ($\omega=-2.0110$)	30.8 6	1.3065e-02	1311 ($\omega=-2.0204$)	27.7 0	1.2480e-03
	HSKSOR	1621 ($\omega=-2.0151$)	16.2 7	2.5972e-02	1136 ($\omega=-2.0221$)	15.7 6	1.3065e-02	659 ($\omega=-2.0411$)	15.2 5	1.2480e-03
	QSKSOR	813 ($\omega=-2.0303$)	14.9 6	2.5972e-02	569 ($\omega=-2.0446$)	14.8 5	1.3065e-02	331 ($\omega=-2.0836$)	14.7 3	1.2480e-03
2048	FSKSOR	6365 ($\omega=-2.0076$)	87.0 8	2.5972e-02	4519 ($\omega=-2.0110$)	75.3 6	1.3065e-02	2618 ($\omega=-2.0204$)	63.6 9	1.2480e-03
	HSKSOR	3246 ($\omega=-2.0076$)	36.1 0	2.5972e-02	2269 ($\omega=-2.0110$)	33.9 6	1.3065e-02	1311 ($\omega=-2.0204$)	31.8 9	1.2480e-03
	QSKSOR	1621 ($\omega=-2.0151$)	30.7 9	2.5972e-02	1136 ($\omega=-2.0221$)	30.2 2	1.3065e-02	659 ($\omega=-2.0411$)	29.7 2	1.2480e-03

Table 2. Comparison on Number of Iterations (k), execution time (seconds) and maximum absolute error (MAE) for the iterative methods using example 2 at $\alpha=0.333$, 0.666, 0.999

M	Method	$\alpha=0.333$			$\alpha=0.666$			$\alpha=0.999$		
		k	t	MAE	k	t	MAE	k	t	MAE
128	FSKSOR	385 ($\omega=-2.0585$)	3.23	3.4734e-04	264 ($\omega=-2.0903$)	3.13	5.2935e-04	144 ($\omega=-2.1686$)	3.09	4.5594e-04
	HSKSOR	193 ($\omega=-2.1204$)	1.85	9.3060e-04	135 ($\omega=-2.1889$)	1.84	1.1077e-03	84 ($\omega=-2.3790$)	1.84	1.0292e-03
	QSKSOR	99 ($\omega=-2.2563$)	1.83	3.2676e-03	73 ($\omega=-2.3991$)	1.83	3.4249e-03	40 ($\omega=-2.8606$)	1.83	3.3262e-03
256	FSKSOR	764 ($\omega=-2.0299$)	6.66	2.0160e-04	513 ($\omega=-2.0441$)	6.40	3.8483e-04	291 ($\omega=-2.0811$)	6.20	3.1267e-04
	HSKSOR	385 ($\omega=-2.0585$)	3.80	3.4734e-04	274 ($\omega=-2.0866$)	3.77	5.2935e-04	144 ($\omega=-2.1686$)	3.73	4.5594e-04
	QSKSOR	193 ($\omega=-2.1204$)	3.74	9.3059e-04	135 ($\omega=-2.1889$)	3.73	1.1077e-03	84 ($\omega=-2.3787$)	3.72	1.0292e-03
512	FSKSOR	1467 ($\omega=-2.0147$)	13.99	1.6516e-04	1025 ($\omega=-2.0218$)	13.61	3.4871e-04	576 ($\omega=-2.0397$)	12.74	2.7687e-04
	HSKSOR	764 ($\omega=-2.0298$)	7.65	2.0159e-04	513 ($\omega=-2.0441$)	7.56	3.8483e-04	291 ($\omega=-2.0811$)	7.40	3.1267e-04
	QSKSOR	385 ($\omega=-2.0585$)	7.38	3.4734e-04	274 ($\omega=-2.0866$)	7.31	5.2935e-04	144 ($\omega=-2.1686$)	7.28	4.5594e-04
1024	FSKSOR	2906 ($\omega=-2.0073$)	31.95	1.5605e-04	2049 ($\omega=-2.0096$)	30.49	3.3969e-04	1125 ($\omega=-2.0197$)	27.04	2.6791e-04
	HSKSOR	1467 ($\omega=-2.0147$)	16.23	1.6516e-04	1115 ($\omega=-2.0189$)	15.91	3.4871e-04	654 ($\omega=-2.0388$)	15.44	2.7687e-04
	QSKSOR	764 ($\omega=-2.0298$)	15.16	2.0159e-04	513 ($\omega=-2.0441$)	15.06	3.8483e-04	310 ($\omega=-2.0834$)	14.93	3.1268e-04
2048	FSKSOR	5708 ($\omega=-2.0037$)	80.73	1.5380e-04	4056 ($\omega=-2.0054$)	71.23	3.3742e-04	2155 ($\omega=-2.0098$)	60.45	2.6568e-04
	HSKSOR	2905 ($\omega=-2.0071$)	35.44	1.5605e-04	2049 ($\omega=-2.0103$)	33.81	3.3969e-04	1151 ($\omega=-2.0202$)	31.90	2.6792e-04
	QSKSOR	1467 ($\omega=-2.0147$)	31.03	1.6516e-04	1025 ($\omega=-2.0218$)	30.63	3.4871e-04	599 ($\omega=-2.0408$)	30.18	2.7687e-04

