IOP Conf. Series: Journal of Physics: Conf. Series **1489** (2020) 012025 doi:10.1088/1742-6596/1489/1/012025

Grünwald Implicit Solution of One-Dimensional Time-Fractional Parabolic Equations Using HSKSOR Iteration

F A Muhiddin[1], J Sulaiman[2] and A Sunarto[3]

[1] Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA Cawangan Sabah, Malaysia.
[1,2] Mathematics with Economics Programme,
Universiti Malaysia Sabah, Malaysia.
[3] Faculty of Islamic Economics and Business,
IAIN Bengkulu, Indonesia.

E-mail: [1]fatihah.anas@uitm.edu.my, [2]jumat@ums.edu.my, [3]andang99@gmail.com

Abstract. This paper presents the application of a half-sweep iteration concept to the Grünwald implicit difference schemes with the Kaudd Successive Over-Relaxation (KSOR) iterative method in solving one-dimensional linear time-fractional parabolic equations. The formulation and implementation of the proposed methods are discussed. In order to validate the performance of HSKSOR, comparisons are made with another two iterative methods, full-sweep KSOR (FSKSOR) and Gauss-Seidel (FSGS) iterative methods. Based on the numerical results of three tested examples, it shows that the HSKSOR is superior compared to FSKSOR and FSGS iterative methods.

1. Introduction

Numerical solution of fractional partial differential equations (FPDEs) could be found in many application areas especially in the fields of engineering, physics and economics. Their broad applications have been mentioned by many researchers, refer [1,2].

In this paper, we focus on the numerical solution of one-dimensional time-fractional parabolic equations (TFPEs), which can generally be defined as follows

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha}+p(x)\frac{\partial U(x,t)}{\partial x}+q(x)\frac{\partial^2 U(x,t)}{\partial x^2}=f(x,t), \qquad t>0, \qquad x\in\Omega=[a,b], \qquad 0<\alpha\le 1, \tag{1}$$

subject to the initial and boundary conditions

$$\begin{aligned} &U(x,0)=g_0(x), && a\le x\le b,\\ &U(a,t)=g_1(t);U(b,t)=g_2(t) && 0\le t\le T,\end{aligned}$$

where $p(x)$ represents a convection coefficient, $q(x)$ a diffusion coefficient and $f(x,t)$ a source term.

In order to solve FPDEs, many numerical methods have been developed. Solving problem (1) using numerical techniques will lead to large sparse linear systems. Normally, it requires the application of iterative solvers. In 2012, Youssef [3] introduced a new version of the Successive Over-Relaxation (SOR) iterative method, which known as Kaudd Successive Over-Relaxation (KSOR) iterative method. Prior to that, in 1991, Abdullah [4] has introduced the half-sweep iteration concept in order to reduce the computational complexities during the iteration process. Due to the reduction of the computational complexities, many investigations on the effectiveness of this iteration concept have been discussed extensively by [5,6,7,8]. Inspired by their works, this paper extended the application of the half-sweep iteration concept to the Kaudd Successive Over-Relaxation (HSKSOR) iterative method by solving the Grünwald implicit approximation equations of the problem (1). Previously, the same method has been applied by [9,10] in their respective problems.

Knowing for the unconditionally stable in both time and space, the implicit finite difference method has been extensively discussed by many researchers before, such as [11,12,13]. To derive the approximation equations of the problem (1), we also have applied the implicit finite difference scheme. However, most of the previous discussions were done based on the Caputo fractional derivative. Meanwhile, in this study, we used the Grünwald fractional operator, which defined by [14,15] as

Definition 1: Grünwald fractional derivative of order-α

$$D_G^{\alpha} f(t) = \frac{1}{(\Delta t)^{\alpha}} \lim_{M\to\infty} \sum_{k=0}^{M} g_{\alpha,k} f(t-k\Delta t), \qquad 0<\alpha<1 \tag{2}$$

where the Grünwald weights are $g_{\alpha,k} = \dfrac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}$.

