Asia Pac. J. Math. 2026 **13**:100

HALF-SWEEP GAUSS-SEIDEL ITERATIVE METHOD FOR SOLVING TWO-DIMENSIONAL TIME-FRACTIONAL DIFFUSION EQUATIONS

ANDANG SUNARTO[1,*], JACKEL VUI LUNG CHEW[2,*], JUMAT SULAIMAN[3]

[1]Departemen Matematika, UIN Fatmawati Sukarno, 38211, Bengkulu, Indonesia

[2]Faculty of Computing and Informatics, Universiti Malaysia Sabah Labuan International Campus, 87000 Sg. Pagar, Labuan, Malaysia

[3]Faculty of Science and Technology, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

[*]Corresponding authors: andangs@mail.uinfasbengkulu.ac.id, jackelchew93@ums.edu.my

Received Apr. 27, 2026

Abstract. This study presents an efficient iterative method for solving two-dimensional time-fractional diffusion equations governed by the Caputo fractional derivative. The discretization process is carried out using the implicit finite difference method with the L1 approximation. The outcome of the discretization process leads to large and sparse penta-diagonal matrices which need to be solved at each time level. To efficiently solve these systems, a half-sweep Gauss-Seidel iterative method is developed. Numerical experiments are conducted on two benchmark test problems with known solutions, across a range of grid resolutions and fractional orders. The results show that the half-sweep Gauss-Seidel iterative method consistently outperforms the conventional Gauss-Seidel iterative method in terms of number of iterations and computation time, while preserving comparable numerical accuracy. These findings show the half-sweep Gauss-Seidel method as an effective iterative method for two-dimensional time-fractional diffusion problems arising in scientific and engineering applications.

2020 Mathematics Subject Classification. 65M06; 65M22.

Key words and phrases. time-fractional diffusion equation; Caputo fractional derivative; half-sweep Gauss-Seidel; implicit finite difference method; linear system.

1. Introduction

The time-fractional diffusion equation (TFDE) is a generalization of the classical diffusion equation, where fractional derivatives are used to model diffusion processes with long-term memory effects. This equation is particularly useful in describing anomalous diffusion, where the rate of diffusion deviates from the classical Brownian motion model. Due its effectiveness in modeling, TDPE has been adapted in various applications such as image denoising [1,2], shipping water events simulation [3], signal

DOI: 10.28924/APJM/13-100

smoothing [4], and drug eluting stent [5]. Fractional derivatives in TFDE were evident to be able to produce a mathematically rigorous framework in accurately modeling complex diffusion dynamics.

Among the various fractional derivatives, the Caputo fractional derivative is a common choice to approximate the time fractional derivative in TFDE. It is preferred due to its ability to handle initial conditions [6]. The Caputo fractional derivative is also known for its consistency in differentiating constant functions, which helps in the numerical treatment of differential equations [7]. Then, various numerical methods have been used to solve TFDE involving the Caputo fractional derivative, such as finite difference method [2,8–11], finite element method [12,13], and pseudo spectral method [14]. Among these methods, implicit finite difference method with the L1 approximation for the Caputo fractional derivative have been extensively studied due to their simplicity, unconditional stability, and reliable accuracy [15–18].

When implicit finite difference method with the L1 approximation is applied to two-dimensional (2D) TFDE problems, such schemes produce large sparse penta-diagonal linear systems at each time step, reflecting the coupling between grid points and the nonlocal temporal history. The efficient solution of these linear systems constitutes the primary computational challenge in fully implicit 2DTFDE models. Classical iterative solvers, such as the Gauss-Seidel method, are frequently employed because of their straightforward implementation and low memory requirements [17,18]. However, in 2D settings with fine spatial discretization, the need to update all grid points during each iteration leads to substantial computational cost, making the conventional Gauss-Seidel method increasingly inefficient for large-scale problems.

To overcome this computational limitation, the half-sweep iterative strategy has been introduced. By exploiting the structure and sparsity pattern of the coefficient matrix, half-sweep strategies significantly reduce arithmetic operations and memory access while preserving convergence properties [16–20]. Despite their demonstrated effectiveness in reducing computational effort, the application of half-sweep techniques to 2DTFDE remains relatively underexplored.

Therefore, in this study, a half-sweep Gauss-Seidel (HSGS) iterative method is developed for the numerical solution of 2DTFDE involving the Caputo fractional derivative. This paper is organized as follows. In Section 2, the proposed iterative method is formulated in conjunction with an implicit half-sweep finite difference method with the L1 temporal approximation. Then, in Section 3, numerical results from experiments using benchmark test problems are discussed to assess the performance of the proposed HSGS iterative method compared the conventional Gauss-Seidel iterative method. Section 4 concludes this study with significant findings and future work.

