Caputo Finite Difference Solution for solving Time-Fractional Diffusion Equations via weighted point iteration

Mohd Usran Alibubin[1], Jumat Sulaiman[1#], Fatihah Anas Muhiddin[2], Andang Sunarto[3]

1 Faculty Science and Natural Resources, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu, Sabah, MALAYSIA.
2 College of Computing, Information and Mathematics, Universiti Teknologi MARA Sabah Branch, 88997 Kota Kinabalu, Sabah, MALAYSIA.
3 Tadris Matematika, Universitas Islam Negeri (UIN), Fatmawati Sukarno, Bengkulu, 38211, INDONESIA.
Corresponding author: Email: jumat@ums.edu.my; Tel: +601131907643.

ABSTRACT Time-fractional diffusion equations (TFDEs) are widely used in modeling anomalous diffusion processes, which occur in various fields such as physics, engineering, and economics. These equations offer a more accurate representation of systems where classical diffusion models fall short, particularly in capturing memory and hereditary properties of materials. In this paper, we employ the Caputo finite difference approximation equation for TFDEs by applying a discretization scheme based on the second-order implicit finite difference and Caputo fractional derivative operator. To solve these equations numerically, the one-dimensional TFDEs are discretized using Caputo's implicit finite difference approximation. The corresponding system of linear approximation equations is then solved using weighted point iteration methods, specifically Successive Overrelaxation (SOR) and Gauss-Seidel (GS). Three examples are provided to evaluate the performance of these iterative methods. The numerical results demonstrate that the SOR method requires fewer iterations and reduces computational time, proving to be more efficient compared to the Gauss-Seidel method.

KEYWORDS: Finite Difference Scheme; Caputo Derivative Operator; Time-Fractional Diffusion Equations; Weighted point iteration.
Received 24 July 2024 **Revised** 19 September 2024 **Accepted** 23 September 2024 **In press** 25 September 2024 **Online** 26 September 2024

Original Article

INTRODUCTION

Presently, many researchers have studied fractional differential equations (FDEs) and their application in various fields of physics, such as fractional kinetics (Cen *et al.*, 2018), measurement of visco-elastic material properties (Xu & Xu, 2018), anomalous diffusion (Chen *et al.*, 2015), fluid mechanics (Paliivets *et al.*, 2021), and image processing (Khalid *et al.*, 2020). Miller & Rose (1993), Podlubny (1999), Diethelm & Ford (2002), Diethelm (2010) and others have all worked on foundational works that solve fractional differential equations. Recent applications have included numerically solving several kinds of linear fractional differential equations. Apart from that, the area of research in time-fractional diffusion equations (TFDEs) has evolved as a useful mathematical tool for explaining time-fractional events where the derivative order is non-integer. This is due to the fact that it may produce superior models that capture non-classical occurrences for complex physical real-world problems in particular cases (Rashid *et al.*, 2021). Fractional operators are important for understanding a wide range of complicated mechanical and physical behaviors, as well as problem solving involving non-Markovian random walks (Ford *et al.*, 2011), which involve systems with long-term memory. However, there are significant practical difficulties in solving the related fractional differential equation. It should be noted that only a few fractional differential equations may be solved analytically using complex functions, such as the Mittag-Lefer function (Kurulay & Bayram, 2012), H-function (Kilbas *et al.*, 2004), and Wright function (Wright, 1935). Therefore, various numerical algorithms for solving TFDEs have recently been developed, which appear to be better capable of dealing with the complexities of fractional-order equations. Recent works have used the reduced spline (RS) method based on a proper orthogonal decomposition (POD) technique (Ghaffari &

Ghoreishi, 2019), the Crank-Nicholson strategy employs the finite element approach (Ali *et al.*, 2017). The following techniques have been proposed and discussed by researchers in the literature: An approach called Method for alternating segment explicit-implicit/implicit explicit parallel difference (Wu *et al.*, 2018), a new method based on fractional finite differences (Zhang, 2009), the use of localized radial basis functions (RBFs) (Ford *et al.*, 2011), and the application of the fractional differential quadrature (FDQ) method ((Yuste, 2006). Previously, other researchers have focused on the implicit scheme (Muhiddin *et al.*, 2020) to discretize the TFDEs problem. They introduced the Caputo finite difference scheme and the Caputo fractional operator into the approximation equations, resulting in a linear system at each time step. Solving the TFDEs numerically leads to a large and sparse system of linear equations (SLEs), which requires iterative methods for efficient computation. While the Gauss-Seidel (GS) technique, belonging to the point iterative family, has a slow convergence rate, the Successive Over-Relaxation (SOR) iterative method has emerged as a prominent solution for addressing this problem (Young, 1973; Alibubin *et al.*, 2018).

