ISSN 1823-626X

Journal of Fundamental Sciences

available online at http://jfs.ibnusina.utm.my

Solving second kind linear Fredholm integral equations via Quarter-Sweep SOR iterative method

Mohana Sundaram Muthuvalu* and Jumat Sulaiman

School of Science and Technology, Universiti Malaysia Sabah, Locked Bag 2073, 88999 Kota Kinabalu, Sabah, Malaysia.

Received 14 May 2010, Revised 20 June 2010, Accepted 1 July 2010, Available online 10 October 2010

ABSTRACT

In this paper, we consider the numerical solutions of linear Fredholm integral equations of the second kind. The Quarter-Sweep Successive Over-Relaxation (QSSOR) iterative method is applied to solve linear systems generated from discretization of the second kind linear Fredholm integral equations using quadrature method. In addition, the formulation and implementation of the proposed method to solve the problem are also presented. Numerical tests and comparisons with other existing methods are given to illustrate the effectiveness of the proposed method.

| Linear Fredholm equations | Quarter-sweep iteration | Quadrature | Successive Over-Relaxation |

http://dx.doi.org/10.11113/mjfas.v6n2.191

1. INTRODUCTION

The theory and applications of the integral equations have been one of the principal tools in various areas of science such as applied mathematics, physics, biology, chemistry and engineering. On the other hand, integral equations are encountered in numerous applications in many fields including continuum mechanics, potential theory, geophysics, electricity and magnetism, kinetic theory of gases, hereditary phenomena in physics and biology, renewal theory, quantum mechanics, radiation, optimization, optimal control systems, communication theory, mathematical economics, population genetics, queuing theory, medicine, mathematical problems of radiative equilibrium, particle transport problems of astrophysics and reactor theory, acoustics, fluid mechanics, steady state heat conduction, fracture mechanics, and radiative heat transfer problems [1]. Basically, integral equations can be classified according to the integration domain. Integral equations in which the integration domain varies with the independent variable in the equation are known as Volterra equations and those with fixed integration domain are Fredholm equations. In this paper, second kind linear integral equations type of Fredholm is considered.

Generally, second kind linear Fredholm integral equations can be written as

$$\lambda y(x) - \int_{\Gamma} K(x,t)y(t)dt = f(x), \Gamma = [a,b] \; \lambda \neq 0 \qquad (1)$$

where the parameter λ, kernel $K \in L^2(\Gamma \times \Gamma)$ and free term $f \in L^2(\Gamma)$ are given, $y \in L^2(\Gamma)$ is the unknown function to be determined and L^2 represents the function space of the solution domain. The kernel function $K(x,t)$ is assumed to be absolutely integrable and satisfy other properties that are sufficient to imply the Fredholm alternative theorem. Meanwhile, Eq. (1) also can be rewritten in the equivalent operator form

$$(\lambda - \kappa)y = f \qquad (2)$$

Fredholm Alternative Theorem [2]

Let χ be a Banach space and let $\kappa : \chi \rightarrow \chi$ be compact. Then the equation $(\lambda - \kappa)y = f$, $\lambda \neq 0$ has a unique solution $x \quad \chi$ if and only if the homogeneous equation $(\lambda - \kappa)z = 0$ has only the trivial solution $z = 0$. In such a case, the operator $\lambda - \kappa : \chi \underset{onto}{\overset{1-1}{\rightarrow}} \chi$ has a bounded inverse $(\lambda - \kappa)^{-1}$.

A numerical approach to the solution of integral equations is an essential branch of scientific inquiry. As a matter of fact, some valid methods of solving linear Fredholm integral equations have been developed in recent years. To solve Eq. (1) numerically, we either seek to determine an approximate solution by using the quadrature method [2-8], or use the projection method [9-15]. Such discretizations of integral equations lead to dense linear

Corresponding author at: School of Science and Technology, Universiti Malaysia Sabah, Locked Bag 2073, 88999 Kota Kinabalu, Sabah, Malaysia.
E-mail addresses: sundaram_at2@yahoo.com or jumat@ums.edu.my

systems and can be prohibitively expensive to solve using direct methods as the order of the linear system increases. Thus, iterative methods are the natural options for efficient solutions.