Throughout the application of the half-sweep concept, only half of all interior node points in the solution domain are considered. Figure 1 shows the difference between the execution of full-and half-sweep iterations. Whereby, the applications are onto the node points of type • only until it reaches a convergence state.

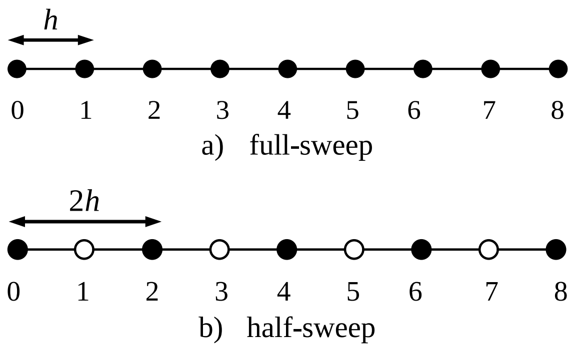

Figure 1: a) and b) show the distribution of uniform node points for the full- and half-sweep cases respectively

2. Half-sweep Grünwald Implicit finite difference approximation equations

In this paper, Grünwald Implicit finite difference scheme is applied to discretize problem (1). To derive the Grünwald implicit approximation equations, first, we divided the solution domain into uniform step-size $h = \dfrac{b-a}{N} = \Delta x$ and $\Delta t = \frac{T}{M}$ in x and t directions respectively, where N and M are some positive integers. Based on these uniformly divided finite grid sizes, the solution domain on the time interval $[0,T]$ and space interval $[a,b]$ are denoted as $t_j = j\Delta t,$ at $j = 0,1,2,\dots,M$ and $x_i = a + ih,$ at $i = 0,1,2,\dots,N$ respectively. Hence, by applying the Grünwald fractional derivative operator to the time and implicit difference scheme in space, the discrete form of problem (1) can be expressed generally as

$$\sum_{k=0}^{j} G_k U_{i,j-k} + \mu_i \left(U_{i+p,j} - U_{i-p,j}\right) + \eta_i \left(U_{i+p,j} - 2U_{i,j} + U_{i-p,j}\right) = f_{i,j} \tag{3}$$

where

$$G_k = \frac{g_{\alpha,k}}{(\Delta t)^{\alpha}},$$

and where, for the full-sweep case at i= 1,2,3,…,N-1 and p=1,

$$\mu_i = \frac{a(x)}{2\Delta x}, \qquad \eta_i = \frac{b(x)}{(\Delta x)^2},$$

while, for the half-sweep case at i=2,4,6,…,N-4,N-2 and p=2,

$$\mu_i = \frac{a(x)}{4\Delta x}, \qquad \eta_i = \frac{b(x)}{(2\Delta x)^2}.$$

Hence, by using the half-sweep Grünwald implicit finite difference scheme, the approximation equation (3) can be simplified as

$$\alpha_i U_{i-2,j} + \beta_i U_{i,j} + \gamma_i U_{i+2,j} = F_{i,j} \tag{4}$$

where

$$\alpha_i = \eta_i - \mu_i, \qquad \beta_i = G_0 - 2\eta_i, \quad \gamma_i = \mu_i + \eta_i,$$

and where

$$F_{i,j} = \begin{cases} f_{i,1} - G_1 U_{i,0} & j = 1 \\ f_{i,j} - \sum_{k=1}^{j} G_k U_{i,j-k} & j = 2,3,\cdots,M \end{cases}$$

Next, the approximation equations (4) can be easily shown in matrix form as follows

$$A\underline{U}_j = \underline{F}_j \tag{5}$$

where

$$A = \begin{bmatrix} \beta_2 & \gamma_2 & & & & & \\ \alpha_4 & \beta_4 & \gamma_4 & & & & \\ & \alpha_6 & \beta_6 & \gamma_6 & & & \\ & & \alpha_8 & \beta_8 & \gamma_8 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \alpha_{N-4} & \beta_{N-4} & \gamma_{N-4} \\ & & & & & \alpha_{N-2} & \beta_{N-2} \end{bmatrix}_{(N-2)\times(N-2)},$$

$$\underline{U}_j = \begin{bmatrix} U_{2,j} & U_{4,j} & U_{6,j} & \cdots & U_{N-4,j} & U_{N-2,j} \end{bmatrix}^T,$$

$$\underline{F}_j = \begin{bmatrix} F_{2,j} - \alpha_2 U_{0,j} & F_{4,j} & F_{6,j} & \cdots & F_{N-4,j} & F_{N-2,j} - \gamma_{N-2} U_{N,j} \end{bmatrix}^T,$$

