

## SOLUTION OF FOURTH-ORDER TWO-POINT BVPS WITH CUBIC NON-POLYNOMIAL SPLINE AND SOR ITERATIVE METHOD

H. Justine\* and J. Sulaiman

Mathematics with Economics Programme, Faculty of Science and Natural Resources,  
Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia

Published online: 17 October 2017

---

### ABSTRACT

Two-point boundary value problems (BVPs) have numerous applications especially in the modeling of most physical phenomena. In this study, the fourth order two-point BVPs were solved by using Successive Over-Relaxation (SOR) iterative method after discretized with cubic non-polynomial spline scheme to generate its corresponding sparse linear system. Then, a numerical experiment was conducted to determine the performances of the SOR iterative method and the spline scheme in terms of iterations number, execution time and maximum absolute error at different mesh sizes. In order to assess the performances of this proposed idea, another iterative method was also conducted which is Gauss-Seidel (GS). Based on the numerical analysis, the two-point BVPs together with the cubic non-polynomial spline scheme were found to be best solved by using the SOR iterative method.

**Keywords:** successive over-relaxation; cubic non-polynomial spline; fourth-order two-point boundary value problem.

---

Author Correspondence, e-mail: [cherryjust20@gmail.com](mailto:cherryjust20@gmail.com)

doi: <http://dx.doi.org/10.4314/jfas.v9i5s.41>

## 1. INTRODUCTION

This current study aims to solve for fourth-order two-point BVPs as given by Equation (1)

$$y^{(4)} + a_1(x)y^{(3)} + a_2(x)y'' + a_3(x)y' + a_4(x)y = f(x), \quad x \in [a, b] \quad (1)$$

with its boundary condition as shown in Equation (2) as follows

$$y(a) = \alpha_1, \quad y(b) = \beta_1 \quad (2)$$

$$y''(a) = \alpha_2, \quad y''(b) = \beta_2$$

where  $a_1(x)$ ,  $a_2(x)$ ,  $a_3(x)$ ,  $a_4(x)$  and  $f(x)$  are known functions of domain  $[a, b]$  whereas  $\alpha_i$  and  $\beta_i$  are constants with  $i = 1, 2$ .

Equation (1) has to undergo a process lowering of order to reduce the single equation of fourth-order problem into two separated equations of second-order problems as below

$$v''(x) + a_1(x)v'(x) + a_2(x)v(x) = G(x), \quad x \in [a, b] \quad (3) \quad \text{with}$$

$$G(x) = f(x) - a_3(x)y'(x) - a_4(x)y(x), \quad (4)$$

$$V(a) = y''(a) = \alpha_2$$

$$V(b) = y''(b) = \beta_2$$

and

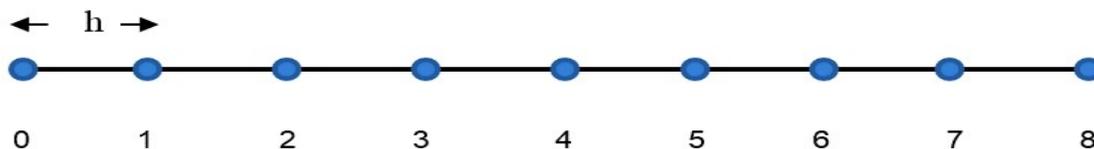
$$y''(x) = v(x) \quad (5)$$

where

$$y(a) = \alpha_1, \quad y(b) = \beta_1 \quad (6)$$

The process was completely done by using simple mathematical differentiation and substitution. Then, by considering [1-5], the cubic non-polynomial spline scheme was implemented to discretize both equations (3) and (5) to generate two systems of linear equations. There are many other numerical methods which have sought wide interest among researchers and some of them are collocation method [6], adomian decomposition method [7], finite element and finite volume methods [8], finite-difference method [8], non-linear shooting method [9] and precise time integration method [10]. Nonetheless, BVPs were usually solved with three standard approaches which are shooting, finite differences and projections [11].

To make the discretization process simpler, the solution domain  $[a,b]$  was divided uniformly into subintervals,  $m$  as shown in Fig. 1.