Extensive research has been conducted in the literature to explore point iterative techniques for solving SLEs resulting from the discretization of differential equations with integer-order. However, there is limited research on the application of these methods to fractional differential equations (Sunarto *et al.*, 2014; Alibubin *et al.*, 2018). Currently, most of the existing work in this area has focused on utilizing the Caputo fractional derivative operator. Therefore, the purpose of this paper is to investigate the performance of the weighted point iteration family, namely SOR iterative method in solving time-fractional diffusion equations using Caputo's implicit finite difference approximation equation. Also, we have developed the GS iterative methods as a benchmark to compare and demonstrate the capabilities of the SOR approach. To evaluate the performance of the SOR method, we consider TFDEs defined as the target equations in our analysis as follows (Podlubny, 1999).

$$\frac{d^{\alpha}u(x,t)}{dt^{2}} = \gamma\frac{d^{2}u(x,t)}{dx^{2}} + \rho\frac{du(x,t)}{dx} + \phi u(x,t) + f(x,t), \quad x \in [\rho_0, \rho_1], \quad 0 \le t \le T \tag{1}$$

subject to the following initial and boundary conditions

$$U(x,0) = U_0(x),$$
and
$$U(x,t) = f(x)$$

γ, ρ and $\emptyset$ were arbitrary constants, and $f(x,t)$ was a known function, whereas α is a parameter that refers to the fractional order of time derivative.

PRELIMINARIES

Before constructing the finite difference approximation of Equation (1), we introduce some basic definitions.

Definition 2.1 The Riemann-Liouville fractional integral operator, $J^{\alpha}$of order-α is defined as (Zhang, 2009)

$$J^{\alpha}f(x) = \frac{1}{\Gamma(\alpha)}\int_0^x (x-t)^{\alpha-1} f(t)dt, \quad \alpha > 0, x > 0. \tag{2}$$

Definition 2.2 The Caputo's fractional partial derivative operator, $D^{\propto}$ of order -α is defined as (Zhang, 2009)

$$D^{\alpha}f(x) = \frac{1}{\Gamma(m-\alpha)}\int_0^x \frac{f^{(m)}(t)}{(x-t)^{\alpha-m+1}}dt, \quad \alpha > 0. \tag{3}$$

with $m-1 < \alpha \le m, m \in N, x >$. This study conducts a comparison between the SOR algorithm and the GS iterative method for solving Problem (1), which involves variable coefficients. To solve Problem (1) numerically, we establish numerical approximations using Caputo's derivative formulation, incorporating Dirichlet boundary conditions, and considering the non-local fractional derivative operator. The proposed approximation equation belongs to the category of unconditionally stable schemes. According to previous research, many studies have been done to demonstrate the efficiency of the SOR iterative method (Youssef, 2012; Youssef & Taha (2012); Alibubin *et al.* (2016). However, there is no SOR iterative that exists in the literature for solving the time-fractional diffusion problem especially combining the Caputo implicit finite difference scheme. As a result, this study compares the SOR iterative approach to the GS iterative method for solving Problem (1) with variable coefficients.

By applying Problem (1), the solution domain is confined to a finite space domain, specifically within the range $0 \le x \le \alpha$,with $0 \le \alpha \le 1$, and the parameter α is associated with the fractional order of the space derivative. To obtain the solution, we consider the initial boundary conditions of Problem (1).

$$U(x,0) = U_0(x), \text{ and } U(x,t) = f(x)$$

where $U_0(x)$, and *f(x)* are given functions. To formulate the discrete approximation to the time fractional derivative in Eq. (1), we consider Caputo's fractional partial derivative of order α, defined by (Sunarto *et al.*, 2018; Alibubin *et al.*, 2024).

$$\frac{\partial^{\alpha} U(x_i,t_n)}{\partial x^{\alpha}} = \frac{1}{\Gamma(2-\alpha)}\int_0^{t_n} \frac{\partial^2 U(x_i,t_n)}{\partial x^2}(t_n - s)^{1-\alpha}ds \tag{4}$$

The following is how the paper is organized: Section 2 provides an approximation formula for the fractional derivative as well as a numerical strategy for solving the TFDEs (1) using Caputo's implicit finite difference method. Section 3 contains the formulation of the SOR iterative method, while Section 4 presents the numerical experiment and conclusions given in Section 5.