2. Half-Sweep Finite Difference Approximation for 2DTFDE

In this study, we consider the following 2DTFDE of the form [21],

$$\frac{\partial^\alpha u}{\partial t^\alpha} = a(x,y,t)\frac{\partial^2 u}{\partial x^2} + b(x,y,t)\frac{\partial^2 u}{\partial y^2} + s(x,y,t), \tag{1}$$

where $u = u(x,y,t) > 0$, $a = a(x,y,t) > 0$, $b = b(x,y,t) > 0$, and $s(x,y,t)$ is the source function.

We supplement Equation (1) with Dirichlet boundary conditions and a prescribed initial condition,

$$u(x,y,0) = u_0(x,y), \quad (x,y) \in \Omega, \tag{2}$$

where $\Omega = \{(x,y)\,|\, 0 \le x \le L,\ 0 \le y \le M\}$. Here, $u_0(x,y)$ is assumed to be sufficiently smooth to ensure the well-posedness of the continuous problem.

To solve Equation (1), we approximate the time-fractional derivative using the Caputo fractional derivative of order α [22],

$$\frac{\partial^\alpha u}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)}\int_0^t \frac{\partial u(x,y,\phi)}{\partial \phi}(t-\phi)^{-\alpha}\, d\phi, \quad 0 < \alpha < 1, \tag{3}$$

with the discrete approximation given by

$$\frac{\partial^\alpha U_{p,q}^n}{\partial t^\alpha} \approx \tau \sum_{r=1}^{n} \sigma(\alpha, r)\left(U_{p,q}^{n-r+1} - U_{p,q}^{n-r}\right), \tag{4}$$

where the approximate solution $U_{p,q}^n = U(x_p, y_q, t_n)$ corresponds to the exact solution $u(x_p, y_q, t_n)$, with $p, q, n \in \mathbb{Z}^+$ such that $1 \le p, q \le M-1$ and $1 \le n \le N$.

The coefficient τ is defined as

$$\tau = \frac{1}{\Gamma(2-\alpha)\, k^\alpha}, \quad k = \Delta t, \tag{5}$$

and

$$\sigma(\alpha, r) = r^{1-\alpha} - (r-1)^{1-\alpha}. \tag{6}$$

Then, to derive computational schemes for spatial derivatives, we consider the Taylor series expansion as follows. Using Taylor series expansion, we obtain

$$U_{p+1,q}^n = U_{p,q}^n + h\frac{\partial U_{p,q}^n}{\partial x} + \frac{h^2}{2}\frac{\partial^2 U_{p,q}^n}{\partial x^2} + \cdots, \tag{7}$$

$$U_{p-1,q}^n = U_{p,q}^n - h\frac{\partial U_{p,q}^n}{\partial x} + \frac{h^2}{2}\frac{\partial^2 U_{p,q}^n}{\partial x^2} + \cdots, \tag{8}$$

$$U_{p,q+1}^n = U_{p,q}^n + h\frac{\partial U_{p,q}^n}{\partial y} + \frac{h^2}{2}\frac{\partial^2 U_{p,q}^n}{\partial y^2} + \cdots, \tag{9}$$

and

$$U_{p,q-1}^n = U_{p,q}^n - h\frac{\partial U_{p,q}^n}{\partial y} + \frac{h^2}{2}\frac{\partial^2 U_{p,q}^n}{\partial y^2} + \cdots. \tag{10}$$

The sum of Equations (7) and (8) yields the conventional second-order central difference approximation in the x-direction,

$$\frac{\partial^2 U^n_{p,q}}{\partial x^2} \approx \frac{U^n_{p+1,q} - 2U^n_{p,q} + U^n_{p-1,q}}{h^2} + O(h^2). \tag{11}$$

Similarly, the sum of Equations (9) and (10) gives the second-order central difference approximation in the y-direction,

$$\frac{\partial^2 U^n_{p,q}}{\partial y^2} \approx \frac{U^n_{p,q+1} - 2U^n_{p,q} + U^n_{p,q-1}}{h^2} + O(h^2). \tag{12}$$

To construct the half-sweep computational scheme, diagonal Taylor series expansions are utilized. These expansions exploit grid points along the diagonal directions to approximate second-order derivatives.

