Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 12, Number 4 (2016), pp. 3469-3479

http://www.ripublication.com/gjpam.htm

Caputo's Implicit Solution of Time-Fractional Diffusion Equation Using Half-Sweep AOR Iteration

[1]A. Sunarto*, [2]J. Sulaiaman and [3]A. Saudi

[1,2] Mathematics with Economic Programme,
Faculty of Science and Natural Resources,
Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia
**Corresponding Author:*
[3]Software Engineering Programme,
Faculty of Computing and Informatics
Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

Abstract

The aim of this paper deals with the application of Half-Sweep Accelerated Over- Relaxation (HSAOR) iterative method using an unconditionally implicit finite difference approximation equation from the discretization of the one-dimensional linear time-fractional diffusion equations by using the Caputo's time fractional derivative. The Caputo's implicit finite difference approximation equation leads a linear system which is solved by using the proposed HSAOR iterative method. Throughout implementations of two numerical experiments conducted, it has shown that the HSAOR method is superior as compared with FSAOR and FSOR methods.

Keywords: Caputo's fractional derivative; Implicit finite difference; HSAOR method.

1. Introduction

From previous studies [1,2,3,4,5] many scientific and engineering have proposed to get fractional partial differential equations (FPDE's) a numerical and/or analytical approximate solutions of the fractional problems. Actually various numerical techniques that can be used to solve the time fractional diffusion equations (TFDE's), such as transform methods [6], finite elements together with the method of lines [3], explicit and implicit finite difference methods [3,7]. Based on the finite difference methods, there exist three discretization schemes can be constructed such as explicit, semi-implicit and implicit.

To get a finite difference solution of the time-fractional diffusion equations (TFDE's), problem needs to be discretized via finite difference discretization scheme by imposing the implicit finite difference scheme and Caputo's fractional operator. Then corresponding approximation equations can be used to construct a linear system at each time level. To solve the linear system, various iterative methods have been proposed and discussed by Young [8], Hackbusch [9] and Saad [10]. From the previous studies of iterations, there exists several families of iterative methods. In addition to these iterative methods, the concept of block iteration has also been introduced by Evans [11], and furthermore explanation of this block concept have been extended by Ibrahim and Abdullah [12], Yousif and Evans [13,14] in which these block iterative methods can be one of the efficient iterative methods.
To improve the convergence rate of iterative process the half-sweep iteration concept is introduced by Abdullah [15] via Explicit Decoupled Group (EDG), iterative method to solve two dimensional Poisson equations. Due to the advantage of this concept, this half-sweep iterative methods have been extensively studied by many researchers; see Ibrahim and Abdulah [12], Yousif and Evans [13,14], Othman and Abdullah [16], Sulaiman, Hasan and Othman [17], Aruchunan and Sulaiman [18,19], Muthuvalu and Sulaiman [20], Saudi and Sulaiman [21,22,23], Fauzi and Sulaiman [24], Hasan and YitHoe [25], Sulaiman [26] and Akhir [27].
As mentioned the advantages of AOR method [28] and half-sweep iteration from previous studies, we examine the application of the Half-Sweep Accelerated Over-Relaxation (HSAOR) iterative method for solving time-fractional diffusion equations (TFDE's) by using the Caputo's implicit finite difference approximation equation. To test performance of the HSAOR method, we also implement the Full-Sweep Successive Over-Relaxation (FSSOR) iterative method being used as control methods and Full-Sweep Accelerated Over-Relaxation (FSAOR) iterative method.
To investigate the performance of HSAOR method, let us consider time-fractional diffusion equation (TFDE's) be defined as

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

where *a(x), b(x)* and *c(x)* are known functions or constants, whereas α is a parameter which refers to the fractional order of time derivative.
The outline of this paper is as follows: In Sections 2 and 3 the preliminaries and the Half-Sweep Caputo's implicit approximation equation are presented. Then Section 4 deals with derivation of family of AOR methods. In which, formulation of the HSAOR iterative method is introduced. In Section 5 shows numerical example and its results and conclusion is given in Section 6.

