Approximation Solution of the Fractional Parabolic Partial Differential Equation by the Half-Sweep and Preconditioned Relaxation

Article *in* Symmetry · June 2021

DOI: 10.3390/sym13061005

4 authors:

Andang Sunarto
Universitas Islam Negeri (UIN) Fatmawati Sukarno, Bengkulu, Indonesia
62 PUBLICATIONS **113** CITATIONS
SEE PROFILE

Praveen Agarwal
Anand International College of Engineering
343 PUBLICATIONS **3,646** CITATIONS
SEE PROFILE

Jackel Vui Lung Chew
Universiti Malaysia Sabah Labuan International Campus
23 PUBLICATIONS **35** CITATIONS
SEE PROFILE

Jumat Sulaiman
Universiti Malaysia Sabah (UMS)
359 PUBLICATIONS **1,557** CITATIONS
SEE PROFILE

symmetry

MDPI

Article

Approximation Solution of the Fractional Parabolic Partial Differential Equation by the Half-Sweep and Preconditioned Relaxation

Andang Sunarto [1,*], Praveen Agarwal [2,3], Jackel Vui Lung Chew [4,*] and Jumat Sulaiman [5]

[1] Department Tadris Matematika, IAIN Bengkulu, Kota Bengkulu 38211, Indonesia
[2] Department of Mathematics, Anand International College of Engineering, Jaipur 303012, India; goyal.praveen2011@gmail.com
[3] Nonlinear Dynamics Research Center (NDRC), Ajman University, Ajman 346, United Arab Emirates
[4] Faculty of Computing and Informatics, Universiti Malaysia Sabah Labuan International Campus, Labuan F.T. 87000, Malaysia
[5] Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Kota Kinabalu 88400, Malaysia; jumat@ums.edu.my
* Correspondence: author: andang99@gmail.com (A.S.); jackelchew93@ums.edu.my (J.V.L.C.)

Abstract: In this study, the numerical solution of a space-fractional parabolic partial differential equation was considered. The investigation of the solution was made by focusing on the space-fractional diffusion equation (SFDE) problem. Note that the symmetry of an efficient approximation to the SFDE based on a numerical method is related to the compatibility of a discretization scheme and a linear system solver. The application of the one-dimensional, linear, unconditionally stable, and implicit finite difference approximation to SFDE was studied. The general differential equation of the SFDE was discretized using the space-fractional derivative of Caputo with a half-sweep finite difference scheme. The implicit approximation to the SFDE was formulated, and the formation of a linear system with a coefficient matrix, which was large and sparse, is shown. The construction of a general preconditioned system of equation is also presented. This study's contribution is the introduction of a half-sweep preconditioned successive over relaxation (HSPSOR) method for the solution of the SFDE-based system of equation. This work extended the use of the HSPSOR as an efficient numerical method for the time-fractional diffusion equation, which has been presented in the 5th North American International Conference on industrial engineering and operations management in Detroit, Michigan, USA, 10–14 August 2020. The current work proposed several SFDE examples to validate the performance of the HSPSOR iterative method in solving the fractional diffusion equation. The outcome of the numerical investigation illustrated the competence of the HSPSOR to solve the SFDE and proved that the HSPSOR is superior to the standard approximation, which is the full-sweep preconditioned SOR (FSPSOR), in terms of computational complexity.

Keywords: implicit finite difference scheme; Caputo's partial derivative; HSPSOR; space-fractional; fractional diffusion equation

Citation: Sunarto, A.; Agarwal, P.; Chew, J.V.L.; Sulaiman, J. Approximation Solution of the Fractional Parabolic Partial Differential Equation by the Half-Sweep and Preconditioned Relaxation. *Symmetry* **2021**, *13*, 1005. https://doi.org/10.3390/sym13061005

Academic Editor: Alexander Shapovalov

Received: 30 April 2021
Accepted: 29 May 2021
Published: 3 June 2021

1. Introduction

In recent years, many effective mathematical physics models have been developed using the theory and applying the partial fractional derivatives. In most of the fractional partial-differential-equation-related works of literature, partial fractional derivatives appear in many anomalous phenomena modelling and in complex systems theory. Examples include a fractional mathematical model of the dynamics of the cancer chemotherapy effect based on both the time instant and the time history [1]; a fractional mathematical model of the population dynamics among cancer stem cells, tumor cells, healthy cells, the

effects of excess estrogen and the body's natural immune response on the cell populations [2]; a fractional mathematical model to investigate the dynamics of tuberculosis between the children and the adults [3]; a fractional model an of energy supply-demand system [4]; and a fractional mathematical model of the COVID-19 pandemic [5].

