Quarter-sweep Nonlocal Discretization Scheme with QSSOR Iteration for Nonlinear Two-point Boundary Value Problems

Quarter-sweep Nonlocal Discretization Scheme with QSSOR Iteration for Nonlinear Two-point Boundary Value Problems

M. U. Alibubin[1] , A. Sunarto[2] , and J. Sulaiman[3]
[1,2,3] Faculty of Science and Natural Resources,
Universiti Malaysia Sabah, Malaysia.

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

Abstract. The aim of this paper is to consider the Quarter-sweep Successive Over Relaxation (QSSOR) iteration for solving nonlinear two-point boundary value problems. The second order finite difference (FD) method is applied to derive the quarter-sweep nonlocal discretization scheme for the sake of transforming the system of nonlinear approximation equations into the corresponding system of linear equations. The formulation and the implementation of the methods are discussed. In addition, the numerical results by solving the proposed problems using QSSOR method are included and compared with the Full-sweep Successive Over Relaxation (FSSOR) and Half-sweep Successive Over Relaxation (HSSOR) methods.

1. Introduction

For the numerical solution of nonlinear two-point boundary value problems (TPBVP), the paper deals with the nonlocal finite difference methods which are used to determine the numerical value at the whole grid points with difference grid sizes. In fact again, the solution to a nonlinear TPBVP must satisfy the boundary conditions. Actually, the nonlinear TPBVP plays the important role to describe various physical problems in sciences, economics and engineering which are designed mathematically by using nonlinear equation model [1-6]. However in many cases, the nonlinear two-point boundary value problem does not have exact analytic solution. Therefore numerical techniques must be used to get the approximate solution of the problems. Nonlinear boundary value problems have attracted much attention from many researchers. For instance, a finite difference (FD) method has been proposed in recent works [1], finite element method [2], shooting method [3], spline approximation method [4] and Sinc-Galerkin method [5]. Literally, there are various iterative methods also have been studied to yield the fast numerical solution of linear systems [7, 8, 9]. Recently, Akhir et al [10] suggested a Half-Sweep Modified SOR (HSMSOR) method by bringing together the concept of the half-sweep iterations and the modified SOR method to solve two-dimensional Helmholtz equation. Actually, the concept of the half-sweep (HS) iteration method has been introduced by Abdullah [11] via the Explicit Decoupled Group (EDG) method to solve two-dimensional Poisson equations. Similarly, further investigations have been extensively conducted in [12, 13, 14, 15, 16]. In 2000, Othman and Abdullah continued this concept by introducing quarter-sweep (QS) iterative method via the Modified Explicit Group (MEG) method to solve two-dimensional Poisson equations [17]. Further studies to prove the effectiveness of the QS iterative methods have been carried out by [18, 19, 20, 21]. The fundamental concept of the HS and QS iterative methods is to reduce the computational complexities during iteration process in which HS and QS iterations will only consider nearly half and quarter of all

interior node points in a solution domain respectively. Thus, in this paper we examined the applications of the QS iteration concept with SOR iterative method by using approximation equation based on quarter-sweep nonlocal discretization scheme for solving proposed problems. The standard SOR iterative method is also called as the Full-Sweep Successive Over Relaxation (FSSOR) method. Meanwhile, combinations of the SOR method with HS and QS iterations are called as Half-Sweep Successive Over-Relaxation (HSSOR) and Quarter-Sweep Successive Over-Relaxation (QSSOR) methods respectively.
To investigate the performance of these three proposed iterative methods, let us consider the general form of a nonlinear two-point boundary value problem being defined as

$$-\frac{d^2U}{dx^2} = g(x,U,U'),\ a \le x \le b \tag{1}$$

subject to the boundary conditions

$$U(a) = \beta_0, \quad U(b) = \beta_1$$

and β_0, β_1, and $g(x,U,U')$ are constants and a nonlinear continuous function**,** respectively. As mentioned in the first paragraph, the importance of problem (1) deals with many physical problems. Therefore, the study investigates the effectiveness of QS nonlocal discretization scheme in solving the problem (1) via QSSOR iteration.

