Block Iterative Method for the Solution of Fractional Two-Point Boundary Value Problems

Rostang Rahman¹, Nur Afza Mat Ali², Jumat Sulaiman³, and Fatihah Anas Muhiddin⁴

¹,²,³Mathematics with Economics Programme, Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia.
⁴Faculty of Computing and Mathematical Sciences, Universiti Teknologi MARA Sabah, Malaysia.

E-mail: rostang054@gmail.com

Abstract. The goal of this paper deals with 4-Point Explicit Group (4-EG) iterative method as solver to solve the linear system which is generated by using Caputo’s finite difference approximation equation. Actually, this approximation equation was generated from the discretization process of the fractional two-point boundary value problems via finite difference scheme and Caputo’s fractional operator. The formulation and implementation for 4-EG method have been also included. To investigate the efficiency of 4-EG method, this paper considered three examples numerical experiment. Based on numerical results, the finding showed that the 4-EG method is more efficient as compared with GS method in good agreement.

1. Introduction

From preceding studies, numerous researchers have been discussed on how to clear up the fractional two point boundary value problems in view that this hassle will become more attractive and popular within the current years [1, 2]. The fractional two-point boundary value problems (TPBVPs) related to many applications such as economics, physics, and engineering [3-5]. Because of this matters, various numerical methods had been proposed to get the solutions such as Quadrature Tau method [6], finite element method [7], Homotopy analysis method [8], and Crank-Nicolson finite difference method (C-N-FDM) [9]. Other researchers additionally have talked about the finite difference method is one of the pleasant methods to remedy the same problem [10-13].

From the previous researcher, this paper inspired to use the finite difference method to solve the fractional TPBVPs since none of the previous researcher have been applied this method to the problem by consider ordinary differential equation. Following to that, discretization method want to be performed through finite difference scheme and Caputo’s fractional operator to derive the approximation equation. Then corresponding second order Caputo’s approximation equations from finite different solution can be used to construct a linear system. Having a linear system, the numerical solution of this linear system can be calculate via family of iterative methods because of the characteristic linear system.

Simply, there are numerous iterative methods from preceding research can be used particularly to solve the linear system since it is large-scale in size and has sparse matrix in terms of the characteristics. [14-16]. Because of that this paper focuses to investigate the performance of 4-point Explicit Group (4-EG) iterative method which was developed by Evans [17] in order to increase the iteration convergence
rate compare to point iterative method. In addition, none of researcher have been applied the EG iterative method to solve the problem included fractional. Actually, this method uses small fixed size groups of mesh point strategy in order to speed up the convergence rate in solving the problem. From previous studies, Explicit Group (EG) method is more superior compared to the point iterative method because the EG iterative method requires less number of iteration and mush faster in term of execution time compare to the point iterative method [18]. To analyze the performance of 4-EG iterative method in solving fractional TPBVPs, the implementation of the Gauss Seidel (GS) iterative method acts as control method and To analyze the overall performance 4-EG iterative method, let the fractional TPBVPs are defined as [19],

\[ d(x)D^\beta y(x) + a(x)y(x) + b(x)y(x) + c(x) = f(x), x \in [\eta, \gamma] \]  

subject to the Dirichlet boundary conditions,

\[ y(\eta) = \eta_0, \quad y(\gamma) = \gamma_1. \]

where \( a(x), b(x), c(x) \) and \( d(x) \) are known functions or constants respectively with \( D^\beta y(x) \) refer to fractional derivative and \( \beta \) is a parameter which refers to the fractional order.

To facilitate discretizing process problem (1) via the finite difference scheme, let the solution domain divide with equal and sparse which is \( h \) like Figure 1, in which its spacing is define as

\[ h = \Delta x = \frac{\gamma - \eta}{N} \]  

(2)

Figure 1. The finite grid network for the solution domain of problem 1.

