Investigation of Fractional Diffusion Equation via QSGS iterations

A Sunarto\(^1\), J. Sulaiman\(^2\)

\(^1\)Faculty of Economic and Islamic Business, IAIN Bengkulu, Indonesia.
\(^2\)Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia.

Abstract. We investigate the application of Quarter-Sweep Gauss-Seidel (QSGS) iterative method for solving (SFPDE’s) space-fractional partial diffusion equations with Dirichlet boundary condition. To do this, implicit finite difference scheme and Caputo’s derivative operator are used to discretize one-dimensional linear space-fractional equation to form system of linear equations. Then basic ideas formulation and application of the suggested iterative method are also introduced. Numerical examples of tested problems were carried out to demonstrate advantages of the proposed iterative method beside to the HSGS and FSGS as control method. Based on computational results, the QSGS method is shown to be the most superior than the HSGS and FSGS iterative methods.

1. Introduction
Space-fractional partial diffusion equations (SFPDE’s) for first-order linear can be represented mathematically as,

\[
\frac{\partial Y(x,t)}{\partial t} = a(x)\frac{\partial^\beta Y(x,t)}{\partial x^\beta} + b(x)\frac{\partial Y(x,t)}{\partial x} + c(x)Y(x,t) + f(x,t)
\]

with initial and the Dirichlet boundary condition given as

\[
Y(x,0) = f(x), \quad 0 < x < \ell, \quad Y(0,t) = g_0(t), \quad Y(\ell,t) = g_1(t), \quad 0 < t \leq T.
\]

Many natural phenomena in physic, engineering and other sciences can be presented very successfully by mathematical models using fractional calculus [1,2,3]. Space-fractional diffusion equations are solved numerically by implementing fractional derivative and finite difference schemes in which both schemes are used to discretize in the problem in Eq.(1) to generate the system of linear equations. Recent studies showed that the solution of problem in Eq.(1) can be obtained via fast solution method [1], compact finite difference approximation method [2], a fast second-order finite difference method [3] and modified decomposition method [4]. By considering the Caputo’s fractional derivatives and implicit Caputo’s finite difference schemes over the problem in Eq.(1), the corresponding approximation equation can constructed and used to generate linear system. As a result calculation of the numerical experiments can be prohibitively expensive for solving the linear system of order n. This is due to the Caputo’s operator fractional derivative and implicit Caputo’s finite difference schemes based on the full-sweep iterative concept. To evaulation the computational complexity of linear
systems, a discretization scheme namely Quarter-Sweep (QS) implicit Caputo’s finite difference scheme has been applied to discretize Eq.(1) to create a Quarter-Sweep (QS) system of linear diffusion equations. The concept of QS iterative method has been initiated by Othman and Abdullah [5,6]. Actually, the concept of this method is extension of the Half-Sweep (HS) iterative method, which is inspired by Abdullah [7] via the Explicit Decoupled Group (EDG) iterative method for solving the two-dimensional Poisson equation. Due to advantages of half-sweep (HS) iteration, many studies have been conducted [8,9]. Apart from the half-sweep (HS) iterations, the study of the effectiveness QS iterative methods have also been carried out by many researches [10,11,12].

However this research, is to evaluation the applications of the QSGS iteration which has been applied for solving problem in Eq.(1) based on QS implicit Caputo’s finite difference approximation equation. Actually, this QSGS method is a combination of quarter-sweep iteration and Gauss-Seidel Method. For comparison purpose, we implement the Full-Sweep Gauss-Seidel (FSGS) and Half-Sweep Gauss-Seidel (HSGS) iterative methods being used as a control method.

In the following statement, we give some basic definitions and properties of the fractional derivative theories which are used in this paper:

**Definition 1.**[13,14]The Riemann-Liouville operator, is defined as

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

**Definition 2.**[14,15] The Caputo’s operator, is defined as

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

with \( k - 1 < \beta \leq k \), \( k \in \mathbb{N} \), \( x > 0 \)

We have the following properties when \( k - 1 < \beta \leq k \), \( x > 0 \):

\[ D^\beta_p = 0, \; (p \text{ is a constant}), \; D^\beta x^n = \begin{cases} 0, & \text{for } n \in \mathbb{N}_0 \text{ and } n < \lfloor \beta \rfloor \\ \frac{\Gamma(n+1)}{\Gamma(n+1-\beta)} x^{n-\beta}, & \text{for } n \in \mathbb{N}_0 \text{ and } n \geq \lfloor \beta \rfloor \end{cases} \]

with function \( \lfloor \beta \rfloor \) to show the smallest integer \( \geq \beta \), \( \mathbb{N}_0 = \{0,1,2,...\} \) and \( \Gamma() \) is the gamma function.