First, to approximate the derivative with respect to x, we consider the diagonal points $(p+1, q+1)$ and $(p-1, q-1)$. Expanding $U^n_{p+1,q+1}$ about (p, q) yields

$$\begin{aligned} U^n_{p+1,q+1} = U^n_{p,q} &+ h\left(\frac{\partial U^n_{p,q}}{\partial x} + \frac{\partial U^n_{p,q}}{\partial y}\right) \\ &+ \frac{h^2}{2}\left(\frac{\partial^2 U^n_{p,q}}{\partial x^2} + \frac{\partial^2 U^n_{p,q}}{\partial y^2}\right) + h^2\frac{\partial^2 U^n_{p,q}}{\partial x \partial y} + \cdots, \end{aligned} \tag{13}$$

while the expansion at $(p-1, q-1)$ is given by

$$\begin{aligned} U^n_{p-1,q-1} = U^n_{p,q} &- h\left(\frac{\partial U^n_{p,q}}{\partial x} + \frac{\partial U^n_{p,q}}{\partial y}\right) \\ &+ \frac{h^2}{2}\left(\frac{\partial^2 U^n_{p,q}}{\partial x^2} + \frac{\partial^2 U^n_{p,q}}{\partial y^2}\right) + h^2\frac{\partial^2 U^n_{p,q}}{\partial x \partial y} + \cdots. \end{aligned} \tag{14}$$

By adding Equations (13) and (14), the first-order terms are eliminated, yielding a symmetric expression. Rearranging gives the following half-sweep approximation,

$$\frac{\partial^2 U^n_{p,q}}{\partial x^2} \approx \frac{U^n_{p+1,q+1} - 2U^n_{p,q} + U^n_{p-1,q-1}}{2h^2} + O(h^2). \tag{15}$$

It is important to note that this approximation is constructed along the diagonal direction $(1, 1)$, and therefore incorporates variations in both x and y.

Using a similar procedure, we derive the half-sweep scheme associated with the opposite diagonal direction. Specifically, we consider the points $(p-1, q+1)$ and $(p+1, q-1)$. The corresponding Taylor expansions are given by

$$\begin{aligned} U^n_{p-1,q+1} = U^n_{p,q} &- h\frac{\partial U^n_{p,q}}{\partial x} + h\frac{\partial U^n_{p,q}}{\partial y} \\ &+ \frac{h^2}{2}\left(\frac{\partial^2 U^n_{p,q}}{\partial x^2} + \frac{\partial^2 U^n_{p,q}}{\partial y^2}\right) - h^2\frac{\partial^2 U^n_{p,q}}{\partial x \partial y} + \cdots, \end{aligned} \tag{16}$$

and

$$U^n_{p+1,q-1} = U^n_{p,q} + h\frac{\partial U^n_{p,q}}{\partial x} - h\frac{\partial U^n_{p,q}}{\partial y} + \frac{h^2}{2}\left(\frac{\partial^2 U^n_{p,q}}{\partial x^2} + \frac{\partial^2 U^n_{p,q}}{\partial y^2}\right) - h^2\frac{\partial^2 U^n_{p,q}}{\partial x \partial y} + \cdots . \tag{17}$$

By combining Equations (16) and (17), the first-order terms cancel, leading to

$$\frac{\partial^2 U^n_{p,q}}{\partial y^2} \approx \frac{U^n_{p-1,q+1} - 2U^n_{p,q} + U^n_{p+1,q-1}}{2h^2} + O(h^2). \tag{18}$$

However, it should be emphasized that this approximation is derived along the diagonal direction $(-1, 1)$. Consequently, Equation (18) does not strictly represent the pure Cartesian second-order derivative with respect to y, but rather a second-order directional derivative that includes contributions from $\partial^2 U^n_{p,q}/\partial x^2$, $\partial^2 U^n_{p,q}/\partial y^2$, and the mixed derivative $\partial^2 U^n_{p,q}/\partial x \partial y$.