The success of fractional partial differential equations was attributed to the derivative order's generalization, from integer-order to arbitrary order. The fractional derivatives in the fractional partial differential equations have an effective memory function that enables many physical phenomena to be described effectively. Fractional partial differential equations can be categorized as a time-fractional type, a space-fractional type, and a time- and space-fractional type. Our research focused on investigating the numerical solution of the space-fractional partial differential equation, particularly the parabolic type equation such as the space-fractional diffusion equation (SFDE). This research focus came from the need to discover an efficient solution for the SFDE problem.

Based on our short review of the existing solution methods, many researchers have suggested the finite-difference method. The method of finite difference can be implicit or explicit, depending on its stability to obtain the solution [6–10]. Since SFDE involves changing the time and space of a continuous variable, numerical treatment such as finite difference discretization is necessary to transform the differential equation into a finite system of linear equations that a computer can solve. Although SFDE can be solved without discretization, this research focused on improving an iterative method that is mostly formulated based on a discretized equation. Furthermore, the iterative method works best in obtaining an accurate approximate solution when a finite system of equations is large and complex. Other than the method of finite difference, several methods have been proposed to solve the space-fractional problems, such as the spectral collocation method [11], the boundary value method [12], the finite volume method [13], and the method of lines and splines [14].

In this study, an unconditionally stable implicit finite difference scheme and the β order Caputo fractional partial derivative were executed to discretize the SFDE and to obtain the correct approximation equation. The generated system of equation after the approximation equation was used on the solution domain, which led to a tridiagonal linear system. Since the sparse and large-scale linear system's coefficient matrix is difficult to be solved analytically, an iterative method is employed as an alternative. In terms of the efficiency of the iterative processes, many authors [8,15,16] have suggested and debated over several numerical iterative methods. Besides that, the block iteration has been presented [17] to show the computation cost efficiency improvement by separating systems of equations. The preconditioned iterative methods have also been widely recognized as the competent methods for solving linear systems. Thus, the symmetry in solving the SFDE problem via a numerical method exists in the use of the finite difference discretization scheme and the iterative method. A finite difference scheme needs to be unconditionally stable so that the solution can be obtained regardless of the time and space step sizes used. On the other hand, an iterative method needs to be efficient, which means the accurate solution of a highly complex linear system can be computed in a short time.

The contribution of this study was to build and examine the efficacy of the half-sweep preconditioned SOR (HSPSOR) method, which was formulated from the use of implicit finite difference and the Caputo fractional derivative, for resolving the SFDE implicitly. We also used the full-sweep preconditioned SOR (FSPSOR) iterative techniques as a control method to analyze the efficacy of the HSPSOR method. To begin the formulation and the investigation of the HSPSOR method, we considered the following general form of the SFDE as follows:

$$\frac{\partial U(x,t)}{\partial t} = A_1 \frac{\partial^\beta U(x,t)}{\partial x^\beta} + A_2 \frac{\partial U(x,t)}{\partial x} + A_3 U(x,t) + B(x,t), \quad (1)$$

where $A_i, i = 1, 2,$ and 3 were arbitrary constants, and $B(x,t)$ was a known function. Notice that when the value of β in Equation (1) equals 2, one can get a usual diffusion

equation with a second-order derivative in space. The derivative of order-β for an SFDE usually lies between $1 < \beta \leq 2$ to provide flexibility in the study of the effect of the medium and the space interaction with the fluid [18]. The solution of Equation (1), which is $U(x,t)$, can be approximated using the method of finite difference subject to the initial and boundary conditions as given by:

$$U(x,0) = U_0, 0 \leq x \leq l, \tag{2}$$

and

$$U(0,t) = U_L, U(l,t) = U_R, 0 < t < T. \tag{3}$$

Before the space-fractional term in Equation (1) is discretized by the finite difference mean, the following established definitions from the theory of fractional derivatives must be defined as follows [19]:

Definition 1. *Let a real number* $\alpha > 0$*, and let the function* f *be continuous on* $F' = (0,\infty)$ *and integrable on any finite subinterval of* $F = [0,\infty)$*. Then, for* $x > 0$*, the Riemann–Liouville fractional integral,* f *with the order* α *is defined as*:

$$J^{\alpha} = \frac{1}{\Gamma(\alpha)} \int_0^x (x-\xi)^{\alpha-1} f(\xi)\, d\xi. \tag{4}$$

