Journal of Mathematical Problems, Equations and Statistics

E-ISSN: 2709-9407
P-ISSN: 2709-9393
JMPES 2024; 5(2): 106-113
© 2024 JMPES
www.mathematicaljournal.com
Received: 24-07-2024
Accepted: 26-08-2024

Mahesh T Holambe
Shrikrishna Mahavidyalaya Gunjoti, Maharashtra, India

Vinod B Kulkarni
Swa. Sawarkar Mahavidyalaya, Beed, Maharashtra, India

Corresponding Author:
Mahesh T Holambe
Shrikrishna Mahavidyalaya Gunjoti, Maharashtra, India

Iterative method for solving fractional mathematical physics model

Mahesh T Holambe and Vinod B Kulkarni

DOI: https://dx.doi.org/10.22271/math.2024.v5.i2b.157

Abstract

This paper will solve one of the fractional mathematical physics models, a one-dimensional time-fractional differential equation, by utilizing the second-order quarter-sweep finite-difference scheme and the preconditioned accelerated over-relaxation method. The proposed numerical method offers an efficient solution to the time-fractional differential equation by applying the computational complexity reduction approach by the quarter-sweep technique. The finite-difference approximation equation will be formulated based on the Caputo's time-fractional derivative and quarter-sweep central difference in space. The developed approximation equation generates a linear system on a large scale and has sparse coefficient in terms of the number of iterations and computation time. The quarter-sweep computes a quarter of the total mesh points using the preconditioned iterative method while maintaining the solution's accuracy. A numerical example will demonstrate the efficiency of the proposed quarter-sweep preconditioned accelerated over-relaxation method against the half-sweep preconditioned accelerated over-relaxation, and the full-sweep preconditioned accelerated over-relaxation method. The numerical finding showed that the quarter sweep finite difference scheme and preconditioned accelerated over-relaxation method can serve as an efficient numerical method to solve fractional differential equations.

Keywords: Caputo's fractional derivative, implicit finite-difference scheme, QSPAOR, TFDE

Introduction

The growing interest in the theory and applications of fractional calculus has become the motivation for many researchers in recent years. Fractional calculus has attracted attention of experts from all over the world. Various fractional operators have been introduced in the literature such as [3, 9-11, 27, 30, 31], and this encourages more extensive researches to be conducted. Solving fractional differential equations (FDEs) using numerical methods has been: seen as an ongoing research trend. 'The analytical s solutions of most FDEs are challenging compared to the ordinary (ODEs) and partial differential equations (PDEs) in general. Therefore, numerical solutions are: actively being found by proposing new numerical approximation techniques to solve the FDEs. Some notable numerical methods have been developed to solve the fractional partial derivatives problems [1, 2, 14, 19, 20, 29]. Besides that, [12] has presented several interesting MATLAB routines for solving FDEs. The author has provided many solution techniques for solving three identified FDE problems such as the standard FDEs, the multiorder systems of FDEs, and the multiterm FDEs One of the studies [13] presented several computational cost evaluations for the numerical solutions of FDEs from the point of view of computer science. Based on that work the computational complexities for the time-fractional, space-fractional, and space-time FDEs are known to be $O(N^2 M)$, $O(NM^2)$, and O(NM(M+N)). The authors have also compared the three mentioned computational costs against O(MN), which is the cost of finding solutions for the classical partial differential equations using finite-difference methods. Moreover, here, M the and N denote the number ofs spatial grid points and time: steps, respectively authors have mentioned that the preconditioner technique is a good alternative to accelerate the computational process in solving FDEs

In our development of the numerical method to solve FDEs, we are interested in applying the second-order quarter-sweep finite-difference scheme with a preconditioning technique to solve the time-fractional FDE (TFDE). 'There are several finite difference scheme applications t to solve the TFDE [5, 7, 12,1 25, 26]. However, the investigation on the efficiency of the numerical method used to: solve the TFDE is quite limited. The quarter-Sweep finite- difference scheme has been a good computation complexity reduction approach, especially when large number of mesh points are considered [4, 22, 28].

The quarter-sweep is able to reduce the computational complexity of computing the solutions of a large linear system by computing a quarter of the total number of mesh points without offsetting the solutions' accuracy. Therefore, this paper investigates the efficiency of the quarter-sweep finite difference scheme with a preconditioning technique called PAOR [26] to solve the TFDE. The PAOR iterative method will be used to compute a quarter of mesh points after quarter-sweep implementation. The remaining mesh points will be estimated by averaging. This efficient numerical method is important to the physicists to aid their investigation on the time-fractional mathematical model, arising from the necessity to sharpen the concepts of equilibrium, stability states, and time evolution in the long-time limit [8, 17, 18].

