International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 7, No. 2, June, 2020, pp. 201-204
Journal homepage: http://iieta.org/journals/mmep

Performance Numerical Method Half-Sweep Preconditioned Gauss-Seidel for Solving Fractional Diffusion Equation

Andang Sunarto[1*], Jumat Sulaiman[2]

[1] IAIN Bengkulu, Jln Raden Fatah, Pagar Dewa Kota Bengkulu, Indonesia
[2] UMS Malaysia, Jl UMS Kota Kinabalu, Sabah, Malaysia

Corresponding Author Email: andang99@gmail.com

https://doi.org/10.18280/mmep.070205

Received: 17 January 2020
Accepted: 28 March 2020

Keywords:
HSPGS, space-fractional, Caputo's, implicit finite difference

ABSTRACT

The main purpose, we derive a finite difference approximation equation from the discretization of the one-dimensional linear space-fractional diffusion equations by using the space fractional derivative of Caputo's. The linear system will be generated by the Caputo's finite difference approximation equation. The resulting linear system was then resolved using Half-Sweep Preconditioned Gauss-Seidel (HSPGS) iterative method, which compares its effectiveness with the existing Preconditioned Gauss-Seidel (PGS) or call named (Full-Sweep Preconditioned Gauss-Seidel (FSPGS)) and Gauss-Seidel (HSPGS) methods. Two examples of the issue are provided in order to check the performance efficacy of the proposed approach. The findings of this study show that the proposed iterative method is superior to FSPGS and GS.

1. INTRODUCTION

From The previous studies in [1-5] many successful mathematical models, which are based on fractional partial derivative equations (FPDEs), have been developed. Following to that, there are several methods used to solve these models. For instance, we have transform method [6], which is used to obtain analytical and/or numerical solutions of the fractional diffusion equations (FDE). Other than this method, other researchers have proposed finite difference methods such as explicit, implicit and fast method [7-9] Also it is pointed out that the explicit methods are conditionally stable. Therefore, we discretize the space-fractional diffusion equation via the implicit finite difference discretization scheme and Caputo's fractional partial derivative of order β in order to derive a Caputo's implicit finite difference approximation equation.

This approximation equation leads a tridiagonal linear system. Due to the properties of the coefficient matrix of the linear system which is sparse and large scale, iterative methods are the alternative option for efficient solutions. Among the existing iterative methods, the preconditioned iterative methods [10-12] have been widely accepted to be one of the efficient methods for solving linear systems.

Because of the advantages of these iterative methods, the aim of this paper is to construct and investigate the performance effectiveness of the Half-Sweep Preconditioned Gauss-Seidel (HSPGS) iterative method for solving space-fractional parabolic partial differential equations (SPPDE's) based on the Caputo's implicit finite difference approximation equation. To investigate the effectiveness of the HSPGS method, we also implement the Gauss Seidel (GS) and FSPGS iterative methods being used a control method.

To performance the effectiveness of HSPGS method, let space-fractional parabolic partial differential equation (SPPDE's) be defined as:

$$\frac{\partial U(x,t)}{\partial t} = a(x)\frac{\partial^{\beta} U(x,t)}{\partial x^{\beta}} + b(x)\frac{\partial U(x,t)}{\partial x} + c(x)U(x,t) + f(x,t) \quad (1)$$

with initial condition $U(x,0) = f(x),\ 0 \le x \le \ell$, and boundary conditions $U(0,t) = g_0(t),\ 0 < t \le T,\ \ U(\ell,t) = g_1(t),\ 0 < t \le T$.

Then to develop the linear systems, some definitions that can be applied for fractional derivative theory need to developing the approximation equation of Eq. (1) in:

Definition 1. [13] The Riemann-Liouville fractional integral operator, J^{β} of order-β is defined as:

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

Definition 2. [13] The Caputo's fractional partial derivative operator, D^{β} of order - β is defined as:

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

with $m-1 < \beta \le m,\ m \in N,\ x > 0$.