**Fig.1.** Separation of uniform subintervals,  $m = 8$

where the length,  $\Delta x$  of each subinterval is given as follows

$$\Delta x = \frac{b - a}{m} = h \tag{7}$$

Then, Equation (3) and (5) were solved iteratively at once by using SOR iterative method by referring to [12-15]. In order to assess the performance of this proposed idea clearly in term of iterations number, execution time and maximum absolute error, GS iterative method was also conducted for comparison purpose.

In response to the vast applications of two-point BVPs as stated by [16-20] regardless of fields, it is the main focus of this present study to initiate a numerical idea in term of technique and algorithm, as well as to provide a reference for future research. To be precise, a preparation for application of spline scheme at a different degree for solving two-point BVPs of high-order with different iterative methods.

## 2. METHOD OF SOLUTION

The general function of spline which was given consideration is as follows

$$S(x) = Q_i(x), \quad x \in [x_i, x_{i+1}] \quad i = 0, 1, \dots, n \tag{8}$$

and the approach starts with a derivation of non-polynomial spline equations from the general function of cubic non-polynomial spline as given by Equation (9)

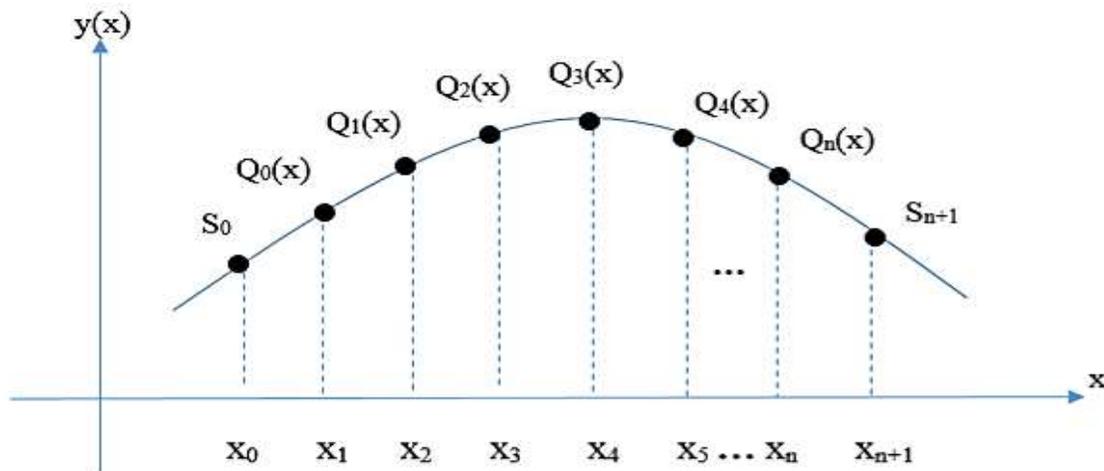
$$Q_i(x) = a_i \cos k(x - x_i) + b_i \sin k(x - x_i) + c_i(x - x_i) + d_i \tag{9}$$

where  $a_i, b_i, c_i$  and  $d_i$  are constants for  $i = 0, 1, \dots, n$  and  $k$  is the frequency for the trigonometric function in Equation (8) as  $k \rightarrow 0$ . Before discretized, Equation (8) was first defined into several variables as below.

$$Q_i(x_i) = y_i, \quad Q_i(x_{i+1}) = y_{i+1}, \quad Q_i'(x_i) = D_i, \quad (10)$$

$$Q_i'(x_{i+1}) = D_{i+1}, \quad Q_i''(x_i) = S_i, \quad Q_i''(x_{i+1}) = S_{i+1}.$$

By assuming the value of  $y(x)$  as the exact solution, a set of non-polynomial spline equations,  $S_i$  for  $y_i = y(x_i)$  was obtained from all the segments of  $Q_i(x)$  which passing through the points  $(x_i, S_i)$  and  $(x_{i+1}, S_{i+1})$  as shown in Fig. 2.