Table 3. Comparison on Number of Iterations (k), execution time (seconds) and maximum absolute error (MAE) for the iterative methods using example 3 at $\alpha=0.333, 0.666, 0.999$

M	Method	$\alpha=0.333$			$\alpha=0.666$			$\alpha=0.999$		
		k	t	MAE	k	t	MAE	k	t	MAE
128	FSKSOR	423 ($\omega=-2.0606$)	3.16	1.1757e-03	295 ($\omega=-2.0905$)	3.13	2.5780e-03	172 ($\omega=-2.1743$)	3.11	2.2070e-03
	HSKSOR	209 ($\omega=-2.1252$)	1.85	1.1785e-03	145 ($\omega=-2.1892$)	1.84	2.5807e-03	89 ($\omega=-2.3794$)	1.84	2.2096e-03
	QSKSOR	111 ($\omega=-2.2578$)	1.83	1.1897e-03	78 ($\omega=-2.4024$)	1.82	2.59138e-03	46 ($\omega=-2.8662$)	1.84	2.2200e-03
256	FSKSOR	853 ($\omega=-2.0300$)	6.69	1.1750e-03	592 ($\omega=-2.0444$)	6.44	2.5773e-03	342 ($\omega=-2.0837$)	6.27	2.2064e-03
	HSKSOR	423 ($\omega=-2.0606$)	3.82	1.1757e-03	295 ($\omega=-2.0906$)	3.78	2.5780e-03	172 ($\omega=-2.1743$)	3.75	2.2071e-03
	QSKSOR	209 ($\omega=-2.1252$)	3.73	1.1785e-03	145 ($\omega=-2.1892$)	3.74	2.5807e-03	89 ($\omega=-2.3794$)	3.72	2.2097e-03
512	FSKSOR	1707 ($\omega=-2.0150$)	14.43	1.1748e-03	1181 ($\omega=-2.0221$)	13.68	2.5772e-03	681 ($\omega=-2.0411$)	12.89	2.2063e-03
	HSKSOR	853 ($\omega=-2.0300$)	7.70	1.1750e-03	592 ($\omega=-2.0445$)	7.59	2.5773e-03	342 ($\omega=-2.0837$)	7.43	2.2064e-03
	QSKSOR	423 ($\omega=-2.0606$)	7.41	1.1757e-03	295 ($\omega=-2.0906$)	7.37	2.5780e-03	172 ($\omega=-2.1743$)	7.33	2.2071e-03
1024	FSKSOR	3392 ($\omega=-2.0075$)	33.80	1.1748e-03	2353 ($\omega=-2.0110$)	30.74	2.5771e-03	1356 ($\omega=-2.0204$)	27.92	2.2062e-03
	HSKSOR	1707 ($\omega=-2.0150$)	16.51	1.1748e-03	1181 ($\omega=-2.0221$)	16.02	2.5772e-03	681 ($\omega=-2.0411$)	15.46	2.2063e-03
	QSKSOR	853 ($\omega=-2.0300$)	15.21	1.1750e-03	592 ($\omega=-2.0444$)	15.1	2.5773e-03	342 ($\omega=-2.0837$)	14.97	2.2064e-03
2048	FSKSOR	6774 ($\omega=-2.0038$)	88.57	1.1747e-03	4680 ($\omega=-2.0055$)	75.59	2.5771e-03	2708 ($\omega=-2.0102$)	63.53	2.2062e-03
	HSKSOR	3392 ($\omega=-2.0075$)	36.43	1.1748e-03	2353 ($\omega=-2.0110$)	34.29	2.5771e-03	1356 ($\omega=-2.0204$)	32.55	2.2062e-03
	QSKSOR	1707 ($\omega=-2.0150$)	31.22	1.1748e-03	1181 ($\omega=-2.0221$)	30.73	2.5772e-03	681 ($\omega=-2.0411$)	30.20	2.2063e-03