We have the following properties when $m-1 < \beta \le m$, $x > 0$: $D^{\beta}{}_k = 0$, (k is a constant),

$$D^{\beta} x^n = \begin{cases} 0, & \text{for } n \in N_0 \text{ and } n < [\beta] \\ \dfrac{\Gamma(n+1)}{\Gamma(n+1-\beta)} x^{n-\beta}, & \text{for } n \in N_0 \text{ and } n \ge [\beta] \end{cases}$$

where, function $\lceil\beta\rceil$ denotes the smallest integer greater than or equal to β, $N_0=\{0,1,2,\ldots\}$ and $\Gamma(.)$ is the gamma function.

2. CAPUTO APPROXIMATION DERIVATIVE

Assume that $h=\frac{\ell}{k}$, k is positive integer and using second order approximation, we get

$$\frac{\partial^{\beta}U(x_i,t_n)}{\partial x^{\beta}}=\frac{1}{\Gamma(2-\beta)}\int_0^{t_n}\frac{\partial^2 U(x_i,s)}{\partial x^2}(t_n-s)^{1-\beta}\partial s$$
$$=\frac{1}{\Gamma(2-\beta)}\sum_{j=0,2,4..}^{i-2}\int_{jh}^{(j+1)h}\left(\frac{U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}}{2h^2}\right)(nh-s)^{\beta}\partial s \tag{4}$$

Let us define

$$\sigma_{\beta,h}=\frac{2h^{-\beta}}{\Gamma(3-\beta)} \quad \text{and} \quad g_j^{\beta}=\left(\frac{j}{2}+1\right)^{2-\beta}-\frac{j}{2}^{2-\beta}$$

then the discrete approximation of Eq. (4).

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

Now we approximate Eq. (1) by using Caputo's implicit finite difference approximation:

$$\lambda\left(U_{i,n}-U_{i,n-2}\right)=a_i\sigma_{\beta,h}\sum_{j=0}^{i-1}g_j^{\beta}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right)$$
$$+b_i\frac{\left(U_{i+2,n}-U_{i-2,n}\right)}{4h}+C_iU_{i,n}+f_{i,n}$$

for *i=2,4,...,m-2*. Then we can simplify the scheme approximation equation as:

$$\lambda U_{i,n-2}=-a_i\sigma_{\beta,2h}\sum_{j=0,2,4.}^{i-2}g_j^{\beta}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right)$$
$$-\frac{b_i}{4h}\left(U_{i+2,n}-U_{i-2,n}\right)-C_iU_{i,n}+\lambda U_{i,n}-f_{i,n}$$

So, we get:

$$\therefore b_i^*U_{i-2,n}+\left(\lambda-c_i^*\right)U_{i,n}-b_i^*U_{i+2,n}$$
$$-a_i^*\sum_{j=0,2,4}^{i-2}g_j^{\beta}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right)=f_i \tag{5}$$

where, $a_i^*=a_i\sigma_{\beta,2h}$, $b_i^*=\frac{b_i}{4h}$, $c_i^*=c_i$, $F_i^*=f_{i,n}$ and $f_i=\lambda\left(U_{i,n-2}\right)+F_i^*$.

For simplicity, let Eq. (5) for $n>3$ be rewritten as:

$$-R_i+\alpha_iU_{i-6,n}+s_iU_{i-4}+p_iU_{i-2,n}+q_iU_{i,n}+r_iU_{i+2,n}=f_i \tag{6}$$

where,

$$R_i=a_i^*\sum_{j=6}^{i-2}g_j^{\beta}\left(U_{i-j+2,n}-2U_{i-j,n}+U_{i-j-2,n}\right),\ \alpha_i=\left(-a_i^*g_2^{\beta}\right),$$
$$s_i=\left(-a_i^*g_1^{\beta}+2a_i^*g_2^{\beta}\right),$$
$$p_i=\left(b_i^*-a_i^*g_2^{\beta}+2a_i^*g_1^{\beta}-a_i^*\right),$$
$$q_i=\left(-a_i^*g_1^{\beta}+2a_i^*+\left(\lambda-c_i^*\right)\right),\ r_i=\left(-a_i^*-b_i^*\right)$$

