Progress in Fractional Differentiation and Applications
An International Journal

http://dx.doi.org/10.18576/pfda/080208

Numerical Investigation on the Solution of a Space-Fractional via Preconditioned SOR Iterative Method

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

[1] Department Tadris Matematika, IAIN Bengkulu, Indonesia
[2] Department Mathematics, Anand International College of Engineering, Rajasthan 303012, India
[3] Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia
[4] Faculty of Computing and Informatics, Universiti Malaysia Sabah Labuan International Campus, Malaysia

Received: 2 Jun. 2020, Revised: 18 Sep. 2020, Accepted: 20 Sep. 2020
Published online: 1 Apr. 2022

Abstract: This paper considers a numerical investigation on the solution of a one-dimensional linear space-fractional partial differential equation with the application of an unconditionally implicit finite difference method and the Caputo's space-fractional derivative. We formulate the Caputo'simplicit finite difference approximation equation to form a corresponding linear system in which its coefficient matrix is large-sized and has a great sparsity. We construct a preconditioned linear system intending to speed up the convergence rate in computing the solutions of the linear system using the SOR iterative method. We present two examples of the one-dimensional linear space-fractional partial differential equation problem to illustrate the effectiveness and efficiency of our proposed PSOR iterative method. Through the investigation, the numerical results show that the proposed PSOR iterative method is superior to the preconditioned Gauss-Seidel and Gauss-Seidel iterative methods in terms of efficiency.

Keywords: Space fractional diffusion equation, implicit finite difference, Caputo's derivative, PSOR method.

1 Introduction, motivation and preliminaries

To the best of our literature review, many successful fractional partial differential equations or in short FPDEs have been formulated for the best mathematical model in describing physical phenomena such as fluid mechanics, solid-state physics, and chemical kinematics. For instance, a fractional advection-dispersion equation has been used in the modeling of the transport of the passive tracers in the fluid flow through a porous medium [1]. Then, a fractional diffusion equation has been applied to model the process of heat transfer in hybrid nanofluid [2]. Other than that, the theory of FPDEs has been applied in developing an image denoising model which can suppress noise and maintain the image edge better than the use of PDE [3]. The application of FPDEs is not limited to the science areas such as physics, biology, and chemistry. The FPDE of parabolic type like fractional diffusion equation can also be extended to the theory of option pricing and generalized the exponential market models [4]. From this review, it can be seen that the FPDEs able to give more realistic results in the description of any phenomena because the fractional parameter can be adjusted to the extent it fits the experimental data.

Since the solutions of the FPDEs are important to understand the theory or behavior of any phenomena, many methods, either via analytical or numerical mean, have been developed. Some of the analytical methods that have been proposed to solve the application of FPDE are the transform method [5] and the collocation method [6]. Besides that, several researchers have proposed the finite difference methods, which can be categorized as explicit, implicit, and the mixture of these, to solve the FPDEs numerically [7–11]. It was pointed out that the implicit method has the properties of unconditionally stable and has a lower computational cost compared to the mixture of explicit-implicit, for instance, the Crank-Nicolson method. Due to the properties of the implicit finite difference method, we apply an implicit finite

* Corresponding author e-mail: andang99@gmail.com

difference method for the discretization of the space-fractional partial differential equation (SFPDE) with the use of Caputo's fractional partial derivative of order β.

From the application of the implicit finite difference method and the Caputo's fractional partial derivative of order β, we formulate an approximation equation that leads to a tridiagonal linear system. The linear system's coefficient matrix has properties such as large-sized and has great sparsity which encourages the use of iterative methods as the alternative option to obtain efficient solutions. As far as the iterative methods are concerned, many researchers, for instance [12–15], have proposed and discussed several iterative method families to solve a large and sparse system of linear equations. From our recent literature review on the numerical solution of the fractional partial differential equation, [16] proposed a circulant preconditioner technique for the solution of the fractional diffusion equation. And then [17] used a preconditioned fast parallel method to solve the space-time fractional diffusion equation. The preconditioning technique is widely known in solving a large and sparse linear system of equations. It is used to transform a linear system into another form of a linear system which is equivalent but the solutions can converge faster especially when a particular iterative method is used.