Definition 2. *Let a real number* $\beta > 0$*, such that* $m - 1 < \beta \leq m$*, where* m *is a natural number element. Let the function* f *be continuous on* $F' = (0,\infty)$ *and integrable on any finite subinterval of* $F = [0,\infty)$*. Then, for* $x > 0$*, the Caputo fractional,* f *with the order* β *is defined as:*

$$D_x^{\beta} f(x) = \frac{1}{\Gamma(m-\beta)} \int_0^x \frac{f^m(\xi)}{(x-\xi)^{\beta-m+1}} d\xi. \tag{5}$$

From Definition 2, the following properties hold:

$$D_x^{\beta} C = 0, \tag{6}$$

and

$$D_x^{\beta} x^{\eta} = \begin{cases} 0, \\ \dfrac{\Gamma(\eta+1)}{\Gamma(\eta+1-\beta)} x^{\eta-\beta}, \end{cases} \tag{7}$$

where η is an element of natural number and $\Gamma(.)$ is a gamma function.

2. Approximation to a Space-Fractional Diffusion Equation

Equation (1) approximates the space-fractional term using the Caputo definition and the second-order half-sweep finite-difference. For more details about half-sweep see[17,20–22]. Suppose that $h = \frac{\ell}{P}$, with P as any positive integer. Then, the space-fractional term can be equated as:

$$\frac{\partial^{\beta} U(x,t)}{\partial x^{\beta}} = \frac{1}{\Gamma(2-\beta)} \int_0^x \frac{\partial^2 U(\xi,t)}{\partial x^2} (x-\xi)^{1-\beta} d\xi, \tag{8}$$

which is also equivalence to:

$$\frac{1}{\Gamma(2-\beta)} \sum_{j=0,2,4,\ldots}^{i-2} \int_{jh}^{(j+1)h} \frac{U_{i+2-j,n} - 2U_{i-j,n} + U_{i-2-j,n}}{2h^2} (Ph-\xi)^{1-\beta} d\xi. \tag{9}$$

Let $\varphi = \frac{2h^{-2}}{\Gamma(3-\beta)}$ and $g_j^\beta = \left(\frac{j}{2}+1\right)^{2-\beta} - \left(\frac{j}{2}\right)^{2-\beta}$. Then, from Equation (9), we obtain:

$$\frac{\partial^\beta U(x_i, t_n)}{\partial x^\beta} = \varphi \sum_{j=0,2,4,\ldots}^{i-2} g_j^\beta \left(U_{i+2-j,n} - 2U_{i-j,n} + U_{i-2-j,n}\right). \tag{10}$$

Using Equation (10) and the implicit finite difference scheme for the remaining derivatives in Equation (1), the simplified approximation to the SFDE via the finite difference and Caputo partial derivative can be formulated into:

$$\begin{gathered}\gamma U_{i,n} - A_1\varphi \sum_{j=0,2,4,\ldots}^{i-2} g_j^\beta \left(U_{i+2-j,n} - 2U_{i-j,n} + U_{i-2-j,n}\right) - \frac{A_2}{4h}\left(U_{i+2,n} - U_{i-2,n}\right) \\ -A_3 U_{i,n} - B_{i,n} = \gamma U_{i,n-2},\end{gathered} \tag{11}$$

for $i = 2,4,\ldots,P-2$. Equation (11) can be rearranged neatly into:

$$b_i U_{i-2,n} + (\gamma_i - c_i)U_{i,n} - b_i U_{i+2,n} - a_i^* = f_i, \tag{12}$$

where $b_i = \frac{A_2}{4h}, \gamma_i = \Delta t, c_i = A_3, a_i^* = a_i \sum_{j=0,2,4,\ldots}^{i-2} g_j^\beta \left(U_{i+2-j,n} - 2U_{i-j,n} + U_{i-2-j,n}\right)$, $a_i = A_1\varphi$, and $f_i = \gamma U_{i,n-2} + B_{i,n}$. Based on Equation (12), we can simply state an equation for $n > 3$ as:

$$p_i U_{i-6,n} + q_i U_{i-4,n} + r_i U_{i-2,n} + s_i U_{i,n} + v_i U_{i+2,n} - R_i = f_i, \tag{13}$$