Then Eq. (6) can be used to construct a linear system in matrix form as:

$$A\underset{\sim}{U}=\underset{\sim}{f} \tag{7}$$

where,

$$A=\begin{bmatrix} q_2 & r_2 & & & & & & \\ p_4 & q_4 & r_4 & & & & & \\ s_6 & p_6 & q_6 & r_6 & & & & \\ \alpha_8 & s_8 & p_8 & q_8 & r_8 & & & \\ & \alpha_{10} & s_{10} & p_{10} & q_{10} & r_{10} & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & \alpha_{m-4} & s_{m-4} & p_{m-4} & q_{m-4} & r_{m-4} \\ & & & & \alpha_{m-2} & s_{m-2} & p_{m-2} & q_{m-2} \end{bmatrix}_{(m-2)x(m-2)}$$

$$\underset{\sim}{U}=\left[U_{2,1}\quad U_{4,1}\quad U_{6,1}\quad\cdots\quad U_{m-4,1}\quad U_{m-2,1}\right]^T,$$

$$\underset{\sim}{f}=\left[f_2-p_2U_{2,1}\quad f_4+s_4U_{41}\quad f_6+\alpha_6U_{6,1}\quad f_8+R_i\quad\cdots\quad f_{m-4,1}+R_{m-4}\quad f_{m-2,1}-p_{m-2}U_{m,1}+R_{m-2}\right]^T$$

3. HSPGS METHODS

Before applying the HSPGS iterative method, we need to transform the original linear system (7) into the preconditioned linear system.

$$A^*\underset{\sim}{x}=\underset{\sim}{f}^* \tag{8}$$

where,

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

Actually, the matrix *P* is called a preconditioned matrix and defined as [14-16] $P=I+S$.
where,

$$S=\begin{bmatrix} 0 & -r_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -r_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -r_3 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & -r_{m-1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{(m-1)x(m-1)}$$

and the matrix I is an identical matrix. To formulate HSPGS method, let the coefficient matrix A^* in (7) be expressed as summation of the three matrices

$$A^*=D-L-V \tag{9}$$

where, *D*, *L* and *V* are diagonal, lower triangular and upper triangular matrices respectively. By using Eq. (9) and (11), the formulation of HSPGS iterative method can be defined generally as [11, 17, 18]:

$$\underset{\sim}{x}^{(k+1)} = (D-L)^{-1} V \underset{\sim}{x}^{(k)} + (D-L)^{-1} \underset{\sim}{f}^{*} \qquad (10)$$

where, $\underset{\sim}{x}^{(k+1)}$ represents an unknown vector at *(k+1)*th iteration.

The implementation of the HSPGS iterative method can be described in Algorithm 1.

Algorithm 1: HSPGS method

i. Initialize $\tilde{U} \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$.
ii. For $j = 1,2,\ldots,n$ Implement
For $i = 1,2,\ldots,m-1$ calculate
$\underset{\sim}{x}^{(k+1)} = (D-L)^{-1} V \underset{\sim}{x}^{(k)} + (D-L)^{-1} \underset{\sim}{f}^{*}$
$\underset{\sim}{U}^{(k+1)} = P^T \underset{\sim}{x}^{(k+1)}$
Convergence test. If the convergence criterion i.e $\left\| \underset{\sim}{U}^{(k+1)} - \underset{\sim}{U}^{(k)} \right\| \le \varepsilon = 10^{-10}$ is satisfied, go to Step (iii). Otherwise go back to Step (ii).
iii Display approximate solutions.

4. INDING NUMERICAL

We have examples of the SFPDE's to verify the effectiveness of the HSPGS methods. In comparison, three criteria such as number iterations, the execution time (seconds) and maximum error at three different values of $\beta = 1.2$, $\beta = 1.5$ and $\beta = 1.8$ .During the implementation of the point iterations, the convergence test considered the tolerance error, $\varepsilon = 10^{-10}$.

Example 1 [19]:
Let us consider the following space-fractional initial boundary value problem

$$\frac{\partial U(x,t)}{\partial t} = d(x)\frac{\partial^{\beta} U(x,t)}{\partial x^{\beta}} + p(x,t), \qquad (11)$$

Example 2 [19] :
Let us consider the following space-fractional initial boundary value problem

$$\frac{\partial U(x,t)}{\partial t} = \Gamma(1.2)x^{\beta}\frac{\partial^{\beta} U(x,t)}{\partial x^{\beta}} + 3x^{2}(2x-1)e^{-t}, \qquad (12)$$

All numerical results for Eqns. (11) and (12), obtained from application of GS, FSPGS and HSPGS iterative methods are recorded in Table 1 and 2 by using the different value of mesh size, M=128, 256, 512, 1024 and 2048.