2. Preliminaries

Before constructing the approximation equation of problem (1), the following are Half-Sweep Caputo's implicit finite difference and some basic definitions given for fractional derivative theory

Definition 1. [29] The Riemann-Liouville fractional integral operator, J^{α} of order-α is defined as

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

Definition 2.[29] The Caputo's fractional partial derivative operator, D^{α} of order-α is defined as

$$D^{\alpha} f(x) = \frac{1}{\Gamma(m-\alpha)} \int_{0}^{x} \frac{f^{(m)}(t)}{(x-t)^{\alpha-m+1}} dt, \alpha > 0 \tag{3}$$

with $m-1 < \alpha \le m, \mathrm{m} \in \mathrm{N},\ x > 0$

As mentioned in the previous section of getting numerical solution of problem (1), firstly, we construct implicit finite difference approximate equation based on the Caputo's derivative definition with Dirichlet boundary conditions and consider the non-local fractional derivative operator. This approximation equation can be categoried as unconditionally stable scheme. To solve problem (1), the solution domain of the problem has been restricted to the finite space domain $0 \le x \le \gamma$, with $0 < \alpha < 1$, whereas the parameter α refers to the fractional order of time derivative. Consider problem (1), associated with the boundary and initial conditions as follow

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

and the initial condition

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

where $g_0(t), g_1(t)$ and $f(x)$ are given functions. To get a discretize approximation to the time fractional derivative in Eq. (1), we consider Caputo's fractional partial derivative of order α, which is defined as [29,8]

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

3. Half-Sweep Caputo's Implicit Finite Difference Approximation

To derive the Caputo's implicit finite difference approximation section, let us consider Eq.(4) to show the formulation of Caputo's fractional partial derivative of the first order approximation method which is given as

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

where

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

and

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

To facilitate us in 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 $x_i = ih,\ i = 0,1,2,\ldots,m$ and the grid points in the time interval $[0,T]$ are labeled $t_j = jk,\ j = 0,1,2,\ldots,n$. Then the values of the function $U(x,t)$ at the grid points are denoted as $U_{i,j} = U(x_i, t_j)$. By using the formulation in Eq. (5) and the implicit finite difference discretization scheme, the Half-Sweep Caputo's implicit finite difference approximation equation of problem (1) to the reference grid point at $(x_i, t_j) = (ih, jk)$ can be shown as

$$\sigma_{\alpha,k} \sum_{j=1}^{n} \omega_j^{(\alpha)} \left(U_{i,n-j+1} - U_{i,n-j}\right)$$
$$= a_i \frac{1}{4h^2}\left(U_{i-2,n} - 2U_{i,n} + U_{i+2,n}\right) + b_i \frac{1}{4h}\left(U_{i+2,n} - U_{i-2,n}\right) + c_i U_{i,n}, \quad (6)$$

for $i=2,4...,m-2$.

Actually, this half-sweep approximation equation is classified as the fully implicit finite difference approximation equation which is consistent first order accuracy in time and second order in space. To construct the linear system, then the approximation equation (6) can be rewritten based on the specified time level. For instance, we have for $n \geq 2$:

$$\sigma_{\alpha,k} \sum_{j=1}^{n} \omega_j^{(\alpha)} \left(U_{i,n-j+1} - U_{i,n-j}\right) = p_i U_{i-2,n} + q_i U_{i,n} + r_i U_{i+2,n}, \quad (7)$$

where

$$p_i = \frac{a_i}{4h^2} - \frac{b_i}{4h},$$

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

$$r_i = \frac{a_i}{4h^2} + \frac{b_i}{4h}$$

Also, we get for $n = 1$,

$$-p_i U_{i-2,1} + q_i^* U_{i,1} - r_i U_{i+2,1} = f_{i,0}, \quad i = 2,4...,m-2 \quad (8)$$

where

$$\omega_j^{(\alpha)} = 1,\ q_i^* = \sigma_{\alpha,k} - q_i,\ f_{i,0} = \sigma_{\alpha,k} U_{i,0}.$$

For the convenience in solving linear systems in Eq. (7) and (8), Let us consider the approximation equation (8) being used in constructing the tridiagonal matrix form as

$$A\underset{\sim}{U} = \underset{\sim}{f} \quad (9)$$

where

$$A = \begin{bmatrix} q_2^* & -r_2 & & & & \\ -p_4 & q_4^* & -r_4 & & & \\ & -p_6 & q_6^* & -r_6 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -p_{m-4} & q_{m-4}^* & -r_{m-4} \\ & & & & -p_{m-2} & q_{m-2}^* \end{bmatrix}_{\left(\left(\frac{m}{2}\right)-1\right)x\left(\left(\frac{m}{2}\right)-1\right)},$$

$$\underset{\sim}{U} = \begin{bmatrix} U_{2,1} & U_{4,1} & U_{6,1} & \cdots & U_{m-4,1} & U_{m-2,1} \end{bmatrix}^T,$$

$$\underset{\sim}{f} = \begin{bmatrix} f_{21} + p_1 U_{01} & f_{41} & f_{61} & \cdots & f_{m-4,1} & f_{m-2,1} + p_{m-2} U_{m,1} \end{bmatrix}^T,$$

