NUMERICAL SOLUTIONS OF THE TIME-FRACTIONAL DIFFUSION EQUATIONS BY USING QUARTER-SWEEP SOR ITERATIVE METHOD

Andang Sunarto *1, Jumat Sulaiman †1 and Azali Saudi ‡2

[1]Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia
[2]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia

January 28, 2015

Communicated by *S.A. Edalatpanah*

Abstract

The main objective of this paper is to describe the formulation of Quarter-Sweep Successive Over-Relaxation (QSSOR) iterative method using the Caputos time fractional derivative together with Quarter-Sweep implicit finite difference approximation equation for solving one-dimensional linear time-fractional diffusion equations. To solve the problems, a linear system will be constructed via discretization of the one-dimensional linear time fractional diffusion equations by using the Caputos time fractional derivative. Then the generated linear system has been solved by using the proposed QSSOR iterative method. Computational results are provided to demonstrate the effectiveness of the proposed methods as compared with the FSSOR and HSSOR methods.

Keywords : Caputos fractional derivative, Implicit finite difference, QSSOR method.

1 Introduction

Based on previous studies, many problems in engineering and science involve fractional partial differential equations (FPDEs) ([1], [2], [3], [4],[5]). The applications of fractional partial differential equations are encountered in many fractional problems to obtain numerical and/or analytical solutions. To solve one-dimensional diffusion model with constant coefficients a fractional derivative,

*andang99@gmail.com
†jumat@ums.edu.my
‡azali60@gmail.com

which replaces the first-order space partial derivative in a diffusion model will lead to slower diffusion. Actually, many numerical methods have been proposed.

For instance, Yuste and Acedo [6] proposed a numerical methods for solving the time fractional diffusion equations (TFDE), such explicit and implicit finite difference methods [7]. Nevertheless the explicit methods are conditionally stable, this finite difference schemes are available in the literature [8].

For solving the problems of the time-fractional diffusion equations (TFDEs) numerically, the problems need to be discretized. By using the implicit finite difference scheme and Caputo fractional operator, a linear system at each time level can be constructed through the Caputos implicit finite difference approximation equations. In order to solve a linear systems, many researchers have also discussed the concept of iterative methods such as Young [9], Hackbush [10] and Saad [11]. Besides these iterative methods, the concept of block iteration has also been introduced by Evans [12], Ibrahim and Abdullah [13],Yousif and Evans ([14],[15]) to show the efficiency of its computation cost. Actually for solving the large linear system, Abdullah [16] has initiated Half-Sweep iteration via Explicit Decoupled Group (EDG) method for solving two dimensional Poisson equation, which is one of the most known and widely used iterative techniques to solve in solving any linear systems. Motivated by this finding, extension of the concept of half sweep iterations has been used to introduce quarter-sweep iteration via the Modified Explicit Group (MEG) iterative method [17] to solve two-dimensional Poisson equations. Further studies to verify the effectiveness of the quarter-sweep iterations have been carried out, see (See [18], [19], [20], [21], [22])

In this paper, we examine the applications of Quarter-Sweep Successive Over-relaxation (QSSOR) iterative method to solve time-fractional parabolic partial differential equations (TPPDEs) based on the Caputos implicit finite difference approximation equation. To show the performance of the QSSOR method, we implement the Full-Sweep Successive Over-relaxation (FSSOR) and Half-Sweep Successive Over-relaxation (HSSOR) iterative methods being used as a control method.

Firstly, for the derivation of the QSSOR iterative method, consider time-fractional diffusion equation (TFDEs) be defined as

$$\frac{\partial^\alpha U\left(x,t\right)}{\partial^\alpha t} = a\left(x\right)\frac{\partial^2 U\left(x,t\right)}{\partial^2 t} + b\left(x\right)\frac{\partial U\left(x,t\right)}{\partial t} + c\left(x\right)U\left(x,t\right), \tag{1}$$

where a(x), b(x) and c(x) are known functions or constants, whereas is a parameter which refers to the fractional order of time derivative.