Throughout this paper, we discretized TFDE using the unconditionally stable second-order quarter-sweep implicit finite-difference (QSIFD) scheme. We used Caputo's fractional partial derivative to form the approximation equation. Usually, the finite-difference approximation equation's implementation leads to a tridiagonal matrix of the linear system due to its characteristics. The discretized finite-difference approximations also form a large and sparse matrix which is the best alternative to be solved using the iterative method. We have observed the successful iterative methods from many researchers. From many discussions and extensions made ins several categories of i iterative methods, we find that the preconditioned iterative methods have the unique properties to solve a linear system efficiently [15, 16, 23].

This paper's main contribution is to present the efficiency of our proposed numerical method, which can be called the quarter-sweep preconditioned accelerated over-relaxation (QSPAOR) iterative method for solving TFDEs. In this paper, the numerical method's efficiency is evaluated based o the number of iterations and the computation time. The QSPAOR iterative method's over-relaxation and full sweep preconditioned accelerated over-relaxation and the improvement in terms of the reduction of both the quarter-sweep finite difference approximation equation and the AOR iterative method's convergence analysis are also provided.

The general TFDE that we consider as the main problem to be solved can be written as

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

Where $p(x), q(x)\ and\ r(x)$ are known functions or coefficients; meanwhile, α is a parameter that determines the degree of fractional order for the time derivative. For the formulation pf the finite-difference approximation with Caputo's derivative, here are the important definitions that we use:

Definition 1: the Rirmann-Liouville fractional integral operator J^α, of order α is defined as

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

Definition 2: the Caputo's fractional partial derivative operator, D^α of order α is defined as

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

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

Research methodology

To solve the fractional differential problem shown in Eq.(1) we assume that the solutions exist and satisfy the Dirichlet boundary conditions. Therefore, using eq (2) the time fractional derivative term in eq(1) is discretized using

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{1}{\Gamma(n-1)}\int_0^\infty \frac{\partial u(x-s)}{\partial t}(t-s)^{-\alpha}ds,\ t > 0, 0 < \alpha < 1. \tag{4}$$

Using the approximation equation to Eq. (1) employing the finite-difference method and Caputo's fractional derivative, we develop a C++ code for the simulation of the approximate solutions. We have two examples of the TFDE to examine the iterative methods, i.e., the proposed QSPAOR, HSPAOR, and FSPAOR. The proposed numerical method's efficiency is examined using the number of iterations (K) and the computation time measured in seconds. The maximum absolute error (MAE) is also observed for accuracy checking. These criteria are compared by using three different order parameters α, i.e., α = 0.25, α = 0.50, and α = 0.75. The convergence tolerance, $\varepsilon = 10^{-10}$ is set to terminate the iteration process.

Approximation to the time fractional differential equation

The first-order approximation to the Caputo's fractional derivative, which is derived from the discrete approximation to the time-fractional derivative term shown in Eq. (4), can be written as

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

Where, from eq(5), we define two representations for the sake of simplicity as follows:

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

And,

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

Then we use the common discretization by partitioning the solution domain of eq(1) uniformly, subjected by the Dirichlet boundary conditions. The numbers $m \, and \, n \, (m, n \ni N^*)$ are defined so that the grid framework in space and time is fixed everywhere and has increments denoted as $h = \Delta x = \frac{\gamma - 0}{m} \, and \, k = \Delta t = \frac{T}{n}$, respectively. Based on the developed uniform grid network, the grid points in the space interval [0, γ] are represented by $x_i = ih, for \, i = 0, 1, 2, \ldots, m.$ Meanwhile the grid points in the time interval [0, T] are labelled as $t_j = jk \, for \, j = 0, 1, 2, \ldots n.$ therefore, the values of the function U(x, t) at the grid points are expressed as $U_{i,j} = U(x_i, t_j)$.
The implementation of QSIFD discretization scheme for eq(5) produced the Caputo's approximation to Eq(1) at the grid point $(x_i, t_j) = (ih, jk)$ which can be formulated as,

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

$$= \frac{pi}{16h^2}\left(U_{i-4,n} - 2U_{i,n} + U_{i+4,n}\right) + \frac{qi}{8h}\left(U_{i+4,n} - U_{i-4,n}\right) + r_i U_{i,n} \quad (6)$$

For i=4, 8, . . . , m-4.