CAPUTO'S IMPLICIT FINITE DIFFERENCE APPROXIMATION EQUATION

In this section, we provide a concise overview of the discretization process for Problem (1). The formulation of Caputo's fractional partial derivative is represented by Equation (4), which corresponds to the first-order approximation approach.

$$D_t^{\alpha} U_{i,n} \cong \sigma_{\alpha,k}\sum_{j=1}^{n}\omega_j^{(\alpha)}\left(U_{i,n-j+1} - U_{i,n-j}\right) \tag{5}$$

where

$$\sigma_{\alpha,k} = \frac{1}{\Gamma(1-\alpha)(1-\alpha)k^{\alpha}}$$

and

$$\omega_j^{(\alpha)} = j^{1-\alpha} - (j-1)1^{1-\alpha}$$

Before discretizing Equation (1), we assume that the solution domain of the problem is uniformly partitioned. To achieve this, we consider positive integers m and n, which define the grid sizes in the space and time directions for the finite difference algorithm. These grid sizes are denoted as $h = \Delta x =$

$\frac{\gamma-0}{m}$ and $k = \Delta t = \frac{T}{n}$ respectively. Based on these grid sizes, we construct a uniformly divided grid network for the solution domain. The grid points in the space interval $[0,\gamma]$are represented by the numbers $x_i = ih,\ \ i = 0,1,2, \dots, m$. Similarly, the grid points in the time interval are labeled $t_i = jk$, $j = 0,1,2, \dots, n$. The values of the function $U(x,t)$ at these grid points are denoted as $U_{i,j} = U(x_1,t_j)$. Utilizing Eq. (5) and employing the implicit finite difference discretization scheme, we obtain the Caputo's implicit finite difference approximation equation of Problem (1) for the grid point centered $(x_i,t_j) = (ih,nk)$. This equation is expressed as follows:

$$\sigma_{\alpha,k}\sum_{j=1}^{n}\omega_j^{(\alpha)}\left(U_{i,n-j+1}-U_{i,n-j}\right) = \frac{\gamma}{h^2}\left(U_{i-1,n}-2U_{i,n}+U_{i+1,n}\right)+\frac{\rho}{2h}\left(U_{i+1,n}-U_{i-1,n}\right)+\emptyset U_{i,n}+f_{i,n} \tag{6}$$
$$i = 1,2,\dots,m-1$$

The obtained approximation equation, referred to as Caputo's implicit finite difference approximation equation, exhibits consistent first-order accuracy in time and second-order accuracy in space, as stated in Eq. (6). It should be noted that the form of this approximation equation can be adjusted based on the selected time level. For instance, let's consider the case where $n \geq 2$:

$$\sigma_{\alpha,k}\sum_{j=1}^{n}\omega_j^{(\alpha)}\left(U_{i,n-j+1}-U_{i,n-j}\right) = \left(\frac{\gamma}{h^2}-\frac{\rho}{2h}\right)U_{i-1,n}+\left(\emptyset-\frac{2\gamma}{h^2}\right)U_{i,n}+\left(\frac{\gamma}{h^2}+\frac{\rho}{2h}\right)U_{i+1,n}+f_{i,n}$$
$$\therefore\ \sigma_{\alpha,k}\sum_{j=1}^{n}\omega_j^{(\alpha)}\left(U_{i,n-j+1}-U_{i,n-j}\right) = \beta_0U_{i-1,n}+\beta_1U_{i,n}+\beta_2U_{i+1,n}+f_{i,n}$$

where

$$\beta_0 = \frac{\gamma}{h^2}-\frac{\rho}{2h},\qquad \beta_1 = \emptyset-\frac{2\gamma}{h^2},\qquad \beta_2 = \frac{\gamma}{h^2}+\frac{\rho}{2h}$$

Finally, we get for $n = 1, \omega_j^{(\alpha)} = 1$

$$\sigma_{\alpha,k}\left(U_{i,1}-U_{i,0}\right) = \beta_0U_{i-1,1}-\beta_1U_{i,1}+\beta_2U_{i+1,1}+f_{i,1} \tag{7}$$

The approximation Equation (7) can be rewritten as follows.

$$-p_iU_{i-1,1}+q_iU_{i,1}-r_iU_{i+1,1} = f_{i,1}^{*},\quad i = 1,2,\dots,m-1 \tag{8}$$

where

$$p_i = \sigma_{\alpha,k}-\beta_0,\ \ q_i = -\beta_1,\ \ r_i = \sigma_{\alpha,k}-\beta_2,\ f_{i,1}^{*} = f_{i,1}-\sigma_{\alpha,k},$$