Now, let us define the solution space with M grid points and N time levels. The distribution of grid points is equidistant with

$$x_p = ph, \quad y_q = qh, \quad h = \frac{1}{M},$$

and the time discretization is uniform with

$$t_n = nk, \quad k = \frac{1}{N}.$$

Substituting Equations (4), (15), and (18) into Equation (1), we obtain the half-sweep finite difference approximation for the 2DTFDE as follows,

$$\tau \sum_{r=1}^{n} \sigma(\alpha, r)\left(U^{n-r+1}_{p,q} - U^{n-r}_{p,q}\right) = a^*\left(U^n_{p+1,q+1} - 2U^n_{p,q} + U^n_{p-1,q-1}\right) + b^*\left(U^n_{p-1,q+1} - 2U^n_{p,q} + U^n_{p+1,q-1}\right) + s^n_{p,q}, \tag{19}$$

where $1 \leq p, q \leq M-1$, $0 \leq n \leq N$, and

$$a^* = \frac{a}{2h^2}, \quad b^* = \frac{b}{2h^2}. \tag{20}$$

The computation at each grid point leads to a system of linear equations of the form,

$$\mathbf{F} = \begin{bmatrix} F^n_{1,1} \\ F^n_{2,1} \\ \vdots \\ F^n_{1,2} \\ F^n_{2,2} \\ \vdots \\ F^n_{M-1,M-1} \end{bmatrix} = \mathbf{0}, \tag{21}$$

where each equation can be generalized as

$$\begin{aligned} F_{p,q}^{n} &= (\tau + 2a^* + 2b^*)U_{p,q}^{n} - a^*U_{p+1,q+1}^{n} - a^*U_{p-1,q-1}^{n} \\ &- b^*U_{p-1,q+1}^{n} - b^*U_{p+1,q-1}^{n} \\ &- s_{p,q}^{n} - G_{p,q}^{n-1} = 0, \end{aligned} \tag{22}$$

with $G_{p,q}^{n-1}$ defined as

$$G_{p,q}^{n-1} = \tau \sum_{r=2}^{n} \sigma(\alpha, r) \left(U_{p,q}^{n-r+1} - U_{p,q}^{n-r} \right). \tag{23}$$

To solve the linear system presented in Equation (21) using an iterative method, we rewrite it in matrix form as

$$A\mathbf{U}^n = \mathbf{b}^n, \tag{24}$$

where

$$\mathbf{b}^n = \mathbf{s}^n + \mathbf{G}^{\,n-1}. \tag{25}$$

Based on Equation (24), $\mathbf{U}^n$ is the vector of unknowns at time level n, A is the penta-diagonal coefficient matrix, and $\mathbf{b}^n$ is the right-hand side vector at time level n. In this matrix formulation, $\mathbf{b}^n$ aggregates contributions from the source function and the history term arising from the nonlocal nature of Equation (4). Solving Equation (24) efficiently therefore becomes the primary computational challenge. To address this, we propose the HSGS iterative method.

By decomposing the matrix A into five diagonal components, we write

$$A = D - K - L - V - W, \tag{26}$$

where D is the main diagonal matrix, K and L represent the backward (lower) diagonal components, V and W represent the forward (upper) diagonal components corresponding to diagonal stencil interactions.

Accordingly, the HSGS iterative scheme is derived as

$$\mathbf{U}^{(l+1)} = (D - K - L)^{-1} \Big[(V + W)\mathbf{U}^{(l)} - \mathbf{F} \Big], \tag{27}$$

where l denotes the iteration index and $\mathbf{F}$ is the vector form of the residual terms defined in Equation (22).

To facilitate numerical experimentation using the proposed HSGS iterative method for solving the 2DTFDE, the algorithmic procedure is presented in the following section.

Algorithm 1 HSGS Iterative Method for Solving 2DTFDE

Require: Problem parameters a, b, discretization parameters h, k, tolerance $\varepsilon_{\max}$

Ensure: Numerical solution $\mathbf{U}$, number of iterations $l_{\max}$, execution time (s), and maximum absolute error (MAE)

Initialize 2D spatial grid using h
Initialize approximation functions using a, b
Initialize solution variables $\mathbf{U}$
Set initial condition and apply boundary conditions
Compute matrix coefficients and fractional weights $\sigma(\alpha, r)$
for $n = 1, 2, \ldots, N$ **do**
    Initialize iterative solution $\mathbf{U}^{(0)}$ at time level n
    $l \leftarrow 0$
    while convergence criterion not satisfied **do**
        Update half-sweep grid points using Equation (22)
        Update remaining grid points using standard implicit approximation
        Compute difference $\|\mathbf{U}^{(l+1)} - \mathbf{U}^{(l)}\|$
        if $\|\mathbf{U}^{(l+1)} - \mathbf{U}^{(l)}\| < \varepsilon_{\max}$ **then**
            break
        else
            $l \leftarrow l + 1$
        end if
    end while
    Set $l_{\max} \leftarrow l$
    Compute fractional memory term $\mathbf{G}^{n-1}$
    Update right-hand side vector $\mathbf{b}^{n+1}$
    Set $\mathbf{U}^{(0)} \leftarrow \mathbf{U}^{n}$ for next time level
    Apply boundary conditions
    if exact solution is available **then**
        Compute maximum absolute error (MAE)
    end if
end for
return $\mathbf{U}$, $l_{\max}$, execution time, MAE

3. Results and Discussion

In this section, two benchmark test problems are presented to assess the stability, convergence behavior, accuracy, effect of fractional orders, and computational efficiency of the proposed HSGS method for solving the 2DTFDE. The performance of the HSGS method is compared with that of the conventional Gauss–Seidel (GS) method.

