SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Implementation of the KSOR Method for Solving One-Dimensional Time-Fractional Parabolic Partial Differential Equations with the Caputo Finite Difference Scheme Title of Manuscript

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

[1] Faculty Science and Natural Resources, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia
[2] Mathematical Sciences Studies, Universiti Teknologi MARA Sabah Branch, 88997 Kota Kinabalu, Sabah, Malaysia
[3] Tadris Matematika,Universitas Islam Negeri (UIN) Fatmawati Sukarno, Bengkulu 38211, Indonesia

ARTICLE INFO

Article history:
Received 25 September 2023
Received in revised form 11 January 2024
Accepted 19 January 2024
Available online 5 July 2024

Keywords:

Caputo finite difference scheme; time-fractional parabolic equations; KSOR method

ABSTRACT

This study presents numerical solution of time-fractional linear parabolic partial differential equations (PDEs) using the Caputo finite difference scheme. The discretization process is based on the second-order implicit finite difference scheme and the Caputo fractional derivative operator. The resulting system of linear approximation equations is solved using the Kaudd Successive Over Relaxation (KSOR) iterative method. A comparison is made with the Gauss-Seidel (GS) iterative method through three numerical examples. The results demonstrate that the KSOR method requires fewer iterations and reduced computational time compared to the GS method.

1. Introduction

Fractional differential equations (FDEs) have recently garnered substantial attention due to their capacity to simulate various complex phenomena in fields such as sciences, engineering and physics including anomalous diffusion [1,2], fluid mechanics [3,4], and image processing [5]. The research area in time-fractional parabolic equations (TFPEs) has evolved as a useful mathematical tool for explaining time-fractional events where the derivative order is non-integer. This is because it may produce superior models that capture non-classical occurrences for complex physical real-world problems in particular cases [6]. Moreover, fractional operators are crucial for understanding a wide range of complicated mechanical and physical behaviours, as well as problem-solving involving non-Markovian random walks [6], which involve systems with long-term memory. However, there are major practical challenges in solving the corresponding fractional differential equation. It should be emphasized that only a few fractional differential equations may be solved analytically utilizing complex functions such as the Mittag-Lefer function [7], H-function [8], and Wright function [9]. As a

* *Corresponding author.*
E-mail address: jumat@ums.edu.my

https://doi.org/10.37934/araset.48.1.168179

result, various numerical techniques for solving TFPE have recently been devised, which appear to be more capable of dealing with the complexities of fractional-order equations. Recent studies have employed the reduced spline (RS) method based on a proper orthogonal decomposition (POD) technique [10,11], while the Crank-Nicholson strategy leverages the finite element approach [12,13].

Researchers have proposed and explored the following strategies in the literature: Method for alternating segment explicit-implicit/implicit explicit parallel difference method [14] is one way. A new method based on fractional finite differences [15], the use of localized radial basis functions (RBFs) [16], and the fractional differential quadrature (FDQ) method [17]. Other researchers had previously concentrated on the implicit scheme [10-12] to discretize the TFPEs problem. They implemented the Caputo finite difference scheme and the Caputo fractional operator into the approximation equations, generating a linear system at each time step. The numerical solution of TFPE generated a large and sparse system of linear equations (SLE), which demands iterative methods for effective computation. Due to the slow convergence rate of the point iterative family, such as GS technique, the SOR iterative method has arisen as a notable option for resolving this problem [22-27].

Extensive study has been undertaken in the literature to investigate point iterative approaches for solving SLE deriving from the discretization of differential equations with integer-order. However, there has been limited study on the application of these approaches to fractional differential equations [25,26]. Currently, most of the previous research in this field has been employing the Caputo fractional derivative operator. The primary goal of this study is to evaluate the effectiveness of the KSOR iterative approach in solving TPPDEs using Caputo's implicit finite difference approximation equation. In addition, we developed the GS iterative methods as a benchmark to compare and demonstrate the capabilities of the KSOR approach. To assess the efficiency of the KSOR technique, we consider TFPEs, which are defined as the target equations in our analysis as follows:

$$\frac{\partial^{\alpha} u}{\partial t^{2}} = \gamma \frac{\partial^{2} u}{\partial x^{2}} + \rho \frac{\partial u}{\partial x} + \theta u(x,t) + f(x,t),\ x \in [\rho_0, \rho_1], 0 \le t \le T \tag{1}$$

subject to the following initial and boundary conditions

$$u(x,0) = u_0(x),$$

and

$$u(x,t) = f(x)$$

where γ, ρ and θ were arbitrary constants, and $f(x,t)$ was a known function. The parameter α characterizes the fractional order of the time derivative, taking a value in the inclusive range of $0 \le \alpha \le 1$. The discretization of the time component in Eq. (1) involves the use of a fractional operator. In this study, we employ the Caputo-type fractional derivative operator, as defined in [10,26,33,]. Additionally, we provide a comprehensive exposition of our numerical approach, emphasizing its theoretical foundation, implementation strategy, and validation through numerical experiments.

2. Preliminaries

We begin with some fundamental definitions before constructing the finite difference approximation equation of Eq. (1).

2.1 Definition

The Riemann-Liouville [28] fractional integral operator, J^{α} of order-α is defined as

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

2.2 Definition

The Caputo's [28] fractional partial derivative operator, $D^{\propto}$ of order-α is defined as

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

with $m-1 < \alpha \leq m, m \in N, x > 0$.

This paper compares the KSOR with the GS iterative methods for solving Eq. (1), incorporating variable coefficients. The numerical solution employs Caputo's derivative formulation, integrating Dirichlet boundary conditions, and the non-local fractional derivative operator. The proposed approximation equation belongs to a category of schemes that are unconditionally stable. By utilizing Eq. (1), the solution domain is confined to a finite space domain, of $0 \leq x \leq \alpha$, with $0 \leq \alpha \leq 1$, and the parameter α correlates with the fractional order of the space derivative. The initial boundary conditions of Eq. (1) are considered to obtain the solution.

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

where $U_0(x)$, and *f(x)* are given functions. Caputo's fractional partial derivative of order α, defined by [25,26], is used for constructing the discrete approximation to the time fractional derivative in Eq. (1).

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

The subsequent sections of the paper are organized as follows: Section 2 provides an approximation formula for the fractional derivative and outlines a numerical strategy for solving the time fractional parabolic Eq. (1) using Caputo's implicit finite difference method. Section 3 details the formulation of the KSOR iterative method, and Section 4 presents the numerical experiments. Conclusions are drawn in Section 5.

3. Derivation of Caputo's Implicit Finite Difference Approximation Equation

Before constructing the Caputo's implicit finite difference approximation equation, we provide a succinct overview of the discretization methodology applied to address Eq. (1). The formulation of Caputo's fractional partial derivative is encapsulated in Eq. (4), adhering to the first-order approximation framework as elucidated by Murio (2008) [34]

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

where

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

and

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

We assume that the problem's solution domain is uniformly partitioned before discretizing Eq. (1). To do 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 labelled $t_i = jk,\ j = 0,1,2,\dots,n$. The values of the function $U(x,t)$ at these grid points are denoted as $U_{i,j} = U(x_1, t_j)$. We obtain Caputo's implicit finite difference approximation equation of Eq. (1) by utilizing Eq. (5) and employing the implicit finite difference discretization scheme, for the grid point centered $(x_i, t_j) = (ih, nk)$. This equation is expressed as follows:

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

For $i = 1,2,\dots,m-1$.

The Eq. (6) highlights that 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. Importantly, the structure of this approximation equation is adaptable based on the chosen time level. For instance, let's consider the case where $n \geq 2$:

$$\sigma_{\alpha,k} \sum_{j=1}^{n} \omega_j^{(\alpha)} \left(U_{i,n-j+1} - U_{i,n-j}\right) = \left(\frac{\gamma}{h^2} - \frac{\rho}{2h}\right) U_{i-1,n} + \left(\theta - \frac{2\gamma}{h^2}\right) U_{i,n} + \left(\frac{\gamma}{h^2} + \frac{\rho}{2h}\right) U_{i+1,n} + f_{i,n} \quad (7)$$