Again, Equation (8) can be expressed in a matrix form as

$$A\underset{\sim}{U} = \underset{\sim}{f} \tag{9}$$

where

$$A = \begin{bmatrix} q & -r & & & & \\ -p & q & -r & & & \\ & -p & q & -r & & \\ & & \ddots & \ddots & \ddots & \\ & & & -p & q & -r \\ & & & & -p & q \end{bmatrix}_{(m-1)\times(m-1)}$$
$$\underset{\sim}{U} = \begin{bmatrix} U_{11} & U_{21} & U_{31} & \cdots & U_{m-2,1} & U_{m-1,1} \end{bmatrix}^T,$$
$$\underset{\sim}{f} = \begin{bmatrix} U_{11}+p_1U_{01} & U_{21} & U_{31} & \cdots & U_{m-2,1} & U_{m-1,1}+p_{m-1}U_{m,1} \end{bmatrix}^T.$$

FORMULATION OF SOR ITERATIVE METHOD

In this section, we investigate the performance of the SOR method as studied by Young (1971) for solving the linear system resulting from the discretization of the problem (1). As a benchmark, we also consider the GS iterative method which is equivalent to the SOR iterative method when the relaxation parameter $\omega = 1$. The objective of this study is to showcase the efficiency of the SOR iterative method for solving problems (1). This method is specifically designed to handle the second-order implicit finite difference scheme and the Caputo fractional derivative operator. To establish the formulation of the SOR iterative method, we decompose the coefficient matrix A in Equation (9) mentioned above as:

$$A = D + L + V \tag{10}$$

where D, L and V are the diagonals, lower triangulation, and upper triangulation matrices respectively. The SOR iterative method can be obtained and presented in matrix form using the decomposition matrix in Equation (10) as shown in several studies (Ford *et al.*, 2011; Yuste, 2006; Zhang, 2009).

$$\underset{\sim}{U}^{(k+1)} = (D - \omega L)^{-1}[\omega V + (1-\omega)D]\underset{\sim}{U}^{(k)} + (D - \omega L)^{-1} f, \tag{11}$$

where $\underset{\sim}{U}_j^{(k)}$ represents the unknown vector at the k^{th} iteration and relaxation parameter $\omega \in [1,2)$. Meanwhile, by referring to Equations (8) and (11), the SOR scheme can be expressed based on the point iteration as

$$U_{i,j}^{(k+1)} = (1-\omega)U_{i+1,j}^{(k)} + \frac{\omega}{q_i}\left(p_i U_{i-1,1} + r_i U_{i+1,1} - f_{i,1}^*\right) \quad i = 1,2,\dots,n; \quad j = 1,2,3,\dots,M \tag{12}$$

Remember that the relaxation parameter for the conventional SOR iterative approach is $0 \le \omega < 1$. Algorithm 1 summarizes the general algorithm of the SOR iterative technique for solving SLE (9).

Table 1. Algorithm for SOR scheme

Algoritm 1: SOR scheme	
i.	Initialize $\underset{\sim}{U}_j^{(k+1)} \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$
ii.	Assign the optimal value of ω
iii.	For $i = 1,2,\dots,n-1$ and $j = 1,2,3,\dots,m-1$ assign $U_{i,j}^{(k+1)} = (1-\omega)U_{i+1,j}^{(k)} + \frac{\omega}{q_i}(p_i U_{i-1,1} + r_i U_{i+1,1} - f_{i,1}^*)$
iv.	Check the convergence test. If the convergence criterion i.e $\left\Vert \underset{\sim}{U}^{(k+1)} - \underset{\sim}{U}^{(k)} \right\Vert \le \varepsilon = 10^{-10}$ is satisfied, go to step (v). Otherwise, go back to step (iii)
v.	Display approximate solutions.

NUMERICAL EXPERIMENTS

To investigate the performance of the SOR and the GS iterative methods, we evaluated three examples of TFDEs. The goal was to validate the efficiency of both iterative approaches based on three criteria: the number of iterations (K), the execution time in seconds, and the maximum error. The evaluation was conducted at three different values of $\alpha = 0.25$, $\alpha = 0.50$, and $\alpha = 0.75$. Throughout the implementation of the point iterations, a convergence test was performed considering a tolerance error, $\varepsilon = 10^{-10}$. This ensured that the iterative methods continued until the desired level of accuracy was achieved.