When the approximation in eq(6) is applied on the specified time level $n \geq 2$ eq(6) can be expressed as,

$$\sigma_{\alpha,k} \sum_{j=1}^{n} \omega_j^{(\alpha)} \left(U_{i,n-j+1} - U_{i,n-j}\right) = p_i' U_{i-4,n} + q_i' U_{i,n} + r_i' U_{i+4,n} \quad (7)$$

And the coefficients are represented by

$$p_i' = \frac{pi}{16h^2} - \frac{qi}{8h} \; q_i' = r_i - \frac{pi}{8h^2} \, , r_i' = \frac{pi}{16h^2} + \frac{qi}{8h}.$$

In addition to this for n=1 we have

$$-p_i' U_{i-4,1} + q_i^* U_{i,1} - r_i' U_{i+4,1} = f_{i,1} \; i = 4, 6, \ldots, m-4 \quad (8)$$

Where $\omega_j^{(\alpha)} = 1, q_i^* = \sigma_{\alpha,k} - q_i', and \; f_{i,1} = \sigma_{\alpha,k} U_{i,1}$

When a certain number of grid points is considered based on eq(8) a system of linear equations is obtained, which can be expressed in the form matrix as follows:

$$AU = f$$

$$A = \begin{bmatrix} q_4^* \; -r_4' \\ -p_8' \; q_8^* \; -r_8' \\ -p_{12}' \; q_{12}^* \; -r_{12}' \\ \cdot \\ \cdot \\ \cdot \\ -p_{m-8}' \; q_{m-8}^* \; -r_{m-8}' \\ -p_{m-4}' \; q_{m-4}^* \end{bmatrix}_{\left(\left(\frac{m}{4}\right)-1\right) x \left(\left(\frac{m}{4}\right)-1\right)}$$

$$U = \left[U_{4,1}, U_{8,1}, U_{12,1}, \ldots., U_{m-8,1}, U_{m-4,1}\right]^T$$

And

$$f = \left[U_{4,1} + p_1' \; U_{0,1} \; U_{8,1}, U_{12,1}, \ldots., U_{m-8,1}, U_{m-4,1} + p_{m-4}' \; U_{m,1}\right]^T$$

Analysis of stability
In this section the stability analysis on the formulated Caputo's finite difference approximation in eq(6) is considered based on von Neumann's approach and the equivalence theorem [21, 24, 33].

Theorem 4.1. the fully IFd approximation to the solution of eq(1) with $0 < \alpha < 1$ on the finite domain $0 \leq x \leq 1$ with zero boundary condition $U(0,t) = U(l,t) = 0 \; for \; all \; t \geq 0$ is consistent and unconditionally stable.

Proof: Writing the solution of eq(1) in the form $U_j^m = \xi_n e^{i\omega jh}, i = \sqrt{-1}, \omega$ is real eq(1) becomes

$$\sigma_{\alpha,k}\xi_{n-1}e^{i\omega jh} - \sigma_{\alpha,k}\sum_{j=2}^{n}\omega_j^{(\alpha)}(\xi_{n-j+1}e^{i\omega jh} - \xi_{n-j}e^{i\omega jh})$$

$$= -pi\xi_n e^{i\omega(j-4)h} + (\sigma_{\alpha,k} - q_i)\xi_n e^{i\omega jh} - ri\xi_n e^{i\omega(j+4)h} \quad (10)$$

By simplifying and reordering eq(10) we get

$$\sigma_{\alpha,k}\xi_{n-1} - \sigma_{\alpha,k}\sum_{j=2}^{n}\omega_j^{(\alpha)}(\xi_{n-j+1} - \xi_{n-j}) = (((-pi - ri)\cos(\omega h)) + (\sigma_{\alpha,k} - qi)) \quad (11)$$

Eventually from eq(11) we reduce to

$$\xi_n = \frac{\xi_{n-1} + \sum_{j=2}^{n}\omega_j^{(\alpha)}(\xi_{n-j} - \xi_{n-j+1})}{\left(1 + \frac{(pi+ri)}{\sigma_{\alpha,k}}\cos(\omega h) + \frac{q_i}{\sigma_{\alpha,k}}\right)} \quad (12)$$

Hence, from eq(12) it can be observed that

$$\left(1 + \frac{(pi+ri)}{\sigma_{\alpha,k}}\cos(\omega h) + \frac{q_i}{\sigma_{\alpha,k}}\right) \geq 1 \quad (13)$$

For all $\alpha, n, \omega, h, and\ k,$ then we have the inequality $\xi_1 \leq \xi_0$ and

$$\xi_n \leq \xi_{n-1} + \sum_{j=2}^{n}\omega_j^{(\alpha)}(\xi_{n-j} - \xi_{n-j+1}), n \geq 2 \quad (14)$$