The concept of the half-sweep iteration has been inspired by Abdullah [16] via the Explicit Decoupled Group (EDG) iterative method to solve two-dimensional Poisson equations. Following to that, applications of the half-sweep iterative methods have been reviewed in [17-21]. Meanwhile, Othman and Abdullah [22] extended the half-sweep iteration concept by introducing quarter-sweep iterative method via the Modified Explicit Group (MEG) iterative method to solve two-dimensional Poisson equations. Further studies to verify the effectiveness of the quarter-sweep iterative methods have been carried out; see [23-27]. In this paper, we examined the applications of the half- and quarter-sweep iterative concepts with Successive Over-Relaxation (SOR) iterative method by using approximation equation based on quadrature method for solving problem (1). The standard SOR iterative method is also called as the Full-Sweep Successive Over-Relaxation (FSSOR) method. Meanwhile, combinations of the SOR method with half- and quarter-sweep iterations are called as Half-Sweep Successive Over-Relaxation (HSSOR) and Quarter-Sweep Successive Over-Relaxation (QSSOR) methods respectively.

The outline of this paper is organized in following way. In Section 2, the formulations of the full-, half- and quarter-sweep quadrature approximation equations are elaborated. In Section 3, we discuss the formulations of the FSSOR, HSSOR and QSSOR methods. Some numerical results are shown in Section 4 to assert the performance of the iterative methods. Analysis on computational complexity is mentioned in Section 5 and some concluding remarks are given in Section 6.

2. DISCRETIZATION OF INTEGRAL EQUATIONS

As afore-mentioned, a discretization scheme based on method of quadrature was used to construct approximation equations for problem (1) by replacing the integral to finite sums. In order to facilitate the formulation of the full-, half- and quarter-sweep quadrature approximation equations for problem (1), further discussion will be restricted onto repeated trapezoidal (RT) scheme, which is based on linear interpolation formula with equally spaced data.

Let the interval $[a,b]$ be partitioned into n subintervals of length

$$h=\frac{b-a}{n}. \tag{3}$$

Then, trapezoidal scheme for approximating definite integral $\int_a^b y(t)\,dt$ can be defined as

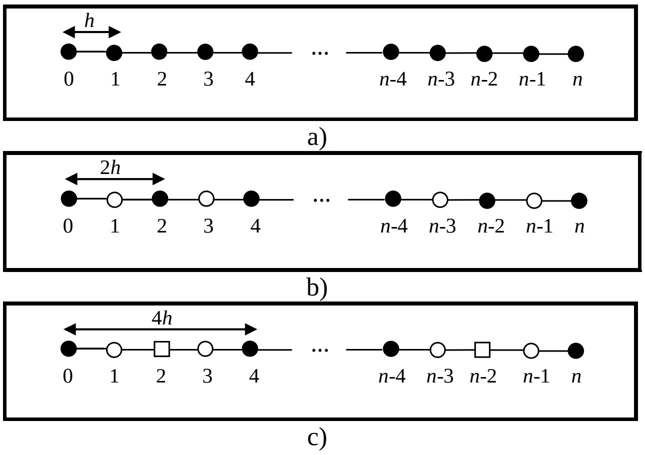
$$\int_a^b y(t)dt=\frac{h}{2}\big(y(a)+y(b)\big)+\varepsilon_n(y) \tag{4}$$

and its repeated formula can be shown as

$$\int_a^b y(t)dt=\frac{h}{2}y(a)+h\sum_{j=1}^{n-1}y(t_j)+\frac{h}{2}y(b)+\varepsilon_n(y) \tag{5}$$

where t_j and $\varepsilon_n(y)$ are abscissas of the partition points of the integration interval $[a,b]$ and truncation error respectively.