Table 1. Comparison between number of iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example at $\beta = 1.2, 1.5, 1.8$

M	Method	$\beta = 1.2$			$\beta = 1.5$			$\beta = 1.8$		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	FSPGS	36	1.09	2.37e-02	104	2.83	6.20e-04	345	9.48	3.99e-02
	HSPGS	19	0.26	2.37e-02	42	1.24	6.20e-04	108	3.25	4.60e-02
256	FSPGS	72	7.23	2.44e-02	272	27.00	5.69e-04	1123	111.98	3.97e-02
	HSPGS	36	2.50	2.44e-02	104	11.33	5.69e-04	345	44.05	4.59e-02
512	FSPGS	151	58.11	2.47e-02	723	276.20	5.36e-04	3659	1398.43	3.96e-02
	HSPGS	72	23.35	2.47e-02	272	124.86	5.36e-04	1123	478.23	4.55e-02
1024	FSPGS	328	492.56	2.49e-02	1935	945.20	5.13e-04	11836	2138.11	3.95e-02
	HSPGS	151	193.63	2.49e-02	724	473.13	5.13e-04	3657	1054.31	4.53e-02
2048	FSPGS	1547	1227.21	2.50e-02	8320	4348.68	5.02e-04	47322	8979.18	3.93e-02
	HSPGS	327	472.53	2.50e-02	1938	3120.96	5.02e-04	22152	4335.75	4.51e-02

Table 2. Comparison between number of iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example at $\beta = 1.2, 1.5, 1.8$

M	Method	$\beta = 1.2$			$\beta = 1.5$			$\beta = 1.8$		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	FSPGS	27	0.72	1.80e-01	75	1.83	5.44e-02	213	5.27	8.88e-04
	HSPGS	15	0.25	1.80e-01	30	0.54	5.44e-02	67	2.94	8.88e-04
256	FSPGS	55	4.72	1.84e-01	197	17.11	5.58e-02	686	59.48	4.09e-04
	HSPGS	27	1.38	1.84e-01	75	7.83	5.58e-02	213	20.45	4.09e-04
512	FSPGS	116	37.86	1.86e-01	522	170.92	5.65e-02	2213	737.50	1.54e-04
	HSPGS	55	10.51	1.86e-01	197	77.58	5.65e-02	686	331.95	1.54e-04
1024	FSPGS	250	322.55	1.89e-01	1435	443.81	5.69e-02	3452	820.62	1.49e-04
	HSPGS	116	147.81	1.89e-01	522	299.59	5.69e-02	1224	411.91	1.49e-04
2048	FSPGS	518	413.21	1.88e-01	4125	713.64	5.85e-02	5127	3173.73	1.20e-04
	HSPGS	251	207.81	1.88e-01	1437	311.27	5.85e-02	2253	1062.72	1.20e-04

5. DISCUSSION AND CONCLUSION

In order to get the numerical solution of the space-fractional diffusion problems, the paper presents the derivation of the Caputo's implicit finite difference approximation equations in which this approximation equation leads a linear system. From observation of all experimental results by imposing the GS, FSPGS and HSPGS iterative methods, it is obvious at $\beta = 1.2$ that number of iterations have declined approximately by 41.30-82.45% corresponds to the HSPGS iterative method compared with the GS and FSPGS method. Again, in terms of execution time, implementations of HSPGS method are much faster about 51.18-92.43% than the GS and FSPGS method. It means that the HSPGS method requires the least amount for number of iterations and computational time at $\beta = 1.2$ as compared with GS and FSPGS iterative methods. Based on the accuracy of both iterative methods, it can be concluded that their performance numerical solutions are in good agreement.