where $R_i = a_i \sum_{j=0,2,4,\ldots}^{i-2} g_j^\beta \left(U_{i+2-j,n} - 2U_{i-j,n} + U_{i-2-j,n}\right), p_i = -a_i g_2^\beta, q_i = -a_i g_1^\beta + 2a_i g_2^\beta, r_i = b_i - a_i g_2^\beta + 2a_i g_1^\beta - a_i, s_i = -a_i g_1^\beta + 2a_i + \gamma - c_i$, and $v_i = -a_i - b_i$. From Equation (13), the system of the linear equation can be expressed as follows:

$$\underset{\sim}{A}\underset{\sim}{U} = \underset{\sim}{f}, \tag{14}$$

where

$$A = \begin{bmatrix} s_2 & v_2 & & & & & & \\ r_4 & s_4 & v_4 & & & & & \\ q_6 & r_6 & s_6 & v_6 & & & & \\ p_8 & q_8 & r_8 & s_8 & v_8 & & & \\ & p_{10} & q_{10} & r_{10} & s_{10} & v_{10} & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & p_{P-4} & q_{P-4} & r_{P-4} & s_{P-4} & v_{P-4} \\ & & & & p_{P-2} & q_{P-2} & r_{P-2} & s_{P-2} \end{bmatrix}_{(P-2)\times(P-2)},$$

$$\underset{\sim}{U} = [U_2 \quad U_4 \quad U_6 \quad \cdots \quad U_{P-4} \quad U_{P-2}]^T,$$

and

$$\underset{\sim}{f} = \left[f_2 - r_2 U_2 \quad f_4 + q_4 U_4 \quad f_6 + p_6 U_6 \quad \cdots \quad f_{P-4} + R_{P-4} \quad f_{P-2} - r_{P-2}U_P + R_{p-2}\right]^T$$

3. Half-Sweep Preconditioned SOR Formulation

The system (Equation (14)) is transformed in the form of the linear equation's preconditioned system as follows:

$$A^* \underset{\sim}{x} = \underset{\sim}{f^*}, \tag{15}$$

where $A^* = \rho A \rho^T, \underset{\sim}{f^*} = \rho \underset{\sim}{f}$, and $\underset{\sim}{U} = \rho^T \underset{\sim}{x}$.

Further, to discover the numerical tridiagonal linear system solutions of Equation (15), we considered the half-sweep preconditioned SOR (HSPSOR) iterative method. The ρ was the preconditioned matrix and was given as [23]:

$$\rho = I + S, \tag{16}$$

where

$$S = \begin{bmatrix} 0 & -v_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -v_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -v_3 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & -v_{P-1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{(P-1)\times(P-1)}.$$

Meanwhile, I was an identity matrix. The generated coefficient matrix A^*, as in Equation (15), can be split as follows:

$$A^* = D - L - V, \tag{17}$$

where D, L, and V were the diagonal matrices, the lower triangular, and the upper triangular, respectively. Considering Equation (17), the HSPSOR iterative method can be rewritten in the form of:

$$\underset{\sim}{x}^{(k+1)} = (D - \omega L)^{-1}[\omega V + (1-\omega)D]\underset{\sim}{x}^{(k)} + (D - \omega L)^{-1}\underset{\sim}{f^*}, \tag{18}$$

where the unknown vector, $\underset{\sim}{x}^{(k+1)}$, was given the solution at the $(k+1)$th iteration, and the relaxation parameters were chosen within the range, $1 < \omega < 2$. The computational algorithm of the HSPSOR iterative method was as follows.

Algorithm 1 The HSPSOR iterative method.

i. Initialize $\underset{\sim}{U} \leftarrow 0$and $\varepsilon \leftarrow 10^{-10}$.

ii. For $j = 0, 1, \ldots, n$, and for $i = 0, 1, 2, \ldots, P$, calculate Equation (18), then approximate solutions:

$$\underset{\sim}{U}^{(k+1)} = \rho^T \underset{\sim}{x}^{(k+1)}.$$

If the convergence criterion is satisfied, that is:

$$\left\| \underset{\sim}{U}^{(k+1)} - \underset{\sim}{U}^{(k)} \right\| \leq \varepsilon,$$

go to Step (iii). Otherwise, go back to Step (ii).

iii Stop.