3. Formulation of the family of Kaudd Successive Over-Relaxation iterative methods

As explained in Section 2, the characteristic of the coefficient matrix A is large and sparse. In this paper, we solve equation (5) using FSKSOR and HSKSOR iterative methods. To verify their effectiveness, the FSGS iterative method performed as a control method.

To derive the FSKSOR and HSKSOR method, first, let the coefficient matrix in equation (5) be decomposed as

$$A = D + L + V \tag{6}$$

where D, L and V represent the diagonal, lower triangular and upper triangular respectively.

Then, recall the family of SOR iterative method which generally stated in [16,17,18,19] as

$$\underline{U}_j^{(k+1)} = (1-\omega)\underline{U}_j^{(k)} + \omega(D+L)^{-1}\left(\underline{F} - V\underline{U}_j^{(k)}\right) \tag{7}$$

Therefore, by using the same the decomposition matrix in equation (6), the general scheme of FSKSOR and HSKSOR iterative method can be written as [3,10]

$$\underline{U}^{(k+1)} = (1+\omega)^{-1}\left(\underline{U}^{(k)} + \omega(D+L)^{-1}\left(\underline{F} - V\underline{U}^{(k)}\right)\right) \tag{8}$$

where ω and $\underline{U}_j^{(k+1)}$ respectively represent the relaxation factor and the unknown vector at k^{th} iteration.

Therefore, by determining the values of matrices D, L and V, the proposed algorithm for the HSKSOR iterative method to solve equation (6) could be described as in Algorithm 1.

Algorithm 1: Half-sweep KSOR scheme

i. Initialize $\underline{U}_i^{(0)} \leftarrow 0$ and $\varepsilon \leftarrow 10^{-10}$.
ii. Assign the optimal value of ω,
iii. For $i = 2,4,6,\cdots,N-4,N-2,$ and $j = 1,2,3,\cdots,M$. Compute:

$$\underline{U}^{(k+1)} = (1+\omega)^{-1}\left(\underline{U}^{(k)} + \omega(D+L)^{-1}\left(\underline{F} - V\underline{U}^{(k)}\right)\right)$$

iv. Perform the convergence test, $\left|\underline{U}_{i,j}{}^{(k+1)} - \underline{U}_{i,j}{}^{(k)}\right| \le \varepsilon = 10^{-10}$. If yes, go to step (v). Otherwise, repeat step (iii).
v. Compute the remaining points (i.e. ○) using, direct method.
vi. Display approximate solutions.

4. Numerical examples

In this section, three examples are tested in order to demonstrate the effectiveness of FSKSOR and HSKSOR iterative methods with the Grünwald implicit difference scheme at three different values of alfas (α=0.333, 0.666, 0.999).

Example 1 [20] Consider the following one-dimensional linear inhomogeneous fractional Burger's equation

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} + \frac{\partial U(x,t)}{\partial x} - \frac{\partial^2 U(x,t)}{\partial x^2} = \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} + 2x - 2.$$

with p(x) =1 and q(x) = -1, and subject to the initial condition $U(x,0) = x^2$. The exact solution is $U(x,t) = x^2 + t^2$.

Example 2 [21] Consider the following one-dimensional inhomogeneous time-fractional parabolic equation

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} - \frac{\partial^2 U(x,t)}{\partial x^2} = f(x,t),$$

with p(x) =0 and q(x) = -1, while $f(x,t) = \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} t^{2-\alpha}\sin(2\pi x) + 4\pi^2 t^2 \sin(2\pi x),$

and subject to the initial condition $U(x,0) = 0$. The exact solution is given by $U(x,t) = t^2\sin(2\pi x)$.