$$\therefore \sigma_{\alpha,k} \sum_{j=1}^{n} \omega_j^{(\alpha)} \left(U_{i,n-j+1} - U_{i,n-j}\right) = \beta_0 U_{i-1,n} + \beta_1 U_{i,n} + \beta_2 U_{i+1,n} + f_{i,n}$$

where

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

Finally, by rearranging Eq. (7), we arrive at the case where the value of $n = 1, \omega_j^{(\alpha)} = 1$

$$\sigma_{\alpha,k}\left(U_{i,1} - U_{i,0}\right) = \beta_0 U_{i-1,1} - \beta_1 U_{i,1} + \beta_2 U_{i+1,1} + f_{i,1} \quad (8)$$

Therefore, the approximation Eq. (8) can be rewritten as follows.

$$-p_i U_{i-1,1} + q_i U_{i,1} - r_i U_{i+1,1} = f_{i,1}^{*}, \quad i = 1,2,\dots,m-1 \quad (9)$$

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 Eq. (9) can be expressed in a matrix form as

$$\underset{\sim}{AU} = \underset{\sim}{f} \tag{10}$$

where,

$$A = \begin{bmatrix} q* & -r* & & & & \\ -p* & q* & -r* & & & \\ & -p* & q* & -r* & & \\ & & \ddots & \ddots & \ddots & \\ & & & -p* & q* & -r* \\ & & & & -p* & q* \end{bmatrix}_{(m-1)\times(m-1)},$$

$$\underset{\sim}{U} = [U_{11} \quad U_{21} \quad U_{31} \quad \cdots \quad U_{m-2,1} \quad U_{m-1,1}]^T,$$

$$\underset{\sim}{f} = [U_{11} + p_1 U_{01} \quad U_{21} \quad U_{31} \quad \cdots \quad U_{m-2,1} \quad U_{m-1,1} + p_{m-1} U_{m,1}]^T.$$

The structure of Eq. (9) was identified as a tridiagonal matrix, given that its nonzero elements are present solely on the main diagonals immediately above and below it. This tridiagonal matrix representation facilitated the application of KSOR iterative method.

4. Results Implementation of KSOR Iterative Method

In this part, we investigate the application KSOR method [17] for solving the linear system that results from discretizing Eq. (1). The KSOR method is a modified version of SOR iterative method. As a comparison, we consider the GS iterative method as a benchmark to assess the efficiency of KSOR iterative method. When the relaxation parameter $\omega = 1$, GS iterative method is equivalent to the SOR iterative method. The purpose of this study is to demonstrate the efficiency of the KSOR iterative method for solving Eq. (1). This method is specifically designed to handle the second-order implicit finite difference scheme and the Caputo fractional derivative operator. To develop the formulation of the KSOR iterative method, we decompose the coefficient matrix A in Eq. (9) as follows:

$$A = D + L + V \tag{11}$$

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 Eq. (11) as [19-21].

$$\underset{\sim}{U}_j^{(k+1)} = (D - \omega L)^{-1}[\omega V + (1-\omega)D]\underset{\sim}{U}_j^{(k)} + (D - \omega L)^{-1} f, \tag{12}$$

where $\underset{\sim}{U}_j^{(k)}$ represents the unknown vector at the k^{th} iteration and relaxation parameter $\omega \in [1,2)$.

Meanwhile, Using Eq. (8) and Eq. (12), the SOR scheme based on the point iteration can be expressed as

$$U_{i,j}^{(k+1)} = (1-\omega)U_{i+1,j}^{(k)} + \frac{\omega}{q_i}(p_i U_{i-1,1} + r_i U_{i+1,1} - f_{i,1}^{*}) \quad i = 1,2,\dots,n; \quad j = 1,2,3,\dots,M \tag{13}$$

Recently, [21] introduced a new type of SOR iterative method called the KSOR iterative method, in which the relaxation parameter ω* within the range of R − [−2,0]. In matrix representation, the generic form of KSOR is written as

$$\underset{\sim}{U}_j^{(k+1)} = \left(\left((1-\omega^*)D - \omega^* L\right)^{-1}(D + \omega^* V)\right)\underset{\sim}{U}_j^{(k)} + \left((1-\omega^*)D - \omega^* L\right)^{-1}(\omega^* \underset{\sim}{f}_j) \tag{14}$$

For the KSOR iterative approach, the relaxation parameter ω is extended to R- [-2,0]. Remember that the relaxation parameter for the conventional SOR iterative approach is $0 \leq \omega < 2$. Algorithm 1 summarises the general algorithm of the KSOR iterative technique for solving SLE (9).