Table 4. Percentage reduction on the iteration numbers (Iter) and execution time for QSKSOR in comparison to the FSKSOR iterative method.

α		Example		
		1	2	3
0.33	Iter	74.53-74.75	73.71-74.74	73.76-75.50
		%	%	%
	Tim e	41.88-64.64	43.34-61.56	42.09-64.75
0.66	Iter	73.50-75.40	72.35-74.96	73.56-75.51
		%	%	%
	Tim e	39.87-59.90	41.53-57.00	41.85-59.35
0.99	Iter	72.89-74.83	71.13-75.00	73.26-74.85
		%	%	%
	Tim e	40.76-53.34	40.00-50.07	40.67-52.46
		%	%	%

Table 5. Percentage reduction on the iteration numbers (Iter) and execution time for HSKSOR in comparison to the FSKSOR iterative method.

α		Example		
		1	2	3
0.33	Iter	49.00-50.74	47.92-49.87	49.68-50.59
		%	%	%
	Tim e	41.25-58.54	42.72-56.10	41.46-58.87
0.66	Iter	49.79-50.53	45.58-49.95	49.72-50.85
		%	%	%
	Tim e	39.87-54.94	41.09-52.53	41.21-54.64
0.99	Iter	48.19-49.92	41.67-50.52	48.26-49.93
		%	%	%
	Tim e	40.45-49.93	39.84-47.23	40.19-48.76
		%	%	%

From the numerical results obtained and recorded in Tables 1 to 3, it clearly shows that, by implementing the quarter-sweep concept to the standard KSOR iteration technique, the required iteration numbers needed to solve problem (1) can be reduced at all mesh sizes that are considered. Meanwhile, as compared to the other iterative methods tested, the time execution of the QSKSOR iterative method has shown improvement as the mesh size increases

V. CONCLUSION

This paper concerns on the implementation of quarter-sweep iteration concept with KSOR iterative method in solving the linear systems produced from discretization of fractional parabolic equations using the Grünwald type of time-fractional derivative with the implicit difference scheme. It shows, based on the numerical results, that QSKSOR iterative method yields the best results with the least iteration numbers and time of execution amongst the tested family of KSOR iterative methods. Overall, the numerical results have shown that in term of iteration numbers and the time of execution, the QSKSOR scheme is better than the other methods.

ACKNOWLEDGEMENT

This study was supported by Universiti Malaysia Sabah (Postgraduate research grant: GUG0221-1/2018).