REFERENCES

[1] Fallahgoul, H.A., Focardi, S.M., Fabozzi, F.J. (2017). Fractional partial differential equation and option pricing. In Fractional Calculus and Fractional Processes with Applications to Financial Economics, pp. 59-80. https://doi.org/10.1016/b978-0-12-804248-9.50006-1

[2] KAY, E. (1968). The mathematics of the model. In A Mathematical Model for Handling in a Warehouse, pp. 58–68. https://doi.org/10.1016/b978-0-08-103792-8.50014-0

[3] Menni, Y., Chamkha, A.J., Lorenzini, G., Ameur, H., Salmi, M., Fridja, D. (2020). Numerical simulation of dynamic pressure and kinetic energy fields of turbulent oil flow in staggered baffled pipes. Mathematical Modelling of Engineering Problems, 7(1): 10-16. https://doi.org/10.18280/mmep.070102

[4] Salehizadeh, M.R., Nouri, H. (2020). Circuit modelling by difference equation: Pedagogical advantages and perspectives. Mathematical Modelling of Engineering Problems, 7(1): 26–30. https://doi.org/10.18280/mmep.070104

[5] Sweilam, N.H., Abou Hasan, M.M., Baleanu, D. (2017). New studies for general fractional financial models of awareness and trial advertising decisions. Chaos, Solitons and Fractals, 104. https://doi.org/10.1016/j.chaos.2017.09.013

[6] Chaves, A.S. (1998). A fractional diffusion equation to describe Lévy flights. Physics Letters A, 239(1–2): 13-16. https://doi.org/10.1016/s0375-9601(97)00947-x

[7] Agrawal, O.P. (2002). Solution for a fractional diffusion-wave equation defined in a bounded domain. Nonlinear Dynamics, 29(1–4): 145–155. https://doi.org/10.1023/A:1016539022492

[8] Scherer, R., Kalla, S.L., Tang, Y., Huang, J. (2011). The Grnwald-Letnikov method for fractional differential equations. Computers and Mathematics with Applications, 62(3): 902–917. https://doi.org/10.1016/j.camwa.2011.03.054

[9] Zheng, Q.Q., Lu, L.Z. (2019). On parameterized matrix splitting preconditioner for the saddle point problems. International Journal of Computer Mathematics. https://doi.org/10.1080/00207160.2017.1420179

[10] Bo, L.Y., Yang, J. (2012). A new preconditioned Gauss-Seidel iterative method and its convergence. Zhongbei Daxue Xuebao (Ziran Kexue Ban)/Journal of North University of China (Natural Science Edition), 33(1). https://doi.org/10.3969/j.issn.1673-3193.2012.01.017

[11] Gunawardena, A.D., Jain, S.K., Snyder, L. (1991). Modified iterative methods for consistent linear systems. Linear Algebra and Its Applications, 154–156(C): 123–143. https://doi.org/10.1016/0024-3795(91)90376-8

[12] Saad, Y. (2003). Preconditioned Iterations. In Iterative Methods for Sparse Linear Systems, pp. 261-281. https://doi.org/10.1137/1.9780898718003.ch9

[13] Young, D. (1971). Iterative solution of large linear systems.

[14] Kohno, T., Kotakemori, H., Niki, H., Usui, M. (1997). Improving the modified Gauss-Seidel method for Z-matrices. Linear Algebra and Its Applications, 267: 113-123. https://doi.org/10.1016/s0024-3795(97)80045-6

[15] Morimoto, M. (2010). Study on the preconditioners (I + Sm). Journal of Computational and Applied Mathematics, 234(1): 209-214. https://doi.org/10.1016/j.cam.2009.12.016

[16] Morimoto, M., Kotakemori, H., Kohno, T., Niki, H. (2003). The Gauss-Seidel method with preconditioner (I+R). Transactions of the Japan Society for Industrial and Applied Mathematics, 13(4): 439–445. https://doi.org/10.11540/jsiamt.13.4_439

[17] Honghao, H., Dongjin, Y., Yi, H., Jinqiu, X. (2009). Preconditioned gauss-seidel iterative method for linear systems. In Proceedings - 2009 International Forum on Information Technology and Applications, IFITA 2009, 1: 382-385. https://doi.org/10.1109/IFITA.2009.339

[18] Shen, H., Shao, X., Huang, Z., Li, C. (2011). Preconditioned Gauss-Seidel iterative method for Z-matrices linear systems. Bulletin of the Korean Mathematical Society, 48(2): 303-314. https://doi.org/10.4134/BKMS.2011.48.2.303

[19] Sunarto, A., Sulaiman, J., Saudi, A. (2015). Numerical solution for solving space-fractional diffusion equations using half-sweep gauss-seidel iterative method. In International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2: 20.