### 2. Derivation of Quarter-Sweep (QS) Caputo’s Approximation Equations

In the section, we derive a numerical approximation to the fractional derivative term problem in Eq.(1). Consider the Caputo’s derivative operator and assume \( h = \frac{\ell}{\phi} \), with \( \phi \) is positive integer. With

Apply the second-order finite difference scheme, we derive the second-order QS implicit Caputo’s finite difference scheme as follows:

\[ \frac{\partial^\beta Y(x_i,t_n)}{\partial x^\beta} = \frac{(4h)^\beta}{\Gamma(3-\beta)} \sum_{j=0,4,8}^4 \left( Y_{i-j+4,n} - 2Y_{i-j,n} + \frac{\sigma_{j,4h}}{4} \left( \frac{j}{4} + 1 \right)^{2-\beta} - \frac{j^{2-\beta}}{4} \right) \] (4)

with \( \sigma_{j,4h} = \frac{(4h)^\beta}{\Gamma(3-\beta)} \) \( g_j^\beta = \left( \frac{j}{4} + 1 \right)^{2-\beta} - \frac{j^{2-\beta}}{4} \).

Therefore we obtain Eq.(5) by simplifying the discrete approximation of Eq.(4) as follows

\[ \frac{\partial^\beta Y(x_i,t_n)}{\partial x^\beta} = \sigma_{j,4h} \sum_{j=0,4,8}^4 g_j^\beta \left( Y_{i-j+4,n} - 2Y_{i-j,n} + Y_{i-j-4,n} \right) \] (5)
With apply Eq.(5) and QS implicit Caputo’s finite difference scheme, we approximate the problem in Eq.(1) in order to derive the QS implicit Caputo’s finite difference approximation equation as follows

\[ a_i \sigma \frac{1}{h^2} \sum_{j=0,4,8} g_j \left( Y_{j-4n} - 2Y_{j-4n}^* + Y_{j+4n} - 2Y_{j+4n}^* \right) + b_i \frac{1}{8h} \left( Y_{i-4n} - Y_{i+4n} \right) - C_i Y_{i,n} + f_{i,n} \]  

for \( i = 4, 8, \ldots, m-4 \). Again based on the approximation equation (6), we have

\[ b_i^* Y_{i-4n} + \left( \lambda - c_i^* \right) Y_{i,n} - b_i^* Y_{i+4n} - a_i \sum_{j=0,4,8} g_j^* \left( Y_{j-4n} - 2Y_{j-4n}^* + Y_{j+4n} - 2Y_{j+4n}^* \right) = f_i^* \]  

with \( a_i^* = a_i \sigma \frac{1}{h^2} \), \( b_i^* = b_i \frac{1}{8h} \), \( c_i^* = c_i \), \( F_i^* = f_{i,n} \), \( f_i = \lambda \left( Y_{i-4n} - Y_{i+4n} \right) + F_i^* \).

Let us notice the approximation equation (8) being rewritten in the following form

\[ -R_i + \alpha_i Y_{i+12n} + s_i Y_{i+8} + p_i Y_{i+4n} + q_i Y_{i,n} + r_i Y_{i+4n} = f_i \]  

with \( R_i = a_i \sum_{j=12}^{4} g_j^* \left( Y_{j+i} - 2Y_{j+i}^* + Y_{j+i+4n} - 2Y_{j+i+4n}^* \right) \), \( \alpha_i = \left( -a_i^* g_j^* \right) \), \( s_i = \left( -a_i^* g_j^* + 2a_i^* g_j^* - a_i^* \right) \), \( p_i = \left( b_i^* - a_i^* g_j^* + 2a_i^* g_j^* - a_i^* \right) \), \( q_i = \left( -a_i^* g_j^* + 2a_i^* + \left( \lambda - c_i^* \right) \right) \), \( r_i = \left( -a_i^* - b_i^* \right) \).