**Fig.2.** Illustration of non-polynomial spline equations for  $m=8$

Then, this discretization process ended with all the values of constant  $a_i, b_i, c_i$  and  $d_i$  were expressed in term of  $y_i, y_{i+1}, D_{i+1}, D_i, S_i, S_{i+1}$  in the following form

$$a_i = h^2 \frac{-S_{i+1} + S_i \cos(\theta)}{\theta^2 \sin(\theta)}, \quad (11)$$

$$b_i = -h^2 \frac{S_i}{\theta^2},$$

$$c_i = \frac{y_{i+1} - y_i}{h} + h \frac{(S_{i+1} + S_i)}{\theta^2},$$

$$d_i = y_i + h^2 \frac{S_i}{\theta^2}$$

where  $\theta = kh$  and  $i = 0, 1, 2, \dots, N$ .

However, all the constant values must be substituted in the continuity equation  $Q_{i-1}^m(x) = Q_i^m(x)$  where  $m = 0, 1$  and has to be solved simultaneously before the general approximation equation can be formed as below

$$-\left(\frac{1}{4h}\right)y_{i-1} + \left(\frac{1}{2h}\right)y_i - \left(\frac{1}{4h}\right)y_{i+1} + h\alpha S_{i-1} + 2h\beta S_i + h\alpha S_{i+1} = 0 \tag{12}$$

where

$$\alpha = \left(\frac{1}{\theta \sin 4\theta} - \frac{1}{4\theta^2}\right), \quad \beta = \left(\frac{1}{4\theta^2} - \frac{4\theta \cos 4\theta}{\theta \sin 4\theta}\right) \quad \text{and } i = 0,1,2,\dots, N.$$

In order to eliminate all the variables  $S$  in Equation (12), finite difference approach in Equation (13) composes of central finite difference, backward finite difference, and forward finite difference were used to replace all the variables.

$$\begin{aligned} S_{i-1} &= -f_{i-1}y'_i - q_{i-1}y_{i-1} + g_{i-1} \\ S_i &= -f_iy'_i - q_iy_i + g_i \end{aligned} \tag{13}$$

$$S_{i+1} = -f_{i+1}y'_{i+1} - q_{i+1}y_{i+1} + g_{i+1}$$

where,

$$\begin{aligned} y'_i &= \frac{y_{i+1} - y_{i-1}}{2h}, \\ y'_{i-1} &= \frac{-y_{i+1} + 4y_i - 3y_{i-1}}{2h}, \\ y'_{i+1} &= \frac{3y_{i+1} - 4y_i + y_{i-1}}{2h} \end{aligned}$$

The general approximation equation was finally obtained as given by Equation (14) as follows

$$a_iy_{i-1} + b_iy_i + c_iy_{i+1} = F_i, \quad i = 0,1,2,\dots, n \tag{14}$$

where

$$a_i = -\mu_0 + \gamma\left(\frac{3p_{i-1}}{2h} - q_{i-1}\right) + \theta\frac{p_i}{2h} - \gamma\frac{p_{i+1}}{2h}, \quad b_i = 2\mu_0 - \gamma\frac{2p_{i-1}}{h} - \theta q_i + \gamma\frac{2p_{i+1}}{h}$$

$$c_i = -\mu_0 + \gamma\frac{p_{i+1}}{2h} - \theta\frac{p_i}{2h} + \gamma\left(\frac{3p_{i+1}}{2h} - q_{i+1}\right), \quad F_i = -\gamma f_{i-1} - \frac{\theta}{2h} f_i + \gamma f_{i+1}$$

$$\gamma = \mu_1 h^2, \quad \phi = 2\mu_2 h^2,$$

$$\mu_1 = \theta - \sin \theta, \quad \mu_2 = \sin \theta - \theta \cos \theta.$$

By using computer program, Equation (14) was used to construct a system of linear equations in a matrix form of Equation Equation(15) as below

$$Ay = \vec{F} \tag{15}$$



**Table 1.** Algorithm of SOR iterative method

- 
- i. Initialize  $U_i^{(0)} \leftarrow 0, \varepsilon \leftarrow 10^{-10}$
  - ii. Assign the optimal value of  $\omega$
  - iii. Solve the system by calculating  $U_i^{(k+1)}$  using
 
$$\tilde{S}^{(k+1)} = (1-\omega)\tilde{S}^{(k)} + \omega(D+L)^{-1}(-U\tilde{S}^{(k)} + F)$$
  - iv. Check the convergence test,
 
$$|U_i^{(k+1)} - U_i^{(k)}| \leq \varepsilon = 10^{-10}$$

If yes, go to step (v). Otherwise go back to step (iii).
  - v. Display the approximate solutions.
- 