---

4. Numerical Evaluation via C++

In this section, we implemented two examples of SFDE, using C++ programming to verify the HSPSOR method's effectiveness. The C++ programming language was used to code Algorithm 1 because of a better coding organization and comprehension, which enabled numerical computation to be conducted accurately. Furthermore, C++ programming was preferred for this work compared to other mathematics applications such as

Mathematica and Maple because the number of iterations and the execution time could be recorded and optimized according to the code arrangement.

For the comparison purpose, we considered a standard or full-sweep preconditioned SOR by taking into account the number of iterations, the execution time (seconds), and the maximum error at the values of $\beta = 1.2$, $\beta = 1.5$, and $\beta = 1.8$. The tolerance error considered in Algorithm 1 and the C++ simulation was set at $\varepsilon = 10^{-10}$. This stopping condition was fixed for the different mesh sizes used, M; that is, 128, 256, 512, 1024, and 2048. There was no best value of ε because, when a large value of ε was chosen such as $\varepsilon = 10^{-2}$, the number of iterations became small, and the result became inaccurate. Moreover, when a small value of ε was used, such as $\varepsilon = 10^{-100}$, the number of iterations became large without improving the accuracy. Therefore, $\varepsilon = 10^{-10}$ was selected arbitrarily to check the convergence of the solution. The solution can only be obtained through the simulation of Algorithm 1 after the iteration process was completed successfully.

Example 1. *Let us consider the SFDE's initial boundary value problem*:

$$\frac{\partial U(x,t)}{\partial t} = A_1 \frac{\partial U^{\beta}(x,t)}{\partial x^{\beta}} + B(x,t), \tag{19}$$

Example 2. *Let us consider the SFDE's initial boundary value problem*:

$$\frac{\partial U(x,t)}{\partial t} = \Gamma(1.2)x^{\beta} \frac{\partial U^{\beta}(x,t)}{\partial x^{\beta}} + 3x^2(2x-1)e^{-t}, \tag{20}$$

Based on Tables 1 and 2, it can be observed that for the five different mesh sizes and the three different β used for the simulation of FSPSOR and HSPSOR to solve Example 1 and 2, respectively, the number of iterations and the execution time required by HSPSOR are always smaller than FSPSOR. These results illustrated the success of the Half-Sweep finite difference scheme with Caputo's space fractional to approximate the solution of SFDE with lower computational complexity. The maximum errors produced by PSOR iteration to solve both Example 1 and 2 decreases with the increasing mesh sizes for $\beta = 1.8$ and 1.5. This means that with these values of β, the numerical solution of SFDE via HSPSOR can be computed nearly to its exact solution with a sufficiently large mesh size is used. However, for $\beta = 1.2$, the maximum error is getting larger when the mesh is made to be narrower. This shows the sign of limitation from the use of PSOR iteration, and a modification on PSOR iteration needs to be conducted to handle this problem.

Table 1. Computing the results with $\beta = 1.2, 1.5,$ and 1.8.

M	Method	$\beta = 1.2$			$\beta = 1.5$			$\beta = 1.8$		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	FSPSOR	34	0.84	2.37×10^{-2}	80	1.90	6.20×10^{-4}	246	5.76	3.99×10^{-2}
	HSPSOR	23	0.38	2.37×10^{-2}	37	0.54	6.99×10^{-4}	94	2.36	3.99×10^{-2}
256	FSPSOR	67	5.33	2.44×10^{-2}	211	17.84	5.69×10^{-4}	806	67.75	3.97×10^{-2}
	HSPSOR	34	2.73	2.44×10^{-2}	94	6.90	6.21×10^{-4}	303	34.65	3.97×10^{-2}
512	FSPSOR	129	41.43	2.47×10^{-2}	566	182.83	5.36×10^{-4}	2635	843.91	3.96×10^{-2}
	HSPSOR	67	22.65	2.47×10^{-2}	246	86.09	5.69×10^{-4}	988	421.58	3.96×10^{-2}
1024	FSPSOR	278	472.35	2.49×10^{-2}	1514	898.29	5.13×10^{-4}	11,829	2099.87	3.95×10^{-2}
	HSPSOR	141	206.58	2.49×10^{-2}	655	434.72	5.36×10^{-4}	5413	1033.78	3.95×10^{-2}
2048	FSPSOR	608	1219.76	2.50×10^{-2}	4052	4299.73	5.02×10^{-4}	47,289	8852.28	3.93×10^{-2}
	HSPSOR	305	608.80	2.50×10^{-2}	2188	2133.43	5.13×10^{-4}	23,143	4425.90	3.93×10^{-2}

Table 2. Computing the results with $\beta = 1.2, 1.5,$ and 1.8.