Example 1 *[Ford et al, 2011] Consider the following time fractional initial boundary value problem*

$$\frac{d^{\alpha}U(t,x)}{\mathrm{dt}^2}-\frac{d^2U(t,x)}{\mathrm{dx}^2}=f(x,t),\qquad t\in[0,1],\qquad t\geq 0,\qquad 0<x<1, \tag{13}$$

where the boundary conditions are given in

$$\mathrm{u}(0,\mathrm{t})=\mathrm{u}(1,\mathrm{t})=0,\quad t\in[0,1],$$

and initial condition

$$\mathrm{u}(\mathrm{t},0)=0,\ \ u(t,1)=0,\ \ 0<x<1$$

The exact solution is written as

$$u(x,t)=t^2\sin 2\pi x.$$

and

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

Example 2 *[Karatay et al, 2011] Consider the following time fractional initial boundary value problem*

$$\frac{d^{\alpha}U(t,x)}{\mathrm{dt}^2}-\frac{d^2U(t,x)}{\mathrm{dx}^2}=3.009011112t^{\frac{3}{2}}\sin(\pi x)\cos(\pi x)+4t^2\sin(2\pi x)\,\pi^2,\ \ (0<x<1,0<<1) \tag{14}$$

where the boundary conditions are given in

$$\mathrm{u}(0,\mathrm{x})=\mathrm{u}(1,\mathrm{x})-\sin(2\pi x),\quad 0\leq x\leq 1,$$

with initial condition as

$$\mathrm{u}(\mathrm{t},0)=0,\ \ u(t,1)=0,\ \ 0\leq x\leq 1$$

The exact solution is

$$u(t,x)=t^2\sin(2\pi x).$$

Example 3 *[Mohammad et al, 2021] Consider the following time fractional initial boundary value problem*

$$\frac{d^{\alpha}U(t,x)}{\mathrm{dt}^2}=\frac{d^2U(t,x)}{\mathrm{dx}^2}+f(x,t),\qquad x\in[0,1],\qquad t\geq 0,\qquad 0<\alpha<1, \tag{15}$$

where the exact solution is

$$u(x,t)=t^2\,(\mathrm{x}-1)^2\sin(2\pi x).$$

and

$$f(x,t)=0.5t^2e^2x^2(\mathrm{x}-1)^2\Gamma(\alpha+3)-t^{(2+\alpha)}e^x(x^4+6x^3+x^2-8x+2).$$

Table 1-3 presents the results of numerical experiments for numerical experiments given in Example 1 - 3 acquired by the implementation of GS and SOR iterative methods at various mesh sizes, m = 512, 1024, 2048, 4096, and 8192.

Table 1. Comparison of the number of iterations (K), execution time (Seconds), and maximum errors for iterative algorithms using Example 1 at $\alpha = 0.25$, $\alpha = 0.50$, and $\alpha = 0.75$.

M	Method	α = 0.25			α = 0.50			α = 0.75		
		K	t	Max Error	K	t	Max Error	K	t	Max Error
512	GS	53857	114.98	1.2810e-03	24085	89.63	4.4632e-03	6330	52.28	7.9839e-03
	SOR	2364	43.97	1.2802e-03	513	7.75	4.4628e-03	605	8.12	7.9837e-03
		ω=1.9665			ω=1.9665			ω=1.9665		
1024	GS	173277	517.97	1.2831e-03	82433	265.18	4.4645e-03	21924	152.50	7.9844e-03
	SOR	8200	112.38	1.2801e-03	2537	97.32	4.4632e-03	733	92.77	7.9839e-03
		ω=1.9665			ω=1.9665			ω=1.9665		
2048	GS	569412	4134.55	1.2898e-03	276232	1662.40	4.4687e-03	74187	536.333	7.9857e-03
	SOR	26390	324.55	1.2805e-03	8528	237.90	4.4633e-03	2000	187.07	7.9840e-03
		ω=1.9665			ω=1.9665			ω=1.9665		
4096	GS	968304	6223.24	1.2910e-03	893663	7568.23	4.4851e-03	242796	2772.33	7.9911e-03
	SOR	78937	1145.75.85	1.2814e-03	27712	642.17	4.4637e-03	6907	423.12	7.9841e-03
		ω=1.9665			ω=1.9665			ω=1.9665		
8192	GS	1292468	15634.89	1.2981e-03	1093664	13246.30	4.4913e-03	755078	9497.12	8.0123e-03
	SOR	213681	2461.79	1.2849e-03	91185	1466.14	4.4649e-03	23736	1163.51	7.9845e-03
		ω=1.9665			ω=1.9665			ω=1.9665		

Table 2. Comparison of the number of iterations (K), execution time (Seconds), and maximum errors for iterative algorithms using Example 1 at $\alpha = 0.25$, $\alpha = 0.50$, and $\alpha = 0.75$.