All computations are carried out using the C programming language in Visual Studio Code on a Lenovo IdeaPad Slim 5 laptop. The stopping tolerance for both iterative solvers is fixed at $\varepsilon_{\max} = 10^{-10}$. Uniform spatial grid resolutions with sizes $M \times M = 8 \times 8, 16 \times 16, 32 \times 32, 64 \times 64, 128 \times 128$ are considered. The fractional orders are taken as $\alpha = 0.25, 0.50, 0.75$, and the number of time steps is fixed at $N = 100$.

3.1. **Test Problem 1 [21].** Consider the following 2DTFDE with initial and boundary conditions,

$$\frac{\partial^\alpha u}{\partial t^\alpha} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \left(\Gamma(2+\alpha)t - 2t^{1+\alpha}\right) e^{x+y}, \quad 0 < \alpha < 1, \tag{28}$$

which satisfies the exact solution,

$$u(x,y,t) = e^{x+y} t^{1+\alpha}, \quad 0 < x, y < 1,\ 0 < t < 1. \tag{29}$$

3.2. **Test Problem 2 [21].** Consider another 2DTFDE with a different source term,

$$\frac{\partial^\alpha u}{\partial t^\alpha} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \left(\frac{2}{\Gamma(3-\alpha)} t^{2-\alpha} + 2t^2\right) \sin x \sin y, \quad 0 < \alpha < 1, \tag{30}$$

subject to the initial and boundary conditions,

$$u(x,y,0) = 0, \quad 0 < x, y < 1, \tag{31}$$

$$u(0,y,t) = u(x,0,t) = 0, \quad 0 < t < 1, \tag{32}$$

$$u(1,y,t) = t^2 \sin(1) \sin y, \quad u(x,1,t) = t^2 \sin x \sin(1), \tag{33}$$

which satisfies the exact solution,

$$u(x,y,t) = t^2 \sin x \sin y. \tag{34}$$

The numerical results obtained from the experiments are presented in Tables 1 and 2. These results are analyzed and discussed based on observed accuracy, convergence behavior, and computational performance of the proposed HSGS method in comparison with the classical GS method.

TABLE 1. Results of Solving Test Problem 1

Order α	Size	Method	Iteration	Time (s)	MAE
0.25	8×8	GS	102	0.052	5.174×10^{-4}
		HSGS	54	0.040	2.148×10^{-3}
	16×16	GS	379	0.357	1.349×10^{-4}
		HSGS	198	0.183	5.358×10^{-4}
	32×32	GS	1390	4.012	3.606×10^{-5}
		HSGS	727	2.306	1.357×10^{-4}
	64×64	GS	5064	48.509	1.125×10^{-5}
		HSGS	2655	28.146	3.616×10^{-5}
	128×128	GS	18280	958.483	4.920×10^{-6}
		HSGS	9631	646.507	1.121×10^{-5}
0.50	8×8	GS	80	0.032	5.461×10^{-4}
		HSGS	43	0.030	2.162×10^{-3}
	16×16	GS	292	0.229	1.683×10^{-4}
		HSGS	153	0.174	5.653×10^{-4}
	32×32	GS	1072	2.908	7.054×10^{-5}
		HSGS	560	1.775	1.692×10^{-4}
	64×64	GS	3911	36.079	4.604×10^{-5}
		HSGS	2048	21.149	7.072×10^{-5}
	128×128	GS	14160	729.076	3.978×10^{-5}
		HSGS	7447	443.976	4.602×10^{-5}
0.75	8×8	GS	50	0.038	8.086×10^{-4}
		HSGS	28	0.034	2.412×10^{-3}
	16×16	GS	176	0.152	4.438×10^{-4}
		HSGS	93	0.112	8.375×10^{-4}
	32×32	GS	639	1.779	3.484×10^{-4}
		HSGS	335	1.095	4.462×10^{-4}
	64×64	GS	2333	21.390	3.247×10^{-4}
		HSGS	1220	13.341	3.492×10^{-4}
	128×128	GS	8478	435.292	3.186×10^{-4}
		HSGS	4449	469.248	3.248×10^{-4}