Besides that, the concept of block iteration, which was extended from single point iteration in the implementation of an iterative method, has been introduced [18]. Due to its potential to reduce the computation cost as well as to improve the efficiency of an iterative method, several researchers have applied the concept of block iteration in their studies [19, 20]. Other than the block iteration, the preconditioned iterative methods [13–15, 21] have also been widely known as one of the techniques that are efficient in solving large and sparse linear systems. By utilizing the unique properties and the potential of preconditioned type iterative methods, we propose a Preconditioned SOR (PSOR) iterative method for the efficient solution of one-dimensional linear SFPDE with the use Caputo's implicit finite difference approximation. Both the effectiveness and efficiency of the PSOR method in solving the transformed linear system will be investigated by comparing its numerical result against the Preconditioned Gauss-Seidel (PGS) method and Gauss-Seidel (GS) which acts as the control method.

The objective of this paper is to construct the Caputo's implicit finite difference approximation equation, let us consider the following SFPDE of parabolic type:

$$\frac{\partial}{\partial t}u(x,t) = p(x)\frac{\partial^\beta}{\partial x^\beta}u(x,t) + q(x)\frac{\partial}{\partial x}u(x,t) + r(x)u(x,t) + w(x,t), \tag{1}$$

The solution of (1) can be obtained subjects to the initial condition $u(x,0) = u_0, 0 \le x \le l$, and the boundary conditions $u(0,t) = g_0$ and $u(l,t) = g_1, 0 < t \le T$. We apply the following definitions from fractional derivative theory in the construction of the Caputo's finite difference approximation equation.

Definition 1. [22] The Riemann-Liouville fractional integral operator, J^β of order $-\beta$ can be defined as follows:

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

where $\beta > 0$ and $x > 0$.

Definition 2. [22] The Caputo's fractional partial derivative operator, D^β of order $-\beta$ can be defined as follows:

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

where $\beta > 0$, $m-1 < \beta \le m$, $m \in \mathbb{N}$, and $x > 0$.

Using these definitions shown in (2) and (3), we can have the following properties as follows.

$$D^\beta k = 0, \quad \text{and} \tag{4}$$

$$D^\beta x^n = \begin{cases} 0, & \text{for} \quad n \in \mathbb{N}_0 \quad \text{and} \quad n < [\beta], \\ \frac{\Gamma(n+1)}{\Gamma(n+1-\beta)}x^{n-\beta}, & \text{for} \quad n \in \mathbb{N}_0 \quad \text{and} \quad n \ge [\beta], \end{cases} \tag{5}$$

where k is a constant, $[\beta]$ is the smallest value of an integer which is greater than or equal to β, $\mathbb{N}_0$ is the natural number and $\Gamma(.)$ is the gamma function.

2 Caputo's implicit finite difference approximation equation

Now, the Caputo's implicit finite difference approximation equation is constructed in an ordered fashion as follows. Firstly, we define $h = lM$, with M is any positive integer. Then, using the implicit finite difference method, we get

$$\frac{\partial^\beta}{\partial x^\beta}u(x_i,t_n) = \frac{1}{\Gamma(2-\beta)}\int_0^{t_n}\frac{\partial^2}{\partial x^2}u(x_i,s)(t_n-s)^{1-\beta}\partial s \tag{6}$$

$$= \frac{1}{\Gamma(2-\beta)} \sum_{j=0}^{i-1} \int_{jh}^{(j+1)h} \left(\frac{U_{i-j+1,n} - 2U_{i-j,n} + U_{i-j-1,n}}{h^2} \right) (nh-s)^{\beta} \partial s \tag{7}$$

$$= \frac{h^{-\beta}}{\Gamma(3-\beta)} \sum_{j=0}^{i-1} (U_{i-j+1,n} - 2U_{i-j,n} + U_{i-j-1,n})((j+1)^{2-\beta} - j^{2-\beta}). \tag{8}$$