Example 3 [22] Consider the following one-dimensional linear inhomogeneous time-fractional parabolic equation

$$\frac{\partial^\alpha U(x,t)}{\partial t^\alpha} - \frac{\partial^2 U(x,t)}{\partial x^2} = f(x,t), \qquad t > 0, x \in R, 0 < \alpha \le 1,$$

with p(x) =0 and q(x) = -1, while $f(x,t) = \frac{2e^x t^{2-\alpha}}{\Gamma(3-\alpha)} - t^2 e^x,$

subject to the initial condition $U(x,0) = 0$. The exact solution is given by $U(x,t) = t^2 e^x$.

Three parameters are considered for comparison which are the iteration numbers (k), computation time in seconds (time) and maximum absolute error. In the implementation of the iterative methods, the convergence test considered the tolerance error, $\varepsilon=10^{-10}$. Meanwhile, the optimal values of ω were pre-selected by choosing the one with the smallest iteration numbers. Then, the numerical results obtained from the implementation of FSKSOR and HSKSOR to examples 1 to 3 are recorded as in Tables 1 to 3 respectively. Whereas, Table 4 shows the decrement percentage of the iteration numbers and computation time.

Table 1. Comparison of Number Iterations (k), execution time (in seconds) and maximum absolute error for the iterative methods using example 1 at α=0.333, 0.666, 0.999.

M	Method	α=0.333			α=0.666			α=0.999		
		k	t	Max Error	k	t	Max Error	k	t	Max Error
128	FSGS	18325	7.89	2.5972e-02	8957	5.36	1.3065e-02	2772	3.69	1.2478e-03
	FSKSOR	404 (ω=-2.0613)	3.20	2.5972e-02	283 (ω=-2.0909)	3.11	1.3065e-02	166 (ω=-2.1743)	3.14	1.2480e-03
	HSKSOR	199 (ω=-2.1261)	1.88	2.5971e-02	140 (ω=-2.1899)	1.87	1.3065e-02	86 (ω=-2.3794)	1.87	1.2480e-03
256	FSGS	67139	41.90	2.5973e-02	32944	23.62	1.3066e-02	10244	11.42	1.2473e-03
	FSKSOR	813 (ω=-2.0303)	6.67	2.5972e-02	569 (ω=-2.0446)	6.47	1.3065e-02	331 (ω=-2.0836)	6.33	1.2480e-03
	HSKSOR	404 (ω=-2.0613)	3.76	2.5972e-02	283 (ω=-2.0909)	3.72	1.3065e-02	166 (ω=-2.1743)	3.68	1.2480e-03
512	FSGS	243922	276.40	2.5975e-02	120271	142.56	1.3067e-02	37649	53.24	1.2454e-03
	FSKSOR	1621 (ω=-2.0151)	14.51	2.5972e-02	1136 (ω=-2.0221)	13.63	1.3065e-02	659 (ω=-2.0411)	13.08	1.2480e-03
	HSKSOR	813 (ω=-2.0303)	7.71	2.5972e-02	569 (ω=-2.0446)	7.59	1.3065e-02	331 (ω=-2.0836)	7.45	1.2480e-03
1024	FSGS	877165	1914.74	2.5981e-02	435083	965.24	1.3074e-02	137338	323.96	1.2376e-03
	FSKSOR	3246 (ω=-2.0076)	34.07	2.5972e-02	2269 (ω=-2.0110)	30.86	1.3065e-02	1311 (ω=-2.0204)	27.70	1.2480e-03
	HSKSOR	1621 (ω=-2.0151)	16.27	2.5972e-02	1136 (ω=-2.0221)	15.76	1.3065e-02	659 (ω=-2.0411)	15.25	1.2480e-03
2048	FSGS	3114564	13356.94	2.6008e-02	1556326	6730.67	1.3102e-02	496352	2216.78	1.2066e-03
	FSKSOR	6365 (ω=-2.0076)	87.08	2.5972e-02	4519 (ω=-2.0110)	75.36	1.3065e-02	2618 (ω=-2.0204)	63.69	1.2480e-03
	HSKSOR	3246 (ω=-2.0076)	36.10	2.5972e-02	2269 (ω=-2.0110)	33.96	1.3065e-02	1311 (ω=-2.0204)	31.89	1.2480e-03

Table 2. Comparison of Number Iterations (k), execution time (in seconds) and maximum absolute error for the iterative methods using example 2 at α=0.333, 0.666, 0.999.