TABLE 2. Results of Solving Test Problem 2

Order α	Size	Method	Iteration	Time (s)	MAE
0.25	8×8	GS	91	0.040	4.586×10^{-5}
		HSGS	48	0.035	1.868×10^{-4}
	16×16	GS	334	0.277	1.260×10^{-5}
		HSGS	175	0.226	4.675×10^{-5}
	32×32	GS	1208	3.820	4.185×10^{-6}
		HSGS	636	2.774	1.266×10^{-5}
	64×64	GS	4329	70.512	2.051×10^{-6}
		HSGS	2288	40.996	4.182×10^{-6}
	128×128	GS	15327	1297.082	1.425×10^{-6}
		HSGS	8154	1011.849	2.018×10^{-6}
0.50	8×8	GS	71	0.046	5.206×10^{-5}
		HSGS	38	0.029	1.916×10^{-4}
	16×16	GS	256	0.234	1.926×10^{-5}
		HSGS	135	0.161	5.307×10^{-5}
	32×32	GS	924	2.928	1.096×10^{-5}
		HSGS	486	1.768	1.935×10^{-5}
	64×64	GS	3314	51.363	8.852×10^{-6}
		HSGS	1750	28.625	1.096×10^{-5}
	128×128	GS	11764	1324.546	8.229×10^{-6}
		HSGS	6249	705.322	8.820×10^{-6}
0.75	8×8	GS	45	0.041	8.325×10^{-5}
		HSGS	25	0.026	2.223×10^{-4}
	16×16	GS	154	0.231	5.114×10^{-5}
		HSGS	82	0.118	8.486×10^{-5}
	32×32	GS	550	2.178	4.301×10^{-5}
		HSGS	290	1.129	5.138×10^{-5}
	64×64	GS	1972	35.120	4.094×10^{-5}
		HSGS	1040	19.953	4.305×10^{-5}
	128×128	GS	7027	677.218	4.033×10^{-5}
		HSGS	3723	413.072	4.092×10^{-5}

In terms of stability of the iteration process, this study found that both the GS and HSGS methods remain stable under all tested conditions, including fine grid resolutions and varying fractional orders. No divergence or oscillatory behavior is observed in the numerical solutions for every time step. As a result, the numerical solutions are successfully computed from the initial condition to the final time level of $t = 1$. Despite the reduced update strategy by the HSGS method, its stability indicates that it preserves the essential numerical properties of the computational scheme.

In terms of convergence behavior of the approximation equation, the total number of iterations reported in Tables 1 and 2 shows that the HSGS method consistently requires fewer iterations than the GS method across all grid sizes and fractional orders. On average, the HSGS method reduces the number of iterations by approximately 45-50%, regardless of the mathematical structure of 2DTFDE problem or source function. This improvement can be attributed to the effective half-sweep updating strategy that reduces redundant computations by updating only a subset of grid nodes per iteration while maintaining good propagation of solution updates. Consequently, the HSGS method exhibits faster convergence behavior compared to the conventional implicit finite difference approximation used by the GS method.

In terms of accuracy, both the GS and HSGS methods exhibit convergence toward the known solution as the grid is refined. The MAE decreases consistently with increasing grid resolution for all fractional orders. This result confirms the numerical consistency of implicit finite difference approximation. However, the GS method produces lower MAE values compared to HSGS, particularly for smaller fractional orders. This discrepancy is primarily due to the reduced number of updates per iteration in the HSGS method that slows the propagation of local error corrections across the computational domain. Despite this, the accuracy gap between the two tested methods diminishes as the grid is refined. At finer grid resolutions, the MAE values of HSGS approach those of GS. Moreover, for higher fractional orders, the difference in accuracy becomes negligible. For instance, in Problem 2 with $\alpha = 0.75$ and grid size 128×128, both methods produce nearly identical MAE values.

In terms of computational efficiency, the reduction in the number of iterations directly translates into improved computational efficiency. Across all test cases, the HSGS method consistently achieves lower execution times than the GS method. For example, in Test Problem 2 with $\alpha = 0.50$ and grid size 128×128, the execution time is reduced from 1324.546 seconds (GS) to 705.322 seconds (HSGS), representing a reduction of approximately 45%. Similar performance gains are observed across other grid sizes and fractional orders. Although the time reduction is less than the iteration reduction, the overall computational savings remain substantial. This indicates that the HSGS method is more efficient for large-scale problems where computational cost becomes a critical factor.