Meanwhile, Fig. 1 shows the finite grid networks in order to form the full-, half- and quarter-sweep quadrature approximation equations.



a)

b)

c)

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

Based on Fig. 1, the full-, half- and quarter-sweep iterative methods will compute approximate values onto node points of type ● only until the convergence criterion is reached. Then, other approximate solutions at remaining points (points of the different type) can be computed using the direct method [16, 22].

By applying Eq. (5) into Eq. (1) and neglecting the error term $\varepsilon_n(y)$, a system of linear equations can be formed for approximation values of $y(t)$. Therefore, the full-, half- and quarter-sweep repeated trapezoidal approximation equations for Eq. (1) can be generally shown as follows

$$\lambda y_i-\left(\frac{ph}{2}K_{i,0}y_0+ph\sum_{j=p}^{n-p}K_{i,j}y_j+\frac{ph}{2}K_{i,n}y_n\right)=f_i$$

$$i=0,1p,2p,\cdots,n \quad j=0,1p,2p,\cdots,n. \tag{6}$$

The value of p, which corresponds to 1, 2 and 4, represents the full-, half- and quarter-sweep cases respectively. From Eq. (6), it is obvious that discretization of the Eq. (1) using RT scheme leads to the dense linear systems as follows

$$M\underset{\sim}{y} = \underset{\sim}{f} \tag{7}$$

where

$$M = \begin{bmatrix} \lambda - \frac{ph}{2}K_{0,0} & -phK_{0,p} & -phK_{0,2p} & \cdots & -\frac{ph}{2}K_{0,n} \\ -\frac{ph}{2}K_{p,0} & \lambda - phK_{p,p} & -phK_{p,2p} & \cdots & -\frac{ph}{2}K_{p,n} \\ -\frac{ph}{2}K_{2p,0} & -phK_{2p,p} & \lambda - phK_{2p,2p} & \cdots & -\frac{ph}{2}K_{2p,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{ph}{2}K_{n,0} & -phK_{n,p} & -phK_{n,2p} & \cdots & \lambda - \frac{ph}{2}K_{n,n} \end{bmatrix}_{\left(\left(\frac{n}{p}\right)+1\right)\times\left(\left(\frac{n}{p}\right)+1\right)}$$

$$\underset{\sim}{y} = \begin{bmatrix} y_0 & y_p & y_{2p} & \cdots & y_{n-2p} & y_{n-p} & y_n \end{bmatrix}^T$$

and

$$\underset{\sim}{f} = \begin{bmatrix} f_0 & f_p & f_{2p} & \cdots & f_{n-2p} & f_{n-p} & f_n \end{bmatrix}^T .$$

3. ITERATIVE METHODS

As mentioned in Section 1, a family of SOR iterative method consists of FSSOR, HSSOR and QSSOR methods will be applied to solve dense linear system generated from the discretization of the problem (1), as shown in Eq. (7). To develop the formulation for SOR methods, let the coefficient matrix M be decomposed into

$$M = D - L - U \tag{8}$$

where D, L and U are diagonal, strictly lower triangular and strictly upper triangular matrices respectively. Thus, the general scheme for SOR iterative methods can be written as follows

$$D\underset{\sim}{y}^{(k+1)} = \omega L\underset{\sim}{y}^{(k+1)} + \omega U\underset{\sim}{y}^{(k)} + \omega \underset{\sim}{f} + (1-\omega)D\underset{\sim}{y}^{(k)} \tag{9}$$

where ω is a relaxation parameter. Actually, the iterative methods attempt to find a solution by repeatedly solving the linear system using approximations to the vector $\underset{\sim}{y}$. Iterations for iterative methods continue until the solution is within a predetermined acceptable bound on the error.