Based on eq(14) for n=2 we obtain

$$\xi_2 \leq \xi_1 + \omega_2^{(\alpha)}(\xi_0 - \xi_1) \quad (15)$$

Then by repeating the same process as in eq(15) we can get

$$\xi_j \leq \xi_{j-1}, j = 1, 2, \ldots, n-1 \quad (16)$$

From eq(16) we finally have

$$\xi_n \leq \xi_{n-1} + \sum_{j=2}^{n}\omega_j^{(\alpha)}(\xi_{n-j} - \xi_{n-j+1}) \leq \xi_{n-1} \quad (17)$$

Since each term in the sum shown in eq(17) is negative, it implies that the inequalities in eq(16) & eq(17) can be generalized into

$$\xi_n \leq \xi_{n-1} \leq \xi_{n-2} \leq \ldots \leq \xi_1 \leq \xi_0 \quad (18)$$

Thus $\xi_n = |U_j^n| \leq \xi_0 = |U_j^0| = |f_j|$ which entails $\|U_j^n\| \leq \|f_j\|$ and we have stability. It follows that the numerical solution of the approximation equation to eq(1) converges to the exact solution as $h, k \to 0$.

QSPAOR iterative method

In this section, we discuss solving the tridiagonal linear system as in eq(9). To formulate the QSPAOR iterative method first convert the initial linear system into the preconditioned system in the form of

$$A^* x = f^* \quad (19)$$

Referring to eq(19) the new coefficient matrix is obtained by

$$A^* = PAP^T \quad (20)$$

Then the right-hand side functional vector is

$$f^* = Pf \quad (21)$$

And lastly the approximate solutions are calculated using

$$U = P^T x \quad (22)$$

Based on the transformation that we use in eq(20)-(22) the matrix P is defined as a preconditioning matrix that is,

$$P = I + S \quad (23)$$

Where,

$$S = \begin{bmatrix} 0 \; -r_1' \; 0\;0\;0\;0 \\ 0\;0 \; -r_2' \; 0\;0\;0 \\ 0\;0\;0 \; -r_3' \; 0\;0 \\ 0\;0\;0\;0\;0 \\ 0\;0\;0\;0\;0 \; -r_{m-1}' \\ 0\;0\;0\;0\;0\;0 \end{bmatrix}_{(m-1)\times m-1)}$$

And the matrix I is an identity matrix.
Next, we let the coefficient matrix A^* in eq (19) be given in the form of a sum as follows:

$$A^* = D - L - V \tag{24}$$

Based on the sum of their matrices in eq (24) we represent D, L, & V as the diagonal, the lower and the upper triangular matrices, respectively. Hence using the preconditioned system in eq(19) and matrices in eq(24) the proposed iterative method for solving TFDE, QSPAOR can be generally formulated as

$$x^{(k+1)} = (D - \omega L)^{-1}[\beta V + (\beta - \omega)D + (1-\beta)D]x^k + \beta(D - \omega L)^{-1}f^* \tag{25}$$

Where $x^{(k+1)}$ denotes the vector to be determined at the (k+1) th iteration.
The operation of the QSPAOR method is executed as in Algorithm 1.

Algorithm 1 (QSPAOR method)
i) Initialize $U \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$.
ii) For $j = 4, 8, \ldots n-4$ and $i = 4, 8, \ldots, m-4$ calculate
iii) $x^{(k+1)} = (D - \omega L)^{-1}[\beta V + (\beta - \omega)D + (1-\beta)D]x^k + \beta(D - \omega L)^{-1}f^*$ and then $U^{k+1} + P^T x^{(k+1)}$.
iv) Convergence criterion $\|U^{k+1} - U^k\| \leq \varepsilon$. If the process converged, go to step (iv) otherwise, repeat step (1).
v) Display approximate solutions

Convergence of AOR method
As the QSPAOR iterative method has been formulated, in this section, we discuss the convergence of AOR method that we implement for the solution process to solve eq(1) therefore let us consider the AOR method [32].

$$x^{(k+1)} = (D - \omega L)^{-1}[\beta V + (\beta - \omega)D + (1-\beta)D]x^k + \beta(D - \omega L)^{-1}f^* \tag{26}$$

With n=0, 1, 2, ...