Additionally, the fractional order α has a significant influence on both convergence and accuracy. As α increases, the number of iterations required for convergence decreases for both methods. This indicates that problems with higher fractional orders exhibit faster convergence behavior. Furthermore, the accuracy difference between HSGS and GS becomes less pronounced as α increases. When α is small, the strong memory effect inherent in fractional derivatives amplifies the impact of incomplete updates in HSGS, leading to larger errors. In contrast, for larger values of α, the system behaves more like a classical 2D diffusion equation, allowing HSGS to achieve accuracy comparable to GS.

Overall, our numerical results demonstrate that the HSGS method provides a favorable balance between computational efficiency and numerical accuracy. While it introduces a modest increase in error compared to the conventional GS method, this drawback is mitigated by grid refinement and becomes negligible for higher fractional orders. The significant reduction in the number of iterations and execution time makes HSGS particularly suitable for large-scale simulations of 2DTFDE. Therefore, the proposed HSGS method can become an efficient alternative method to the conventional GS method in applications where computational efficiency is prioritized over marginal improvements in accuracy.

4. Conclusion

This paper presented an efficient numerical method for solving 2DTFDE based on the Caputo fractional derivative. An implicit finite difference scheme using the L1 approximation was employed to discretize the governing equations, resulting in sparse linear systems that are penta-diagonal. To efficiently solve these systems, the HSGS iterative method was systematically formulated. By updating only a subset of the grid points at each iteration through an appropriate ordering strategy, the proposed method significantly reduces computational effort while preserving the convergence characteristics of the conventional GS method. The numerical experiments demonstrated that HSGS achieves numerical accuracy comparable to GS, while substantially reducing the number of iterations and execution time. The results indicate that the HSGS method provides a simple yet effective acceleration technique for implicit solvers of 2DTFDE. Due to its ease of implementation and significant efficiency gains, the proposed HSGS method is well suited for large-scale simulations arising in scientific and engineering applications. Future work may focus on extending this method to solve problems with variable coefficients, nonlinear source terms, or space-time fractional operators.

Acknowledgements. The authors wish to acknowledge the financial support provided by Universiti Malaysia Sabah for the publication of this research.

Authors' Contributions. All authors have read and approved the final version of the manuscript. The authors contributed equally to this work.

Conflicts of Interest. The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] X. Liao, M. Feng, Time-Fractional Diffusion Equation-Based Image Denoising Model, Nonlinear Dyn. 103 (2021), 1999–2017. https://doi.org/10.1007/s11071-020-06136-x.

[2] H. Chen, H. Qiao, W. Wei, J. Li, Time Fractional Diffusion Equation Based on Caputo Fractional Derivative for Image Denoising, Opt. Laser Technol. 168 (2024), 109855. https://doi.org/10.1016/j.optlastec.2023.109855.

[3] M. González-Olvera, L. Torres, J. Hernández-Fontes, E. Mendoza, Time Fractional Diffusion Equation for Shipping Water Events Simulation, Chaos Solitons Fractals 143 (2021), 110538. https://doi.org/10.1016/j.chaos.2020.110538.

[4] A.P. Jayaraj, K.N. Gounder, J. Rajagopal, Optimizing Signal Smoothing Using HERS Algorithm and Time Fractional Diffusion Equation, Expert Syst. Appl. 238 (2024), 122250. https://doi.org/10.1016/j.eswa.2023.122250.

[5] D. Sundriyal, P. Saxena, Mathematical Model of Time-Fractional Diffusion Equation in Drug-Eluting Stent, in: J. Singh, G.A. Anastassiou, D. Baleanu, D. Kumar (Eds.), Advances in Mathematical Modelling, Applied Analysis and Computation, Springer Nature Switzerland, Cham, 2024: pp. 206–218. https://doi.org/10.1007/978-3-031-56307-2_14.

[6] R. Gorenflo, Y. Luchko, M. Yamamoto, Time-Fractional Diffusion Equation in the Fractional Sobolev Spaces, Fract. Calc. Appl. Anal. 18 (2015), 799–820. https://doi.org/10.1515/fca-2015-0048.

[7] I.K. Youssef, A.R.A. Ali, A Multilevel Approach for Stability Conditions in Fractional Time Diffusion Problems, J. Phys. Conf. Ser. 1591 (2020), 012085. https://doi.org/10.1088/1742-6596/1591/1/012085.