M	Method	α=0.333			α=0.666			α=0.999		
		k	t	Max Error	k	t	Max Error	k	t	Max Error
128	FSGS	14148	6.40	3.4745e-04	7120	4.72	5.2947e-04	2358	3.57	4.5604e-04
	FSKSOR	385 (ω=-2.0585)	3.23	3.4734e-04	264 (ω=-2.0903)	3.13	5.2935e-04	144 (ω=-2.1686)	3.09	4.5594e-04
	HSKSOR	193 (ω=-2.1204)	1.85	9.3060e-04	135 (ω=-2.1889)	1.84	1.1077e-03	84 (ω=-2.3790)	1.84	1.0292e-03
256	FSGS	47194	28.00	2.0203e-04	24125	17.50	3.8529e-04	8332	10.12	3.1294e-04
	FSKSOR	764 (ω=-2.0299)	6.66	2.0160e-04	513 (ω=-2.0441)	6.40	3.8483e-04	291 (ω=-2.0811)	6.20	3.1267e-04
	HSKSOR	385 (ω=-2.0585)	3.80	3.4734e-04	274 (ω=-2.0866)	3.77	5.2935e-04	144 (ω=-2.1686)	3.73	4.5594e-04

512	FSGS	151187	149.13	1.6691e-04	79109	86.99	3.5049e-04	29799	42.17	2.7708e-04
	FSKSOR	1467 (ω=-2.0147)	13.99	1.6516e-04	1025 (ω=-2.0218)	13.61	3.4871e-04	576 (ω=-2.0397)	12.74	2.7687e-04
	HSKSOR	764 (ω=-2.0298)	7.65	2.0159e-04	513 (ω=-2.0441)	7.56	3.8483e-04	291 (ω=-2.0811)	7.40	3.1267e-04
1024	FSGS	454367	812.72	1.6304e-04	251077	509.06	3.4511e-04	107228	240.64	2.6674e-04
	FSKSOR	2906 (ω=-2.0073)	31.95	1.5605e-04	2049 (ω=-2.0096)	30.49	3.3969e-04	1125 (ω=-2.0197)	27.04	2.6791e-04
	HSKSOR	1467 (ω=-2.0147)	16.23	1.6516e-04	1115 (ω=-2.0189)	15.91	3.4871e-04	654 (ω=-2.0388)	15.44	2.7687e-04
2048	FSGS	1241856	4591.56	1.7665e-04	839911	3310.98	3.3991e-04	384794	1578.31	2.5732e-04
	FSKSOR	5708 (ω=-2.0037)	80.73	1.5380e-04	4056 (ω=-2.0054)	71.23	3.3742e-04	2155 (ω=-2.0098)	60.45	2.6568e-04
	HSKSOR	2905 (ω=-2.0071)	35.44	1.5605e-04	2049 (ω=-2.0103)	33.81	3.3969e-04	1151 (ω=-2.0202)	31.90	2.6792e-04

Table 3. Comparison of Number Iterations (k), execution time (in seconds) and maximum absolute error for the iterative methods using example 3 at α=0.333, 0.666, 0.999.