Where

$$L_{\omega,\beta} = (D - \omega L)^{-1}[\beta V + (\beta - \omega)D + (1-\beta)D] = D - \beta(D - \omega L)^{-1}A \tag{27}$$

Theorem If the AOR method (16) converges $(\rho(L_{\omega,\beta}) < 1)$ for some $\beta, \omega \neq 0$, then exactly one of the following statements holds:
i) $\omega \in (0,2) and\ \beta \in (-\infty, 0) \cup (0, +\infty)$
ii) $\omega \in\in (-\infty, 0) \cup (2, +\infty) and \beta \in \left(\frac{2\omega}{2-\omega}, 0\right) \cup (0,2)$.

Proof: It is known that the eigenvalues $\lambda_i\ of\ L_{\omega,\beta}(\beta, \omega \neq 0)$ are connected with the eigenvalues $\xi_i\ of\ L_{\omega,\omega} \equiv L_\omega (L_\omega\ is\ the\ SOR\ iteration\ matrix) by\ the\ relationship.$

$$\lambda_j = \left(1 - \frac{\beta}{\omega}\right) + \frac{\beta}{\omega}\xi_j, j = 2(2)m-2 \tag{28}$$

From eq(28) we get $\xi_j = 1 - \frac{\omega}{\beta} + \frac{\omega}{\beta}\lambda_j, j = 2(2)m-2.$ We also note that $\prod_{j=2,4,..}^{m-2} \xi_j = (1-\omega)^n$. Therefore $\prod_{j=2,4,..}^{m-2} \left(1 - \frac{\omega}{\beta} + \frac{\omega}{\beta}\lambda_j\right) - (1-\omega)^n$ and since $|\lambda_j| < 1, j = 2(2)m-2$ from hypothesis we obtain

$$|(1-\omega)^n| = \prod_{j=2,4,..}^{m-2} \left|1 - \frac{\omega}{\beta} + \frac{\omega}{\beta}\lambda_j\right| \leq \prod_{j=2,4,..}^{m-2} \left(\left|1 - \frac{\omega}{\beta}\right| + \left|\frac{\omega}{\beta}|\lambda_j|\right|\right)$$

$$< \prod_{j=2,4,..}^{m-2} \left(\left|1 - \frac{\omega}{\beta}\right| + \left|\frac{\omega}{\beta}\right|\right) = \left(\left|1 - \frac{\omega}{\beta}\right| + \left|\frac{\omega}{\beta}\right|\right)^n \tag{29}$$

That is,

$|1-\omega| < \left|1-\frac{\omega}{\beta}\left|\frac{\omega}{\beta}\right|\right|$

Or equivalently,

$$|\beta(1-\omega)| < |\beta-\omega| + |\omega| \quad (31)$$

It can be shown that eq(31) holds if and only if exactly one of the following statements holds:
i) $\omega \in (0,2) and\ \beta \in (-\infty,0) \cup (0,+\infty)$
ii) $\omega \in\in (-\infty,0) \cup (2,+\infty) and \beta \in \left(\frac{2\omega}{2-\omega},0\right) \cup (0,2)$

And the proof is complete.

Theorem If the AOR method with $\omega = 0$ converges $(\rho(L_{0,\beta}) < 1)$ then $0 < \beta < 2$.

Proof: If $\omega = 0$ then $L_{0,\beta} = (1-\beta)D + \beta(L+U) = (1-\beta)D + \beta B\ if\ \mu_j\ j = 2(2)m-2$ are the eigenvalues of B, then for the eigenvalues λ_j of $L_{0,\beta}$ we get.

$$\lambda_j = 1-\beta+\beta\mu_j, j = 2(2)m-2 \quad (32)$$

Which implies

$$\mu_j = \frac{1}{\beta}(\beta-1+\lambda_j), j = 2(2)m-2 \quad (33)$$

But since B = 0 we get

$$\sum_{j-2,4,\dots}^{m-2} \mu_j = 0 = \sum_{j-2,4,\dots}^{m-2} (\beta-1+\lambda). \quad (34)$$

From eq(34) we have

$$\sum_{j-2,4,\dots}^{m-2} \lambda_j = \left(\frac{m}{2}-1\right).(1-\beta), \quad (35)$$

And consequently,

$$\left|\left(\frac{m}{2}-1\right)(1-\beta)\right| = \left|\sum_{j-2,4,\dots}^{m-2} \lambda_j\right| \le \sum_{j-2,4,\dots}^{m-2} |\lambda_j < n| \quad (36)$$

Since

$|\lambda_j| < 1, j = 2(2)m-2$ from the hypothesis, $\left|\left(\frac{m}{2}-1\right)(1-\beta)\right| < n, or\ 0 < \beta < 2.$

