Advanced Science Letters
Vol. 24, No. 3, 1297-1931, 2018

Caputo's Implicit Solution of Space-Fractional Diffusion Equations by QSSOR Iteration

Andang Sunarto[1,*], Jumat Sulaiman[2], Azali Saudi[3]
[1]Faculty of Economic and Islamic Business, IAIN Bengkulu, Indonesia
[2]Mathematics with Economics Program, Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia
[3]Knowledge Technology Research Unit, Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia

In this study, we present the Caputo's implicit finite difference solution for the one-dimensional linear space-fractional diffusion equations based on the quarter-sweep Caputo's implicit finite difference approximation equation. To derive the approximation equation, firstly the proposed problem will be discretized by using the Caputo's space fractional derivative and the second-order central difference scheme. Then we can construct a linear system which has been solved by using Quarter-Sweep SOR (QSSOR)iterative method. The numerical results of this iterative method will be compared with other existing SOR methods such as Full-Sweep SOR (FSSOR) and Half-Sweep SOR (HSSOR). Finally it can be concluded that the proposed iterative method is superior compared with the FSSOR and HSSOR methods.

Keywords: Caputo's fractional derivative;Implicit Scheme; QSSOR method

1. INTRODUCTION

Over recent years, many researchers have paid attention to study of fractional partial differential equations (FPDEs). Fractional partial differential equations (FPDE's) arise in many science problems such as biology[1], chemistry[2], and hydrology[3]. One of its advantages is that it can be seen as a super set of ordinary derivatives that give the fractional derivatives to accomplish what integer-order derivatives cannot[4]. One may refer for the history of the development of fractional differential operator[5,6].

Several definitions of fractional calculus have been proposed in the last two centuries. Here, we give some basic definitions and properties of the fractional calculus theory which are used further in this paper

Definition 1[7,8,9]**.**The Riemann-Liouville fractional integral operator , J^{β} of order-β is defined as

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

*Email Address: andang99@gmail.com

Definition 2[9,10]**.**The Caputo's fractional partial derivative operator, D^{β} of order - β is defined as

$$D^{\beta}f(x)=\frac{1}{\Gamma(m-\beta)}\int_0^x \frac{f^{(m)}(t)}{(x-t)^{\beta-m+1}}dt, \qquad (2)$$

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

We have the following properties when $m-1<\beta\le m,\ x>0$:

$$D^{\beta}{}_{k}=0, (k \text{ is a constant}),$$

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

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

In this paper, we consider a Caputo's implicit finite difference scheme to discretize the one-dimensional space-fractional diffusion equation of the form

 doi:10.1166/asl.2018.11190

$$\frac{\partial U(x,t)}{\partial t}=a(x)\frac{\partial^{\beta}U(x,t)}{\partial x^{\beta}}+b(x)\frac{\partial U(x,t)}{\partial x}+c(x)U(x,t)+f(x,t) \quad (3)$$

with initial condition

$$U(x,0)=f(x),\ 0<x<\ell,$$

and the Dirichlet boundary conditions

$$U(0,t)=g_0(t),\quad U(\ell,t)=g_1(t),\ 0<t\le T.$$

Actually the problem in Eq.(3) has been solved by many authors. Khader[11] used the properties of chebyshev polynomial method to discretize Eq.(3) .For instance, and then to get a linear system which was solved by finite difference method. Next Mehdi Safari[12] used He's variational iteration method to solve Eq(3).