[8] S. Arshad, M. Wali, J. Huang, S. Khalid, N. Akbar, Numerical Framework for the Caputo Time-Fractional Diffusion Equation with Fourth Order Derivative in Space, J. Appl. Math. Comput. 68 (2022), 3295–3316. https://doi.org/10.1007/s12190-021-01635-5.

[9] S. Niazai, A.A. Rahimzai, M. Danesh, B. Safi, Numerical Solution of Diffusion Equation with Caputo Time Fractional Derivatives Using Finite-Difference Method with Neumann and Robin Boundary Conditions, Aust. J. Eng. Innov. Technol. 4 (2022), 95–108. https://doi.org/10.34104/ijmms.022.0950108.

[10] P. Roul, V.M.K.P. Goura, R. Cavoretto, A Numerical Technique Based on B-spline for a Class of Time-fractional Diffusion Equation, Numer. Methods Partial Differ. Equ. 39 (2021), 45–64. https://doi.org/10.1002/num.22790.

[11] S. Kumari, M. Mehra, Numerical Solution to Loaded Difference Scheme for Time-Fractional Diffusion Equation with Temporal Loads, J. Math. Chem. 63 (2024), 105–131. https://doi.org/10.1007/s10910-024-01658-w.

[12] A. Achabbak, M. Seaid, N. Yebari, A Highly Efficient Method for Numerical Solution of Time-Fractional Partial Differential Equations, Moroccan J. Pure Appl. Anal. 10 (2024), 25–47.

[13] Y. Zhao, P. Chen, W. Bu, X. Liu, Y. Tang, Two Mixed Finite Element Methods for Time-Fractional Diffusion Equations, J. Sci. Comput. 70 (2015), 407–428. https://doi.org/10.1007/s10915-015-0152-y.

[14] H.B. Jebreen, A Highly Accurate Computational Approach to Solving the Diffusion Equation of a Fractional Order, Mathematics 12 (2024), 1965. https://doi.org/10.3390/math12131965.

[15] P. Roul, V.M.K.P. Goura, R. Agarwal, A High-Order Compact Finite Difference Scheme and Its Analysis for the Time-Fractional Diffusion Equation, J. Math. Chem. 61 (2023), 2146–2175. https://doi.org/10.1007/s10910-023-01510-7.

[16] P. Agarwal, A. Sunarto, J.V.L. Chew, J. Sulaiman, S. Momani, New Preconditioning and Half-Sweep Accelerated Overrelaxation Solution for Fractional Differential Equation, J. King Saud Univ. - Sci. 35 (2023), 102461. https://doi.org/10.1016/j.jksus.2022.102461.

[17] A. Sunarto, J. Sulaiman, J. Lung, Numerical Solution of the Time-Fractional Diffusion Equations via Quarter-Sweep Preconditioned Gauss-Seidel Method, Int. J. Appl. Math. 34 (2021), 111–125. https://doi.org/10.12732/ijam.v34i1.5.

[18] A. Sunarto, J. Sulaiman, J.V.L. Chew, Computational Half-Sweep Preconditioned Gauss–Seidel Method for Time-Fractional Diffusion Equations, in: Fractional Order Systems and Applications in Engineering, Elsevier, 2023: pp. 285–300. https://doi.org/10.1016/b978-0-32-390953-2.00024-4.

[19] J.V.L. Chew, J. Sulaiman, Implicit Finite Difference Solution of One-Dimensional Porous Medium Equations Using Half-Sweep Newton-Explicit Group Iterative Method, J. Eng. Appl. Sci. 13 (2018), 1286–1290.

[20] J.V.L. Chew, E. Aruchunan, J. Sulaiman, Efficiency Evaluation of Half-Sweep Newton-EGSOR Method to Solve 1D Nonlinear Porous Medium Equations, in: S.A. Abdul Karim, A. Shafie, (eds) Towards Intelligent Systems Modeling and Simulation. Studies in Systems, Decision and Control, Vol 383, Springer, Cham, (2022). https://doi.org/10.1007/978-3-030-79606-8_25.

[21] A.T. Balasim, N.H.M. Ali, A Rotated Crank-Nicolson Iterative Method for the Solution of Two-Dimensional Time-Fractional Diffusion Equation, Indian J. Sci. Technol. 8 (2015), 1–7. https://doi.org/10.17485/ijst/2015/v8i32/92045.

[22] I. Podlubny, Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications, Academic Press, 1998.