Algorithm 1: KSOR scheme

i. Initialize $\underset{\sim}{U}_j^{(k+1)} = 0$ and $\varepsilon = 10^{-10}$
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
$\left\| \underset{\sim}{U}^{(k+1)} - \underset{\sim}{U}^{(k)} \right\| \leq \varepsilon = 10^{-10}$ is satisfied, go to step (v). Otherwise, go back to step (iii)
v. Display approximate solutions.

5. Numerical Experiments

In this paper, numerical experiments were performed to assess the efficiency of the suggested method utilizing the C programming language, which is a versatile and efficient tool for computational tasks. To do this, we evaluated three examples of time fractional parabolic partial differential equations. The goal was to validate the efficiency of the KSOR and GS iterative methods, based on the three criteria: the number of iterations (K), the computational time in seconds (t), and the maximum error at three different values of α = 0.25, α = 0.50, and α = 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.

5.1 Example 1 [29]

Consider the following time fractional initial boundary value problem

$$\frac{\partial^{\alpha} u(t,x)}{\partial t^2} - \frac{\partial^2 u(t,x)}{\partial x^2} = f(x,t), \ t \in [0,1], \ t \geq 0, \ 0 < x < 1, \tag{15}$$

Where the boundary conditions are given in $u(0,t) = u(1,t) = 0, \quad t \in [0,1]$, and initial condition is $u(t,0) = 0, u(t,1) = 0, \quad 0 < x < 1$

Where the exact solution is $u(x,t) = t^2 \sin 2\pi x$ and $f(x,t) = \frac{2}{\Gamma(3-\alpha)} t^{2-\alpha} \sin(2\pi x) + 4\pi^2 \sin(2\pi x) t^2$.

5.2 Example 2 [30]

Consider the following time fractional initial boundary value problem

$$\frac{\partial^\alpha u(t,x)}{\partial t^2} - \frac{\partial^2 u(t,x)}{\partial x^2} = 3.009011112 t^{\frac{3}{2}} \sin(\pi x) \cos(\pi x) + 4t^2 \sin(2\pi x)\, \pi^2, \quad (0 < x < 1, 0 < 1) \quad (16)$$

Where the boundary conditions are given in

$$u(0,x) = u(1,x) - sin(2\pi x), \quad 0 \le x \le 1,$$

and initial condition $u(t,0) = 0, \quad u(t,1) = 0, \quad 0 \le x \le 1$. The exact solution is $u(t,x) = t^2 \sin(2\pi x)$.

5.3 Example 3 [31]

Consider the following time fractional initial boundary value problem

$$\frac{\partial^\alpha u(t,x)}{\partial t^2} = \frac{\partial^2 u(t,x)}{\partial x^2} + f(x,t), \ x \in [0,1], \ t \ge 0, \ 0 < \alpha < 1, \quad (17)$$

Where the exact solution is $u(x,t) = t^2 \, (\mathrm{x} - 1)^2 \sin(2\pi x)$ and

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

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

Table 1
Comparison of the number of iterations (K), computational time, t (Seconds), and maximum errors for iterative algorithms using Example 1 at α = 0.25, α = 0.50, and α = 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
	KSOR	2364	9.50	1.2801E-03	513	7.90	4.4628E-03	605	8.12	7.9838E-03
		ω=-2.0346			ω=-2.0346			ω=-2.0346		
1024	GS	173277	517.97	1.2831E-03	82433	265.18	4.4645E-03	21924	152.50	7.9844E-03
	KSOR	26385	86.41	1.2801E-03	2537	35.83	4.4632E-03	733	16.10	7.9839E-03