The main idea of the current work is to apply of the quarter-sweep SOR (QSSOR) iteration for solving problem Eq.(3) based on Caputo's implicit finite difference approximation equations. Basically, the combination of quarter-sweep iteration with SOR method is called as Quarter-Sweep SOR (QSSOR) method. To show the performance of the QSSOR method, we implement the FSSOR and Half-Sweep SOR (HSSOR) iterative methods being used as a control method. From other observation of previous study, the proposed quarter-sweep iteration concept on QSSOR is inspired by the concept of the half-sweep iterative method. Firstly, this concept has been introduced by Abdullah via the Explicit Decoupled Group (EDG) iterative method to solve two-dimensional Poisson equations[13]. The main characteristics of the half-sweep iteration is to reduce the computational complexities during iteration process, in which it will only consider nearly half of all interior node points in the solution domain in Eq.(3). Thus, the applications of half-sweep iterative methods have been discussed in[14,15]. Later, Othman and Abdullah[16] extended concept of half-sweep iteration by introducing the quarter-sweep iterative method via the modified explicit group (MEG) iterative method to solve two-dimensional Poisson Equations . Furthermore, to verify the effectiveness of the quarter-sweep iteration many studies have also been carried out[17,18]. Actually, quarter-sweep iteration inherits the characteristics of the half-sweep iteration in which its implementation process will consider approximately a quarter of all interior node points of the solution domains in Eq.(3).

2. CAPUTO'S IMPLICIT FINITE DIFFERENCE APPROXIMATION

Based on Eq. (3) and assume $h=\frac{\ell}{k}$, where k is positive integer, the formulation of Caputo's fractional partial derivative of the second order approximation method is given as

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

$$=\frac{1}{\Gamma(2-\beta)}\sum_{j=0,4,8}^{i-4}\int_{jh}^{(j+1)h}\left(\frac{U_{i-j+4,n}-2U_{i-j,n}+U_{i-j-4,n}}{4h^2}\right)(nh-s)^{\beta}\partial s$$

$$=\frac{(2h)^{\beta}}{\Gamma(3-\beta)}\sum_{j=0,4,8}^{i-4}\left(U_{i-j+4,n}-2U_{i-j,n}+U_{i-j-4,n}\right)\left(\left(\frac{j}{4}+1\right)^{2-\beta}-\frac{j}{4}^{2-\beta}\right) \quad (4)$$

and we have the following expressions

$$\sigma_{\beta,4h}=\frac{(2h)^{-\beta}}{\Gamma(3-\beta)}$$

and

$$g_j^{\beta}=\left(\frac{j}{4}+1\right)^{2-\beta}-\frac{j}{4}^{2-\beta}$$

By simplifying the discrete approximation in Eq.(4), we have

$$\frac{\partial^{\beta}U(x_i,t_n)}{\partial x^{\beta}}=\sigma_{\beta,2h}\sum_{j=0,4,8}^{i-4}g_j^{\beta}\left(U_{i-j+4,n}-2U_{i-j,n}+U_{i-j-4,n}\right) \quad (5)$$

By using Eq. (5) and the implicit finite difference discretization scheme, the Caputo's implicit finite difference approximation equation of problem in Eq (3) can be given a

$$\lambda(U_{i,n}-U_{i,n-4})=a_i\sigma_{\beta,4h}\sum_{j=0.4.8}^{i-4}g_j^{\beta}(U_{i-j+4,n}-2U_{i-j,n}+U_{i-j-4,n})+b_i\frac{(U_{i+4,n}-U_{i-4,n})}{8h}$$
$$+C_iU_{i,n}+f_{i,n} \quad (6)$$

for $i=4,8...,m-4$. We obtain Eq.(7) let manipulating the approximation equation (6) as follows

$$\lambda U_{i,n-4}=-a_i\sigma_{\beta,4h}\sum_{j=0,4,8}^{i-4}g_j^{\beta}(U_{i-j+4,n}-2U_{i-j,n}+U_{i-j-4,n})-\frac{b_i}{8h}(U_{i+4,n}-U_{i-4,n})-C_iU_{i,n}+\lambda U_{i,n}-f_{i,n}$$

(7)

Again from simplification of Eq.(7), we can get

$$\therefore b_i^{*}U_{i-4,n}+(\lambda-c_i^{*})U_{i,n}-b_i^{*}U_{i+4,n}-a_i^{*}\sum_{j=0,4,8}^{i-4}g_j^{\beta}(U_{i-j+4,n}-2U_{i-j,n}+U_{i-j-4,n})=f_i$$