The outline of this paper is organized as follows: In Sections 2 and 3, the approximate equation of the Caputos fractional derivative operator and numerical procedure for solving time fractional diffusion equation(1)by means of the implicit finite difference method is given. The analysis of stability of method is determined in Section 4, formulation of the QSSOR iterative method is introduced in Section 5. Then, In Section 6 shows numerical examples and its results and conclusion is given in Section 7.

2 Preliminaries

This section gives some definitions that can be applied for fractional derivative theory before developing the approximation equation of Problem (1).

Definition (1). [8] The Riemann-Liouville fractional integral operator, J^{α} of order- α is defined as

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

Definition (2) [8]The Caputos fractional partial derivative operator, D^{α} of order-α is defined as

$$J^{\alpha} f(x,t) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-t)^{\alpha} f(t)\, dt\, \alpha > 0,\ x > 0, \tag{3}$$

with $m-1 < \alpha \leq m$, $m \in N$, $x > 0$.

To solve TFDEs in Eq.(1) as mentioned in the first section, we get numerical approximations by using implicit finite difference scheme via the Caputos derivative definition with Dirichlet boundary conditions and the non-local fractional derivative operator. This Caputos implicit finite difference approximation equation can be categorized as unconditionally stable scheme. Before solving Problem (1), the solution domain of the problem has been restricted to the finite space domain$0 \leq x \leq \gamma$, with$0 < \alpha < 1$, whereas the parameter α refers to the fractional order of space derivative. Also, let us consider the initial and boundary conditions of Problem (1) be given as

$$U(0,t) = g_0(t),\ U(\ell,t) = g_1(t),$$

and the initial condition

$$U(x,0) = f(x),$$

where $g_0(t)$, $g_1(t)$, and $f(x)$, are given functions. To discretize the time-fractional derivative in Eq.(1), we consider Caputos fractional partial derivative of order α defined by ([8],[9]):

$$\frac{\partial^{\alpha} u(x,t)}{\partial t^{\alpha}} = \frac{1}{\Gamma(n-1)} \int_0^{\infty} \frac{\partial u(x-s)}{\partial t} (t-s)^{-\alpha}\, ds, \quad t > 0, \quad 0 < \alpha < 1. \tag{4}$$

3 Caputos Implicit Finite Difference Approximation

By applying Eq.(4), the formulation of Caputos fractional partial derivative of the first order approximation method is given as [23] :

$$D_t^{\alpha} U_{i,n} \cong \sigma_{\alpha,k} \sum_{j=1}^{n} \omega_j^{(\alpha)} \left(U_{i,n-j+1} - U_{i,n-j}\right), \tag{5}$$

and we have the following expressions

International Journal of Mathematical Engineering and Science (IJMES)
Volume 3 Issue 2 (February 2014) ISSN : 2277-6982
http://www.ijmes.com/

$$\sigma_{\alpha,k}=\frac{1}{\Gamma\left(1-\alpha\right)\left(1-\alpha\right)k^{\alpha}}\quad ,$$

and

$$\omega_j^{(\alpha)}=j^{1-\alpha}-\left(j-1\right)^{1-\alpha}.$$

Before discretizing Problem (1), let the solution domain of the problem be partitioned uniformly. To do this, we consider some positive integers m and n in which the grid sizes in space and time directions for the finite difference algorithm are defined as $h=\Delta x=\frac{\gamma-0}{m}$ and $k=\Delta t=\frac{T}{n}$ respectively. According to these grid sizes, we develop the uniformly grid network of the solution domain where the grid points in the space interval $[0,\gamma]$ are shown as the numbers $x_i=ih$,$i=0,1,2,...,m$ and the grid points in the time interval $[0,T]$ are labeled $t_j=jk$,$j=0,1,2,...,n$. Then the values of the function $U\left(x,t\right)$ at the grid points are denoted as $U_{i,j}=U\left(x_i,t_j\right)$. As mentioned in Eq.(5)and using the Quarter-Sweep implicit finite difference discretization scheme, the Caputos Quarter-Sweep implicit finite difference approximation equation of Problem (1) the grid point, $(x_i,t_j)=(ih,nk)$ is given as