		ω=-2.0346			ω=-2.0346			ω=-2.0346		
2048	GS	569412	4134.55	1.2898E-03	276232	1662.40	4.4687E-03	74187	536.333	7.9857E-03
	KSOR	78926	745.85	1.2805E-03	8527	239.90	4.4633E-03	1999	36.52	7.9840E-03
		ω=-2.0346			ω=-2.0346			ω=-2.0346		
4096	GS	968304	6223.24	1.2910E-03	893663	7568.23	4.4851E-03	242796	2772.33	7.9911E-03
	KSOR	78926	1088.85	1.2814e-03	27707	754.98	4.4637E-03	6906	107.61	7.9841E-03
		ω=-2.0346			ω=-2.0346			ω=-2.0346		
8192	GS	1292468	15634.89	1.2981E-03	1093664	13246.30	4.4913E-03	755078	9497.12	8.0123E-03
	KSOR	213651	2345.88	1.2850E-03	91168	1030.26	4.4649-03	23731	444.45	7.9845E-03
		ω=-2.0346			ω=-2.0346			ω=-2.0346		

Table 2
Comparison of the number of iterations (K), computational time, t (Seconds), and maximum errors for iterative algorithms using Example 1 at α = 0.25, α = 0.50, and α = 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
	KSOR	21583	17.44	7.4557E-04	2462	6.63	4.4632E-03	1863	2.06	8.1974E-03
		ω=-2.7088			ω=-2. 7088			ω=-2. 7088		
1024	GS	174667	444.32	7.4834E-04	82432	207.83	4.4647E-03	21651	61.32	8.1980E-03
	KSOR	61284	99.40	7.4663E-04	25350	44.14	4.4637E-03	6534	12.19	8.1977E-03
		ω=-2. 7088			ω=-2. 7088			ω=-2. 7088		
2048	GS	574443	2851.48	7.5512E-04	276231	1314.58	4.4689E-03	73097	384.56	8.1994E-03
	KSOR	185180	611.75	7.4894E-04	86153	288.38	4.4649E-03	22557	76.37	8.1981E-03
		ω=-2. 7088			ω=-2. 7088			ω=-2. 7088		
4096	GS	1035653	6225.50	7.5511E-04	564235	3576.54	4.4689E-03	189546	752.56	8.1980E-03
	KSOR	299989	1321.12	7.5604E-04	157696	1021.95	4.4693E-03	51235	128.65	8.1995E-03
		ω=-2. 7088			ω=-2. 7088			ω=-2. 7088		
8192	GS	2054687	9216.66	7.5532E-04	1307158	7635.36	4.4688E-03	702565	1532.56	8.1980E-03
	KSOR	497983	2523.11	7.4663E-04	331544	2143.41	4.4693E-03	114080	246.64	8.1995E-03
		ω=-2. 7088			ω=-2. 7088			ω=-2. 7088		

Table 3
Comparison of the number of iterations (K), computational time, t (Seconds), and maximum errors for iterative algorithms using Example 1 at α = 0.25, α = 0.50, and α = 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
	KSOR	21631	23.48	1.8086E-04	5833	11.63	2.1951E-04	1197	8.31	1.6509E-04
		ω=-2.6999			ω=-2.6999			ω=-2.6999		
1024	GS	308691	1076.17	2.5983E-03	80866	265.29	4.6239E-03	17184	124.50	2.5799E-03
	KSOR	69441	110.17	1.8203E-04	18673	39.68	2.1993E-04	3866	19.86	1.6518E-04
		ω=-2.6999			ω=-2.6999			ω=-2.6999		
2048	GS	55978	2537.32	2.5983E-03	254484	1222.56	4.6386E-03	55978	140.95	2.5983E-03
	KSOR	20936	512.31	1.8667E-04	56184	159.39	2.2162E-04	11800	56.67	1.6555E-04
		ω=-2.6999			ω=-2.6999			ω=-2.6999		
4096	GS	174175	4869.22	2.6075E-03	750184	5845.53	4.6459E-03	174175	691.32	2.6075E-03
	KSOR	56246	1194.00	2.0515E-04	151274	611.69	2.2809E-04	32395	173.13	1.6703E-04
		ω=-2.6999			ω=-2.6999			ω=-2.6999		
8192	GS	2361262	53241.22	2.6075E-03	1990784	19038.98	4.6496E-03	506195	5638.00	2.6121E-03
	KSOR	843690	5856.00	2.0515E-04	314776	2173.42	2.5148E-04	66565	475.05	1.7284E-04
		ω=-2.6999			ω=-2.6999			ω=-2.6999		