(8)

where

$$a_i^{*}=a_i\sigma_{\beta,4h}\ ,\ b_i^{*}=\frac{b_i}{8h}\ ,\ c_i^{*}=c_i\ ,\ F_i^{*}=f_{i,n}\ ,$$
$$f_i=\lambda(U_{i,n-4})+F_i^{*}.$$

By considering Eq.(8), this approximation equation can be shown as follows

$$-R_i+\alpha_iU_{i-12,n}+s_iU_{i-8}+p_iU_{i-4,n}+q_iU_{i,n}+r_iU_{i+4,n}=f_i \quad (9)$$

where

$$
\begin{aligned}
R_i &= a_i^* \sum_{j=12}^{i-4} g_j^{\beta}\left(U_{i-j+4} - 2U_{i-j,n} + U_{i-j-4,n}\right),\\
\alpha_i &= \left(-a_i^* g_2^{\beta}\right),\\
s_i &= \left(-a_i^* g_1^{\beta} + 2a_i^* g_2^{\beta}\right),\\
p_i &= \left(b_i^* - a_i^* g_2^{\beta} + 2a_i^* g_1^{\beta} - a_i^*\right),\\
q_i &= \left(-a_i^* g_1^{\beta} + 2a_i^* + \left(\lambda - c_i^*\right)\right),\\
r_i &= \left(-a_i^* - b_i^*\right).
\end{aligned}
$$

Imposing Eq.(9) into all interior node point in solution domain in Eq.(3), it can be seen that the linear system can be constructed in matrix form as

$$A\underset{\sim}{U} = \underset{\sim}{f} \qquad (10)$$

where

$$
A = \begin{bmatrix}
q_4 & r_4 & & & & & & \\
p_8 & q_8 & r_8 & & & & & \\
s_{12} & p_{12} & q_{12} & r_{12} & & & & \\
\alpha_{16} & s_{16} & p_{16} & q_{16} & r_{16} & & & \\
 & \alpha_{20} & s_{20} & p_{20} & q_{20} & r_{20} & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & & \alpha_{m-8} & s_{m-8} & p_{m-8} & q_{m-8} & r_{m-8} \\
 & & & & \alpha_{m-4} & s_{m-4} & p_{m-4} & q_{m-4}
\end{bmatrix}_{(m-4)X(m-4)}
$$

$$\underset{\sim}{U} = \begin{bmatrix} U_{4,1} & U_{8,1} & U_{12,1} & \cdots & U_{m-4,1} & U_{m-2,1} \end{bmatrix}^T ,$$

$$\underset{\sim}{f} = \begin{bmatrix} f_4 + p_4 U_{4,1} & f_8 + s_8 U_{8,1} & f_{12} + \alpha_{12} U_{12,1} & f_{16} + R_i & \cdots f_{m-8,1} + R_{m-8} & f_{m-4,1} + p_{m-4} U_{m,1} + R_{m-4} \end{bmatrix}^T$$

3. FORMULATION OF QSSOR ITERATIVE METHOD

According to the linear system in Eq. (10), it is clear that the main characteristic of its coefficient matrix has large scale and sparse. Therefore, the iterative methods will be used to get the approximate solution of Eq.(10). From the previous studies, the concept of various iterative methods has been initiated and conducted by many researchers such as, Young[19,20] and Hackbusch[21]. In this section, we illustrate on how to develop the formulation of QSSOR iterative method being applied to solve linear system in Eq.(10). As we mention in the previous section, The main advantage of the Qurter-Sweep iteration is to reduce the computational complexities during iteration process. Due to the advantage of this concept, we apply QSSOR method, the over linear system in Eq.(10). Before that, let the coefficient matrix, A in Eq.(10) be expressed as summation of the three matrices

$$A = D - L - V \qquad (11)$$

where *D*, *L* and *V* are diagonal, lower triangular and upper triangular matrices respectively.

Thus, QSSOR iterative method can be defined generally as[16,17]

$$\underset{\sim}{U}^{(k+1)} = \left(D - \omega L\right)^{-1}[\omega V + (1-\omega)D]\underset{\sim}{U}^{(k)} + \left(D - \omega L\right)^{-1}\omega f \qquad (12)$$

where $\underset{\sim}{U}^{(k)}$ represents an unknown vector at k[th] iteration. The implementation of the QSSOR iterative method can be described in Algorithm 1.