Examples
For the numerical simulation, we consider two examples of the TFDE problems to evaluate the efficiency of the proposed QSPAOR against the previously developed iterative methods in our research, namely HSPAOR and FSPAOR. The three criteria, as mentioned in Sect. 2, are compared for each of the three different values of a, i.e., α = 0.25, α = 0.50, and α = 0.75. The iteration cycle for the running program based on Algorithm 1 is limited by the tolerance $\varepsilon = 10^{-10}$ We consider the following two TFDE examples, namely the time-fractional initial boundary value problems from [6]:

Example 1

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{\partial^2 u(x,t)}{\partial x^2}, 0 < \alpha \le 1, 0 \le x \le y, t > 0. \quad (37)$$

The boundary conditions that we implement are stated in fractional terms as follows:

$$u(0,t) = \frac{2kt^\alpha}{\Gamma(\alpha+1)}, u(l,t) = l^2 + \frac{2kt^\alpha}{\Gamma(\alpha+1)} \quad (38)$$

And to initiate the approximate solutions we set the initial condition

$$u(x,0) = x^2 \quad (39)$$

Example 2

$$\frac{\partial u(x,t)}{\partial t} = \Gamma(1.2)x^{\beta}\frac{\partial^{\beta}u(x,t)}{\partial x^{\beta}} + 3x^2(2x-1)e^{-t} \quad (40)$$

Table 1: Numerical results for Example 1

M	Method	$\alpha = 0.25$ K	Seconds	MAE	$\alpha = 0.50$ K	Seconds	MAE	$\alpha = 0.75$ K	Seconds	MAE
128	FSPAOR	1351	5.80	9.97e-05	694	0.92	9.86e-05	318	0.37	1.30e-04
	HSPAOR	409	2.47	9.97e-05	253	0.46	9.85e-05	93	0.12	1.30e-04
	QSPAOR	200	0.92	9.96e-05	124	0,24	9.84e-05	44	0.07	1.29e-04
256	FSPAOR	4192	24.95	9.97e-05	2694	7.52	9.90e-05	1307	2.92	1.30e-04
	HSPAOR	1605	11.00	9.97e-05	1027	4.47	9.89e-05	492	1.46	1.29e-04
	QSPAOR	784	4.19	9.95e-05	502	2.10	9.88e-05	241	0.87	1.28e-04
512	FSPAOR	15608	236.25	9.9e-05	10,085	62.59	9.90e-05	4947	29.56	1.32e-04
	HSPAOR	6029	114.76	9.97e-05	3887	32.61	9.90e-05	1900	15.05	1.31e-04
	QSPAOR	2950	43.02	9.95e-05	1540	15.21	9.89e-05	928	7.21	1.30e-04
1024	FSPAOR	54130	337.36	9.99e-05	33652	432.78	9.90e-05	16609	215.41	1.40e-04
	HSPAOR	21478	1613.83	9.97e-05	13387	212.95	9.88e-05	6498	113.40	1.40e-04
	QSPAOR	10640	898.67	9.95e-05	5531	103.96	9.87e-05	3479	53.67	1.39e-04
2048	FSPAOR	196523	14378.36	9.99e-05	121947	3026.56	9.90e-05	59500	1211.32	1.71e-04
	HSPAOR	77153	7189.71	9.97e-05	47933	1349.79	9.90e-05	23344	694.40	1.71e-04
	QSPAOR	38471	3078.90	9.96e-05	19711	601.76	9.88e-05	11740	321.85	1.70e-04

Table 2 Numerical results for example 2

M	Method	$\alpha = 0.25$ K	Seconds	MAE	$\alpha = 0.50$ K	Seconds	MAE	$\alpha = 0.75$ K	Seconds	MAE
128	FSPAOR	406	3.32	1.95e-02	153	2.27	8.29e-05	142	1.62	1.37e-01
	HSPAOR	136	1.48	1.94e-02	77	1.30	8.30e-05	71	0.67	1.36e-01
	QSPAOR	49	0.72	1.94e-02	34	0.64	8.29e-05	19	0.33	1.35e-01
256	FSPAOR	1270	14.75	1.95e-02	591	8.21	8.29e-05	236	4.28	1.37e-01
	HSPAOR	618	7.21	1.94e-02	287	4.33	8.30e-05	111	2.33	1.36e-01
	QSPAOR	270	3.35	1.94e-02	141	2.03	8.29e-05	81	1.96	1.35e-01
512	FSPAOR	4841	91.72	1.95e-02	2330	53.97	8.29e-05	1064	31.84	1.37e-01
	HSPAOR	2365	44.07	1.95e-02	1139	23.24	8.30e-05	519	12.77	1.36e-01
	QSPAOR	1044	21.10	1.94e-02	592	11.87	8.29e-05	324	5.25	1.35e-01
1024	FSPAOR	16373	152.97	1.94e-02	8471	428.76	8.29e-05	4029	323.97	1.37e-01
	HSPAOR	8816	61.07	1.94e-02	4273	21334	8.30e-05	1987	148.63	1.36e-01
	QSPAOR	3908	29.58	1.94e-02	1895	106.90	8.29e-05	1219	51.76	1.35e-01
2048	FSPAOR	59608	853.87	1.95e-02	31048	1121.34	8.29e-05	14899	614.63	1.37e-01
	HSPAOR	29771	426.83	1.95e-02	15340	511.24	8.30e-05	7344	253.97	1.36e-01
	QSPAOR	13203	209.50	1.94e-02	6852	251.99	8.29e-05	4497	123.18	1.35e-01