Based on equation (2), the node grid of the solution domain will be indicated as

\[ x_i = \eta + ih, \quad i = 0, 1, 2, \ldots \]  

(3)

and the values of function \( y(x) \) at point \( x_i \) are donated as \( y(x_i) = y_i \).

2. Preliminaries

To discretize problem (1) by using Caputo’s fractional operator, firstly the operator needs to be constructed based on some definition and mathematical preliminaries related to fractional calculus in this section.

**Definition 1** [20]. The fractional integral operator for Riemann-Liouville

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

(4)

**Definition 2** [20]. The fractional partial derivative for Caputo’s operator

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

(5)

Where the value of parameter \( \beta \) can be set based on condition \( m - 1 < \beta \leq m, m \in N \), and \( x > 0 \).

**Definition 3** [20]. If \( m \) to be the smallest integer that exceeds \( \beta \), then

\[ D^\beta y(x,t) = \frac{d^\beta y(x,t)}{dx^\beta} = \begin{cases} 
\frac{1}{\Gamma(m - \beta)} \int_0^x \frac{(x-s)^{m-\beta-1}}{(x-t)^{\beta+m+1}} y(x,t)ds, & \text{for } m - 1 < \beta \leq m \\
\frac{\partial^m}{\partial x^m} y(x,t), & \text{for } \beta = m \in N 
\end{cases} \]  

(6)
To get the numerical solution of problem (1), the proposed finite difference scheme with Caputo’s fractional operator were considered to construct the Caputo’s finite difference approximation equation in Section 3.

3. Second-Order Caputo’s Finite Difference Approximation Equation

To get the approximation equation of problem (1) based on the second order Caputo’s finite difference, preliminaries in Section 2 will be considered to derive the formulation of second order Caputo’s fractional derivative operator which is given as:

\[ D^{\beta}y_i \equiv \sigma_{\beta,h} \sum_{j=0}^{N_i} g^\beta_j (y_{i-j+1} - 2y_{i-j} + y_{i-j-1}) \]  

(7)

where

\[ \sigma_{\beta,h} = \frac{h^{-\beta}}{\Gamma(3 - \beta)} \]  

(8)

and

\[ g^\beta_j = (j+1)^{2-\beta} - j^{2-\beta} \]  

(9)

According to Figure 1, we develop the uniformly grid network of the solution domain where the polar grid of the solution domain will be indicated as \( x_i = \eta + ih \), \( i = 0,1,2, \ldots \) and the values of function \( y(x) \) at point \( x_i \) are donated as \( y(x_i) = y_i \). As mentioned in section 2, using the finite difference discretization scheme and second-order Caputo’s fractional derivative operator, the second-order Caputo’s finite difference approximation equation of problem (1) can be given as:

\[ d_i^{\sigma_{\beta,h}} \sum_{j=0}^{N_i} g^\beta_j (y_{i-j+1} - 2y_{i-j} + y_{i-j-1}) + a_i \left( \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \right) + b_i \left( \frac{y_{i+1} - y_{i-1}}{2h} \right) + c_i (y_i) = f_i \]  

(10)

by simplifying equation (10), the approximation equation (10) can be written:

\[ a_i^* y_{i+1} + b_i^* y_i + c_i^* y_{i-1} - R_i = f_i, \quad i=1,2,3 \]  

(11)

where

\[ \lambda_i = d_i^{\sigma_{\beta,h}}, \quad a_i = a_i - \mu_i - \lambda_i, \quad b_i = c_i - 2\alpha_i + 2\lambda_i, \quad c_i = \alpha_i + \mu_i - \lambda_i, \quad R_i = \lambda_i \sum_{j=1}^{N_i} g^\beta_j (y_{i-j+1} - 2y_{i-j} + y_{i-j-1}). \]

Again, besides the values of \( i=1,2, \) and 3 for the equation (11), the derivation of approximation equation (10) needs to be shown for \( i=4,5,6, \ldots \). To do this, the second-order Caputo’s finite difference approximation equation can be stated as follows:

\[ R_i^* + p_i y_{i-3} + q_i y_{i-2} + r_i y_{i-1} + s_i y_i + z_i y_{i+1} = f_i, \quad i=3,4,5, \ldots \]  