In formulating various iterative schemes such as full-, half- and quarter-sweep iterations, finite grid network can be used as a guide as well as to facilitate us in terms of development and implementation of the corresponding proposed algorithms for the proposed methods. According to the point iterations, the implementation of these three iterative methods will be applied onto the node points of the same type • until the iterative convergence fixed is achieved [22]. Based on Fig. 1, the FS , HS and QS 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 [11, 19].

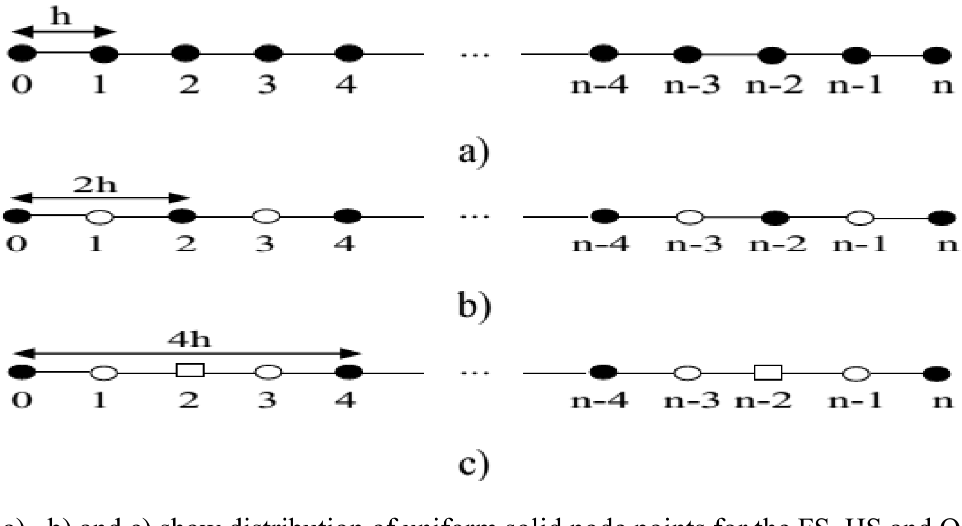

Figure 1 a) , b) and c) show distribution of uniform solid node points for the FS, HS and QS cases respectively.

Based on Figure 1, the following discussion will be restricted to divide the solution domain [a, b] of problem (1) into subintervals in which the distance of the subinterval, Δx is defined in Eq. (2).

$$\Delta x = \frac{(b-a)}{m} = h,\ n = m-1 \tag{2}$$

Based on Figure (1), clearly the distance of solid node points for quarter-sweep is $4h$ meanwhile the full-sweep and half-sweep h and $2h$ respectively as shown in Eq. (2).

2. Formulations of Quarter-sweep Nonlocal Discretization Scheme

Before constructing the Quarter-sweep FD approximation equation of problem (1), let us consider several nonlocal quarter-sweep discretization schemes being given as follows [23]

$$U_i^2 = U_i U_{i+4} \tag{3}$$

$$U_i^2 = \left(\frac{U_{i-4}+U_{i+4}}{2}\right)U_i \tag{4}$$

$$U_i^2 = \left(\frac{U_{i-4}+U_{i+4}}{2}\right)U_i^2 \tag{5}$$

By using the approach of second-order quarter-sweep FD discretization scheme, the corresponding approximation equations for problem (1) can be easily shown as

$$-U_{i-4} + 2U_i - U_{i+4} - (h)^2 f_i(U_4, U_8, \cdots, U_{n-3}) = 0, \quad i = 4, 8, 12, \cdots, n-3 \tag{6}$$

Where

$$f_i(U_4, U_8, \cdots, U_{n-3}) = g\left(x_i, U_{i,} \frac{U_{i+4}-U_{i-4}}{8(h)}\right) \tag{7}$$