$$\sigma_{\alpha,k}\sum_{j=1}^{n}\omega_j^{(\alpha)}\left(U_{i,n-j+1}-U_{i,n-j}\right)$$

$$=a_i\frac{1}{16h^2}\left(U_{i-4,n}-2U_{i,n}+U_{i+4,n}\right)+b_i\frac{1}{8h}\left(U_{i+4,n}-U_{i-4,n}\right)+c_iU_{i,n},\tag{6}$$

for i=4,8,12,...,m-4.

Now, the approximation equation (6) is known as the fully Quarter-Sweep implicit finite difference approximation equation, which is consistent first order accuracy in time and second order in space. Actually, the approximation can be rewritten based on the specified time level. For instance, we have for $n\geq 2$:

$$\sigma_{\alpha,k}\sum_{j=1}^{n}\omega_j^{(\alpha)}\left(U_{i,n-j+1}-U_{i,n-j}\right)=p_iU_{i-4,n}+q_iU_{i,n}+r_iU_{i+4,n},\tag{7}$$

where

$$p_i=\frac{a_i}{16h^2}-\frac{b_i}{8h},$$

$$q_i=c_i-\frac{a_i}{8h^2},$$

$$r_i=\frac{a_i}{16h^2}+\frac{b_i}{8h}.$$

also, we get for n = *1*,

$$-p_i U_{i-4,1} + q_i^* U_{i,1} - r_i U_{i+4,1} = f_{i,0}, \qquad i = 4, 6, ..., m-4 \tag{8}$$

where

$$\omega_j^{(\alpha)} = 1,$$

$$q_i^* = \sigma_{\alpha,k} - q_i,$$

$$f_{i,0} = \sigma_{\alpha,k} U_{i,0}.$$

According to Eq.(8), it can be seen that the tridiagonal linear system can be constructed in matrix form as

$$A \underset{\sim}{U} = \underset{\sim}{f}. \tag{9}$$

where

$$A = \begin{bmatrix} q_4^* & -r_4 & & & & \\ -p_8 & q_8^* & -r_8 & & & \\ & -p_{12} & q_{12}^* & -r_{12} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -p_{m-8} & q_{m-8}^* & -r_{m-8} \\ & & & & -p_{m-4} & q_{m-4}^* \end{bmatrix}_{((\frac{m}{4})-1)x(\frac{m}{4}-1)},$$

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

$$\underset{\sim}{f} = \begin{bmatrix} f_{4,1} + p_1 U_{0,1} & f_{8,1} & f_{12,1} & \cdots & f_{m-8,1} & f_{m-4,1} + p_{m-4} U_{m,1} \end{bmatrix}^T.$$

4 Analysis of Stability

In this section, we have considered the stability analysis of the implicit finite difference approximation equation in Eq.(7). For stability analysis, we use Von-Neumanns [24] and the Lax equivalence theorem [25]. It follows that numerical solution of approximation equation in Eq.(7) converges to the exact solution as $h, k \to 0$.

Theorem 4.1.

The fully implicit numerical method Eq.(7), solution to Eq.(1) with $0 < \alpha < 1$ on the finite domain $0 \leq x \leq 1$, with zero boundary condition $U(0,t) = U(\ell,t) = 0$ for all $t \geq 0$, is consistent and unconditionally stable.