Table 4 summarizes the percentage decrease of the KSOR iterative approach in comparison to the GS iterative method for the Problems 1–3. According to the results, the KSOR iterative method manages to lower the GS iteration numbers. Furthermore, the computational time increases as the mesh size increases. This clearly shows that the KSOR iterative approach is more efficient than the GS iterative method.

Table 4
Reduction in the number of iterations (Iter) and computational time (Time) for the KSOR when compared to the GS iterative approach in percent

Example		α = 0.25	α = 0.50	α = 0.75
1	Iter	83.50% - 95.61%	91.66% - 97.87%	90.44% - 96.85%
	Time	84.94% - 91.74%	91.20% - 91.66%	84.50% - 95.32%
2	Iter	60.30% - 72.62%	64.75% - 74.64%	70.25% - 80.20%
	Time	72.62% - 74.83%	71.62% - 78.76%	74.15% - 83.94%
3	Iter	64.30% - 76.88%	76.27% - 84.18%	76.54% - 86.85%
	Time	86.05% -89.00%	82.50% - 88.60%	81.05% - 91.57%

6. Conclusion

The KSOR method was introduced in the preceding section utilizing a second-order implicit finite difference scheme with the Caputo fractional derivative operator. Based on the numerical results presented in Table 1 to Table 3, the KSOR iterative method performs better than the GS iterative method in terms of iteration numbers and computational time. This improved performance due to the selection of an optimal relaxation parameter, which helps int the achievement of an optimal convergence rate. Furthermore, the numerical solutions obtained from the KSOR iterative method demonstrate a strong correlation with those obtained from the GS iterative method [18,30]. The concept, introduced by [16], presents an alternative method to reduce the computational time for solving TFPEs. This expansion will allow for further enhancements and improvements in solving such equations efficiently. This study can be expanded in the future by considering the utilization of the half-sweep iteration concept as discussed in [18], which was inspired by Abdullah (1991) as an alternative approach to speed up the execution time for solving one-dimensional TFPEs. Other point iterations, such as [19] and red-black half-sweep iteration [33] can be utilized in addition to the half-sweep iteration to solve the proposed problem.

Acknowledgement
This research was supported by Universiti Malaysia Sabah (Postgraduate research grant: GUG0585-1/2023).

References

[1] Xu, Qinwu, and Yufeng Xu. "Extremely low order time-fractional differential equation and application in combustion process." *Communications in Nonlinear Science and Numerical Simulation* 64 (2018): 135-148. https://doi.org/10.1016/j.cnsns.2018.04.021

[2] Edeki, S. O., O. O. Ugbebor, and E. A. Owoloko. "Analytical solutions of a time-fractional nonlinear transaction-cost model for stock option valuation in an illiquid market setting driven by a relaxed Black–Scholes assumption." *Cogent Mathematics* 4, no. 1 (2017): 1352118. https://doi.org/10.1080/23311835.2017.1352118

[3] Paradisi, Paolo. "Fractional calculus in statistical physics: The case of time fractional diffusion equation." *Communications in Applied and Industrial Mathematics* 6, no. 2 (2015).