M	Method	α=0.333			α=0.666			α=0.999		
		k	t	Max Error	k	t	Max Error	k	t	Max Error
128	FSGS	18947	7.94	1.1755e-03	9174	5.54	2.5778e-03	2824	3.78	2.2069e-03
	FSKSOR	423 (ω=-2.0606)	3.16	1.1757e-03	295 (ω=-2.0905)	3.13	2.5780e-03	172 (ω=-2.1743)	3.11	2.2070e-03
	HSKSOR	209 (ω=-2.1252)	1.85	1.1785e-03	145 (ω=-2.1892)	1.84	2.5807e-03	89 (ω=-2.3794)	1.84	2.2096e-03
256	FSGS	69499	40.98	1.1744e-03	33775	23.09	2.5767e-03	10436	11.31	2.2058e-03
	FSKSOR	853 (ω=-2.0300)	6.69	1.1750e-03	592 (ω=-2.0444)	6.44	2.5773e-03	342 (ω=-2.0837)	6.27	2.2064e-03
	HSKSOR	423 (ω=-2.0606)	3.82	1.1757e-03	295 (ω=-2.0906)	3.78	2.5780e-03	172 (ω=-2.1743)	3.75	2.2071e-03
512	FSGS	252889	267.84	1.1724e-03	123467	138.85	2.5746e-03	38383	51.68	2.2036e-03
	FSKSOR	1707 (ω=-2.0150)	14.43	1.1748e-03	1181 (ω=-2.0221)	13.68	2.5772e-03	681 (ω=-2.0411)	12.89	2.2063e-03
	HSKSOR	853 (ω=-2.0300)	7.70	1.1750e-03	592 (ω=-2.0445)	7.59	2.5773e-03	342 (ω=-2.0837)	7.43	2.2064e-03
1024	FSGS	911238	1836.10	1.1650e-03	447430	922.83	2.5670e-03	140195	309.11	2.1956e-03
	FSKSOR	3392 (ω=-2.0075)	33.80	1.1748e-03	2353 (ω=-2.0110)	30.74	2.5771e-03	1356 (ω=-2.0204)	27.92	2.2062e-03
	HSKSOR	1707 (ω=-2.0150)	16.51	1.1748e-03	1181 (ω=-2.0221)	16.02	2.5772e-03	681 (ω=-2.0411)	15.46	2.2063e-03
2048	FSGS	3243868	12617.00	1.1357e-03	1604114	6334.88	2.5367e-03	507555	2083.86	2.1641e-03
	FSKSOR	6774 (ω=-2.0038)	88.57	1.1747e-03	4680 (ω=-2.0055)	75.59	2.5771e-03	2708 (ω=-2.0102)	63.53	2.2062e-03
	HSKSOR	3392 (ω=-2.0075)	36.43	1.1748e-03	2353 (ω=-2.0110)	34.29	2.5771e-03	1356 (ω=-2.0204)	32.55	2.2062e-03

From the recorded results in Tables 1 to 3, it clearly shows that the implementation of the half-sweep concept to the standard KSOR iteration technique managed to reduce the required iteration numbers and the computation time needed to solve the problem (1) at all mesh sizes that are examined. The summary of the decrement percentage of FSKSOR and HSKSOR in comparison to the FSGS iterative method for examples 1 to 3 are recorded in Table 4.

Table 4. Decrement percentage of the number of iterations and computation time for HSKSOR and FSKSOR iterative methods compared to FSGS iterative Method

Example	Method		α =0.333	α=0.666	α=0.999
1	FSKSOR	Iter	97.80–99.80%	96.84–99.71%	94.01–99.47%
		Time	59.44–99.35%	41.98–98.88%	14.91–97.13%
	HSKSOR	Iter	98.91–99.90%	98.44–99.85%	96.90–99.74%
		Time	76.17–99.73%	65.11–99.50%	49.32–98.56%
2	FSKSOR	Iter	97.28–99.54%	96.29–99.52%	93.89–99.44%
		Time	49.53–98.24%	33.69–97.85%	13.45–96.17%
	HSKSOR	Iter	98.64–99.77%	98.10–99.76%	96.44–99.70%
		Time	71.09–99.23%	61.02–98.98%	48.46–97.98%
3	FSKSOR	Iter	97.77–99.79%	96.78–99.71%	93.91–99.47%
		Time	60.20–99.30%	43.50–98.81%	17.72–96.95%
	HSKSOR	Iter	98.90–99.90%	98.42–99.85%	96.85–99.73%
		Time	76.70–99.71%	66.79–99.46%	51.32–98.44%

5. Conclusion

In this work, we implemented the half-sweep iteration concept on Grünwald implicit difference schemes and KSOR iterative method to solve one-dimensional time-fractional parabolic equations. Based on the numerical results, it shows that HSKSOR iterative method has reduced the number of iterations and execution time of the standard FSKSOR and FSGS iterative methods. Meanwhile, from the maximum error, it shows that the accuracy of HSKSOR is in good agreement with FSKSOR and FSGS iterative methods. Overall, the numerical results have shown that the HSKSOR method is more superior in terms of the number of iterations and the execution time than the standard method. For future work, other iterative methods especially on the two-step iteration family [23,24,25,26] will be applied to the proposed approximation equation in order to increase its convergence iteration.