By letting $\sigma_{\beta,h} = h^{\beta}/\Gamma(3-\beta)$ and $g_j^{\beta} = (j+1)^{2-\beta} - j^{2-\beta}$, then the discrete approximation equation that is shown in (8) can be written into

$$\frac{\partial^{\beta}}{\partial x^{\beta}} u(x_i, t_n) = \sigma_{\beta,h} \sum_{j=0}^{i-1} g_j^{\beta} (U_{i-j+1,n} - 2U_{i-j,n} + U_{i-j-1,n}), \tag{9}$$

Eventually, the approximation equation to (1) based on the application of the Caputo's implicit finite difference method becomes

$$\lambda(U_{i,n} - U_{i,n-1}) = p_i \sigma_{\beta,h} \sum_{j=0}^{i-1} g_j^{\beta} (U_{i-j+1,n} - 2U_{i-j,n} + U_{i-j-1,n}) + q_i \left(\frac{U_{i+1,n} - U_{i-1,n}}{2h} \right) + r_i U_{i,n} + w_{i,n}, \tag{10}$$

for $i = 1, 2, ..., M-1$ and $\lambda = 1/k$. (10) can be simplified further into

$$\lambda U_{i,n-1} = -p_i \sigma_{\beta,h} \sum_{j=0}^{i-1} g_j^{\beta} (U_{i-j+1,n} - 2U_{i-j,n} + U_{i-j-1,n}) - \frac{q_i}{2h} (U_{i+1,n} - U_{i-1,n}) - r_i U_{i,n} - w_{i,n} + \lambda U_{i,n}, \tag{11}$$

and finally, we get

$$w_i^* = q_i^* U_{i-1,n} + (\lambda - r_i^*) U_{i,n} - q_i^* U_{i+1,n} - p_i^* \sum_{j=0}^{i-1} g_j^{\beta} (U_{i-j+1,n} - 2U_{i-j,n} + U_{i-j-1,n}), \tag{12}$$

Each of the coefficients in (12) is denoted as $w_i^* = \lambda U_{i,n-1} + w_{i,n}$, $q_i^* = q_i/2h$, $r_i^* = r_i$ and $p_i^* = p_i \sigma_{\beta,h}$. It is important to point out that the approximation equation shown in (12) can be called as the Caputo's fully implicit finite difference approximation equation. (12) has a second-order accuracy in space-fractional. For the sake of simplicity, we rewrite (12) for the case when $n > 3$ as

$$-\rho_i + a_i U_{i-3,n} + b_i U_{i-2,n} + c_i U_{i-1,n} + d_i U_{i,n} + e_i U_{i+1,n} = f_i, \tag{13}$$

where $\rho_i = p_i^* \sum_{j=3}^{i-1} g_j^{\beta} (U_{i-j+1,n} - 2U_{i-j,n} + U_{i-j-1,n}), a_i = (-p_i^* g_2^{\beta}), b_i = (-p_i^* g_1^{\beta} + 2p_i^* g_2^{\beta}), c_i = (q_i^* - p_i^* g_2^{\beta} + 2p_i^* g_1^{\beta} - p_i^*), d_i = (-p_i^* g_1^{\beta} + 2p_i^* + (\lambda - r_i^*)), e_i = (-p_i^* - q_i^*)$, and $f_i = w_i^*$. Hence, the corresponding linear system from (13) can be formed in the matrix form as follows:

$$A\tilde{U} = \tilde{f}, \tag{14}$$

where

$$A = \begin{bmatrix} d_1 & e_1 & & & & & \\ c_2 & d_2 & e_2 & & & & \\ b_3 & c_3 & d_3 & e_3 & & & \\ a_4 & b_4 & c_4 & d_4 & e_4 & & \\ & a_5 & b_5 & c_5 & d_5 & e_5 & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & a_{M-2} & b_{M-2} & c_{M-2} & d_{M-2} & e_{M-2} \\ & & & & a_{M-1} & b_{M-1} & c_{M-1} & d_{M-1} \end{bmatrix}_{(M-1)\times(M-1)}, \tag{15}$$

$$\tilde{U} = \left[U_{1,1}, U_{2,1}, U_{3,1}, \ldots, U_{M-2,1}, U_{M-1,1} \right]^T, \tag{16}$$

$$\tilde{f} = \left[f_1 - c_1 U_{1,1}, f_2 - b_2 U_{2,1}, f_3 - a_3 U_{3,1}, f_4 + \rho_4, \ldots, f_{M-2,1} + \rho_{M-2}, f_{M-1,1} - e_{M-1} U_{M-1,1} + \rho_{M-1} \right]^T. \tag{17}$$

To find the solutions of the linear system shown in (14), we propose the PSOR iterative method which can be explained further in the next section.

3 Formulation preconditioned SOR method

To formulate the PSOR iterative method for computing the approximate solutions of the linear system in (14), we do the formation of the linear system into the corresponding preconditioned linear system as follows.

$$A^*\tilde{U}^* = \tilde{f}^*, \tag{18}$$

Each matrix components from (18) are denoted as $A^* = PAP^T, \tilde{f}^* = P\tilde{f}$, and $\tilde{U} = P^T\tilde{U}^*$. Also,we define a matrix P as the preconditioned matrix and more details can be referred in [15,21]

$$P = I + S, \tag{19}$$

where

$$S = \begin{bmatrix} 0 & -r_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -r_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -r_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & -r_{M-1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{(M-1)\times(M-1)}, \tag{20}$$

Now, to formulate the PSOR iterative method, we let the coefficient matrix A^* in (18) be expressed in the form of summation of the three-component matrices,

$$A^* = D - L - V, \tag{21}$$

Each component D, L, and V represent the diagonal, lower triangular, and upper triangular parts of the matrix A^* respectively. Then, using the preconditioned shown in (19) and the decomposition matrix in (21), the PSOR iterative method is constructed in the form of iterative formula as [13,15],

$$\tilde{U^*}^{(k+1)} = (D - \omega L)^{-1}[\omega V + (1-\omega)D]\tilde{U^*}^{(k)} + (D - \omega L)^{-1}\tilde{f}^*, \tag{22}$$

where $\tilde{U^*}^{(k+1)}$ represents an unknown vector at $(k+1)$th iteration. We accompany the PSOR iterative formula with the following algorithm to facilitate the implementation of the formula to solve the one-dimensional linear SFPDE. Below is the following algorithm.

Algorithm 1: PSOR iterative method
i. Set the initial guess $\tilde{U}^* \leftarrow 0$ and the tolerance $\varepsilon \leftarrow 10^{-10}$,
ii. For $j = 1,2,...,n$, do
For $i = 1,2,...,M-1$, compute formula (22)
then $\tilde{U}^{(k+1)} = P^T\tilde{U^*}^{(k+1)}$
iii. Check convergence $||\tilde{U}^{(k+1)} - \tilde{U}^{(k)}|| \leq \varepsilon$. If it is satisfied, return to Step (ii).
iv. Show approximate solutions.

4 Numerical experiment

In this section, we investigate both the effectiveness and efficiency of the PSOR method using the comparison analysis against the Preconditioned Gauss-Seidel method (PGS) and Gauss-Seidel method (GS). To do so, we have selected two examples of the one-dimensional linear SFPDE. Below are the following examples to be tested.