The performance of the SOR methods can be very often drastically improved with the proper choice of the relaxation parameter ω. Relaxation parameter can be calculated in practice by consecutively choosing values with some precision until the optimal value is obtained. Note that choosing $\omega = 1$ will result in the Gauss-Seidel (GS) method. For a general system matrix M, guarantee that the SOR method converges for relaxation parameters $0 < \omega < 2$ and theoretical optimal value given by the formula [28]

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho^2}} \tag{10}$$

where ρ is the spectral radius of the Jacobi iteration matrix. The previous studies have shown the experimental value of ω to be in good agreement with the theoretical value [29, 30]. By determining values of matrices D, L and U as stated in Eq. (8), the general algorithm for FSSOR, HSSOR and QSSOR iterative methods to solve problem (1) would be generally described in Algorithm 1.

4. NUMERICAL SIMULATIONS

In order to compare the performances of the iterative methods described in the previous section, several experiments were carried out on the following Fredholm integral equations problems.

Example 1 [1] The integral equation

$$y(x) - \int_0^1 \left(4xt - x^2\right) y(t)\,dt = x, \quad 0 \le x \le 1 \tag{11}$$

with the exact solution given by

$$y(x) = 24x - 9x^2 .$$

Example 2 [31] The integral equation

$$y(x)-\int_0^1 (x^2+t^2)y(t)dt = x^6 - 5x^3 + x + 10\,,\;\; 0 \le x \le 1 \qquad (12)$$

Algorithm 1. FSSOR, HSSOR and QSSOR algorithms

For $i=0,p,2p,\cdots,n-2p,n-p,n$ and $j=0,p,2p,\cdots,n-2p,n-p,n$

Calculate

$$y_i^{(k+1)} \leftarrow \begin{cases} (1-\omega)y_i^{(k)} + \dfrac{\omega}{1-\dfrac{ph}{2}K_{i,i}}\left(\displaystyle\sum_{j=p}^{n-p} phK_{i,j}y_j^{(k)} + \frac{ph}{2}K_{i,n}y_n^{(k)} + f_i\right) & ,i=0 \\ (1-\omega)y_i^{(k)} + \dfrac{\omega}{1-phK_{i,i}}\left(\dfrac{ph}{2}K_{i,0}y_0^{(k+1)} + \displaystyle\sum_{j=i+p}^{n-p} phK_{i,j}y_j^{(k)} + \frac{ph}{2}K_{i,n}y_n^{(k)} + f_i\right) & ,i=p \\ (1-\omega)y_i^{(k)} + \dfrac{\omega}{1-phK_{i,i}}\left(\dfrac{ph}{2}K_{i,0}y_0^{(k+1)} + \displaystyle\sum_{j=p}^{i-p} phK_{i,j}y_j^{(k+1)} + \frac{ph}{2}K_{i,n}y_n^{(k)} + f_i\right) & ,i=n-p \\ (1-\omega)y_i^{(k)} + \dfrac{\omega}{1-\dfrac{ph}{2}K_{i,i}}\left(\dfrac{ph}{2}K_{i,0}y_0^{(k+1)} + \displaystyle\sum_{j=p}^{n-p} phK_{i,j}y_j^{(k+1)} + f_i\right) & ,i=n \\ (1-\omega)y_i^{(k)} + \dfrac{\omega}{1-phK_{i,i}}\left(\dfrac{ph}{2}K_{i,0}y_0^{(k+1)} + \displaystyle\sum_{j=p}^{i-p} phK_{i,j}y_j^{(k+1)} + \sum_{j=i+p}^{n-p} phK_{i,j}y_j^{(k)} + \frac{ph}{2}K_{i,n}y_n^{(k)} + f_i\right) & ,others \end{cases}$$

with the exact solution

$$y(x) = x^6 - 5x^3 + \frac{1045}{28}x^2 + x + \frac{2141}{84}$$

Example 3 [32] The integral equation

$$y(x) - \frac{1}{2}\int_0^{\frac{\pi}{2}} xt\,y(t)dt = \sin(x) - \frac{x}{2}\,,\quad 0 \le x \le 1 \qquad (13)$$