Actually, Eq. (7) is called as the nonlinear term of problem (1). To solve the nonlinear system in Eq. (6), the nonlocal discretization scheme is used to transform the nonlinear system into the form of a system of linear equations. In this paper, however, we consider the nonlocal discretization scheme in Eq. (4) being imposed over the nonlinear approximation equation (6). Therefore, Eq. (7) can be rewritten as follows

$$f_i(U_4, U_8, \cdots, U_{n-3}) = g\left(x_i, \frac{U_{i-4}+U_{i+4}}{2}, \frac{U_{i+4}-U_{i-4}}{8(h)}\right) \tag{8}$$

3. Formulations of Quarter-sweep Successive Over Relaxation method

In this section, we present on how to derive the formulation of the QSSOR methods. Based on the approximation equation in Eqs. (6) and (8), the general scheme of the QSSOR method can be stated as

$$U_i^{(k+1)} = (1-\omega)U_i^{(k)} + \frac{\omega}{2}\left(\begin{matrix} U_{i-4}^{(k+1)} + U_{i+4}^{(k)} \\ + h^2 f_i\left(U_4^{(k+1)}, U_8^{(k+1)}, \cdots, U_{i-4}^{(k+1)}, U_{i+4}^{(k)}, \cdots, U_n^{(k)}\right)\end{matrix}\right), i = 4, 8, 12, \cdots, n-3 \tag{9}$$

where ω and $U_i^{(k)}, 4, 8, 12, \cdots n-3$ represent as a relaxation factor and the k[th] estimation for corresponding approximate solutions respectively. Actually the FSSOR iterative method in form of discussed Eq. (9) was by Young [7]. Practically, the optimal value of ω in range $1 \le \omega < 2$ will be obtained by implementing several computer programs and then the best approximate value of ω is chosen in which its number of iterations is the smallest. The general algorithm for the QSSOR iterative methods to solve the linear equations (9) would be generally described in Algorithm 1 [18]

Algorithm 1 : QSSOR scheme

i. Initialize $\underset{\sim i}{U^{(0)}} \leftarrow 0, \varepsilon \leftarrow 10^{-10}$

ii. Assign the value of ω

iii. For $i = 4, 8, 12, \ldots, n-3$, calculate

$$U_i^{(k+1)} = (1-\omega)U_i^{(k)} \frac{\omega}{2}\left(\begin{array}{l} U_{i-4}^{(k+1)} + U_{i+4}^{(k)} \\ + h^2 f_i\left(U_1^{(k+1)}, U_2^{(k+1)}, \cdots, U_{i-4}^{(k+1)}, U_{i+4}^{(k)}, \cdots, U_n^{(k)}\right) \end{array} \right)$$

iv. Check the convergence test, $\left|U_i^{(k+1)} - U_i^{(k)}\right| \le \varepsilon = 10^{-10}$. If yes, go to step (iv). Otherwise go back to step (iii).

v. Display approximate solutions.

4. Numerical Experiments

In order to validate the performance of the FSSOR, HSSOR and QSSOR iterative methods together with the nonlocal approach, three nonlinear example problems were tested. For the sake of comparison, three criteria will be considered for these three proposed iterative methods which are number of iterations, computation time (in seconds) and maximum absolute error.

Example 1 [24]

$$\varepsilon U'' + 2U' - e^U = 0, \qquad \text{for} \qquad 0 \le x \le 1 \tag{10}$$

subject to the boundary conditions

$$U(0) = 0 \qquad U(1) = 0$$

with exact solution were defined by

$$U(x) = \log\left(\frac{2}{1+x}\right) - \exp\left(\frac{-2x}{\varepsilon}\right)\log 2. \tag{11}$$

Example 2 [25]

$$U''(x) = \frac{3}{2}U^2, \quad \text{for} \quad 0 < x < 1 \tag{12}$$

subject to the boundary conditions

$$U(0) = -4, \quad U(1) = 1$$

with exact solution were defined by

$$U(x) = \frac{4}{(1+x)^2}. \tag{13}$$

Example 3 [26]

$$U''(x) + \frac{0.5}{x}U'(x) = e^{U(x)}\left(0.5 - e^{U(x)}\right), \quad \text{for} \quad 0 < x < 1 \tag{14}$$

subject to the boundary conditions

$$U(0) = \log[2], \quad U(1) = 0$$

From above three examples, results of numerical experiments obtained have been summarized in Table 1. In the implementation approach, the convergence test considered the tolerance error $\varepsilon = 10^{-10}$.