By applying Eq.(9) into all interior points of the solution domain problem in Eq (1), the linear system to be expressed in matrix form as

\[ A \ Y = f \]  

with

\[ A = \begin{bmatrix} q_1 & r_1 \\ p_8 & q_8 & r_8 \\ s_{12} & p_{12} & q_{12} & r_{12} \\ \alpha_{16} & s_{16} & p_{16} & q_{16} & r_{16} \\ & \alpha_{20} & s_{20} & p_{20} & q_{20} & r_{20} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m-4} & s_{m-4} & p_{m-4} & q_{m-4} & r_{m-4} & (m-1)x(m-4) \end{bmatrix}, \]

\[ Y = \begin{bmatrix} Y_{4,1} \\ Y_{8,1} \\ Y_{12,1} \\ \vdots \\ Y_{m-4,1} \\ Y_{m-2,1} \end{bmatrix}^T, \]

\[ f = \begin{bmatrix} f_1^* + p_4 Y_{4,1} + f_8 + q_8 Y_{8,1} + f_{12} + \alpha_2 Y_{12,1} + f_{16} + R_{16} + f_{m-8} + R_{m-8} + f_{m-4} + p_{m-4} Y_{m-4} + R_{m-4} \end{bmatrix}^T. \]

### 3. Formulation of QSGS Methods

Such in the up section, the coefficient matrix (A) of linear system in equations (10) is sparse and large. So three proposed iterative methods are natural option for efficient solution of sparse linear system. To develop the formulation of QSGS iterative method being applied for solving linear system (10) explain in this section and to solve the formulation for QSGS method, Coefficient matrix (A) can be decomposed into

\[ A = Di - Lo - Va \]  

with \( Di, Lo \) and \( Va \) are diagonal, lower triangular and upper triangular matrices respectively. Thus, QSGS iterative method can be defined generally as [11,12]

\[ Y^{(i+1)} = \left( Di - Lo \right)^{-1} \left( Va Y^{(i)} + f \right) \]
Actually, this methods attempts to obtain a solution to the system of linear diffusion equations by repeatedly solving the linear system using approximations to the vector $\mathbf{Y}$. The implementation of the QSGS iterative method can be described in QSGS Algorithm.

**QSGS Algorithm:**

i. First $\mathbf{Y} \leftarrow 0$ and $\epsilon \leftarrow \left(10^{-10}\right)$

ii. Start $i=4,8,\ldots,m-4$, $j=0,4,\ldots,n-2$ assign

$$\mathbf{Y}^{(k+1)} = (D_l - L_l)^{-1} \left(V \mathbf{Y}^{(k)} + f\right)$$

iii. Next convergence test.

If the convergence criterion i.e.

$$\|\mathbf{Y}^{(k+1)} - \mathbf{Y}^{(k)}\| \leq \epsilon = 10^{-10}$$

is satisfied, step into (iv)

If not go back to step (ii)

iv. Display approximate Caputo’s solutions

### 4. Numerical Investigation

To investigation the performance of the three proposed iterative methods, three parameters have been considered such as $K$ (number of iterations), Time (in second) and Max Error (maximum error) at three different values ($\beta = 1.2$, $\beta = 1.5$ and $\beta = 1.8$). For performance comparison, the FSGS and HSGS methods were used as control results. In this paper, the simulations were carried out on mesh sizes as 128, 256, 512, 1024 and 2048 with tolerance error ($\epsilon = 10^{-10}$). The results of numerical simulations for problems (13,14) recorded in Tables 1 and 2 respectively.

**Examples 1** : [15]

Consider the space-fractional partial diffusion equation (SFPDE’s) initial boundary value problem

$$\partial_{t}^{\alpha} Y(x,t) = d(x) \frac{\partial^{\beta} Y(x,t)}{\partial x^{\beta}} + p(x,t),$$

with initial condition is $Y(x,0) = (x^2 + 1)\sin(1)$ and the boundary conditions is $Y(0,t) = \sin(t+1)$, $Y(1,t) = 2\sin(t+1)$, for $t>0$.

**Examples 2** : [15]

Consider the space-fractional partial diffusion equations (SFPDE’s) initial boundary value problem

$$\Gamma(1.2) x^{\beta} \frac{\partial^{\beta} Y(x,t)}{\partial x^{1.8}} + 3x^2 (2x - 1)e^{-t},$$

with the initial condition $Y(x,0) = x^2 - x^3$, and zero Dirichlet conditions. The exact solution of problem in Eq.(14) is $Y(x,t) = x^2 (1-x)e^{-t}$.