[4] Almeida, Ricardo, Nuno RO Bastos, and M. Teresa T. Monteiro. "Modeling some real phenomena by fractional differential equations." *Mathematical Methods in the Applied Sciences* 39, no. 16 (2016): 4846-4855. https://doi.org/10.1002/mma.3818

[5] Gaspar, Francisco J., and Carmen Rodrigo. "Multigrid waveform relaxation for the time-fractional heat equation." *SIAM Journal on Scientific Computing* 39, no. 4 (2017): A1201-A1224. https://doi.org/10.1137/16M1090193

[6] Ghaffari, Rezvan, and Farideh Ghoreishi. "Reduced spline method based on a proper orthogonal decomposition technique for fractional sub-diffusion equations." *Applied Numerical Mathematics* 137 (2019): 62-79. https://doi.org/10.1016/j.apnum.2018.11.014

[7] Kumar, Sachin. "Crank-Nicolson Quasi-Wavelet Method for the Numerical Solution of Variable-Order Time-Space Riesz Fractional Reaction-Diffusion Equation." In *Applications of Fractional Calculus to Modeling in Dynamics and Chaos*, pp. 407-428. Chapman and Hall/CRC, 2022. https://doi.org/10.1201/9781003006244-16

[8] Wang, Yuan-Ming, and Lei Ren. "A high-order L2-compact difference method for Caputo-type time-fractional sub-diffusion equations with variable coefficients." *Applied Mathematics and Computation* 342 (2019): 71-93. https://doi.org/10.1016/j.amc.2018.09.007

[9] Wu, Lifei, Yadi Zhao, and Xiaozhong Yang. "Alternating Segment Explicit-Implicit and Implicit-Explicit Parallel Difference Method for Time Fractional Sub-Diffusion Equation." *Journal of Applied Mathematics and Physics* 6, no. 05 (2018): 1017. https://doi.org/10.4236/jamp.2018.65089

[10] Sunarto, Andang, J. Sulaiman, and A. Saudi. "Application of the full-sweep AOR iteration concept for space-fractional diffusion equation." In *Journal of Physics: Conference Series*, vol. 710, no. 1, p. 012019. IOP Publishing, 2016. https://doi.org/10.1088/1742-6596/710/1/012019