Algorithm 1 : QSSOR Scheme

i. Initialize $\underset{\sim}{U} \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$
ii. For i=4,8,…,m-4 and j=0,4,…,n-2 assign

$$\underset{\sim}{U}^{(k+1)} = \left(D - \omega L\right)^{-1}[\omega V + (1-\omega)D]\underset{\sim}{U}^{(k)} + \left(D - \omega L\right)^{-1}\omega f$$

iii. 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 step (iv). Otherwise go back to step (ii)
iv. Display approximate solutions

4. ANALYSIS OF STABILITY

In this section is described in the analysis of stability on the problems of Quarter-Sweep one dimensional linear space-fractional diffusion equation. Referring back and considering the equation (3) such that the equation can be expressed as:

$$a_i^* u_{i+4,n} - \left(1 + 2a_i^*\right)u_{i,n} + a_i^* u_{i-4} + a_i^* \sum_{j=4}^{i-4} g_j^{\beta}\left(u_{i-j+4,n} - 2u_{i-j,n} + u_{i-j-4,n}\right) = -u_{i,n-4} \qquad (11)$$

For $i = 4,8,\ldots,M-4$ and $n = 1,2,3,\ldots N$.

Subject to initial conditions

$$u_{i,0} = g_i, \qquad i = 4,8,\ldots,M-4, M = 2^k, k \ge 1$$

and boundary conditions

$$u_{o,n} = g_{1,i};\ u_{M,n} = g_{2,i};\ \text{and}$$
$$i = 4,8,\ldots,M-4, M = 2^k, k \ge 1$$

with $a_i^* = \Delta t.\sigma_{\beta,h}$ and $a_i^* > 0.$

Equation (11) can be shown in matrix form as:

$$A\underset{\sim}{u_n} = -u_{n-4}, \quad n \ge 4 \qquad (12)$$

with $\underset{\sim}{u_n} = \left(u_{4,n}, u_{8,n}, \ldots, u_{m,n}\right)^T$ and $A = \left(a_{ij}\right)$ is a square matrix on the M-level coefficient. For $i = 4,8,\ldots,M-4, M = 2^k, k \ge 1$ and $j = 4,8,\ldots,M-4$, the coefficient is

$$a_{ij} = \begin{cases} 0,\ jika\ j \ge i+8 \\ b_i,\ jika\ j = i+4 \\ \left(-\left(1+2b_i\right)+b_i g_4\right),\ jika\ \ j = i = 8,12,\ldots,M-4 \\ b_i\left(1-2g_4+g_8\right),\ jika\ j = i-4,\ i = 12,16,\ldots,M-8 \\ b_i\left(g_4-2g_8+g_{12}\right),\ jika\ j = i-8,\ i \ge 16 \end{cases}$$

If:

$$a_{44} = -\left(1+2b_4\right),$$
$$a_{84} = b_8\left(1-2g4\right),$$
$$a_{124} = b_{12}\left(g_4-2g_8\right),$$

Lema 1 (Cases of Quarter-Sweep Matrix Coefficient)

If $\lambda_j(A),\ j = 4,8,\ldots,M-4,$ it represents the eigenvalue of the matrix A it is evidenced the following:

(i). $\left|\lambda_j(A)\right| > 1,\quad j = 4,8,\ldots,M-4,$

(ii). $\left\|-A^{-1}\right\|_2 \le 1$

Proof:
Gerschgorin's theorem states that every eigenvalue of the square matrix A is at least following one of the following disks:

$$\left|\lambda - a_{jj}\right| \le \sum_{\ell=4,\ell\ne j}^{m} a_{\ell j},\ j = 4,8,\ldots,M-4 \tag{13}$$

then, each eigenvalue on matrix A meets one of the inequalities:

$$|\lambda| \le \left|\lambda - a_{jj}\right| \le \sum_{\ell=4,\ell\ne j}^{m}\left|a_{\ell j}\right| \le \sum_{\ell=4}^{m}\left|a_{\ell j}\right|, \tag{14}$$

$$|\lambda| \ge \left|a_{jj}\right| - \left|\lambda - a_{jj}\right| \ge \left|a_{jj}\right| - \sum_{\ell=4,\ell\ne j}^{m}\left|a_{\ell j}\right|. \tag{15}$$