(12)

where

\[ R_i^* = \lambda_i \sum_{j=3}^{N_i} g^\beta_j (y_{i-j+1} - 2y_{i-j} + y_{i-j-1}), \]

\[ p = \lambda g_2^\beta, \quad q = \lambda g_1^\beta - 2\lambda g_2^\beta, \quad r = a_i - 2\alpha_i g_1^\beta + \lambda g_2^\beta, \quad s = b_i^* + \lambda g_1^\beta, \quad z = c_i. \]
To make it accessible in solving second order Caputo’s approximation equations (11) and (12), both these approximation equations being used in constructing the linear system. As a result, the characteristic of approximation equations produce a large-scale and sparse linear system in the form of:

\[ A y = f \]  

(13)

4. Formulation of 4EG Iterative Method

To increase the convergence rate and reduce the computational complexity in solving the proposed problem, this section will show the formulation of EG iterative method which introduced by Evans [17] as mention in Section 1. From this method, the convergence rate can be reduce and the computational time can be much faster by solving the several small groups of points instead of a large linear system per iteration. This method divide the original matrix into several small fixed-sized groups of points.

By uses small fixed size groups of mesh point strategy as mention in section 1, in order to accelerate the convergence rate of point iterative methods, this section will show the formulation of linear system (13) by use size four-point groups strategy.

\[
\begin{bmatrix}
    s_i & z_i & 0 & 0 \\
    r_{i+1} & s_{i+1} & z_{i+1} & 0 \\
    q_{i+2} & r_{i+2} & s_{i+2} & z_{i+2} \\
    p_{i+3} & q_{i+3} & r_{i+3} & s_{i+3}
\end{bmatrix}
\begin{bmatrix}
    y_i \\
    y_{i+1} \\
    y_{i+2} \\
    y_{i+3}
\end{bmatrix}
=
\begin{bmatrix}
    S_1 \\
    S_2 \\
    S_3 \\
    S_4
\end{bmatrix},
\]

(14)

where

\[
S_i = f_i + R_i,
\]

\[
S_i = f_i + R_i,
\]

\[
S_i = f_i + R_i,
\]

\[
S_i = f_i + R_i - z_{i+3}y_{i+4}
\]

Next by simplifying equation (15) we can show the following scheme

\[
y_i = (a_iS_1 + b_iS_2 + c_iS_3 - d_iS_4) / \det
\]

(16)

where the

\[
a_i = s_2s_4 - r_3s_2z_1 + q_2z_3
\]

\[
b_i = r_3s_2z_1 - s_2s_4
\]

\[
c_i = s_4z_2
\]

\[
d_i = z_4z_3
\]

Again, the equation (15) simply for \( y_{i+1} \) can be written as

\[
u_{i+1} = (a_2S_1 + b_2S_2 - c_2S_3 + d_2S_4) / \det
\]
where

\[ a_2 = q_3 s_4 z_2 - r_5 s_3 s_4 + r_2 r_4 z_1 - p_4 z_2 z_3 \]
\[ b_3 = s_4 (s_4 r_3 - p_4 z_1) \]
\[ c_2 = s_4 s_2 z_2 \]
\[ d_2 = s_1 z_2 z_3 \]