M	Method	$\beta = 1.2$			$\beta = 1.5$			$\beta = 1.8$		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	FSPSOR	34	0.84	2.37×10^{-2}	80	1.90	6.20×10^{-4}	246	5.76	3.99×10^{-2}
	HSPSOR	23	0.38	2.37×10^{-2}	37	0.54	6.99×10^{-4}	94	2.36	3.99×10^{-2}
256	FSPSOR	67	5.33	2.44×10^{-2}	211	17.84	5.69×10^{-4}	806	67.75	3.97×10^{-2}
	HSPSOR	34	2.73	2.44×10^{-2}	94	6.90	6.21×10^{-4}	303	34.65	3.97×10^{-2}
512	FSPSOR	129	41.43	2.47×10^{-2}	566	182.83	5.36×10^{-4}	2635	843.91	3.96×10^{-2}
	HSPSOR	67	22.65	2.47×10^{-2}	246	86.09	5.69×10^{-4}	988	421.58	3.96×10^{-2}
1024	FSPSOR	278	472.35	2.49×10^{-2}	1514	898.29	5.13×10^{-4}	11,829	2099.87	3.95×10^{-2}
	HSPSOR	141	206.58	2.49×10^{-2}	655	434.72	5.36×10^{-4}	5413	1033.78	3.95×10^{-2}
2048	FSPSOR	608	1219.76	2.50×10^{-2}	4052	4299.73	5.02×10^{-4}	47,289	8852.28	3.93×10^{-2}
	HSPSOR	305	608.80	2.50×10^{-2}	2188	2133.43	5.13×10^{-4}	23,143	4425.90	3.93×10^{-2}

5. Conclusions

This paper describes the mathematical derivation of the implicit finite difference in Caputo's approximation equations in which this approximation equation leads to a linear system. By imposing the iterative methods of the FSPSOR and the HSPSOR, based on observation of all experimental effects, it was evident that the number of iterations of the HSPSOR decreased by approximately 31.30–85.45 per cent compared with the iterative methods of the FSPSOR. Meanwhile, the execution time was much quicker, by around 41.18–95.33% more than the FSPSOR method. This implies that, relative to the FSPSOR iterative methods, the HSPSOR method needs the minimum number of iterations and computational time. It can be inferred from the precision of both iterative approaches that their numerical resolutions were in the acceptable domain. When the tabulated numerical results of the HSPSOR to solve the SFDE was compared against the results from the application of the HSPSOR on the time-fractional diffusion equation (TFDE) [24], it was found that the overall maximum errors produced by the HSPSOR in solving the SFDE were slightly greater than when the HSPSOR was used to solve the TFDE. A thorough study of the error of the method used will be conducted in the future. From this work, the capability of the half-sweep finite difference scheme to reduce the computational complexity for solving space- and time-fractional diffusion equations and the compatibility of the scheme with the PSOR iteration was shown.

Author Contributions: Writing—original draft preparation, A.S.; writing—review and editing, J.V.L.C.; supervision, P.A. and J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Reference

1. Veeresha, P.; Prakasha, D.G.; Baskonus, H.M. New numerical surfaces to the mathematical model of cancer chemotherapy effect in Caputo fractional derivatives. *Chaos Interdiscip. J. Nonlinear Sci.* **2019**, *29*, 013119, doi:10.1063/1.5074099.
2. Solís-Pérez, J.E.; Gómez-Aguilar, J.F.; Atangana, A. A fractional mathematical model of breast cancer competition model. *Chaos Solitons Fractals* **2019**, *127*, 38–54, doi:10.1016/j.chaos.2019.06.027.
3. Fatmawati, M.A.K.; Bonyah, E.; Hammouch, Z.; Shaiful, E.M. A mathematical model of tuberculosis (TB) transmission with children and adults groups: A fractional model. *Aims Math.* **2020**, *5*, 2813–2842, doi:10.3934/math.2020181.
4. Noeiaghdam, S.; Sidorov, D. Caputo-Fabrizio Fractional Derivative to Solve the Fractional Model of Energy Supply-Demand System. *Math. Model. Eng. Probl.* **2020**, *7*, 359–367, doi:10.18280/mmep.070305.