Next, to prove the condition (i), consider the equation(15) for matrix A, then each eigenvalue of the matrix A satisfies the inequality

$$\begin{aligned} \left|\lambda_4 A\right| &\ge \left|-\left(1+2b_4\right)\right| - \left|b_4\right| \\ &\ge 1+2b_4-b_4 \\ &\ge 1+b_4 > 1,\ \because b_4 > 0 \\ \therefore &\left|\lambda_4(A)\right| > 1 \end{aligned}$$

$$\begin{aligned} \left|\lambda_8(A)\right| &\ge \left|-\left(1+2b_8\right)+b_8g_4\right| - \left|b_8\right| - \left|b_8-2b_8g_4\right| \\ &\ge \left(1+2b_8+b_8g_4\right) - b_8 - \left(b_8-2b_8g_4\right) \\ &\ge 1+2b_8+b_8g_4-b_8-b_8+2b_8g_4 \\ &\ge 1+2b_8-2b_8+b_8g_4+2b_8g_4 \\ &\ge 1+3b_8g_4 > 1,\ \because b_8 > 0 \\ \therefore &\left|\lambda_8(A)\right| > 1 \\ &\vdots \end{aligned}$$

$$\begin{aligned} \left|\lambda_M(A)\right| &\ge \left|-\left(1+2b_M\right)+b_Mg_{M-4}\right| - \left|b_M\right| - \left|b_M-2b_Mg_{M-4}\right| \\ &\ge \left(1+2b_M+b_Mg_{M-4}\right) - \left(b_M\right) - \left(b_M-2b_Mg_{M-4}\right) \\ &\ge 1+2b_M+b_Mg_{M-4}-b_M-b_M+2b_Mg_{M-4} \\ &\ge 1+2b_M-2b_M+b_Mg_{M-4}+2b_Mg_{M-4} \\ &\ge 1+3b_Mg_{M-4} > 1,\ \because b_M > 0 \\ \therefore &\left|\lambda_M(A)\right| > 1 \end{aligned}$$

Next, this proves that $\left|\lambda_j(A)\right| > 1,\ j = 4,8,\ldots,M-4.$

To prove the condition (ii), assume $\left\|-A^{-1}\right\|_2 = \max_{1\le j\le N}\left|\lambda_j\left(-A^{-1}\right)\right| > 1$ and $\rho\left(-A^{-1}\right) = \frac{1}{\lambda}$ so $\left\|-A^{-1}\right\|_2 \le \frac{1}{\left|\lambda_j\left(-A^{-1}\right)\right|} \le 1$.

It can be concluded that the statement

$$\left\|-A^{-1}\right\|_2 \le \frac{1}{\left|\lambda_j\left(-A^{-1}\right)\right|} \le 1$$

is a condition of (ii) the stability of the Gerschgorin theorem[22].

Theorem 1 (Case of Quarter-Sweep Approximation Equation)

The solution on the Quarter-Sweep approximation equation (11) is unconditional stable.

Proof:
To prove that the above scheme is unconditionally stable, then it is shown $\left\|\underset{\sim}{u}_n\right\|_2 \le \left\|\underset{\sim}{u}_0\right\|_2$, for $n \ge 4$

From equation (11), we have

$$A\underset{\sim}{u}_n = -\underset{\sim}{u}_{n-4},\quad n = 4,8,\ldots,N \tag{16}$$