M	Method	α = 0.25			α = 0.50			α = 0.75		
		K	t	Max Error	K	t	Max Error	K	t	Max Error
512	GS	54367	69.28	7.4630e-04	6985	2.36	4.4619e-03	6262	7.97	8.1975e-03
	SOR	21582	14.36	7.4556e-04	2462	6.63	4.4632e-03	1863	2.06	8.1974e-03
		ω=1.5852			ω=1.5852			ω=1.5852		
1024	GS	174667	444.32	7.4834e-04	82432	207.83	4.4647e-03	21651	61.32	8.1980e-03
	SOR	61283	81.77	7.4663e-04	25349	44.14	4.4637e-03	6534	12.19	8.1977e-03
		ω=1.5852			ω=1.5852			ω=1.5852		
2048	GS	574443	2851.48	7.5512e-04	276231	1314.58	4.4689e-03	73097	384.56	8.1994e-03
	SOR	185173	510.38	7.4894e-04	86150	288.38	4.4649e-03	22556	76.37	8.1980e-03
		ω=1.5852			ω=1.5852			ω=1.5852		
4096	GS	1035653	6225.50	7.5511e-04	564235	3576.54	4.4689e-03	189546	752.56	8.1980e-03
	SOR	599967	3349.85	7.5602e-04	287685	1021.95	4.4693e-03	76054	128.65	8.1995e-03
		ω=1.5852			ω=1.5852			ω=1.5852		
8192	GS	2054687	9216.66	7.5532e-04	1307158	7635.36	4.4688e-03	702565	1532.56	8.1980e-03
	SOR	836512	3349.85	7.4894e-04	423977	2143.41	4.4693e-03	247702	246.64	8.1995e-03
		ω=1.5852			ω=1.5852			ω=1.5852		

Table 3. Comparison of the number of iterations (K), execution time (Seconds), and maximum errors for iterative algorithms using Example 1 at $\alpha = 0.25$, $\alpha = 0.50$, and $\alpha = 0.75$.

M	Method	α = 0.25			α = 0.50			α = 0.75		
		K	t	Max Error	K	t	Max Error	K	t	Max Error
512	GS	93567	168.38	8.2271e-03	24584	66.33	4.5947e-03	5102	43.86	2.5436e-03
	SOR	22813	40.59	1.8089e-04	6150	11.63	2.1951e-04	1262	8.14	1.6508e-04
		ω=1.5676			ω=1.5676			ω=1.5676		
1024	GS	308691	1076.17	2.5983e-03	80866	265.29	4.6239e-03	17184	124.50	2.5799e-03
	SOR	73073	110.17	1.8213e-04	19647	127.57	2.1997e-04	4070	19.22	1.6519e-04
		ω=1.5676			ω=1.5676			ω=1.5676		
2048	GS	55978	2537.32	2.5983e-03	254484	1222.56	4.6386e-03	55978	140.95	2.5983e-03
	SOR	219512	1034.97	1.8706e-04	58903	409.33	2.2176e-04	12381	52.73	1.6558e-04
		ω=1.5676			ω=1.5676			ω=1.5676		
4096	GS	174175	4869.22	2.6075e-03	750184	5845.53	4.6459e-03	174175	691.32	2.6075e-03
	SOR	585294	3982.81	2.0672e-04	157479	607.40	2.2864e-04	33752	154.93	1.6715e-04
		ω=1.5676			ω=1.5676			ω=1.5676		
8192	GS	2361262	53241.22	2.6075e-03	1990784	19038.98	4.6496e-03	506195	5638.00	2.6121e-03
	SOR	843690	5494.53.00	2.0536e-04	321261	1282.61	2.5333e-04	50628	475.05	1.6715e-04
		ω=1.5676			ω=1.5676			ω=1.5676		

The comparative results in Tables 1-3 show the substantial advantages of the SOR method over the GS method, particularly in terms of iteration number and execution time. Across all tested fractional orders $\alpha = 0.25, \alpha = 0.50$, and $\alpha = 0.75$ and mesh sizes, the SOR method is consistently demonstrated marked reduction in both iteration number and execution time. For instance, with M=512 and α=0.25, the SOR method achieved 95.6% reduction in iteration numbers compared to the GS method, while maintaining similar accuracy levels, as evidenced by the comparable maximum errors. This improvement is primarily attributed to the optimal selection of the relaxation parameter ω, which significantly accelerates convergence. As the mesh size increases, the scalability of the SOR method becomes increasingly evident, offering reductions in execution time exceeding 80% for larger grids, compared to GS. These results highlight the SOR method as not only an efficient iterative solver but also a scalable solution for handling large, complex systems arising from fractional-order diffusion problems. Given these findings, the SOR method presents a compelling alternative for solving time-fractional partial differential equations, outperforming traditional iterative approaches both in terms of speed and computational cost without compromising accuracy.