4. Formulation of Half-Sweep Accelerated Over-Relaxation

Refer to the linear system (9), it can be seen that the characteristic of the coefficient matrix of the linear system has large scale and sparse. It means that the iterative methods are suitable option to solve the linear system [8]. Therefore, we consider the application of HSAOR method as linear solver (9). Particularly, the HSAOR method is essentially the extension of the FSAOR iterative method. The main purpose of the half-sweep iteration is to reduce the computational complexities during iteration process. Development of the HSAOR method is the combination between the half-sweep iteration concept and Accelerated Over-Relaxation (AOR) method. Let the linear system (9) be expressed as summation of the three matrices

$$A = D - L - V \qquad (10)$$

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

According to Eq. (10), the HSAOR iterative method can be defined generally as [28] :

$$\underset{\sim}{U}^{(k+1)} = (D - \omega L)^{-1}\left[\beta V + (\beta - \omega)L + (1 - \beta)D\right]\underset{\sim}{U}^{(k)} + \beta (D - \omega L)^{-1} f \qquad (11)$$

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

Algorithm 1: HSAOR method

i. Initializing all the parameters. Set $k = 0$.

ii. For $j = 1,2,\ldots,n-1,n$ and

$i = 2,4,,\ldots,m-4,m-2$ Calculate

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

iii. Convergence test. If the convergence criterion i.e

$\left\| \underset{\sim}{U}^{(k+1)} - \underset{\sim}{U}^{(k)} \right\| \leq \varepsilon = 10^{-10}$ is satisfied, go to Step (iv).

Otherwise go back to Step (ii).

iv. Display approximate solutions.

5. Numerical Experiment

In order to investigate the performance of the proposed iterative methods, we consider two examples of the time fractional diffusion equations being used to demonstrate effectiveness of the HSAOR compared with FSAOR and FSSOR iterative methods. To do this, three criteria have been considered such as number of iterations, execution time (in seconds) and maximum absolute error at three 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 $\varepsilon = 10^{-10}$.

Examples 1:[30]
Consider the following time fractional initial boundary value problem be given as

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} = \frac{\partial^2 U(x,t)}{\partial x^2}, \quad 0 < \alpha \le 1, 0 \le x \le \gamma, \quad t > 0, \tag{12}$$

where the boundary conditions are given in fractional terms

$$U(0,t) = \frac{2kt^\alpha}{\Gamma(\alpha+1)}, U(\ell,t) = \ell^2 + \frac{2kt^\alpha}{\Gamma(\alpha+1)}, \tag{13a}$$

and the initial condition

$$U(x,0) = x^2. \tag{13b}$$

From Problem (12), as taking $\alpha = 1$, it can be seen that problem (12) can be reduced to the standard diffusion equation

$$\frac{\partial U(x,t)}{\partial t} = \frac{\partial^2 U(x,t)}{\partial x^2}, \quad 0 \le x \le \gamma, \quad t > 0, \tag{14}$$

with the initial and boundary conditions

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

Then the analytical solution of Problem (14) 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 $0 < \alpha \le 1,$ it can be shown that the analytical solution of problem (14) is given as

$$U(x,t) = x^2 + 2k\frac{t^\alpha}{\Gamma(\alpha+1)}. \tag{15}$$

Examples 2: [30]
Let us consider the following time fractional initial boundary value problem be defined as

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

where the boundary conditions are given in fractional terms

$$U(0,t) = 0, \; U(1,t) = e^t, \tag{17a}$$

and the initial condition

$$U(x,0) = x^2 . \qquad (17b)$$

From problem (16), as taking $\alpha = 1$, it can be shown that Eq. (16) can also be reduced to the standard diffusion equation

$$\frac{\partial U(x,t)}{\partial t} = \frac{1}{2} x^2 \frac{\partial^2 U(x,t)}{\partial x^2}, \quad 0 \le x \le \gamma, \quad t > 0 . \qquad (18)$$