5. Higazy, M. Novel fractional order SIDARTHE mathematical model of COVID-19 pandemic. *Chaos Solitons Fractals* **2020**, *138*, 110007, doi:10.1016/j.chaos.2020.110007.
6. Çelik, C.; Duman, M. Crank–Nicolson method for the fractional diffusion equation with the Riesz fractional derivative. *J. Comput. Phys.* **2012**, *231*, 1743–1750, doi:10.1016/j.jcp.2011.11.008.
7. Guo, X.; Li, Y.; Wang, H. A fourth-order scheme for space fractional diffusion equations. *J. Comput. Phys.* **2018**, *373*, 410–424, doi:10.1016/j.jcp.2018.03.032.
8. Sunarto, A.; Sulaiman, J. Investigation of fractional diffusion equation via QSGS iterations. *J. Phys. Conf. Ser.* **2019**, *1179*, 012014, doi:10.1088/1742-6596/1179/1/012014.
9. Tian, W.; Zhou, H.; Deng, W. A class of second order difference approximations for solving space fractional diffusion equations. *Math. Comput.* **2015**, *84*, 1703–1727, doi:10.1090/S0025-5718-2015-02917-2.
10. Zhou, H.; Tian, W.; Deng, W. Quasi-compact finite difference schemes for space fractional diffusion equations. *J. Sci. Comput.* **2013**, *56*, 45–66, doi:10.1007/s10915-012-9661-0.
11. Li, S.; Zhou, Z. Fractional spectral collocation method for optimal control problem governed by space fractional diffusion equation. *Appl. Math. Comput.* **2019**, *350*, 331–347, doi:10.1016/j.amc.2019.01.018.
12. Chen, H.; Huang, Q. Kronecker product based preconditioners for boundary value method discretizations of space fractional diffusion equations. *Math. Comput. Simul.* **2020**, *170*, 316–331, doi:10.1016/j.matcom.2019.11.007.
13. Fu, H.; Liu, H.; Wang, H. A finite volume method for two-dimensional Riemann-Liouville space-fractional diffusion equation and its efficient implementation. *J. Comput. Phys.* **2019**, *388*, 316–334, doi:10.1016/j.jcp.2019.03.030.
14. Salehi, Y.; Darvishi, M.T.; Schiesser, W.E. Numerical solution of space fractional diffusion equation by the method of lines and splines. *Appl. Math. Comput.* **2018**, *336*, 465–480, doi:10.1016/j.amc.2018.04.053.
15. Hacksbusch, W. *Iterative Solution of Large Sparse Systems of Equations*, 2nd ed.; Springer International Publishing: Cham, Switzerland, 2016.
16. Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2003.
17. Ibrahim, A.; Abdullah, A.R. Solving the two dimensional diffusion equation by the four point explicit decoupled group (EDG) iterative method. *Int. J. Comput. Math.* **1995**, *58*, 253–263, doi:10.1080/00207169508804447.
18. Caputo, M. Diffusion with space memory modelled with distributed order space fractional differential equations. *Ann. Geophys.* **2003**, *46*, doi:10.4401/Ag-3395.
19. Miller, K.S.; Ross, B. *An Introduction to the Fractional Calculus and Fractional Differential Equations*, 1st ed.; Wiley-Interscience: New York, NY, USA, 1993.
20. Aruchunan, E.; Muthuvalu, M.S.; 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 Ma-lays.* **2015**, *44*, 139–146, doi:10.17576/jsm-2015-4401-19.
21. Lung, J.C.V.; Sulaiman, J. On quarter-sweep finite difference scheme for one-dimensional porous medium equations. *Int. J. Appl. Math.* **2020**, *2020*, 33, doi:10.12732/ijam.v33i3.6.
22. Sunarto, A.; Sulaiman, J.; Saudi, A. Solving the time fractional diffusion equations by the half-sweep SOR iterative method. In Proceedings of the 2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA), Bandung, Indonesia, 20–21 August 2014; pp. 272–277, doi:10.1109/ICAICTA.2014.7005953.
23. Gunawardena, A.D.; Jain, S.K.; Snyder, L. Modified iterative methods for consistent linear systems. *Linear Algebra Its Appl.* **1991**, *154–156*, 123–143, doi:10.1016/0024-3795(91)90376-8.
24. Sunarto, A.; Sulaiman, J. Application half-sweep preconditioned SOR method for solving time-fractional diffusion equations. In Proceedings of the International Conference on Industrial Engineering and Operations Management, Detroit, MI, USA, 10–14 August 2020; pp. 3414–3420.