CONCLUSION

This study successfully demonstrated the formulation and application of the SOR method using a second-order implicit finite difference scheme with the Caputo fractional derivative operator. The numerical results clearly indicate that the SOR method outperforms the GS method, both in terms of iteration number and execution time. This superior performance is primarily due to the optimal selection of the relaxation parameter, which significantly accelerates the convergence. The results highlight the potential of the SOR method as an efficient solution for handling large systems of linear equations arising from fractional-order diffusion problems.

ACKNOWLEDEMENT

We gratefully acknowledge Universiti Malaysia Sabah for funding this research under the UMSGreat research grant for a postgraduate student, GUG0585-1/2023.

REFERENCES

[1] Ali, U., Abdullah, F. A. & Ismail, A. I. 2017. Crank-Nicolson finite difference method for two-dimensional fractional sub-diffusion equation. *Journal of Interpolation and Approximation in Scientific Computing*, 2017(2), 18–29.

[2] Alibubin, M. U., Sunarto, A., Akhir, M. K. M. & Sulaiman, J. (2016). Performance analysis of half-sweep SOR iteration with rotated nonlocal arithmetic mean scheme for 2D nonlinear elliptic problems. *Global Journal of Pure and Applied Mathematics*, 12(4), 3415–3424.

[3] Alibubin, M.U., Sunarto, A. & Sulaiman, J. 2016. Quarter-sweep Nonlocal Discretization Scheme with QSSOR Iteration for Nonlinear Two-point Boundary Value Problems. *Journal of Physics: Conference Series*, 710(1), 12023

[4] Alibubin, M.U., Sulaiman, J., Muhiddin, F. A. & Sunarto, A. 2024. Implementation of the KSOR Method for Solving One-Dimensional Time-Fractional Parabolic Partial Differential Equations with the Caputo Finite Difference Scheme Title of Manuscript. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 48(1), 168–179.

[5] Almeida, R., Bastos, N. R. & Monteiro, M. T. 2015. Modeling some real phenomena by fractional differential equations. *Mathematical Methods in the Applied Sciences*, 39(16), 4846–4855.

[6] Baharuddin, S., Sunarto, A. & Dalle, J. 2017. KSOR iterative method for solving Fredholm integral equations of second kind. *Journal of Engineering and Applied Sciences*, 12, 3220–3224.

[7] Basiron, Y. 2007. Palm oil production through sustainable plantations. *European Journal of Lipid Science and Technology*, 109(3), 289–295.

[8] Cen, Z., Huang, J. & Xu, A. 2018. An efficient numerical method for a two-point boundary value problem with a Caputo fractional derivative. *Journal of Computational and Applied Mathematics*, 336, 1–7.

[9] Chen, W., Sun, H., Zhang, X. & Chen, W. 2015. *Fractional calculus in anomalous diffusion and fractional mechanics*. World Scientific.

[10] Diethelm, K. 2010. The analysis of fractional differential equations: An application-oriented exposition using differential operators of Caputo type. *Springer Science & Business Media*.

[11] Diethelm, K. & Ford, N. J. 2002. Analysis of fractional differential equations. *Numerical Algorithms*, *36*(1), 31–52.

[12] Edeki, S., Ugbebor, O. & Owoloko, E. 2017. Analytical solution of the time-fractional order Black-Scholes model for stock option valuation on dividend yield basis. *International Journal of Applied Mathematics*, 47(4), 435–447.

[13] Evans, D. J. 1985. Group explicit iterative methods for solving large linear systems. International *Journal of Computer Mathematics*, 17(1), 81–108.

[14] Ford, J. N., Xiao, J. & Yan, Y. 2011. A finite element method for time fractional partial differential equations. *Fractional Calculus and Applied Analysis*, 8(3), 454–474.

[15] Gaspar, F. J. & Rodrigo, C. 2017. Multigrid waveform relaxation for the time-fractional heat equation. *SIAM Journal on Scientific Computing*, 39(4), A1201–A1224.

[16] Ghaffari, R. & Ghoreishi, F. 2019. Reduced spline method based on a proper orthogonal decomposition technique for fractional sub-diffusion equations. *Applied Numerical Mathematics*, 137, 62–79.

[17] Gunzburger, M. & Wang, J. A. 2019. Second-order Crank-Nicolson method for time-fractional PDES. *International Journal of Numerical Analysis and Modeling*, 16(2), 225–239.