For \( y_{i+2} \), also can be written as

\[ y_{i+2} = (a_3 s_1 + b_3 s_2 + c_3 s_3 - d_3 s_4) / \det \]

where

\[ a_3 = r_5 s_4 - q_4 s_4 + q_5 r_3 s_4 - q_4 r_5 z_3 \]
\[ b_3 = s_4 s_2 - r_2 z_1 \]
\[ c_3 = s_4 (s_4 r_3 - p_4 z_1) \]
\[ d_3 = z_3 (r_2 z_1 - s_4 s_2) \]

and \( y_{i+3} \) written as

\[ y_{i+3} = (a_4 s_1 + b_4 s_2 + c_4 s_3 - d_4 s_4) / \det \]

where

\[ a_4 = q_4 s_4 z_2 - p_4 s_3 s_4 + q_4 r_3 s_4 - r_2 r_4 z_1 + p_4 r_3 z_2 - q_3 q_4 z_2 \]
\[ b_4 = s_4 s_2 - r_2 z_1 \]
\[ c_4 = q_4 s_4 z_2 - r_2 s_2 + r_2 r_4 z_1 - p_4 z_1 z_2 \]
\[ d_4 = s_4 s_3 - r_2 s_3 z_1 - r_2 s_3 z_2 + q_3 z_1 z_2 \]

![Figure 2: Implementation of the four point-EG method at solution domain m=16.](image)

**Algorithm 1:** Four-Point EG iteration

i. Set the value of parameters, \( y^{(0)} \leftarrow 0, \varepsilon \leftarrow 10^{-10} \)

ii. Calculate the coefficient matrix, \( A \) and vector, \( F \).

iii. The implementation calculate equation (16) until equation (19)

iv. Perform the convergence test, \( \left| y^{(k+1)} - y^{(k)} \right| \leq \varepsilon = 10^{-10} \). If yes, move to step (v). Otherwise go back to step (iii).

v. Display approximate solution.

5. Results and Discussion

To examine the efficiency of 4-EG iterative method, this section proposed three examples of fractional two point boundary values problems where each example given the exact solution when parameter set to \( \beta = 1.50 \). However, this experiments also demonstrate two different values of parameter \( \beta \) which is \( \beta = 1.25 \) and \( 1.75 \) since the range of parameter \( 1 < \beta \leq 2 \) refer in Section 2 the condition of parameter to analyze the performance of Caputo’s fractional operator. Beyond that point, three measurement parameters would be considered such as number of iterations (k), computational time in second (s) and

5
maximum absolute error (error) with the convergence test set to tolerance error as \( \varepsilon = 10^{-10} \) for the performance of proposed iterative methods. The numerical experiments as follows:

**Example 1** [21]

\[
D^\beta y(x) + y'(x) + y(x) = f(x), \quad 0 \leq x \leq 1
\]  

(17)

where the function \( f(x) \) is given by

\[
f(x) = x^2 + x + \frac{4}{\sqrt{\pi}} \sqrt{x} + 3
\]  

(18)

and the boundary condition given as

\[
y(0) = 1, \quad y(1) = 3
\]

The analytical solution for equation (17) obtain as follows

\[
y(x) = x^2 + x + 1
\]  

(19)

**Example 2** [22]

\[
D^\beta y(x) + y'(x) = f(x), \quad 0 \leq x \leq 1
\]  

(20)

Where the function \( f(x) \) is given by

\[
f(x) = \frac{15}{4} x^{0.5} + \frac{15}{8} \sqrt{\pi x + x^{2.5}} + 1
\]  

(21)

and the boundary condition being given as

\[
y(0) = 1, \quad y(1) = 2
\]

The analytical solution for equation (20) given by

\[
y(x) = x^{2.5} + 1
\]  

(22)

**Example 3** [23]

\[
D^\beta y(x) + by'(x) + cy(x) = f(x), \quad x \in [0, 1], \quad 0 \leq x \leq 1
\]  

(23)

Where the function \( f(x) \) is given by

\[
f(x) = 4 \sqrt{\frac{x}{\pi} + x^3} + 2
\]  

(24)

and the boundary condition being given as

\[
y(0) = 1, \quad y(1) = 2
\]

The analytical solution for equation (23) obtain as follows

\[
y(x) = x^2
\]  

(25)

This experiment were conducted by C programming language than the results which obtained from the implementation of GS and 4-EG iterative methods have been recorded in Tables 1, 2 and 3 at five different values of mesh sizes which are \( m = 128, 256, 512, 1024, \) and 2048.