5. Numerical Experiments

In this paper, the performance of Quarter-sweep nonlocal discretization scheme with QSSOR method for the solution of nonlinear two-point boundary value problem associated with the second order finite difference approximation scheme has been investigated. Based on Table 1, numerical results showed that QSSOR method solved the proposed problems with least number of iterations as compared to the FSSOR and HSSOR methods. Meanwhile, in terms of computation time, QSSOR method computes with the fastest time for all considered mesh sizes. In the aspect of accuracy, numerical solutions obtained for test problems 1 to 3 are comparable for all the tested iterative methods. Finally, it can be concluded that the QSSOR method is superior to FSSOR and HSSOR methods. This is mainly because of the reduction of computational complexity in which the QSSOR method will only consider approximately quarter of all interior node points in a solution domain during iteration process.

References

[1] E. Doedel, *SIAM Journal. Numerical Analysis*, **16**,173-185 (1979).
[2] A. G. Deacon and S. Osher, *SIAM Journal. Numerical Analysis*. **16**, 756-778 (1979).
[3] S. M. Roberts and J. S. Shipman, *American Elsevier, New York*, (1972).
[4] T.Y. Na, *Academic, New York*, (1979).
[5] M. El-Gamel, S.H. Behiry, H. Hashih, *Appl. Math. Comput*, **145**, 717-734 (2003).
[6] A.A. Salama and A.A. Mansou, *Numer. Heat Tr,* **47**, 383–401 (2005).
[7] D.M. Young, *Trans. Amer. Math. Soc.,* **76**, 92-111(1954).
[8] W. Hackbuusch, *Springer-Verlag, New York,* (1995).
[9] Y. Saad, *United States America:PWS Publishing Company*, (1996).
[10] M. K. Akhir, M. Othman, J. Sulaiman, Z.A Majid, *Proceedings of Fundamental Science (FSC)*, 88-89 (2009).
[11] A.R. Abdullah, *Int. J. Comp. Math*, **38,** 61-70 (1991).
[12] M.K. Hassan, J. Sulaiman, S.A.A. Karim, M. Othman, *Journal of Applied Sciences,* **11**(7), 1255-1260 (2011).
[13] E. Aruchunan and J. Sulaiman, *Australian Journal of Basic and Applied Sciences*, **5** (3), 38-43 (2011).
[14] W. S. Yousif and D. J. Evans, *Implementation. Paral. Algo. App*, **7,** 63-71 (1995).
[15] A. R. Abdullah and A. Ibrahim, *Int. J. Comp. Math*, **58**, 253-256 (1995).
[16] M. Othman and A.R Abdullah. *Int. J. Comp. Math,* **69**, 219-229 (1998).
[17] J. Sulaiman, M.K. Hasan and M. Othman, *In Computational and Information Science, Zhang, J., He, J.H. and Fu. Yu., (Eds). Berlin: Springer-Verlag,* 57-63. (2004).
[18] N.I.M. Fauzi and J, Sulaiman, *Journal of Applied Sciences*, **12**(17), 1817-1824 (2012).
[19] M.K. Akhir, M. Othman, J. Sulaiman and Z.A. Majid, *Proceedings of International Conference on Intelligent Network and Computing, IEEE Press. ISBN*: 978-1-4244-8270-2, 416-419 (2010).
[20] M. Othman and A.R. Abdullah, *International Journal of Computer Mathematics*, **76**, 203-217(2000).
[21] M. Othman and A.R. Abdullah, *In V. Malyshkin (Ed.)*, *Lectures Notes in Computer Science, Berlin: Springer-Verlae,* 480-489, (2001).
[22] M.K. Hasan, M. Othman, R. Johari, Z. Abbas and J. Sulaiman, *In B.M. Ali et al. (Eds.), 13th IEEE International Conference on Network,* 1002-1007 (2005).
[23] J. Sulaiman, M.K. Hasan, M.Othman, *In J. Zhang et al. (Eds.), Lectures Notes in Computer Science,* 57-63 (2004).
[24] M.K. Hasan, J. Sulaiman, and M. Othman, *In H.B. Zaman et al. (Eds.), International Symposium on Information Technology,* 1864-1869 (2008).
[25] M.K. Ahir, M. Othman, J. Sulaiman, Z. A. Majid and M. Suleiman, **5**(12), 3033-3039 (2011).
[26] K. Moaddy, I. Hashim, A. K. Alomari and S. Momani, *Sains Malaysiana,* **40**(5), 515–519 (2011).
[27] M.K. Kadalbajoo and A.S.Yadaw, *Applied Mathematics and Computation*, **201**, 504–513 (2008).
[28] S.N. Ha, *Comput. Math. App*. **42**, 1411-1420 (2001).
[29] H. Jafari, M. Ahmadi and S. Sadeghi, *Applications and Applied Mathematics. An International Journal,* **7**(1), 357-364 (2012).