[18] Karatay, I., Bayramoglu, S. R. & Sahin, A. 2011. Implicit difference approximation for the time fractional heat equation with the nonlocal condition. *Applied Numerical Mathematics*, 61, 1281–1288.

[19] Khalid, M., Ahmad, A. & Qasim, M. 2020. Fractional derivatives in image processing. *Advances in Mathematics: Scientific Journal,* 9(1), 413–421.

[20] Kilbas, A. A., Saigo, M. & Trujillo, J. J. 2004. *Theory and Applications of Fractional Differential Equations*. Elsevier.

[21] Kurulay, M. & Bayram, M. 2012. Solutions to fractional partial differential equations via the generalized Mittag-Leffler function. *Mathematical and Computer Modelling,* 55(3–4), 303–311.

[22] Miller, K. S. & Ross, B. 1993. *An Introduction to Fractional Calculus and Fractional Differential Equations.* Wiley.

[23] Mohammad, T., Neeraj, D., Deependra, N. & Anand, C. 2021. Approximation of Caputo time-fractional diffusion equation using redefined cubic exponential B-spline collocation technique. *AIMS Mathematics*, 6(4), 3805–3820.

[24] Muhiddin, F. A., Sulaiman, J. & Sunarto, A. 2019. Numerical performance of half-sweep SOR iteration with the Grünwald implicit finite difference for time-fractional parabolic equations. *Journal of Advanced Research in Dynamical and Control Systems*, 11(12 Special Issue), 119–125.

[25] Paliivets, S., Shpak, I. & Gromov, V. 2021. Fractional-order fluid mechanics modeling. *Applied Mathematical Modelling,* 97, 201–214.

[26] Paradisi, P. 2015. Fractional calculus in statistical physics: The case of time fractional diffusion equation. *Communications in Applied and Industrial Mathematics*, e-530, 1–25.

[27] Podlubny, I. 1999. *Fractional Differential Equations*. Academic Press.

[28] Radzuan, N. Z., Suardi, F. M. M. & Sulaiman, J. 2017. KSOR iterative method with quadrature scheme for solving system of Fredholm integral equations of second kind. *Journal of Fundamental and Applied Sciences*, 9(5S), 609–623.

[29] Rashid, S., Khalid, M. & Mansoor, S. 2021. Numerical simulation of time-fractional diffusion models using wavelet methods. *Mathematical Methods in the Applied Sciences*, 44(10), 7458–7476.

[30] Sunarto, A., Sulaiman, J. & Saudi, A. 2014. SOR method for the implicit finite difference solution of time-fractional diffusion equations. *Borneo Science*, 34, 34–42.

[31] Sunarto, A., Sulaiman, J. & Saudi, A. 2016. Caputo's implicit solution of time-fractional diffusion equation using half sweep AOR iteration. *Global Journal of Pure and Applied Mathematics*, 12(4), 3469–3479.

[32] Wang, Y. M. & Ren, L. A. 2019. High-order L2-compact difference method for Caputo-type time fractional sub-diffusion equations with variable coefficients. *Applied Mathematics and Computation*, 342, 71–93.

[33] Wright, E. M. 1935. The asymptotic expansion of the generalized Bessel function. *Proceedings of the London Mathematical Society*, 38(1), 257–270.

[34] Wu, L., Zhao, Y. & Yang, X. 2018. Alternating segment explicit-implicit and implicit-explicit parallel difference method for time fractional sub-diffusion equation. *Journal of Applied Mathematics and Physics*, 6(5), 1017–1033.

[35] Xu, Q. & Xu, Y. 2018. Extremely low order time-fractional differential equation and application in combustion process. *Communications in Nonlinear Science and Numerical Simulation*, 64, 135–148.

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

[37] Youssef, I. K. 2012. On the Successive Overrelaxation method. *Journal of Mathematics and Statistics*, 8(2), 176–184.

[38] Youssef, I. K. & Taha, A. A. 2013. A generalization of the Successive Overrelaxation method. *Applied Mathematics and Computation*, 219(9), 4601–4613.

[39] Yuste, S. B. 2006. Weighted average finite difference method for fractional diffusion equations. *Journal of Computational Physics*, 216, 264–274.

[40] Zahra, W. K. & Elkholy, S. M. 2013. Cubic spline solution of fractional Bagley-Torvik equation. *Electronic Journal of Mathematical Analysis and Applications*, 1(2), 230–241.

[41] Zhang, Y. 2009. A finite difference method for fractional partial differential equation. *Applied Mathematics and Computation*, 215, 524–529.