and the exact solution is

$$y(x) = \sin(x)$$

In comparison, the GS method acts as the comparison control of numerical results. There are three parameters considered in numerical comparison such as number of iterations, execution time and maximum absolute error. Throughout the experiments, the convergence test considered the tolerance error $\varepsilon = 10^{-10}$ and carried out on several different values of n. Meanwhile, the experimental values of ω were obtained by running the program for different values of ω and choosing the one(s) that gave the minimum number of iterations.

Results of numerical simulations, which were obtained from implementations of the GS, FSSOR, HSSOR and QSSOR iterative methods for Examples 1 - 3, have been recorded in Tables 1 - 3 respectively. Meanwhile, reduction percentages of the number of iterations and execution time for the FSSOR, HSSOR and QSSOR methods compared with GS method have been summarized in Table 4.

5. COMPUTATIONAL COMPLEXITY ANALYSIS

In order to measure the computational complexity of the FSSOR, HSSOR and QSSOR iterative methods, an estimation amount of the computational work required for both methods have been conducted. The computational work is estimated by considering the arithmetic operations performed per iteration. Based on Algorithm 1, it can be observed that there are $3+\dfrac{n}{p}$ additions/subtractions (ADD/SUB) and $7+3\left(\dfrac{n}{p}\right)$ multiplications/divisions (MUL/DIV) in computing a value for each node point in the solution domain for second kind linear Fredholm integral equations. From the order of the coefficient matrix M in Eq. (7), the total number of arithmetic operations per iteration for the FSSOR, HSSOR and QSSOR iterative methods in solving second kind linear Fredholm integral equations has been summarized in Table 5.

Table 1: Comparison of a number of iterations, execution time (seconds) and maximum absolute error for the iterative methods (Example 1)

	Number of iterations				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	194	194	195	195	195
FSSOR	138	139	140	140	140
	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)
HSSOR	136	138	139	140	140
	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)
QSSOR	136	136	138	139	140
	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)	($\omega = 1.84$)

	Execution time (seconds)				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	2.62	10.77	38.77	145.01	570.58
FSSOR	1.70	8.66	27.67	93.99	439.36
HSSOR	0.81	2.07	9.21	38.93	119.58
QSSOR	0.50	0.93	3.46	18.97	60.55

	Maximum absolute error				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	4.6922 E-4	1.1730 E-4	2.9325 E-5	7.3307 E-6	1.8321 E-6
FSSOR	4.6922 E-4	1.1730 E-4	2.9326 E-5	7.3313 E-6	1.8328 E-6
HSSOR	1.8771 E-3	4.6922 E-4	1.1730 E-4	2.9326 E-5	7.3313 E-6
QSSOR	7.5111 E-3	1.8771 E-3	4.6922 E-4	1.1730 E-4	2.9326 E-5

Table 2: Comparison of a number of iterations, execution time (seconds) and maximum absolute error for the iterative methods (Example 2)

	Number of iterations				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	56	56	56	56	56
FSSOR	44	44	44	44	44
	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)
HSSOR	44	44	44	44	44
	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)
QSSOR	43	44	44	44	44
	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)	($\omega = 1.53$)

	Execution time (seconds)				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	0.89	3.47	17.06	55.85	189.98
FSSOR	0.69	2.56	10.53	35.02	129.04
HSSOR	0.44	0.78	4.24	19.34	58.63
QSSOR	0.31	0.55	1.57	8.91	34.88

	Maximum absolute error				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	4.7770 E-4	1.1942 E-4	2.9856 E-5	7.4639 E-6	1.8659 E-6
FSSOR	4.7770 E-4	1.1942 E-4	2.9856 E-5	7.4640 E-6	1.8660 E-6
HSSOR	1.9108 E-3	4.7770 E-4	1.1942 E-4	2.9856 E-5	7.4640 E-6
QSSOR	7.6441 E-3	1.9108 E-3	4.7770 E-4	1.1942 E-4	2.9856 E-5