TABLE 1. Comparison between number of iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example at grid sizes 1024, 2048, 4096, 8192, 16384

	EXAMPLE 1			EXAMPLE 2			EXAMPLE 3		
Number of iteration									
M	**FSSOR**	**HSSOR**	**QSSOR**	**FSSOR**	**HSSOR**	**QSSOR**	**FSSOR**	**HSSOR**	**QSSOR**
1024	3842	1949	982	3011	1527	769	5751	2880	1418
2048	7526	3842	1949	6029	2914	1527	11819	5748	2880
4096	14625	7526	3842	11463	5413	2914	22356	11297	5748
8192	27951	14625	7526	22096	10331	5413	49154	21793	11297
16384	53169	27951	14625	42248	20031	11255	98306	2880	21793
Execution time (seconds)									
M	**FSSOR**	**HSSOR**	**QSSOR**	**FSSOR**	**HSSOR**	**QSSOR**	**FSSOR**	**HSSOR**	**QSSOR**
1024	1.07	0.44	0.29	0.57	0.39	0.26	1.78	1.15	0.45
2048	2.47	0.95	0.55	1.41	0.72	0.44	6.47	3.06	0.83
4096	7.50	2.35	1.00	5.32	2.15	0.83	24.35	8.63	2.56
8192	24.86	7.85	2.51	17.85	6.10	2.10	95.68	30.36	8.59
16384	91.06	26.63	7.81	65.21	21.84	6.03	387.77	128.83	27.66
Maximum errors									
M	**FSSOR**	**HSSOR**	**QSSOR**	**FSSOR**	**HSSOR**	**QSSOR**	**FSSOR**	**HSSOR**	**QSSOR**
1024	1.787947e-02	1.787945e-02	9.319738e-02	3.669275e-06	1.470019e-05	5.878965e-05	8.344832e-07	3.293855e-06	1.295298e-05
2048	1.787948e-02	1.787947e-02	9.350227e-02	9.338850e-07	3.690722e-06	1.470019e-05	2.540690e-07	8.354497e-07	3.293860e-06
4096	1.787951e-02	1.787949e-02	9.365475e-02	2.701202e-07	9.428743e-07	3.690722e-06	1.740979e-07	2.226486e-07	8.354497e-07
8192	1.787961e-02	1.787951e-02	9.373101e-02	1.420423e-07	2.774780e-07	9.428743e-07	1.293783e-08	1.368791e-07	2.226486e-07
16384	1.787976e-02	1.787961e-02	9.376914e-02	2.192884e-07	2.111116e-07	2.762543e-07	3.207085e-09	1.231990e-08	1.368792e-07