Then the analytical solution of problem (18) is obtained as follows

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

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 $0 < \alpha \le 1$, it can be shown that the analytical solution of problem (18) is stated as

$$U(x,t) = x^2 \left[1 + \frac{t^{\alpha}}{\Gamma(\alpha+1)} + \frac{t^{2\alpha}}{\Gamma(2\alpha+1)} + \frac{t^{3\alpha}}{\Gamma(3\alpha+1)} + \ldots \right] \qquad (19)$$

All the simulations were implemented by C programming language. Results of numerical simulations, which were obtained from implementation of FSSOR, FSAOR and HSAOR iterative methods have been recorded in Tables 1 and 2 at different values of mesh sizes, *m* = 128, 256, 512, 1024, and 2048.

Table 1: Comparison of Number Iterations (K), The Execution Time (seconds) and Maximum Errors for The Iterative Methods Using Examples 1 at $\alpha = 0.25, 0.50, 0.75$.

M	**Method**	**α = 0.25**			**α = 0.50**			**α = 0.75**		
		K	**Time**	**Max Error**	**K**	**Time**	**Max Error**	**K**	**Time**	**Max Error**
128	**FSSOR**	714	2.00	9.95e-05	703	1.97	9.84e-05	705	2.04	1.29e-04
	FSAOR	657	1.93	9.95e-05	615	1.83	9.84e-05	613	1.79	1.29e-04
	HSAOR	517	1.78	9.95e-05	510	1.54	9.84e-05	599	1.39	1.29e-04
256	**FSSOR**	1461	6.80	9.95e-05	769	3.98	9.84e-05	769	3.97	1.29e-04
	FSAOR	870	6.38	9.95e-05	659	2.99	9.84e-05	630	1.72	1.29e-04
	HSAOR	857	3.02	9.95e-05	425	2.93	9.84e-05	413	1.50	1.29e-04
512	**FSSOR**	6239	56.20	9.96e-05	3951	35.05	9.84e-05	1821	16.77	1.29e-04
	FSAOR	4940	31.35	9.96e-05	3080	22.62	9.84e-05	1301	16.67	1.29e-04
	HSAOR	870	6.54	9.05-e05	859	6.37	9.84e-05	830	6.45	1.29e-04
1024	**FSSOR**	23626	418.56	9.97e-05	15229	266.77	9.85e-05	7417	129.5	1.30e-04
	FSAOR	19033	4.72	9.97e-05	12232	101.21	9.80-e05	5911	105.27	1.30e-04
	HSAOR	4940	3.28	9.96e-05	3080	42.68	9.80e-05	1301	17.87	1.29e-04
2048	**FSSOR**	87211	3356.96	1.00e-04	56530	2086.15	9.91e-05	27855	1047.77	1.30e-04
	FSAOR	70547	1142.09	1.00e-04	45700	1342	9.89e-05	22474	1002.85	1.30e-04
	HSAOR	19033	512.59	1.00e-04	12232	338.14	9.85e-05	5911	157.02	1.30e-04

From the numerical result recorded in Table 1 by imposing the FSSOR, FSAOR and HSAOR iterative methods, it is obvious at $\alpha = 0.25$ that number of iterations have declined approximately by 27.59-86.05% corresponds to the HSAOR iterative method as compared with FSSOR methods. Particularly in terms of execution time, implementations of HSAOR method are much faster about 11.10-99.21% than FSSOR methods. It means that the HSAOR method requires the less amount for number of iterations and computational time as compared with FSAOR and FSSOR iterative methods. For other value of $\alpha = 0.50, 0.75$, it can be observed that their conclusions are in line with $\alpha = 0.25$.

Table 2: Comparison of Number Iterations (K), The Execution Time (seconds) and Maximum Errors for The Iterative Methods Using Examples 2 at $\alpha = 0.25, 0.50, 0.75$.