For the example of Eq. (40), we initiate the approximate solutions using the initial condition $U(x,0) = x^2 - x^3$and implement the zero Dirichlet conditions. Meanwhile, the exact solution to this problem is $U(x,t) = x^2(1-x)e^{-t}$.
All-important numerical results from the implementation of QSPAOR, HSPAOR, and FSPAOR methods to solve the numerical examples in Eqs.(37) and (40) are recorded in Tables 1 and 2. For the consistency inspection, we run the numerical simulation by increasing the values of mesh sizes, that is, m = 128,256,512, 1024, and 2048. Based on the results tabulated in Tables 1 and 2, we found that QSPAOR required the least number of iterations and the shortest computation time to finish computing the two examples' solutions compared to the HSPAOR and FSPAOR. The numerical results are similar for all values of mesh sizes and parameter α. These results attribute the significant improvement in computing efficiency to the quarter-sweep technique, which computes a quarter of the total number of mesh points using PAOR instead of all mesh points in the solution domain.
The results can be summarized as follows. For Example, l the number of iterations and computation time have declined by 80.25-85.36% and 71.89-81.78%, respectively, if QSPAOR method is compared to the FSPAOR method. When QSPAOR is compared to HSPAOR, the number of iterations and the computation time have reduced by about 49.20-54.74% and 44.18-57.27%, respectively. For Example 2, QSPAOR has reduced the number of iterations and the computation time of FSPAOR by about 72.13-84.16% and 67.16-80.90% respectively. When compared to HSPAOR, these improvements became 43.57-64.35% and 38.57-57.28%, respectively. Overall, the accuracy of the three numerical methods, i.e., QSPAOR, HSPAOR, and FSPAOR, is comparable.

Concluding Remark

This paper solved a one-dimensional TFDE by applying the quarter-sweep finite-difference scheme and the PAOR iterative method. Using the quarter-sweep technique and PAOR iterative method, the computational complexity of computing the solutions of the TFDE has been successfully reduced. The quarter-sweep calculated only a quarter of the total mesh points by using PAOR while averaging the remaining mesh points, and the result is promising. The numerical experiments demonstrated the efficiency of the pro- posed QSPAOR method, in which the number of iterations and computation time have been reduced significantly, compared to the HSPAOR and FSPAOR methods. In addition to that, the accuracy of the three tested methods is almost identical. The study found that the computational complexity reduction by the quarter-sweep and the PAOR method can be an efficient numerical method to solve TFDE.