#### 4. NUMERICAL EXPERIMENT

The numerical experiments were conducted to solve for three examples of fourth-order two-point BVPs. These three examples together with its exact solution are as follow

##### Example 1 [21]

$$y^{(4)}(x) - 2y''(x) + y(x) = -8(\exp(x)),$$

$$y(0) = y(1) = 0,$$

$$y''(0) = 0,$$

$$y''(1) = -4e$$

with its exact solution

$$y(x) = x(1-x)e^x$$

##### Example 2 [21]

$$y^{(4)}(x) - 2y''(x) + yx = 0,$$

$$y(0) = y''(0) = 1$$

$$yu(1) = y''(1) = e$$

with its exact solution

$$y(x) = e^x.$$

**Example 3** [21]

$$y'' - y = -4(2x\cos(x) + 3\sin(x))$$

$$y(0) = y(1) = 0$$

$$y''(0) = 0$$

$$y''(1) = 2\sin(1) + 4\cos(1)$$

with its exact solution

$$y(x) = (x^2 - 1)\sin(x).$$

**5. RESULTS AND DISCUSSION**

Based on the numerical experiments, the performance results of all examples in terms of iterations number (K), execution time (t) in seconds, maximum absolute error ( $\epsilon$ ) were successfully tabulated in Table 2, 3 and 4 respectively. It can be seen that as the mesh sizes increasing, the iterations number also increasing due to the accumulated round-off error occurred at every iteration. However, the main aim is to compare which method will give the best performance, thus focus should be given on the performance comparison between these two iterative methods.

**Table 2.** Result of performances for Example 1

Size	GS			SOR		
	K	t	$\epsilon$	K	t	$\epsilon$
128	26789	1.72	1.1323e-05	546	1.33	1.1471e-05 $\omega = 1.9574$
256	98198	6.62	2.2716e-06	1024	1.61	2.8723e-06 $\omega = 1.9772$
512	356894	31.0	1.8931e-06	1948	3.43	7.2508e-07 $\omega = 1.9875$
1024	1283254	167.5	1.0200e-05	3963	6.97	1.8881e-07 $\omega = 1.9943$
2048	4551026	1004.2	4.1816e-05	7682	13.94	6.3598e-08 $\omega = 1.9969$

**Table 3.** Result of performances for Example 2

Size	GS			SOR		
	K	t	$\epsilon$	K	t	$\epsilon$
128	24843	1.59	2.1023e-06	513	1.11	1.8742e-06 $\omega = 1.8911$
256	90990	6.13	1.8885e-06	382	1.30	1.4840e-06 $\omega = 1.9439$
512	330472	28.9	2.5537e-06	844	1.48	2.0185e-06 $\omega = 1.9709$
1024	1188037	154.9	9.6770e-06	1438	1.9	6.0855e-07 $\omega = 1.9858$
2048	4216978	938.8	3.8595e-05	2850	3.54	1.4989e-07 $\omega = 1.9931$

**Table 4.** Result of performances for Example 3

Size	GS			SOR		
	K	t	$\epsilon$	K	t	$\epsilon$
128	29891	1.97	4.9993e-06	512	1.13	4.8161e-06 $\omega = 1.9551$
256	109207	7.29	1.9401e-06	996	2.39	1.2057e-06 $\omega = 1.9772$
512	395358	34.3	3.2802e-06	1951	2.85	2.9898e-07 $\omega = 1.9886$
1024	1415064	186.6	1.2053e-05	3763	6.13	6.9839e-08 $\omega = 1.9943$
2048	4991340	1103.1	4.8194e-05	7406	8.55	1.6102e-08 $\omega = 1.9972$

By referring to all the tables as mentioned previously, it was found that SOR method performed better than GS method when solving Example 1, 2 and 3. This can be seen clearly

through the difference of iterations number, execution time and maximum absolute error at different mesh sizes which are 128, 256, 512, 1024 and 2048. This difference was further analyzed in percentage form so that the improvement can be evidently presented.