Based on the results recorded in Tables 1, 2, and 3 by comparing the GS and 4-EG iterative methods, it is clearly at \( \beta = 1.50 \) that number of iterations have declined approximately by 71.65% – 72.39%, 71.73% – 72.32%, and 71.42% – 72.49% compare to the 4-EG iterative method with GS iterative method for each examples. Particularly in terms of execution time, the implementations of 4-EG iterative method are much faster about 83.44% – 89.81%, 71.16% – 88.48%, and 68.63% – 71.37% than GS iterative method. It is obvious that the 4-EG iterative method requires the less amount for number of iterations and computational time as compared with classical GS iterative method since 4-EG method consider small fixed size groups of mesh point strategy and GS iterative method just consider point strategy during the solutions. For other value of \( \beta = 1.25 \) and 1.75 it can be observed that their conclusions are in line with \( \beta = 1.50 \) because the max absolute error lower than another \( \beta \).
Table 1. Result of the comparison between parameter $\beta = 1.25, 1.50$ and $1.75$ for Examples 1.

| $\beta$ | M     | K  | Time (Second) | Max Error |
|---------|-------|----|---------------|-----------|
|         |       | GS | 4EG           | GS        | 4EG       | GS   | 4EG   |
| $\beta = 1.25$ | 128   | 18333 | 5031         | 1.00      | 0.19      | 2.1230e-02 | 2.122990e-02 |
|         | 256   | 67072 | 18410        | 14.06     | 2.32      | 2.1228e-02  | 2.122815e-02 |
|         | 512   | 243716| 67186        | 200.65    | 30.96     | 2.1230e-02  | 2.122914e-02 |
|         | 1024  | 878044| 243871       | 3486.79   | 428.81    | 2.1229e-02  | 2.123133e-02 |
|         | 2048  | 3128970| 878223      | 60236.7   | 6013.99   | 2.1257e-02  | 2.123707e-02 |
| $\beta = 1.50$ | 128   | 18198 | 5028         | 1.03      | 0.20      | 1.3813e-05  | 1.389019e-05 |
|         | 256   | 65479 | 18090        | 22.28     | 2.27      | 1.3524e-05  | 1.382768e-05 |
|         | 512   | 234899| 65120        | 283.34    | 30.45     | 1.2340e-05  | 1.353309e-05 |
|         | 1024  | 838194| 233849       | 3222.64   | 409.18    | 7.7771e-06  | 1.234967e-05 |
|         | 2048  | 2966072| 835317      | 40802.48  | 5721.08   | 1.2308e-05  | 7.797462e-06 |
| $\beta = 1.75$ | 128   | 20575 | 5680         | 1.15      | 0.22      | 1.7118e-02  | 1.711782e-02 |
|         | 256   | 72891 | 20191        | 15.28     | 2.53      | 1.7081e-02  | 1.708140e-02 |
|         | 512   | 257190| 71635        | 212.15    | 33.01     | 1.7061e-02  | 1.706269e-02 |
|         | 1024  | 902430| 253238       | 2973.20   | 443.47    | 1.7047e-02  | 1.705211e-02 |
|         | 2048  | 3141444| 890318      | 46972.44  | 6081.43   | 1.7023e-02  | 1.704250e-02 |

Table 2. Result of the comparison between parameter $\beta = 1.25, 1.50$ and $1.75$ for Examples 2.