Proof. To examine the stability of the proposed method, we find for solution of the form $U_j^n = \xi_n e^{i\omega jh}$, $i = \sqrt{-1}$, ω real. Therefore Eq.(7) becomes

$$\sigma_{\alpha,k}\xi_{n-1}e^{i\omega jh}-\sigma_{\alpha,k}\sum_{j=2}^{n}\omega_j^{(\alpha)}\left(\xi_{n-j+1}e^{i\omega jh}-\xi_{n-j}e^{i\omega jh}\right)=$$

$$-p_i\xi_n e^{i\omega(j-4)h}+\left(\sigma_{\alpha,k}-q_i\right)\xi_n e^{i\omega jh}-r_i\xi_n e^{i\omega(j+4)h} \tag{10}$$

By simplifying and reordering over Eq.(10), we have :

$$\sigma_{\alpha,k}\xi_{n-1}-\sigma_{\alpha,k}\sum_{j=2}^{n}\omega_j^{(\alpha)}\left(\xi_{n-j+1}-\xi_{n-j}\right)=$$

$$\xi_n\left(\left(\left(-p_i-r_i\right)\cos\left(\omega h\right)\right)+\left(\sigma_{\alpha,k}-q_i\right)\right).$$

This above equation can be reduced to :

$$\xi_n=\frac{\xi_{n-1}+\sum_{j=2}^{n}\omega_j^{(\alpha)}\left(\xi_{n-j}-\xi_{n-j+1}\right)}{\left(1+\frac{(p_i+r_i)}{\sigma_{\alpha,k}}\cos\left(\omega h\right)+\frac{q_i}{\sigma_{\alpha,k}}\right)}. \tag{11}$$

From Eq.(10), it can be observed that the conducted as

$$\left(1+\frac{(-p_i-r_i)}{\sigma_{\alpha,k}}\cos\left(\omega h\right)-\frac{q_i}{\sigma_{\alpha,k}}\right)\geq 1,$$

for all $\alpha,\ n,\ \omega,\ h$ and k we have :

$$\xi_1\leq\xi_0. \tag{12}$$

and

$$\xi_n\leq\xi_{n-1}+\sum_{j=2}^{n}\omega_j^{(\alpha)}\left(\xi_{n-j}-\xi_{n-j+1}\right),\ n\geq 2. \tag{13}$$

Thus, for n=*2*, the last inequality implies

$$\xi_2\leq\xi_1+\omega_2^{(\alpha)}\left(\xi_0-\xi_1\right).$$

Again repeating the above process, we can get

$$\xi_j\leq\xi_{j-1}, j\text{=}\mathit{1,2,}\ldots\mathit{n\text{-}1}.$$

From Eq.(13), we finally have

$$\xi_n \leq \xi_{n-1} + \sum_{j=2}^{n} \omega_j^{(\alpha)} \left(\xi_{n-j} - \xi_{n-j+1}\right) \leq \xi_{n-j}.$$

Since each term in the summation is negative, it shows that the inequalities Eq.(12) and Eq.(13) imply

$$\xi_n \leq \xi_{n-1} \leq \xi_{n-2} \leq \ldots \leq \xi_1 \leq \xi_0.$$

Thus,

$$\xi_n = \left|U_j^n\right| \leq \xi_0 = \left|U_j^0\right| = \left|f_j\right|,$$

which entails $\left\|U_j^n\right\| \leq \left\|f_j\right\|$, and we have stability.

5 Formulation of Quarter-Sweep Successive Over-Relaxation

By considering the tridiagonal linear system in Eq.(9), it is clear that the characteristic of its coefficient matrix has large scale and sparse. In this paper, application of the QSSOR method is used to solve linear system Eq.(9). As we know, the main objective of the Quarter-Sweep iteration is to reduce the computational complexities during iteration process. Due to the advantage of the concept of QSSOR method, let the linear system Eq.(9) be expressed as summation of the three matrices

$$A = D - L - V \tag{14}$$

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

From the definition in Eq.(14), QSSOR iterative method can be defined generally as [20]:

$$\underset{\sim}{\overset{(k+1)}{U}} = \left(D - \omega L\right)^{-1} \left(\left[\left(1-\omega\right) D + V\omega\right] \underset{\sim}{\overset{(k)}{U}} + \omega f\right) \tag{15}$$