Table 3: Comparison of a number of iterations, execution time (seconds) and maximum absolute error for the iterative methods (Example 3)

	Number of iterations				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	29	29	29	29	29
FSSOR	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)
HSSOR	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)
QSSOR	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)	15 ($\omega = 1.19$)

	Execution time (seconds)				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	0.53	2.24	11.97	33.79	125.77
FSSOR	0.39	1.65	8.87	22.34	90.31
HSSOR	0.25	0.87	5.23	13.65	58.93
QSSOR	0.16	0.41	2.71	7.33	31.94

	Maximum absolute error				
Methods	***n***				
	512	**1024**	**2048**	**4096**	**8192**
GS	2.2111 E-3	2.2098 E-3	2.2095 E-3	2.2094 E-3	2.2094 E-3
FSSOR	2.2111 E-3	2.2098 E-3	2.2095 E-3	2.2094 E-3	2.2094 E-3
HSSOR	2.2164 E-3	2.2111 E-3	2.2098 E-3	2.2095 E-3	2.2094 E-3
QSSOR	2.2374 E-3	2.2164 E-3	2.2111 E-3	2.2098 E-3	2.2095 E-3

Table 4: Reduction percentages of the number of iterations and execution time for the FSSOR, HSSOR and QSSOR methods compared with GS method

Methods	**Number of iterations**		
	Example 1	**Example 2**	**Example 3**
FSSOR	28.20 – 28.87%	21.42 – 21.43%	48.27 – 48.28%
HSSOR	28.20 – 29.90%	21.42 – 21.43%	48.27 – 48.28%
QSSOR	28.20 – 29.90%	21.42 – 23.22%	48.27 – 48.28%

Methods	**Execution time**		
	Example 1	**Example 2**	**Example 3**
FSSOR	19.59 – 35.19%	22.47 – 38.28%	25.89 – 33.89%
HSSOR	69.08 – 80.78%	50.56 – 77.53%	52.83 – 61.17%
QSSOR	80.91 – 91.37%	65.16 – 90.80%	69.81 – 81.70%

Table 5: Total number of arithmetic operations per iteration for FSSOR, HSSOR and QSSOR methods

Methods	**Arithmetic Operation**	
	ADD/SUB	**MUL/DIV**
FSSOR	$n^2 + 4n + 3$	$3n^2 + 10n + 7$
HSSOR	$\frac{n^2}{4} + 2n + 3$	$\frac{3n^2}{4} + 5n + 7$
QSSOR	$\frac{n^2}{16} + n + 3$	$\frac{3n^2}{16} + \frac{5n}{2} + 7$

6. CONCLUSION

In this paper, we have presented an application of the half- and quarter-sweep iterations with SOR method for solving dense linear systems generated from the discretization of the second kind linear Fredholm integral equations by using RT scheme. Through numerical results obtained for Examples 1 - 3 (Tables 1 - 3), it clearly shows that applying the SOR methods can reduce number of iterations and execution time compared to the GS method. At the same time, it has been shown that applying the half- and quarter-sweep iterations reduces the computational time in the implementation of the iterative method, see Table 4.

Overall, the numerical results show that the QSSOR method is a better method compared to the GS, FSSOR and HSSOR methods in the sense of number of iterations and execution time. This is mainly because of the reduction in term of computational complexity; since the implementations of the QSSOR will only consider approximately quarter of all interior node points in a solution domain (refer Table 5).

ACKNOWLEDGEMENT

The authors acknowledge the Postgraduate Research Grant, Universiti Malaysia Sabah (GPS0003-SG-1/2009) for the completion of this article.