Acknowledgments

The authors would like to express sincere gratitude to Universiti Malaysia Sabah for funding this research under UMSGreat research grant for postgraduate student: GUG0221-1/2018.

References

[1] Ammi M R, Jamiai I and Torres D F 2019 *Comput. Math. Appl.* **78**(5) pp 1334-1344
[2] Xu Q and Xu Y 2018 *Commun. Nonlinear Sci. Numer. Simulat.* **64** pp 135-148
[3] Youssef I K 2012 *J. Math. Stat.* **8**(2) pp 176-184
[4] Abdullah A R 1991 *Int. J. Comput. Math.* **38** pp 61-70
[5] Sunarto A, Sulaiman J and Saudi A 2016 *Jurnal Teknologi,* **78**(6-4) pp 7-12
[6] Saudi A and Sulaiman J 2016. *Jurnal Teknologi* **78**(8-2) pp 12-24
[7] Akhir M K Md, Othman M, Sulaiman J, Majid Z A and Suleiman M 2011 *Communications in Computer and Information Science 253 CCIS(PART 3)* pp 218-227
[8] Rahman R, Ali N A M, Sulaiman J and Muhiddin F A 2019 *Advances in Science, Technology and Engineering Systems Journal* **4**(2) pp 237-243
[9] Musli F A and Saudi A 2019 Proceedings of *IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE)* 2019 pp 335-339
[10] Suardi M N, Radzuan N Z F M and Sulaiman J 2017 *Glob. J. Pure Appl. Math.* **13**(11) pp 7921-7934
[11] Baharuddin, Sunarto A and Dalle J 2017 *J. Eng. Appl. Sci.* **12** pp 3220-3224
[12] Mishra T N and Rai K N 2015 *Int. J. Appl. Math. Res.* **4**(1) p 135
[13] Sunarto A, Sulaiman J and Saudi A 2016 *Global J. Pure Appl. Math.* **12**(4) pp 3469-3479
[14] Podlubny I 1999 *Fractional Differential Equations* (San diego: Academic Press).

[15] Zahra W K and Elkholy S M 2013 *Electr. J. Math. Anal. Appl.* **1**(2) pp 230-241
[16] Young D M 1971 Iterative solution of large linear systems (London: Academic Press)
[17] Young D M 1972 *J. Approx. Theory* **5** pp 137-148
[18] Sunarto A, Sulaiman J and Saudi A 2014 *Australian Journal of Basic and Applied Sciences* **8**(24) pp 153-158
[19] Muhiddin F A and Sulaiman J 2017 *Journal of Physics: Conference Series* **890** pp 012065
[20] Uddin M and Haq S 2011 *Commun. Nonlin. Sci. Numer. Simul.* **16**(11) pp 4208-4214
[21] Jiang Y and Ma J 2011 *J. Comput. Appl. Math.* **235**(11) pp 3285-3290
[22] Ma Y 2014 *Appl. Math. Bioinformatics* **4**(2) pp 125-145
[23] Dahalan A A, Sulaiman J and Muthuvalu M S 2014 *Appl. Math. Sci.* **8** pp 885-899
[24] Rahman R, Ali N A M, Sulaiman J and Muhiddin F A 2018 *AIP Conf. Proc.* **1123**(1) pp 012044
[25] Muthuvalu M S and Sulaiman J 2012 *Proyecciones* **31**(1) pp 65-79
[26] Muthuvalu M S and Sulaiman J 2017 *Lobachevskii Journal of Mathematics* **38**(1) pp73-81