Example 1 [15]: In this example, based on (1), we use $p(x) = \Gamma(\beta)x^{0.5}$ as the diffusion term and $w(x,t) = (x^2+1)cos(t+1) - 2xsin(t+1)$ as the source function. The remaining functions are let to be $q(x) = r(x) = 0$. This example will be solved subjected to the initial condition $u_0 = (x^2+1)sin(1)$ and the boundary conditions $g_0 = sin(t+1)$ and $g_1 = 2sin(t+1)$. The provided exact solution is $u(x,t) = (x^2+1)sin(t+1)$.

Example 2 [15]: For the second example, we consider the space-fractional diffusion equation

$$\frac{\partial}{\partial t}u(x,t) = \Gamma(1.2)x^{\beta}\frac{\partial^{\beta}}{\partial x^{\beta}}u(x,t) + 3x^2(2x-1)\varepsilon^{-t}, \tag{23}$$

with the initial condition $u_0 = x^2 - x^3$ and zero Dirichlet conditions. The exact solution of this example is given by $u(x,t) = x^2(1-x)\varepsilon^{-t}$.

For the comparison analysis, we observe the three criteria such as the number of iterations, the execution time (seconds), and maximum error.The numerical results from the implementation of Algorithm 1 and the developed algorithms for PGS and GS methods are recorded and tabulated in Tables 1and 2.

Table 1: Numerical results for the iterative methods using Example 1 at three different values of β

β	Criteria	Method	128	256	512	1024	2048
1.2	k	GS	57	72	151	328	1547
		PGS	36	67	129	278	608
		PSOR	34	60	68	80	92
	seconds	GS	1.19	7.23	58.11	492.56	1227.21
		PGS	1.09	5.45	41.43	472.35	1219.76
		PSOR	0.84	5.33	25.31	115.89	557.00
	max error	GS	2.37e-02	2.44e-02	2.47e-02	2.49e-02	2.50e-02
		PGS	2.37e-02	2.44e-02	2.47e-02	2.49e-02	2.50e-02
		PSOR	2.37e-02	2.37e-02	2.37e-02	2.37e-02	2.37e-02
1.5	k	GS	150	272	723	1935	8320
		PGS	104	225	566	1514	4052
		PSOR	80	211	331	477	679
	seconds	GS	3.77	27.00	276.20	945.20	4348.68
		PGS	2.83	21.61	182.83	898.29	4299.73
		PSOR	1.90	17.84	124.05	714.51	4259.31
	max error	GS	6.20e-04	5.69e-04	5.36e-04	5.13e-04	5.02e-04
		PGS	6.20e-04	5.69e-04	5.36e-04	5.13e-04	5.02e-04
		PSOR	6.20e-04	5.69e-04	5.36e-04	5.13e-04	5.02e-04
1.8	k	GS	473	1123	3659	11836	47322
		PGS	345	890	2635	11829	47289
		PSOR	246	806	1635	2937	5171
	seconds	GS	11.36	111.98	1398.43	4448.83	55209.81
		PGS	9.48	85.00	843.91	2138.11	8979.18
		PSOR	5.76	67.75	619.64	2099.87	8852.28
	max error	GS	3.99e-02	3.97e-02	3.96e-02	3.95e-02	3.93e-02
		PGS	3.99e-02	3.97e-02	3.96e-02	3.95e-02	3.93e-02
		PSOR	3.99e-02	3.97e-02	3.96e-02	3.95e-02	3.93e-02

Using five different values of mesh size which are $M = 128, 256, 512, 1024$, and 2048 and three different values of β which are $1.2, 1.5$, and 1.8, we found that the PSOR iterative method required the least number of iterations in solving the two tested examples among the three iterative methods. At $\beta = 1.2$, PSOR needs 18.69% to 74.30% lesser number of iterations than GS and 4.71% to 48.80% lesser number of iterations than PGS. While at $beta = 1.5$ and 1.8 respectively, PSOR needs 24.61% to 70.54% and 25.20% to 63.79% lesser number of iterations than GS. Then, when PSOR is compared against with PGS, PSOR needs 3.28% to 50.29% and 6.80% to 59.32% lesser number of iterations than PGS at $\beta = 1.5$ and 1.8 respectively.