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

Algorithm 1:QSSOR Method	
i.	Initializing all the parameters. Set $k = 0$.
ii.	For $i = 4, 8, 12, \ldots, m-8, m-4$ and $j = 1, 2, \ldots, n-1, n$Calculate $\underset{\sim}{\overset{(k+1)}{U}} = (D - \omega L)^{-1}\left([(1-\omega)D + V\omega]\underset{\sim}{\overset{(k)}{U}} + \omega f\right)$
iii.	Convergence test. If the convergence criterion i.e. $\left\Vert \underset{\sim}{\overset{(k+1)}{U}} - \underset{\sim}{\overset{(k)}{U}} \right\Vert \leq \varepsilon = 10^{-10}$is satisfied, go to Step (iv). Otherwise go back to Step (ii).
iv.	Compute the remaining point via second order Lagrange scheme.
v.	Display approximate solutions.

6 Numerical Experiments

For the comparison purpose, two examples of time-fractional diffusion equations were considered. Both examples will be chosen from well-posed equations. Also three different proposed iterative methods such as FSSOR, HSSOR and QSSOR will be implemented. In this paper, we will consider different values of $\alpha = 0.25, 0.50$ and 0.75.For implementation of these three iterative schemes, the convergence test considered the tolerance error, which is fixed as ϵ=10^{-10}.

Examples 1:[26]

Consider the following time fractional initial boundary value problem be given as

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} = \frac{\partial^2 U(x,t)}{\partial x^2}, \quad 0 < \alpha \leq 1,\ 0 \leq x \leq \gamma, \quad t > 0, \tag{16}$$

where the boundary conditions are given in fractional terms

$$U(0,t) = \frac{2kt^\alpha}{\Gamma(\alpha+1)}, U(\ell,t) = \ell^2 + \frac{2kt^\alpha}{\Gamma(\alpha+1)}, \tag{17}$$

and the initial condition

$$U(x,0) = x^2. \tag{18}$$

From Problem (16), as taking $\alpha = 1$, it can be seen that problem (16) can be reduced to the standard diffusion equation

$$\frac{\partial U(x,t)}{\partial t} = \frac{\partial^2 U(x,t)}{\partial x^2}, \quad 0 \leq x \leq \gamma, \quad t > 0, \tag{19}$$

with the initial and boundary conditions

$$U\left(x,0\right)=x^2, U(0,t)=2kt, U(\ell,t)=\ell^2+2kt.$$

Then the analytical solution of Problem (19)is obtained as follows

$$U(x,t)=x^2+2kt.$$

Now by applying the series

$$U(x,t)=\sum_{n=0}^{m-1}\frac{\partial^n U(x,0)}{\partial t^n}\frac{t^n}{n!}+\sum_{n=1}^{\infty}\sum_{i=0}^{m-1}\frac{\partial^{mn+i}U(x,0)}{\partial t^{mn+i}}\frac{t^{n\alpha+i}}{\Gamma(n\alpha+i+1)}$$

to $U(x,t)$ for $0<\alpha\leq 1$, it can be shown that the analytical solution of Problem (16) is given as

$$U(x,t)=x^2+2k\frac{t^\alpha}{\Gamma(\alpha+1)}.$$

Examples 2:[26]

Let us consider the following time fractional initial boundary value problem be defined as

$$\frac{\partial^\alpha U\left(x,t\right)}{\partial t^\alpha}=\frac{1}{2}x^2\frac{\partial^2 U\left(x,t\right)}{\partial x^2},\quad 0<\alpha\leq 1,\ 0\leq x\leq\gamma,\quad t>0, \tag{20}$$

where the boundary conditions are given in fractional terms

$$U(0,t)=0, U(1,t)=e^t, \tag{21}$$

and the initial condition

$$U\left(x,0\right)=x^2. \tag{22}$$

From Problem (20), as taking $\alpha=1$, it can be shown that Eq.(20) can also be reduced to the standard diffusion equation

$$\frac{\partial U\left(x,t\right)}{\partial t}=\frac{1}{2}x^2\frac{\partial^2 U\left(x,t\right)}{\partial x^2},\quad 0\leq x\leq\gamma,\quad t>0. \tag{23}$$