REFERENCES

- [1] Q. Xu, and Y. Xu. "Extremely low order time-fractional differential equation and application in combustion process", Communications in Nonlinear Science and Numerical Simulation, vol. 64, pp. 135-148, 2018.
- [2] M. R. S. Ammi, I. Jamiia, and D. F. Torres. "A finite element approximation for a class of Caputo time-fractional diffusion equations", Computers & Mathematics with Applications, vol. 78, no. 5, pp. 1334-1344, 2019.
- [3] P. Agarwal, and A. El-Sayed. "Non-standard finite difference and Chebyshev collocation methods for solving fractional diffusion equation", Physica A: Statistical Mechanics and Its Applications, vol. 500, pp. 40-49, 2018.
- [4] M. Gunzburger, and J. Wang. "A second-order Crank-Nicolson method for time-fractional PDE", International Journal of Numerical Analysis and Modeling, vol. 16, no. 2, pp. 225-239, 2019.
- [5] T. Zhang, and Q. Guo. "The finite difference/finite volume method for solving the fractional diffusion equation", Journal of Computational Physics, vol. 375, pp. 120-134, 2018.
- [6] A. Sunarto, J. Sulaiman, and A. Saudi. "Numerical solution of the time-fractional diffusion equations by using quarter-Sweep SOR iterative method", International Journal of Mathematical Engineering and Science (IJMES), vol. 3, no. 2, pp. 54-67, 2014.
- [7] I. Podlubny. "Fractional Differential Equations". Academic Press, San Diego, 1999.
- [8] W. K. Zahra, and S. M. Elkholy. "Cubic spline solution of fractional Bagley-Torvik equation", Electronic Journal of Mathematical Analysis and Applications, vol. 1, no. 2, pp. 230-241, 2013.
- [9] I. K. Youssef. "On the successive overrelaxation method", Journal of Mathematics and Statistics, vol. 8, no. 2, pp. 176-184, 2012.
- [10] M. Othman, and A. R. Abdullah. "An efficient four points modified explicit group poisson solver", International Journal of Computer Mathematics, vol. 76, pp. 203-217, 2000.
- [11] A. R. Abdullah. "The four point-Explicit Decoupled Group (EDG) method: A fast poisson solver", International Journal of Computer Mathematics, vol. 38, pp. 61-70, 1991.
- [12] A. Saudi, and J. Sulaiman. "Path planning simulation using harmonic potential fields through four-point EDGSOR method via 9-point Laplacian", Jurnal Teknologi, vol. 78, no. 8-2, pp. 12-24, 2016.
- [13] A. A. Dahalan, J. Sulaiman, and M. S. Muthuvalu. "Performance of HSAGE method with Seikkala derivative for 2-D Fuzzy Poisson equation", Applied Mathematical Sciences, vol. 8, pp. 885-899, 2014.
- [14] M.K.M. Akhir, M. Othman, J. Sulaiman, Z. A. Majid, and M. Suleiman. "The four point-EDGMSOR iterative method for solution of 2D Helmholtz equations", In: Communications in Computer and Information Science, A. Abd Manaf., Berlin: Springer-Verlag, vol. 253, pp. 218-227, 2011.
- [15] 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 equations", Sains Malaysiana, vol. 44, pp. 139-146, 2015.
- [16] M. N. Suardi, N. Z. F. M. Radzuan, and J. Sulaiman. "Performance of quarter-sweep SOR iteration with cubic B-spline scheme for solving two-point boundary value problems", Journal of Engineering and Applied Sciences, vol. 14, no. 3, pp. 693-700, 2019.
- [17] M. S. Muthuvalu, and J. Sulaiman. "Solving second kind linear Fredholm integral equations via quarter-sweep SOR iterative method", Journal of Fundamental Sciences, vol. 6, no. 2, pp. 104-110, 2010.
- [18] N.I.M. Fauzi, and J. Sulaiman. "Quarter-sweep modified SOR Iterative algorithm and cubic spline basis for the solution of second order two-point boundary value problems", Journal of Applied Sciences, vol. 12, pp. 1817-1824, 2012.
- [19] F. A. Muhiddin, J. Sulaiman, and A. Sunarto. "MKSOR iterative method for the Grünwald implicit finite difference solution of one-dimensional time-fractional parabolic equations", AIP Conference Proceedings, vol. 2138, p. 030026, 2019.
- [20] M. Uddin, and S. Haq. "RBF's approximation method for time fractional partial differential equations", Communications in Nonlinear Science and Numerical Simulation, vol. 16, no.11, pp. 4208-4214, 2011.
- [21] Y. Jiang, and J. Ma. "High-order finite element methods for time-fractional partial differential equations", Journal of Computational and Applied Mathematics, vol. 235, no. 11, pp. 3285-3290, 2011.
- [22] Y. Ma. "Two implicit finite difference method for time fractional diffusion equation with source term", Journal of Applied Mathematics & Bioinformatics, vol. 4, no. 2, pp. 125-145, 2014.