[11] Sunarto, Andang, J. Sulaiman, and A. Saudi. "Implicit finite difference solution for time-fractional diffusion equations using AOR method." In *Journal of Physics: Conference Series*, vol. 495, no. 1, p. 012032. IOP Publishing, 2014. https://doi.org/10.1088/1742-6596/495/1/012032
[12] Podlubnv, I. "Fractional differential equations academic press." *San Diego, Boston* 6 (1999).
[13] Zahra, W. K., and S. M. Elkholy. "Cubic spline solution of fractional Bagley-Torvik equation." *Electronic Journal of Mathematical Analysis and Applications* 1, no. 2 (2013): 230-241. https://doi.org/10.21608/ejmaa.2013.309805
[14] Evans, D. J. "Group explicit iterative methods for solving large linear systems." *International Journal of Computer Mathematics* 17, no. 1 (1985): 81-108. https://doi.org/10.1080/00207168508803452
[15] Youssef, I. K. "On the successive overrelaxation method." *Journal of Mathematics and Statistics* 8, no. 2 (2012): 176-184. https://doi.org/10.3844/jmssp.2012.176.184
[16] Radzuan, N. Z. F. M., M. N. Suardi, and J. Sulaiman. "KSOR iterative method with quadrature scheme for solving system of Fredholm integral equations of second kind." *Journal of Fundamental and Applied Sciences* 9, no. 5S (2017): 609-623. https://doi.org/10.4314/jfas.v9i5s.43
[17] Alibubin, Mohd, Andang Sunarto, Jumat Sulaiman, and Azali Saudi. "Half-Sweep AOR Iteration With Rotated Nonlocal Arithmetic Mean Scheme For The Solution Of 2D Nonlinear Elliptic Problems." *Advanced Science Letters* 24, no. 3 (2018): 1922-1926. https://doi.org/10.1166/asl.2018.11189
[18] Alibubin, M. U., A. Sunarto, and J. Sulaiman. "Quarter-sweep nonlocal discretization scheme with QSSOR iteration for nonlinear two-point boundary value problems." In *Journal of Physics: Conference Series*, vol. 710, no. 1, p. 012023. IOP Publishing, 2016. https://doi.org/10.1088/1742-6596/710/1/012023
[19] Young, David M. *Iterative solution of large linear systems*. Elsevier, 2014.
[20] Young, David M. "Second-degree iterative methods for the solution of large linear systems." *Journal of Approximation Theory* 5, no. 2 (1972): 137-148. https://doi.org/10.1016/0021-9045(72)90036-6
[21] Youssef, Ismail K., and A. A. Taha. "On the modified successive overrelaxation method." *Applied Mathematics and Computation* 219, no. 9 (2013): 4601-4613. https://doi.org/10.1016/j.amc.2012.10.071
[22] Baharuddin, Baharuddin, Andang Sunarto, and Juhriyansyah Dale. "QSAOR Iterative Method For The Solution of Time Fractional Diffusions Equation." *Journal Iteratif of Enginering and Applied Science* 12, no. 12 (2017): 3220-3224.
[23] Sunarto, Andang, Jumat Sulaiman, and Azali Saudi. "Implementation of the half-sweep AOR iterative algorithm for space-fractional diffusion equations." *Jurnal Teknologi* 78, no. 64 (2016): 7-12. https://doi.org/10.11113/jt.v78.8968
[24] Demir, Ali, Sertaç Erman, Berrak Özgür, and Esra Korkmaz. "Analysis of fractional partial differential equations by Taylor series expansion." *Boundary Value Problems* 2013 (2013): 1-12. https://doi.org/10.1186/1687-2770-2013-68
[25] Zainal, N. F. A., J. Sulaiman, and M. U. Alibubin. "Application of SOR iteration with nonlocal arithmetic discretization scheme for solving Burger's equation." In *AIP Conference Proceedings*, vol. 2013, no. 1. AIP Publishing, 2018. https://doi.org/10.1063/1.5054234
[26] Zhang, Yang. "A finite difference method for fractional partial differential equation." *Applied Mathematics and Computation* 215, no. 2 (2009): 524-529. https://doi.org/10.1016/j.amc.2009.05.018
[27] Muhiddin, F. A., J. Sulaiman, and A. Sunarto. "Grünwald Implicit Solution for Solving One-Dimensional Time-Fractional Parabolic Equations Using SOR Iteration." In *Journal of Physics: Conference Series*, vol. 1358, no. 1, p. 012055. IOP Publishing, 2019. https://doi.org/10.1088/1742-6596/1358/1/012055
[28] Yuste, Santos B. "Weighted average finite difference methods for fractional diffusion equations." *Journal of Computational Physics* 216, no. 1 (2006): 264-274. https://doi.org/10.1016/j.jcp.2005.12.006
[29] Ford, Neville J., Jingyu Xiao, and Yubin Yan. "A finite element method for time fractional partial differential equations." *Fractional Calculus and Applied Analysis* 14 (2011): 454-474. https://doi.org/10.2478/s13540-011-0028-2
[30] Karatay, Ibrahim, Şerife R. Bayramoğlu, and Ali Şahin. "Implicit difference approximation for the time fractional heat equation with the nonlocal condition." *Applied Numerical Mathematics* 61, no. 12 (2011): 1281-1288. https://doi.org/10.1016/j.apnum.2011.08.007
[31] Tamsir, Mohammad, Neeraj Dhiman, Deependra Nigam, and Anand Chauhan. "Approximation of Caputo time-fractional diffusion equation using redefined cubic exponential B-spline collocation technique." *AIMS Mathematics* 6, no. 4 (2021): 3805-3820. https://doi.org/10.3934/math.2021226
[32] Li, Ruitian, Liang Gong, and Minghai Xu. "A heterogeneous parallel Red–Black SOR technique and the numerical study on SIMPLE." *The Journal of Supercomputing* 76 (2020): 9585-9608. https://doi.org/10.1007/s11227-020-03221-1

[33] Wang, Kaixin, and Hong Wang. "A fast characteristic finite difference method for fractional advection–diffusion equations." *Advances in water resources* 34, no. 7 (2011): 810-816. https://doi.org/10.1016/j.advwatres.2010.11.003

[34] Murio, Diego A. "Implicit finite difference approximation for time fractional diffusion equations." *Computers & Mathematics with Applications* 56, no. 4 (2008): 1138-1145. https://doi.org/10.1016/j.camwa.2008.02.015