| $\beta$ | M     | K  | Time (Second) | Max Error |
|---------|-------|----|---------------|-----------|
|         |       | GS | 4EG           | GS        | 4EG       | GS   | 4EG   |
| $\beta = 1.25$ | 128   | 17860 | 4911         | 0.99      | 0.18      | 3.2820e-02 | 3.2818e-02 |
|         | 256   | 65189 | 17935        | 13.68     | 2.31      | 3.3318e-02 | 3.3317e-02 |
|         | 512   | 236202| 65299        | 194.51    | 30.14     | 3.3572e-02 | 3.3571e-02 |
|         | 1024  | 848050| 236351       | 2905.1    | 414.17    | 3.3705e-02 | 3.3700e-02 |
|         | 2048  | 3009165| 848220      | 50524.12  | 5814.66   | 3.3790e-02 | 3.3770e-02 |
| $\beta = 1.50$ | 128   | 17756 | 4915         | 0.98      | 0.19      | 8.8686e-04 | 8.8989e-04 |
|         | 256   | 63748 | 17651        | 13.58     | 2.24      | 4.5977e-04 | 4.6061e-04 |
|         | 512   | 228091| 63400        | 210.57    | 29.47     | 2.3923e-04 | 2.4055e-04 |
|         | 1024  | 228091| 227075       | 3492.56   | 402.34    | 1.2280e-04 | 1.2764e-04 |
|         | 2048  | 2859540| 808533      | 19257.6   | 5554.26   | 4.7423e-05 | 6.6386e-05 |
| $\beta = 1.75$ | 128   | 20106 | 5560         | 1.74      | 0.21      | 3.2009e-02 | 3.2012e-02 |
|         | 256   | 71089 | 19733        | 23.20     | 2.52      | 3.1690e-02 | 3.1691e-02 |
|         | 512   | 250227| 69868        | 317.07    | 32.26     | 3.1519e-02 | 3.1521e-02 |
|         | 1024  | 875406| 246394       | 4663.88   | 435.40    | 3.1426e-02 | 3.1431e-02 |
|         | 2048  | 3036143| 863693      | 44792.83  | 5904.12   | 3.1360e-02 | 3.1381e-02 |
Table 3. Result of the comparison between parameter $\beta = 1.25, 1.50$ and 1.75 for Examples 3.

| $\beta$   | M     | K     | Time (Second) | Max Error |
|-----------|-------|-------|---------------|-----------|
|           | GS    | 4EG   | GS            | 4EG       | GS         | 4EG       |
| $\beta = 1.25$ |       |       |               |           |            |           |
| 128       | 16630 | 4538  | 0.51          | 0.17      | 2.1230e-02 | 2.1230e-02 |
| 256       | 60242 | 16566 | 6.73          | 1.98      | 2.1228e-02 | 2.1228e-02 |
| 512       | 21638 | 60069 | 94.16         | 27.39     | 2.1230e-02 | 2.1229e-02 |
| 1024      | 76879 | 21595 | 1304.27       | 375.25    | 2.1236e-02 | 2.1231e-02 |
| 2048      | 26922 | 76775 | 18156.89      | 5186.54   | 2.1257e-02 | 2.1237e-02 |
| $\beta = 1.50$ |       |       |               |           |            |           |
| 128       | 16487 | 4536  | 0.51          | 0.16      | 1.3813e-05 | 1.3891e-05 |
| 256       | 58743 | 16249 | 6.53          | 1.96      | 1.3524e-05 | 1.3829e-05 |
| 512       | 20833 | 58079 | 90.32         | 26.13     | 1.2340e-05 | 1.3536e-05 |
| 1024      | 73313 | 20655 | 1245.06       | 356.52    | 7.7771e-06 | 1.2356e-05 |
| 2048      | 25494 | 72854 | 17167.51      | 4954.22   | 1.2308e-05 | 7.8149e-06 |
| $\beta = 1.75$ |       |       |               |           |            |           |
| 128       | 18557 | 5159  | 0.58          | 0.18      | 1.7118e-02 | 1.7118e-02 |
| 256       | 65077 | 18192 | 7.23          | 2.21      | 1.7081e-02 | 1.7081e-02 |
| 512       | 22689 | 63917 | 98.52         | 28.74     | 1.7061e-02 | 1.7063e-02 |
| 1024      | 78465 | 22332 | 1328.64       | 385.43    | 1.7047e-02 | 1.7052e-02 |
| 2048      | 26821 | 77395 | 18067.66      | 5263.36   | 1.7023e-02 | 1.7042e-02 |