Matrix A is inverse matrix, then for $n = 4,8,\ldots$ from equation (16), it is found

$$\begin{aligned} &A\underset{\sim}{u}_4 = -\underset{\sim}{u}_0 \\ &\therefore \underset{\sim}{u}_4 = -A^{-1}\underset{\sim}{u}_0 \\ &A\underset{\sim}{u}_8 = -\underset{\sim}{u}_4, \\ &\underset{\sim}{u}_8 = -A^{-1}\underset{\sim}{u}_4 \\ &\underset{\sim}{u}_8 = -A^{-1}\left(-A^{-1}\underset{\sim}{u}_0\right) \\ &\therefore \underset{\sim}{u}_8 = \left(-A^{-1}\right)^2\underset{\sim}{u}_0 \\ &\vdots \end{aligned}$$

$$\underset{\sim}{u}_n = \left(-A^{-1}\right)^n\underset{\sim}{u}_0,\quad n \ge 4 \tag{17}$$

From equation (17), it is found

$$\left\| \underset{\sim}{u_n} \right\|_2 \leq \left\| -A^{-1} \right\|_2^n \left\| \underset{\sim}{u_0} \right\|_2$$

$$\therefore \left\| \underset{\sim}{u_n} \right\|_2 \leq \left\| \underset{\sim}{u_0} \right\|_2$$

From Lema 1, it can be proved that

$$\therefore \left\| -A^{-1} \right\|_2$$

From the above proof, it can be shown that the finite difference scheme for Quarter-Sweep space-fractional diffusion equation is stable unconditionally. Thus, the proof is complete.

5. NUMERICAL EXPERIMENTS

In order to evaluate the effectiveness performance of QSSOR iterative methods as described in the previous section, numerical experiments were carried out from two problems of Space-fractional diffusion equation. Both problems are classified as well-posed equation. For comparison purpose, three criteria have been considered such as number of iterations (K), execution time (in seconds) and maximum absolute error at three different values of $\beta = 1.2, \beta = 1.5$ and $\beta = 1.8$ For implementation of three iterative schemes, the convergence test considered the tolerance error, which is fixed as $\varepsilon = 10^{-10}$.

Example 1[23] :
First, consider the following space-fractional initial boundary value problem

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

(18)
On finite domain $0 < x < 1$, with the diffusion coefficient $d(x) = \Gamma(\beta) x^{0.5}$, the source function $p(x,t) = (x^2+1)\cos(t+1) - 2x\sin(t+1)$, with the initial condition $U(x,0) = (x^2+1)\sin(1)$ and the boundary conditions $U(0,t) = \sin(t+1)$, $U(1,t) = 2\sin(t+1)$, for t>0. The exact solution of this problem is $U(x,t) = (x^2+1)\sin(t+1)$.

Example 2[23] **:**
Consider the following space-fractional initial boundary value problem

$$\frac{\partial U(x,t)}{\partial t} = \Gamma(1.2) x^{\beta} \frac{\partial^{\beta} U(x,t)}{\partial x^{1.8}} + 3x^2(2x-1)e^{-t}, \quad (19)$$

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

All results of numerical experiments for Problem (18) dan (19), which were obtained from implementation of FSSOR, HSSOR and QSSOR iterative methods have been recorded in Tables 1 and 2 at different values of mesh sizes, *m* = 128, 256, 512, 1024, and 2048.

6. CONCLUSIONS

In this paper, we applied the QSSOR method based on the corresponding Caputo's fractional derivative operator and implicit finite difference scheme to solve space-fractional diffusion equation. Based on numerical simulations, clearly it demonstrates that two promising improvements in the number of iteration (K) and execution time with implementing a QSSOR iterative method have been shown as compared to the FSSOR and HSSOR methods. Overall, the numerical results showed that the quarter-sweep iteration concept in association with the SOR iterative method is superior and it has reduced the computational complexity significantly. Therefore further investigation of two-step[24,25] iterative methods should be considered for future works.

REFERENCES

[1] Yuste, S.B and Lindenberg, K, 2001, Subdiffusion-limited A+A Reactions. *Physic Review Letter,* 87(11), 118301.

[2] Yuste, S.B, Acedo, L and Lindenberg, K, 2004, Reaction Front in an A+B C Reaction-Subdiffusion Process, *Physic Review Letter*, E69(3), 036126.