M	**Method**	**α = 0.25**			**α = 0.50**			**α = 0.75**		
		K	**Time**	**Max Error**	**K**	**Time**	**Max Error**	**K**	**Time**	**Max Error**
128	**FSSOR**	683	8.06	1.95e-02	671	6.04	8.30e-02	671	6.06	1.37e-01
	FSAOR	370	6.19	1.94e-02	260	5.86	8.29e-02	257	5.80	1.37e-01
	HSAOR	199	5.11	1.95e-02	154	5.04	8.30e-02	146	5.02	1.37e-01
256	**FSSOR**	2281	28.42	1.95e-02	962	18.30	8.30e-02	724	8.18	1.37e-01
	FSAOR	1761	19.30	1.95e-02	809	11.82	8.29e-02	291	7.39	1.37e-01
	HSAOR	545	6.25	1.95e-02	328	5.70	8.29e-02	163	5.29	1.37e-01
512	**FSSOR**	8322	127.84	1.95e-02	9245	74.36	8.29e-02	1728	32.08	1.37e-01
	FSAOR	6746	125.09	1.94e-02	3240	63.93	8.29e-02	1471	30.79	1.37e-01
	HSAOR	2246	19.45	1.95e-02	1402	14.27	8.29e-02	625	9.09	1.37e-01
1024	**FSSOR**	29260	980.21	1.95e-02	14391	590.41	8.29e-02	6925	243.11	1.37e-01
	FSAOR	25054	866.80	1.94e-02	12126	435.44	8.29e-02	5644	208.20	1.37e-01
	HSAOR	8478	117.00	1.94e-02	5313	76.75	8.29e-02	2461	38.54	1.37e-01
2048	**FSSOR**	94577	7372.44	1.92e-02	56681	1684.52	8.29e-02	23718	1825.23	1.37e-01
	FSAOR	91984	6092.40	1.94e-02	44563	1428.60	8.29e-02	20921	1609.51	1.37e-01
	HSAOR	40541	1109.00	1.94e-02	19650	557.72	8.29e-02	9198	268.45	1.37e-01

From the numerical result recorded in Table 2, $\alpha = 0.25$ it can be observed $\alpha = 0.25$ that number of iterations have declined approximately by 57.13-76.10% corresponds to the HSAOR iterative method as compared with FSSOR methods. Similar to execution time, implementations of HSAOR method are much faster about 36.60-88.06% than FSSOR methods. It means that the HSAOR method requires the less amount for number of iterations and execution time at as compared with FSAOR and FSSOR iterative methods. For other value of $\alpha = 0.50, 0.75$, it can be concluded that their conclusions are in line with $\alpha = 0.25$.

6. Conclusion

For the numerical solution of the time-fractional diffusion problems, this paper applied the derivation of the Caputo's implicit finite difference approximation equations in which this approximation equation leads a tridiagonal linear system. Based on the numerical result recorded in Tables 1 and 2, it can be pointed out, that the HSAOR method requires the less amount for number of iterations and execution time at as compared with FSAOR and FSSOR iterative methods. The observation on the accuracy of proposed iterative methods show that their numerical solutions are in good agreement.

References

[1] Mainardi, F., 1997, "Fractals and Fractional Calculus Continuum Mechanics," Springer-Verlag, Heidelberg., pp. 291-348.

[2] Diethelm, K., and Freed, A.D., 1999, "Scientific Computing in Chemical Engineering II Computational Fluid Dynamic, Reaction Engineering and Molecular Properties," Springer-Verlag, Heidelberg., pp. 217-224.

[3] Meerschaert, M.M., and Tadjeran, C., 2004, "Finite Difference Approximation for Fractional Advection-Dispersion Flow Equations," Journal Computational and Applied Mathematics., 172**,** pp.145-155.

[4] Chaves, A., 1998, "Fractional Diffusion Equation to Describe Levy flight," Physics Letter., 239**,** pp.13-16.

[5] Agrawal, O.P., 2002, "Solution for a Fractional Diffusion-Wave Equation Defined in a Bounded Domain," Nonlinear Dynamic., 29, pp**.**145-155.

[6] Yuste, S.B., and Acedo, L., 2005, An Explicit Finite Difference Method and a New Von Neumann-type Stability Analysis for Fractional Diffusion Equations," SIAM Journal Numerical Analysis., 42**,** pp.1862-1874.

[7] Yuste, S.B., 2006, "Weighted Average Finite Difference Method for Fractional Diffusion Equations," Journal of Computational Physics., 216, pp.264-274.

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

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

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

[11] Evans, D.J., 1985, "Group Explicit Iterative Methods for Solving Large Linear Systems," International Journal Computer Mathematics., 17**,** pp.81-108.

[12] Ibrahim, A., and Abdullah, A.R., 1995, "Solving the two dimensional diffusion equation by the Four Point Explicit Decoupled Group (EDG) iterative method," International Journal Computer Mathematics., 58, pp.253-256.