Based on Tables 1,2 obtained that $K$ (number of iterations), Time (second) for each mesh size as 128, 256, 512, 1024 and 2048 significantly decreased by applying QS iteration concept with acceptable precisions

### 5. Conclusion

In this research, we investigation the application of QSGS method for solving space-fractional partial diffusion equation (SFPDE’s). From the results in Tables 1,2 presented, it can be explained that QSGS method converges faster than the other two methods (HSGS and FSGS) by having less $K$ (number of iterations) and computational time then with HSGS method In addition, QSGS manages to retain the
accuracy of FSGS (standard Gauss-Seidel method). Overall, it be able summarized that to QSGS is the most outstanding method, with a good accuracy and obvious improvement in $K$ (number of iterations) and $T$ (time) and it has proved to be an efficient quarter-sweep method among the three.

6. References

[1] F. Mainardi. Fractal and fractional calculus continuum mechanics. Heidelberg: Springer-Verlag.,1997,291-348
[2] H. Zhou, W.Y. Tian and W. Deng. Compact finite difference approximation for space-fractional diffusion equations. J. of Scientific Computing., 2013, 56, 45-66
[3] T.S. Basu and H. Wang. A fast second-order finite difference method for space-fractional diffusion equations. Int. J. of Numerical Analysis and Modelling.,2012, 9, 658-666
[4] S.S. Ray, K.S. Chaudhuri and R.K. Bera. Application of modified decomposition method for the analytical solution of space-fractional diffusion equation. Applied Mathematics and Computation., 2008, 196, 294-302
[5] M. Othman, A.R. Abdullah. An efficient multigrid poisson solver. Int. J. of Computer Mathematics., 1998,71, 541-553
[6] M. Othman, A.R. Abdullah. An efficient four points modified explicit group poisson solver. Int. J. Computer Mathematics.,2000,76, 203-217
[7] A.R. Abdullah. The four point explicit decoupled group (edg) method: a fast poisson solver. Int. J. Computer Mathematics.,1991,38,61-70
[8] E. Aruchunan, J. Sulaiman. Application of the central-difference with half-sweep gauss-seidel method for solving first order linear fredholm integro-differential equations. World Academy of Science, Engineering and Technology., 2012, 68, 335-339
[9] E. Aruchunan, J. Sulaiman. Comparison of closed repeated newton-cotes quadrature scheme with half-sweep Gauss-Seidel iteration concept in solving linear fredholm integro-differential equations. Int. J. of Science and Engineering Investigations., 2012, 1, 90-96
[10] A. Sunarto, J. Sulaiman and A. Saudi. Numerical solution of the time-fractional diffusion equations by using quarter-sweep sor iterative method. Int.J.of Mathematical Engineering and Science (IJMES)., 2015, 5, 54-67
[11] A. Sunarto, J. Sulaiman and A. Saudi. Approximation solution for the time-fractional diffusion equations using quarter-sweep Gauss-Seidel method. Proceeding The 1st International Conference on Science and Technology for Sustainability, Batam Indonesia., 2014, 120-128
[12] W.S Koh, J. Sulaiman. Quarter-sweep gauss-seidel method using crank-nicolson approach for european put option pricing. Proceeding of The 5th Asian Mathematical Conference., 2007, 261-267
[13] Y. Zhang. A finite difference method for fractional partial differential equation. Applied Mathematics and Computation., 2009, 215,524-529
[14] C. Li, D. Qian, Y.Q. Chen. On Riemann-Liouville and caputo derivatives. Hindawi Publishing Corporation Discrete Dynamics in Nature and Science.,2011, 1, 1-15
[15] H. Azizi, G.B. Loghmani. Numerical approximation for space-fractional diffusion equations via Chebyshev finite difference method. J. of Fractional and Applications., 2013,4, 303-311
Table 1. Comparison between number of iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example 1 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     |
| 128 | FSGS 4 | 1.42          | 1.80e-01      | 182           | 4.41          | 5.44e-02      | 569           | 13.70         | 8.88e-04      |
|     | FSGS 4 | 0.38          | 5.07e-02      | 83            | 0.82          | 1.13e-02      | 233           | 2.27          | 8.88e-04      |
|     | QSGS 17 | 0.27         | 1.59e-02      | 30            | 0.18          | 4.66e-02      | 57            | 0.20          | 3.29e-04      |
| 256 | FSGS 117 | 10.95        | 1.84e-01      | 481           | 45.32         | 5.58e-02      | 931           | 174.77        | 4.09e-04      |
|     | FSGS 63 | 2.50          | 5.28e-02      | 211           | 8.04          | 1.23e-02      | 746           | 28.91         | 4.09e-04      |
|     | QSGS 30 | 0.38          | 1.73e-02      | 71            | 0.54          | 5.12e-02      | 177           | 1.16          | 1.76e-04      |
| 512 | FSGS 249 | 93.84        | 1.86e-01      | 1277          | 284.4         | 5.65e-02      | 1635          | 427.00        | 1.54e-04      |
|     | FSGS 128 | 20.69        | 5.39e-02      | 553           | 86.22         | 1.28e-02      | 1390          | 374.84        | 1.54e-04      |
|     | QSGS 57 | 1.26          | 1.80e-02      | 182           | 5.23          | 5.44e-02      | 569           | 18.40         | 8.88e-04      |
| 1024| FSGS 480 | 313.89       | 1.89e-01      | 1923          | 714.51        | 5.69e-02      | 5937          | 948.83        | 1.49e-04      |
|     | FSGS 271 | 172.33       | 5.45e-02      | 1463          | 570.00        | 1.32e-02      | 4619          | 810.72        | 1.25e-04      |
|     | QSGS 117 | 9.65         | 1.84e-02      | 481           | 39.95         | 5.58e-02      | 1834          | 156.66        | 4.09e-04      |
| 2048| FSGS 1186 | 557.00       | 1.86e-01      | 6241          | 1259.31       | 5.85e-02      | 6482          | 5345.02       | 1.20e-04      |
|     | FSGS 578 | 234.87       | 5.48e-02      | 3210          | 598.04        | 1.35e-02      | 6833          | 4120.13       | 1.20e-04      |
|     | QSGS 249 | 87.27        | 1.86e-02      | 1277          | 448.7         | 5.56e-02      | 5880          | 2082.93       | 1.54e-04      |