In correspondence to this, the execution time of the developed algorithm for the PSOR iterative method is much faster than the PGS and GS iterative methods. At $\beta = 1.2$, the PSOR algorithm is faster than GS by 45.20% and faster than PGS by 28.96%, on average. Then, at $\beta = 1.5$, the PSOR algorithm is faster than GS and PGS by 35.49% and 13.76% respectively. Last but not least, for $\beta = 1.8$, the PSOR algorithm is faster than GS by 45.54% and it is faster by 12.22% when compared to PGS.

From this numerical finding, we show that through the transformation of an original linear system to a preconditioned linear system, the solution of the SFPDE can be computed numerically with better efficiency.Moreover, a further improvement in terms of efficiency can be made by the implementation of SOR iteration which enhances the rate of convergence of the solution. It is also important to mention that the accuracy of the three iterative methods (PSOR, PGS, and GS) shows a good agreement.

Table 2: Numerical results for the iterative methods using Example 2 at three different values of β

β	Criteria	Method	128	256	512	1024	2048
1.2	k	GS	55	67	142	280	738
		PGS	27	55	116	250	518
		PSOR	25	48	102	222	498
	seconds	GS	0.90	7.29	50.90	375.79	750.90
		PGS	0.72	4.72	37.86	322.55	413.21
		PSOR	0.50	2.88	30.90	310.79	395.90
	max error	GS	1.80e-01	1.84e-01	1.86e-01	1.89e-01	1.88e-01
		PGS	1.80e-01	1.84e-01	1.86e-01	1.89e-01	1.88e-01
		PSOR	1.80e-01	1.84e-01	1.86e-01	1.89e-01	1.88e-01
1.5	k	GS	92	224	838	2211	8098
		PGS	75	197	522	1435	4125
		PSOR	62	164	438	1391	4111
	seconds	GS	2.46	28.66	245.79	880.99	908.32
		PGS	1.83	17.11	170.92	443.81	713.64
		PSOR	1.66	14.66	163.79	432.99	688.32
	max error	GS	5.44e-02	5.59e-02	5.66e-02	5.69e-02	5.85e-02
		PGS	5.44e-02	5.58e-02	5.65e-02	5.69e-02	5.85e-02
		PSOR	5.44e-02	5.58e-02	5.65e-02	5.69e-02	5.85e-02
1.8	k	GS	250	842	2856	3801	6314
		PGS	213	686	2213	3452	5127
		PSOR	166	542	1756	2431	4914
	seconds	GS	9.49	95.40	930.62	999.74	4699.38
		PGS	5.27	59.48	737.50	820.62	3173.73
		PSOR	4.64	51.40	694.62	809.74	3167.38
	max error	GS	8.88e-04	4.09e-04	1.55e-04	1.49e-04	1.20e-04
		PGS	8.88e-04	4.09e-04	1.54e-04	1.49e-04	1.20e-04
		PSOR	8.88e-04	4.09e-04	1.54e-04	1.49e-04	1.20e-04

5 Conclusion

To conclude this paper, we presented the formulation of the Caputo's implicit finite difference approximation equations for the one-dimensional linear SFPDE of parabolic type. We show the formation of a linear system using the approximation equation and transformed it into a preconditioned linear system. The solution of the preconditioned linear system is then solved using the proposed PSOR iterative method. From the numerical finding and observation, the PSOR iterative method requires a lesser number of iterations and execution time compared with PGS and GS iterative methods. It is also found that the accuracy of the three iterative methods is in good agreement.