Then the analytical solution of Problem (23) is obtained as follows

$$U(x,t)=x^2e^t.$$

Now by applying the series

$$U(x,t)=\sum_{n=0}^{m-1}\frac{\partial^n U(x,0)}{\partial t^n}\frac{t^n}{n!}+\sum_{n=1}^{\infty}\sum_{i=0}^{m-1}\frac{\partial^{mn+i}U(x,0)}{\partial t^{mn+i}}\frac{t^{n\alpha+i}}{\Gamma(n\alpha+i+1)}$$

to $U(x,t)$ for $0 < \alpha \leq 1$, it can be shown that the analytical solution of Problem (19) is stated as

$$U(x,t) = x^2\left[1 + \frac{t^{\alpha}}{\Gamma(\alpha+1)} + \frac{t^{2\alpha}}{\Gamma(2\alpha+1)} + \frac{t^{3\alpha}}{\Gamma(3\alpha+1)} + ...\right]$$

All results of numerical experiments for Problem (19) and Problem (23), obtained from implementation of FSSOR, HSSOR and QSSOR iterative methods are recorded in Table 1 and Table 2 at different values of mesh sizes, M = 256, 512, 1024, 2048 and 4096.

7 Conclusions

As a conclusion for the numerical solution of the time fractional diffusion problems, this paper deals with the implementation of QSSOR iterative method to solve a linear system generated by the Quarter-Sweep Caputos implicit approximation equations. Through numerical experiments results from Table 1 by comparing the performance between the QSSOR, HSSOR and FSSOR iterative methods at three different values of $\alpha = 0.25, 0.50$ and 0.75, it can be seen that the percentage reduction of number of iterations for the QSSOR iterative method have declined approximately by $51.12 - 98.45\%$, $51.06 - 98.73\%$ and $51.21 - 98.36\%$ respectively as compared with the FSSOR method. In fact, implementations of computational time for QSSOR method are much faster about $29.32 - 97.97\%$, $33.66 - 98.19\%$ and $32.52 - 98.04\%$ respectively than FSSOR method.

In fact, the numerical experiments results from Table 2 show that the percentage reduction of number of iterations for the QSSOR iterative method have declined approximately by$50.36 - 98.32\%$,$51.26 - 98.58\%$ and $50.27 - 97.64\%$ respectively as compared with the FSSOR methods. Also, implementations of computational time for QSSOR method are much faster about $55.61 - 97.92\%$, $40.32 - 97.85\%$ and $30.13 - 98.07\%$ respectively than the FSSOR method. How it can be concluded that the QSSOR method involves less number of iterations and computational time as compared with HSSOR and FSSOR methods. According to the accuracy of three iterative methods, it can be stated that their numerical solutions are in good agreement

References

[1] Mainardi, "Fractal and fractional calculus continuum mechanics," *Open Access Springer Verlag*, pp. 291–348, 1997.

[2] K.Diethelm and A. Freed, "On the solution of linear fractional order differential equation used in the modelling of viscoelasticity," *Chemical Engineering II Computational Fluid Dynamic, Reaction Engineering and Molecular Properties*, pp. 217–244, 2014.

[3] M. Meerschaert and C. Tadjeran, "Finite difference approximation for fractional advection-dispersion flow equation," *Journal of Computer Applied Mathematics*, pp. 145–155, 2004.

[4] A.Chaves, "Fractional diffuion equation to describe levy fligt," *Physics Letter*, no. A239, pp. 13–16, 1998.

[5] O.P.Agrawal, "Solution for a fractional diffusion-wave equation difine in a bounded domain," *Nonlinear Dynamic*, pp. 145–155, 2002.

[6] S.B.Yuste and L.Acedo, "An explicit finite difference method and a new von neumann-type stability analysis for fractional diffusion equations," *Journal of Numerical Analysis*, pp. 1862–1874, 2014.

[7] S.B.Yuste, "Weighted average finite difference method for fractional diffusion equations," *Journal Computer Physics*, pp. 264–274, 2006.