References

1. Akram T, Abbas M, Ali A, Iqbal A, Baleanu D. A numerical approach of a time fractional reaction-diffusion model with a non-singular kernel. Symmetry. 2020;12(10):16-53.
2. Ali KK, Osman MS, Baskonus HM, Elazab NS, Ilhan E. Analytical and numerical study of the HIV-1 infection of CD4+ T-cells conformable fractional mathematical model that causes acquired immunodeficiency syndrome with the effect of antiviral drug therapy. Mathematical Methods in the Applied Sciences. 2020.
3. Aruchunan E, Muthuvalu MS, Sulaiman J. 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. 2015;44(1):139-146.
4. Cen Z, Huang J, Le A, Xu A. A second-order scheme for a time-fractional diffusion equation. Applied Mathematics Letters. 2019;90:79-85.
5. Demir A, Erman S, Özgür B, Korkmaz E. Analysis of fractional partial differential equations by Taylor series expansion. Boundary Value Problems. 2013;2013(1):68.
6. Duo S, Ju L, Zhang Y. A fast algorithm for solving the space-time fractional diffusion equation. Computers and Mathematics with Applications. 2018;75(6):1929-1941.
7. Fujita Y. Cauchy problems of fractional order and stable processes. Japanese Journal of Applied Mathematics. 1990;7(3):459-476.
8. Gao W, Veeresha P, Prakasha DG, Baskonus HM. New numerical simulation for fractional Benney-Lin equation arising in falling film problems using two novel techniques. Numerical Methods for Partial Differential Equations. 2021;37(1):210-243.
9. Gao W, Veeresha P, Prakasha DG, Baskonus HM, Yel G. New numerical results for the time-fractional Phi-four equation using a novel analytical approach. Symmetry. 2020;12(3):478.
10. Gao W, Yel G, Baskonus HM, Cattani C. Complex solitons in the conformable (2+1)-dimensional Ablowitz-Kaup-Newell-Segur equation. AIMS Mathematics. 2020;5(1):507-521.
11. Garrappa R. Numerical solution of fractional differential equations: a survey and a software tutorial. Mathematics. 2018;6(2):16.
12. Gong C, Bao W, Tang G, Jiang Y, Liu X. Computational challenge of fractional differential equations and the potential solutions: a survey. Mathematical Problems in Engineering. 2015;2015:258-265.
13. González-Olvera MA, Torres L, Hernández-Fontes JV, Mendoza E. Time fractional diffusion equation for shipping water events simulation. Chaos, Solitons and Fractals. 2021;143:110538.
14. Gunawardena AD, Jain SK, Snyder L. Modified iterative methods for consistent linear systems. Linear Algebra and Its Applications. 1991;154-156:123-143.
15. Hacksbusch W. Iterative solution of large sparse systems of equations. Applied Mathematical Sciences. 1994.
16. Hilfer R. Foundations of fractional dynamics. Fractals. 1995;3(3):549-556.
17. Hilfer R. Fractional diffusion based on Riemann-Liouville fractional derivatives. Journal of Physical Chemistry B. 2000;104(16):3914-3917.
18. Jaradat I, Alquran M, Momani S, Baleanu D. Numerical schemes for studying biomathematics model inherited with memory-time and delay-time. Alexandria Engineering Journal. 2020;59(5):2969-2974.
19. Kumar S, Shaw PK, Abdel-Aty A, Mahmoud EE. A numerical study on fractional differential equation with population growth model. Numerical Methods for Partial Differential Equations. 2020.
20. Langlands TAM, Henry BI. The accuracy and stability of an implicit solution method for the fractional diffusion equation. Journal of Computational Physics. 2005;205(2):719-736.
21. Othman M, Abdullah AR. An efficient four points modified explicit group Poisson solver. International Journal of Computer Mathematics. 2000;76(2):203-217.
22. Saad Y. Iterative Methods for Sparse Linear Systems. Philadelphia: SIAM; 2003.
23. Schultz MH. A generalization of the Lax equivalence theorem. Proceedings of the American Mathematical Society. 1966;17(5):1034-1035.
24. Sun H, Cao W. A fast temporal second-order difference scheme for the time-fractional subdiffusion equation. Numerical Methods for Partial Differential Equations. 2020.
25. Sunarto A, Sulaiman J. Computational algorithm PAOR for time-fractional diffusion equations. IOP Conference Series: Materials Science and Engineering. 2020;874:012029.
26. Veeresha P, Baskonus HM, Prakasha DG, Gao W, Yel G. Regarding new numerical solution of fractional Schistosomiasis disease arising in biological phenomena. Chaos, Solitons and Fractals. 2020;133:109661.
27. Vui Lung J, Sulaiman J. On quarter-sweep finite difference scheme for one-dimensional porous medium equations. International Journal of Applied Mathematics. 2020;33(3):439-450.
28. Xu Q, Xu Y. Extremely low order time-fractional differential equation and application in combustion process. Communications in Nonlinear Science and Numerical Simulation. 2018;64:135-148.
29. Yang XJ, Gao F. A new technology for solving diffusion and heat equations. Thermal Science. 2017;21(1 Part A):133-140.
30. Yang XJ, Ragulskis M, Taha T. A new general fractional-order derivative with Rabotnov fractional-exponential kernel. Thermal Science. 2019;23(6 Part B):3711-3718.
31. Yeyios AK. A necessary condition for the convergence of the accelerated overrelaxation (AOR) method. Journal of Computational and Applied Mathematics. 1989;26(3):371-373.
32. Zwillinger D. Handbook of Differential Equations. 3rd ed. San Diego: Academic Press; c1997.