References

[1] M. M. Meerschaert and C. Tadjeran, Finite difference approximation for fractional advection-dispersion flow equations, *J. Comput. Appl. Math.* **172**(1), 145–155 (2004).
[2] M. Saqib, I. Khan and S. Shafie, Application of fractional differential equations to heat transfer in hybrid nanofluid: modeling and solution via integral transforms, *Adv. Differ. Equ.* **2019**, Article number 52 (2019).
[3] W. Jiang, Z. Q. Ding and Y. W. Liu, New image edge detection model based on fractional-order partial differention, *J. Comput. Appl.* **32**(10), 2848–2850 (2012).
[4] J. P. Aguilar, J. Korbel and Y. Luchko, Applications of the fractional diffusion equation to option pricing and risk calculations, *Math.* **7**(9), Article number 796 (2019).
[5] A. Chaves, Fractional diffusion equation to describe Levy flight, *Phys. Lett. A* **239** (1-2), 13–16 (1998).
[6] B. Shiri, G. C. Wu and D. Baleanu, Collocation methods for terminal value problems of tempered fractional differential equations, *Appl. Numer. Math.* **156**, 385–395 (2020).
[7] O. P. Agrawal, Solution for a fractional diffusion-wave equation defined in a bounded domain, *Nonlinear Dynam.* **29**, 145–155 (2002).
[8] J. Jia, X. Zheng, H. Fu, P. Dai and H. Wang, A fast method for variable-order space-fractional diffusion equations, *Numer. Algor.* **7**,1-17 (2020).

[9] S. B. Yuste and L. Acedo, An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations, *SIAM J. Numer. Anal.* **42**(5), 1862–1874 (2005).
[10] S. B. Yuste, Weighted average finite difference method for fractional diffusion equations, *J. Comp. Phys.* **216**, 264–274 (2006).
[11] Y. Zhang, A finite difference method for fractional partial differential equation, *Appl. Math. Comput.* **215**, 524–529 (2009).
[12] W. Hackbusch, *Iterative solution of large sparse systems of equations*, Springer-Verlag, New York, 1995.
[13] A. Li, The new preconditioned SOR method for M-matrices, in *2012 International Conference on Computer Science and Electronics Engineering*, pp. 3–5, IEEE, Hangzhou, 2012.
[14] Y. Saad, *Iterative methods for sparse linear systems*, International Thomas Publishing, Boston, 2003.
[15] A. Sunarto and J. Sulaiman, Preconditioned SOR method to solve time-fractional diffusion equations, *J. Phys. Conf. Ser.* **1179**, Article number 012020 (2019).
[16] N. H. Tuan, D. Baleanu, T. N. Thach, D. O'Regan and N. H. Can, Final value problem for nonlinear time fractional reaction–diffusion equation with discrete data, *J. Comput. Appl. Math.* **376**, Article number 112883 (2020).
[17] H. Fu and H. Wang, A preconditioned fast parareal finite difference method for space-time fractional partial differential equation, *J. Sci. Comput.* **78**(3), 1724–1743 (2018).
[18] D. J. Evans, Group explicit iterative methods for solving large linear systems, *Int. J. Comput. Math.* **17**(1), 81–108 (1985).
[19] A. Ibrahim and A. R. Abdullah, Solving the two dimensional diffusion equation by the four point explicit decoupled group (EDG) iterative method, *Int. J. Comput. Math.* **58**(3-4), 253–256 (1995).
[20] J. V. L. Chew and J. Sulaiman, On quarter-sweep fnite difference scheme for one-dimensional porous medium equations, *Int. J. Appl. Math.* **33**(3), 439–450 (2020).
[21] A. D. Gunawardena, S. K. Jain and L. Snyder, Modified iterative methods for consistent linear systems, *Linear Algebra Appl.* **154–156**, 123–143 (1991).
[22] L. A. Hageman and D. M. Young, Iterative solution of large linear systems, *Am. Math. Mon.* **80**, 92 (1971).