[8] Y. Zhang, "A finite difference method for fractional partial differential equation," *Applied Mathematics and Computation*, pp. 524–529, 2009.

[9] D.M.Young, *Iterative Solution of Large Linear Systems.* Academic Press, 1971.

[10] Hackbusch, *Iterative Solution of Large Sparse Systems Equations.* Springer Verlag, 1995.

[11] Y.Saad, *Iterative Method for Sparse Linear Systems.* International Thomas Publishing, 1996.

[12] D.J.Evans, "Group explicit iterative methods for solving large linear systems," *International Journal Computer Mathematics*, pp. 81–108, 1985.

[13] A.Ibrahim and A.R.Abdullah, "Solving the two dimensional diffusion equation by the four point explicit decoupled group (edg) iterative method," *International Journal Computer Mathematics*, pp. 253–256, 1995.

[14] W.S.Yousif and D.J.Evans, "Explicit group iterative methods for solving elliptic partial differential equations in 3-space dimensions," *International Journal Computer Mathematics*, pp. 323–340, 1986.

[15] ——, "Explicit de-coupled group iterative methods and their implementations, parallel algorithm and applications," *International Journal Computer Mathematics*, pp. 53–71, 1995.

[16] A.R.Abdullah, "The four point explicit de-coupled group (edg) methods : A fast poisson solver," *International Journal Computer Mathematics*, pp. 61–70, 1991.

[17] M.Othman and A.R.Abdullah, "An efficient four points modified explicit group (mdg) poisson solver," *International Journal Computer Mathematics*, pp. 203–217, 2000.

[18] N.I.M.Fauzi and J. Sulaiman, "Quarter-sweep modified sor iterative algorithm and cubic spline basic for the solution on second order two-point boundary value problems," *Journal of Applied Science*, pp. 1–8, 2012.

[19] M. Muthuvalu and J. Sulaiman, "Numerical solutions of first kind linear fredholm integral equations by using quarter-sweep sor with piecewise linear collocation method qssor with piecewise linear collocation method for fredholm equations," *Australian Journal of Basic and Applied Science*, pp. 18–24, 2012.

[20] N.I.M.Fauzi and J. Sulaiman, "Cubic spline solutions for two-point boundary value problems using quarter-sweep sor method," *International Journal Open Problems Comptuter Mathematics*, pp. 199–209, 2012.

[21] M.K.M.Akhir, M.Othman, Z. A.M, M.Suleiman, and J.Suliaman, "An efficient quarter-sweep modified sor iterative method for solving helmholtz equation," *International Journal of Pure and Applied Mathematics*, pp. 751–764, 2012.

[22] J.Sulaiman, M.K.Hasan, M.Othman, and S.A.A.Karim, "Fourth-order qsmsor iterative method for the solution of one-dimensional parabolic pdes," in *Proceeding of Conference on Industrial and Applied Mathematics, Indonesia*, 2010, pp. 34–39.

[23] A.M.Murio, "Implicit finite difference approximation for time fractional diffusion equations," *International Journal of Pure and Applied Mathematics*, pp. 1138–1145, 2008.

[24] T.A.M.Langlands and B.I.Henry, "The accuracy and stability of an implicit solution method for the fractional diffusion equation," *Journal of Computational Physics*, pp. 719–736, 2005.

[25] R.D.Richtmyer and K.W.Mortons, *Difference Methods for Initial-Value Problems.* Inter science Publishers, 1967.

[26] A. Demir, B. Ozgur, S. Eman, and E. Korkmaz, "Analysis of fractional partial differential equations by taylor series expansion, boundary value problem," *Boundary Value Problems a Springer Open Journal*, vol. 1, no. 68, pp. 68–81, 2013.