REFERENCES

[1] W. Wang, Appl. Math. Comput., 172 (2006) 946-962.
[2] K.E. Atkinson, The Numerical Solution of Integral Equations of the Second Kind, Cambridge University Press, United Kingdom, 1997.
[3] M.S. Muthuvalu and J. Sulaiman, Journal of Fundamental Sciences, 5(1) (2009) 7-16.
[4] C. Allouch, P. Sablonnière, D. Sbibih and M. Tahrichi, J. Comput. Appl. Math., 233 (2010) 2855-2866.
[5] G. Mastroianni and G. Monegato, SIAM J. Numer. Anal., 41(5) (2004) 1870-1892.
[6] S.-Y. Kang, I. Koltracht and G. Rawitscher, Math. Comput., 72(242) (2003) 729-756.
[7] J. Saberi-Nadjafi and M. Heidari, Appl. Math. Comput., 189 (2007) 980-985.
[8] F.-R. Lin, Calcolo, 40 (2003) 231-248.
[9] Y. Liu, Math. Comput. Model., 50 (2009) 465-469.
[10] P. M. Prenter, SIAM J. Numer. Anal., 10 (4) (1973) 570-581.
[11] K. Atkinson, I. Graham and I. Sloan, SIAM J. Numer. Anal., 20 (1) (1983) 172-186.
[12] Z. Chen, B. Wu and Y. Xu, Commun. Pure Appl. Anal., 6(3) (2007) 643-666.
[13] C. Cattani and A. Kudreyko, Appl. Math. Comput., 215 (2010) 4164-4171.
[14] G. Long, M.M. Sahani and G. Nelakanti, Appl. Math. Comput., 215 (2009) 147-155.
[15] J.-Y. Xiao, L.-H. Wen and D. Zhang, Appl. Math. Comput., 175 (2006) 508-518.
[16] A.R. Abdullah, Int. J. Comput. Math., 38 (1991) 61-70.
[17] M.S. Muthuvalu and J. Sulaiman, Matematika, 24(1) (2008) 75-84.
[18] M. Othman, J. Sulaiman and A.R. Abdullah, Malaysian Journal of Computer Science, 13(2) (2000) 1-6.
[19] J. Sulaiman, M.K. Hasan and M. Othman, Lect. Notes Comput. Sc., LNCS 4487 (2007) 326-333.
[20] A.R. Abdullah and N.H.M. Ali, Parallel Algorithms and Applications, 10 (1996) 93-103.
[21] W.S. Yousif and D.J. Evans, Parallel Algorithms and Applications, 7 (1995) 53-71.
[22] M. Othman and A.R. Abdullah, Int. J. Comput. Math., 76 (2000) 203-217.
[23] S.I. Rakhimov and M. Othman, Sains Malaysiana, 38(5) (2009) 729-733.
[24] J. Sulaiman, M. Othman and M.K. Hasan, Chamchuri J. Math., 1(2) (2009) 93-103
[25] M. Othman and A.R. Abdullah, Lect. Notes Comput. Sc., LNCS 2127 (2001) 480-489.
[26] J. Sulaiman, M. Othman and M.K. Hasan, Int. J. Comput. Math., 81(12) (2004) 1559-1565.
[27] N. Jha and B.K. Mishra, International Journal of Mathematical Modeling, Simulation and Applications, 1(1) (2008) 41-45.
[28] S. Yang and M.K. Gobbert, Appl. Math. Lett., 22 (2009) 325-331.
[29] N.H.M. Ali and L.S. Chong, Appl. Math. Comput., 191 (2007) 533-542.
[30] S.I. Rakhimov and M. Othman, Sains Malaysiana, 38(5) (2009) 729-733.
[31] A.D. Polyanin and A.V. Manzhirov, Handbook of Integral Equations, CRC Press LCC, Florida, 1998.
[32] S. Rahbar and E. Hashemizadeh, Proceedings of the World Congress on Engineering, 2 (2008) 933-937.