Table 2. Comparison between number of iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example 2 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     |
| 128 | FSGS 57 | 1.42          | 1.80e-01      | 182           | 4.41          | 5.44e-02      | 569           | 13.70         | 8.88e-04      |
|     | FSGS 33 | 0.38          | 5.07e-02      | 83            | 0.82          | 1.13e-02      | 233           | 2.27          | 8.88e-04      |
|     | QSGS 17 | 0.27          | 1.59e-02      | 30            | 0.18          | 4.66e-02      | 57            | 0.20          | 3.29e-04      |
| 256 | FSGS 117 | 10.95        | 1.84e-01      | 481           | 45.32         | 5.58e-02      | 931           | 174.77        | 4.09e-04      |
|     | FSGS 63 | 2.50          | 5.28e-02      | 211           | 8.04          | 1.23e-02      | 746           | 28.91         | 4.09e-04      |
|     | QSGS 30 | 0.38          | 1.73e-02      | 71            | 0.54          | 5.12e-02      | 177           | 1.16          | 1.76e-04      |
| 512 | FSGS 249 | 93.84        | 1.86e-01      | 1277          | 284.4         | 5.65e-02      | 1635          | 427.00        | 1.54e-04      |
|     | FSGS 128 | 20.69        | 5.39e-02      | 553           | 86.22         | 1.28e-02      | 1390          | 374.84        | 1.54e-04      |
|     | QSGS 57 | 1.26          | 1.80e-02      | 182           | 5.23          | 5.44e-02      | 569           | 18.40         | 8.88e-04      |
| 1024| FSGS 480 | 313.89       | 1.89e-01      | 1923          | 714.51        | 5.69e-02      | 5937          | 948.83        | 1.49e-04      |
|     | FSGS 271 | 172.33       | 5.45e-02      | 1463          | 570.00        | 1.32e-02      | 4619          | 810.72        | 1.25e-04      |
|     | QSGS 117 | 9.65         | 1.84e-02      | 481           | 39.95         | 5.58e-02      | 1834          | 156.66        | 4.09e-04      |
| 2048| FSGS 1186 | 557.00       | 1.86e-01      | 6241          | 1259.31       | 5.85e-02      | 6482          | 5345.02       | 1.20e-04      |
|     | FSGS 578 | 234.87       | 5.48e-02      | 3210          | 598.04        | 1.35e-02      | 6833          | 4120.13       | 1.20e-04      |
|     | QSGS 249 | 87.27        | 1.86e-02      | 1277          | 448.7         | 5.56e-02      | 5880          | 2082.93       | 1.54e-04      |