International Journal of Mathematical Engineering and Science (IJMES)
Volume 3 Issue 2 (February 2014) ISSN : 2277-6982
http://www.ijmes.com/

Table 1: Comparison of number iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example at $\alpha = 0.25$, 0.50, 0.75

Example 1										
M	Method	$\alpha = 0.25$			$\alpha = 0.50$			$\alpha = 0.75$		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	FSSOR	714	2.08	9.95e-5	703	1.99	9.84e-5	705	2.03	1.29e-4
	HSSOR	349	1.47	9.95e-5	344	1.32	9.84e-5	344	1.37	1.29e-4
	QSSOR	169	0.54	9.95e-5	169	0.51	9.84e-5	168	0.51	1.29e-4
256	FSSOR	1461	6.90	9.95e-5	769	4.01	9.84e-5	769	4	1.29e-4
	HSSOR	357	2.12	9.95e-5	351	2.09	9.84e-5	352	2.16	1.29e-4
	QSSOR	174	0.68	9.95e-5	174	0.68	9.84e-5	172	0.68	1.29e-4
512	FSSOR	6239	55.97	9.96e-5	3951	35.08	9.84e-5	1821	16.51	1.29e-4
	HSSOR	730	8.22	9.95e-5	384	4.72	9.84e-5	384	4.70	1.29e-4
	QSSOR	178	2.48	9.95e-5	175	1.21	9.84e-5	176	1.20	1.29e-4
1024	FSSOR	23626	415.43	9.97e-5	15229	268.33	9.86e-5	7417	129.41	1.30e-4
	HSSOR	3119	64.5	9.96e-5	1975	41	9.84e-5	910	19.12	1.29e-4
	QSSOR	365	8.42	9.95e-5	192	4.83	9.84e-5	192	4.82	1.29e-4
2048	FSSOR	87221	3204.74	9.95e-5	56530	2058.11	9.91e-5	27855	1006.92	1.30e-4
	HSSOR	11813	489.19	9.97e-5	7614	312.34	9.85e-5	3708	151.33	1.30e-4
	QSSOR	1559	66.44	9.96e-5	987	41.88	9.84e-5	455	19.69	1.29e-4

Table 2: Comparison of number iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example at $\alpha = 0.25$, 0.50, 0.75

Example 1										
M	Method	$\alpha = 0.25$			$\alpha = 0.50$			$\alpha = 0.75$		
		K	Time	Max Error	K	Time	Max Error	K	Time	Max Error
128	FSSOR	683	6.06	1.95e-2	671	6.04	8.30e-2	671	6.06	1.37e-1
	HSSOR	339	2.69	1.95e-2	327	6.29	8.30e-2	323	2.68	1.37e-1
	QSSOR	168	1.26	1.95e-2	162	1.26	8.30e-2	160	0.61	1.37e-1
256	FSSOR	781	8.42	1.95e-2	762	8.30	8.30e-2	724	8.18	1.37e-2
	HSSOR	341	3.03	1.95e-2	335	3.05	8.30e-2	335	3.02	1.37e-2
	QSSOR	169	1.32	1.95e-2	163	1.33	8.30e-2	161	1.33	1.37e-2
512	FSSOR	2322	27.84	1.95e-2	945	14.36	8.30e-2	728	12.08	8.30e-2
	HSSOR	390	9.35	1.95e-2	381	8.57	8.30e-2	362	8.44	1.37e-2
	QSSOR	170	3.15	1.95e-2	167	3.15	8.30e-2	167	3.15	1.37e-2
1024	FSSOR	9260	180.21	1.95e-2	4391	90.41	8.30e-2	1925	43.11	1.37e-2
	HSSOR	1161	28.92	1.95e-2	472	14.97	8.30e-2	364	12.27	1.37e-2
	QSSOR	195	8.82	1.95e-2	190	8.70	8.30e-2	181	4.28	1.37e-2
2048	FSSOR	34577	1372.44	1.95e-2	16881	684.52	8.30e-2	7718	325.23	1.37e-2
	HSSOR	4630	199.11	1.95e-2	2195	99.27	8.30e-2	962	44.93	1.37e-2
	QSSOR	580	28.45	1.95e-2	236	14.69	8.30e-2	182	6.26	1.37e-2