[13] Yousif, W.S., and Evans, D.J., 1986, “Explicit Group Iterative Methods for solving elliptic partial differential equations in 3-space dimensions,” International Journal Computer Mathematics., 18, pp.323-340.

[14] Yousif, W.S., and Evan, D.J., 1995, “Explicit De-coupled Group Iterative Methods and Their Implementations, “ Parallel Algorithm and Applications., 7, pp.53-71 (1995).

[15] Abdulah, A.R., 1991, ” The Four Point Explicit Decoupled group (EDG) Method: A Fast Poisson Solver,” International Journal Computer Mathematics., 38**,** pp.61-70.

[16] Othman, M., and Abdullah, A.R., 1998, “The Half-Sweep Multigrid Method as a fast Multigrid Poisson Solver,” International Journal Computer Mathematics., 69*,* pp.319-329.

[17] Sulaiman, J., Hasan, M.K., and Othman, M., 2005, “The Half-Sweep Iterative Alternating Decomposition Explicit (HSIADE) Method for Diffusion Equations,” Computational and Information Science Lecture Note in Computer Science., 3314**,** pp.57-63.

[18] Aruchunan, E., and Sulaiman, J., 2011, “Half-Sweep Conjugate Gradient method for Solving First Order Linear Fredholm Integro-differential Equations,” Australian Journal of Basic and Applied Sciences., 5, pp.38-43.

[19] Aruchunan, E., and Sulaiman, J., “Applications 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., 6, pp.335-339.

[20] Muthuvalu, M.S., and Sulaiman, J., 2012 “Computational Methods Based on Complexity Reduction Approach for First Kind Linear Fredholm Integral Equations with Semi-Smooth Kernel,” Journal of Modern in Numeric Mathematics., 3*,* pp.64-71.

[21] Saudi, A., and Sulaiman, J., 2013, “Robot Path Planning Using Laplacian Behaviour-Based Control (LBBC) via Half-Sweep SOR,” Technological Advances in Electrical, Electronic and Computer Engineering (TAEECE), 2013 International Conference on (IEEE, Konya), pp. 424-429.

[22] Saudi, A., and Sulaiman, J., 2012, “Path Planning for Indoor mobile Robot Using Half-Sweep SOR via Nine-Point Laplacian (HSSOR9L),” IOSR Journal of Mathematics., 3, pp.01-07.

[23] Saudi, A., and Sulaiman, J., 2009, “Path Planning for mobile Robot with Half-Sweep Successive Over-Relaxation (HSSOR) Iterative Method,” In Proceeding of Symposium ion progress in information & Communication Technology (SPICT 2009), Kuala Lumpur, pp. 57-62.

[24] Fauzi, N.I.M., and Sulaiman, J., 2012, “Half-Sweep Modified Successive Over Relaxation Method for Solving Second Order Two-Point Boundary Value Problems using Cubic Spline,” International Journal Cotemporary Mathematics Sciences., 7, pp.1579-1589.

[25] Hasan, M.K., and Yit Hoe, N., 2013, Impact of Alternating Left Right Strategy on Half-Sweep Successive-Over-Relaxation for The Solution of Two

Dimensional Boundary Value Problems," Australian Journal of Basic and Applied Sciences 7, pp.391-398.

[26] Sulaiman, J., Hasan, M.K., Othman, M., and Karim, S.A.A., 2013, Numerical Solutions of Nonlinear Second-Order Two-Point Boundary Value Problems Using Half-Sweep SOR With Newton Method," Journal of Concrete and Applicable Mathematics., 1557, pp.112-120.

[27] Akhir, M.K.M., Othman, M., Sulaiman, J., and Majid, Z.A., 2011, "Half-Sweep Successive Over Relaxations for Solving Two-Dimensional Helmholtz Equation," Australian Journal of Basic and Applied Science., 5, pp.3033-3039.

[28] Hadjidimos, A., 1978, "Accelerated Over-Relaxation Method," Mathematics of Computation., 32, pp.149-157.

[29] Zhang, Y., 2009, 2009, "A Finite Difference Method For Fractional Partial Differential Equation," Applied Mathematics and Computation., 215, pp.524-529.

[30] Ali, S.E., Ozgur, B.E., and Korkmaz, E., 2013, "Computational Methods Based on Complexity Reduction Approach for First Kind Linear Fredholm Integral Equations with Semi-Smooth Kernel," Boundary Value Problem a Springer Open Journal., 68, pp.1687-2770.