[3] Benson, D.A, Wheatcraft, S.W and Meerschaert, M.M, 2000, Application of a Fractional Advection-Dispersion Equation, *Water Resource Research*, 36(6), 1403-1412. .

[4] Su, L, Wang, W and Xu, Q, 2010, Finite Difference Methods for Fractional Dispersion Equations, *Applied Mathematics and Computation*, 216, 3329-3334.

[5] Oldham, K.B and Spanier, J, 1974.The Fractional Calculus, Academic Press, New York.

[6] Miller, K.S and Ross, B, 1993. An Introduction to The Fractional Calculus and Fractional Differential Equations, Willey, New York.

[7] Zhang, Y, 2009, A Finite Difference Method For Fractional Partial Differential Equation, *Applied Mathematics And Computation 215* (2009),524-529.

[8] Podlubny, I, 1999. Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations to Methods of Their Solution and Some of Their Applications, Academic Press, New York.

[9] Kilbas, A.A, Srivastava, H.M and Trujillo, J.J, 2006. Theory and Applications of Fractional Differential Equations. Elsevier Science Inc. New York, USA.

[10] Caputo, M, 1967. Linear Model of Dissipation Whose *Q* is Almost Frequency Independent-II, *Geophysical Journal of Royal Astronomic Society*, 13, 529-539.

[11] Khader, M.M, 2011, On The Numerical Solutions for The Fractional Diffusion Equation, *Communication in Non-Linear Science and Numerical Simulation,* 16, 2535-2542

[12] Safari, M, 2011.Application of He's Variational Iteration Method for The Analytical Solution of Space- Fractional Diffusion Equation, *Applied Mathematics*, 2, 1091-1095.

[13] Abdulah, A.R,1991.The Four Point Explicit Decoupled Group (EDG) Method: A Fast Poisson Solver, *International Journal of Computer Mathematics*, 61-70.

[14] Sunarto, A, Sulaiman, J & Saudi, A, 2014. Solving The Time-Fractional Diffusion Equations by The Half-Sweep SOR Iterative Method, Proceeding of International Conference of Advanced Informatics: Concept, Theory and application (ICAICTA), Bandung, Indonesia, 272-277.

[15] Sunarto, A, Sulaiman, J & Saudi, A, 2014. Half-Sweep Gauss-Seidel Iteration Applied to Time-Fractional Diffusion Equations, Proceeding of International Conference on Global optimization and Its applications (ICoGOIA 2014), Yogyakarta, Indonesia, 68-74.

[16] Othman, M and Abdullah, A.R ,1998. The Half-Sweep Multigrid Method as a fast Multigrid Poisson Solver, *International Journal of Computer Mathematics*, 319-329

[17] Sunarto, A, Sulaiman, J & Saudi, A, 2015. Numerical Solution of The Time-Fractional Diffusion Equations by Using Quarter-Sweep SOR Iterative Method, *International Journal of Mathematical Engineering and Science (IJMES)*, 5, 54-67.

[18] Sunarto, A, Sulaiman, J & Saudi, A, 2014. 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, 120-128.

[19] Young, D.M, 1971. Iterative Solution of Large Linear Systems, London: Academic Press.

[20] Young, D.M, 1972. Second-Degree Iterative Methods for The Solution of Large Linear Systems*, Journal of Approximation Theory,* 5, 37-148.

[21] Hackbusch, W, 1995, Iterative Solution of Large Sparse Systems of Equations. New York: Springer-Verlag.

[22] Brakken-Thal, Sean, 2007, Gershgorin Theorem for Estimating Eigenvalues, Sthal@ups.edu

[23] Azizi, H & Loghmani, G.B, 2013. Numerical Approximation for Space-Fractional Diffusion Equations via Chebyshev Finite Difference Method, *Journal of Fractional and Applications*, 4(2), 303-311.

[24] Evans, D.J. & Sahimi, M.S, 1988. The Alternating Group Explicit Iterative Method (AGE) To Solve Parabolic And Hyperbolic Partial Differential Equations, *Annual Review of Numeric Fluid Mechanic And Heat Transfer,* 2, 283-389.