6. Summary
As conclusion for this paper show the second order Caputo’s operator can be used to solve the fractional TPBVPs with ordinary differential equation and the performance of the iterative method in solving fractional TPBVPs have been demonstrated successfully through three examples with comparison between 4-EG iterative method where used small fixed size groups of mesh point strategy and GS iterative method means point iterative method. Other than that, the numerical results presented in Tables 1, 2 and 3 showed that the proposed iterative method need less number of iterations and computational time in obtaining numerical solutions as compared to the GS iterative method. By using the GS iterative as a control method, 4-EG has a reduced number of iteration of approximately 71.42% -72.49% and computational time approximately 68.63% -89.81%. The performance showed by the 4-EG method used small fixed size groups of mesh point strategy, the speed up of convergence rate increase and in terms of the accuracy of the iterative methods, both tested numerical methods have good agreement. Therefore, it can be concluded that the 4-EG method can be a promising technique for solving fractional TPBVPs.

7. References
[1] C Kou and X Zhang 2007 Com. Eighth ACIS Int. Conf, 3 pp 1056-1059.
[2] M M Khader 2014 Canadian J. Phys, 92(12) pp 1658-1666.
[3] D Matignon 1996 Comp. eng. Sys. appl, 2 pp 963-968.
[4] H Jafari, H Tajadodi, and D Baleanu 2013 Fractional Cal. and Appl. Analysis 16 (1) pp 109-122.
[5] B R Sontakke and A S Shaikh 2015. Int. J. Eng. Research and Applications. 5(5) pp 22-7.
[6] A H Bhrawy, A S Alofi and S S Ezz-Eldien 2011 Appl. Math. Letters. 24(12) pp 2146-2152.
[7] W Deng. 2008. SIAM J. on numerical analysis. 47(1) pp 204-226.
[8] I Hashim, O Abdulaziz and S Momani 2009 Communications in Nonlinear Science and Numerical Simulation. 14(3) pp 674-684.
[9] N H Sweilam, M M Khader and A M S Mahdy 2012. J. of Fractional Cal. and Appl. 2(2) pp 1-9.
[10] A Sunarto, J Sulaiman and A Saudia 2016 J. Phys. 710(1) 012019.
[11] Y Zhang 2009 Appl. Math. & Comp. 215 pp 524-529.
[12] Hong E J, Saudi A, & Sulaiman J 2017 Intl J. of Imaging & Robotics™. 17(4) pp 15-24.
[13] Muthuvalu M S, Aruchunan E, & Sulaiman J 2016 Technology and Social Sciences RCSTSS 2014 pp 297-305 Springer, Singapore.
[14] D M Young 1971 Iterative Solution of Large Linear Systems (Academic Press, London).
[15] A V Henk and Y Saad 2000 J. Comp. & Appl. Math. 123 pp 1-33.
[16] W Hackbusch 1995 Iterative Solution of Large Sparse Systems of Equations (Springer-Verlag, New York).
[17] D J Evans 1985 Int. J. of Comp. Math. 17(1) pp 81-108.
[18] A Saudi and J Sulaiman 2012 Procedia Engineering. 41 pp 182-188.
[19] S Alkan, K Yildirim and A Secer 2016 Open Physics. 14(1) pp 6-14.
[20] A M A El-Sayed, S H Behiry and W E Raslan 2010 Comp. & Math. with Appl. 59(5) pp 1759-1765.
[21] A El-Ajou, O A Arqub and S Momani 2013 Ain Shams Engineering Journal. 4(3) pp 539-547.
[22] A Pedas and E Tamme 2012 J. of Comp. & Appl. Math. 236(13) pp 3349-3359.
[23] H Azizi 2015 J. of Math. Extension 9 pp 57-71.