[25] Sahimi, M.S., Ahmad, A. & Bakar, A.A, 1993. The Iterative Alternating Decomposition Explicit (IADE) Method To Solve The Heat Conduction Equation, *International Journal of Computer Mathematics*, 47, 219-229.

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

M	Method	β = 1.2			β = 1.5			β = 1.8		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	**FSSOR**	66	1.69	2.37e-02	205	4.08	6.21e-04	733	14.47	2.42e-02
	HSSOR	46	0.47	2.22e-02	110	0.78	6.99e-04	158	6.01	2.42e-02
	QSSOR	21	0.12	1.99e-02	32	0.12	8.19e-04	70	0.19	4.11e-02
256	**FSSOR**	129	10.13	2.44e-02	545	42.29	6.69e-04	1361	107.33	2.39e-02
	HSSOR	77	2.93	2.37e-02	212	17.62	6.77e-04	479	49.23	2.39e-02
	QSSOR	46	0.31	2.24e-02	79	0.48	6.99e-04	225	1.30	4.03e-02
512	**FSSOR**	278	85.9	2.47e-02	1459	144.73	5.35e-04	3472	725.25	2.37e-02
	HSSOR	129	19.47	2.44e-02	550	81.68	6.69e-04	1465	219.71	2.37e-02
	QSSOR	77	1.77	2.37e-02	205	4.34	6.21e-04	633	15.48	3.99e-02
1024	**FSSOR**	607	140	2.49e-02	3906	756.12	5.13e-04	5539	1259.97	2.36e-02
	HSSOR	278	68.3	2.47e-02	1385	330.55	5.36e-04	2472	770.92	2.36e-02
	QSSOR	129	10.56	2.44e-02	540	44.33	5.69e-04	2393	194.41	3.97e-02
2048	**FSSOR**	1230	577.00	2.52e-02	6320	3348.68	5.09e-04	13643	3979.18	2.30e-02
	HSSOR	632	202.34	2.49e-02	3345	1096.00	5.13e-04	7625	1321.10	2.30e-02
	QSSOR	278	95.69	2.47e-02	1459	503	5.35e-04	6786	680.26	3.96e-02

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

M	Method	β = 1.2			β = 1.5			β = 1.8		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	**FSSOR**	49	1.19	1.80e-01	156	3.77	5.44e-02	332	3.24	1.25e-04
	HSSOR	23	0.43	1.80e-01	56	0.66	5.44e-02	103	1.22	1.22e-04
	QSSOR	19	0.08	1.59e-01	23	0.08	4.61e-02	43	0.12	3.29e-04
256	**FSSOR**	103	5.45	1.84e-01	225	14.80	5.58e-02	890	36.00	1.44e-04
	HSSOR	52	2.44	1.84e-01	147	6.69	5.58e-02	313	14.25	1.44e-04
	QSSOR	33	0.25	1.73e-01	56	0.36	5.16e-02	138	0.79	1.76e-04
512	**FSSOR**	221	25.31	1.86e-01	732	153.67	5.65e-02	1490	427	1.47e-04
	HSSOR	99	13.05	1.86e-01	393	72.47	5.65e-02	661	176	1.47e-04
	QSSOR	52	1.19	1.80e-01	147	3.12	5.44e-02	448	9.54	8.88e-04
1024	**FSSOR**	271	172.33	5.45e-01	1463	218	1.32e-02	4619	2210.72	1.25e-04
	HSSOR	212	52.00	1.89e-01	547	120.00	5.69e-02	3823	1423.03	1.49e-04
	QSSOR	99	8.19	1.84e-01	393	32	5.58e-02	2153	920.27	4.09e-04
2048	**FSSOR**	880	424.00	1.92e-01	2530	953.23	5.73e-02	7710	4120.81	2.30e-04
	HSSOR	495	198.22	1.92e-01	1782	523.00	5.73e-02	5482	2740.23	2.30e-04
	QSSOR	212	73.76	1.86e-01	1047	365.48	5.65e-02	3264	1029.27	1.54e-04