The SOR iterative method can be said to perform better compared to GS method when solving the fourth-order two-point BVPs, as it required lesser iteration numbers and shorter time execution in order to iterate and converge to its exact solution. In fact, by looking at the maximum absolute error, it has higher accuracy compared to GS. As to complement the tabulated results of performance, Fig.1, Fig.2 and Fig.3 were presented to clarify that the problems are best solved by using SOR method. All the values of the percentage decrease were tabulated in Table 5, 6 and 7.

**Table 5.** Percentage decrease for Example 1

<b>Improvement of SOR Over GS (%)</b>			
<b>Size</b>	<b>K</b>	<b>t</b>	<b><math>\epsilon</math></b>
128	97.96%	22.67%	1.31%
256	98.96%	75.68%	26.44%
512	99.45%	88.94%	61.69%
1024	99.69%	95.84%	98.15%
2048	99.83%	98.61%	99.84%

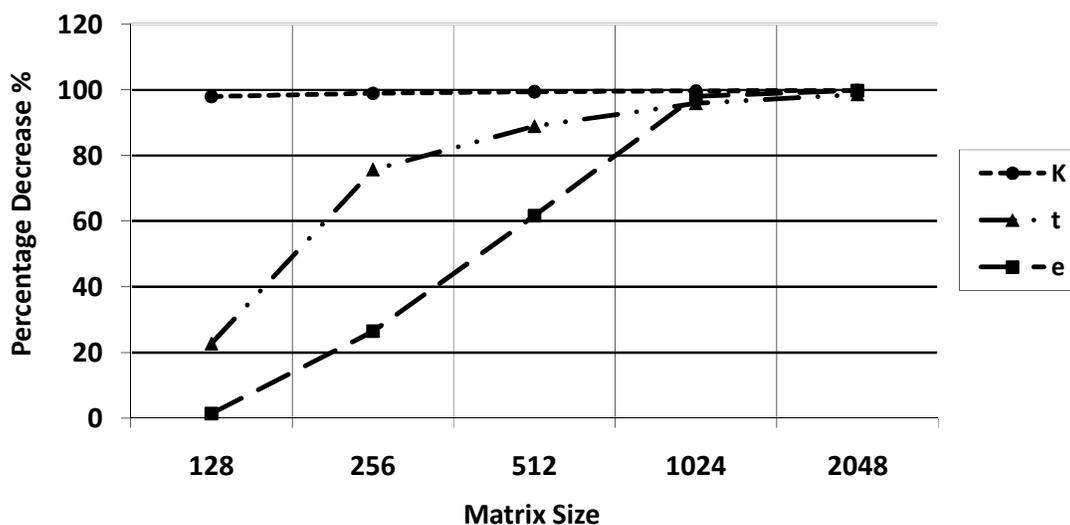
**Table 6.** Percentage decrease for Example 2

<b>Improvement of SOR Over GS (%)</b>			
<b>Size</b>	<b>K</b>	<b>t</b>	<b><math>\epsilon</math></b>
128	97.94%	30.19%	10.85%
256	99.58%	78.79%	21.46%
512	99.74%	94.88%	20.96%
1024	99.88%	98.77%	93.71%
2048	99.93%	99.62%	99.61%

**Table 7.** Percentage decrease for Example 3

Improvement of SOR Over GS (%)			
Size	K	t	$\epsilon$
128	98.29%	42.64%	3.66%
256	99.09%	67.22%	37.85%
512	99.51%	91.69%	90.89%
1024	99.73%	96.71%	99.42%
2048	99.85%	99.22%	99.97%

It should be noted that the percentage decrease was considered as an improvement of SOR method over GS method and was calculated by setting the performance of GS as an initial value, whereas SOR as the final value and the negative sign of percentage decrease was omitted from the graph.



**Fig.1.** Percentage decrease for Example 1

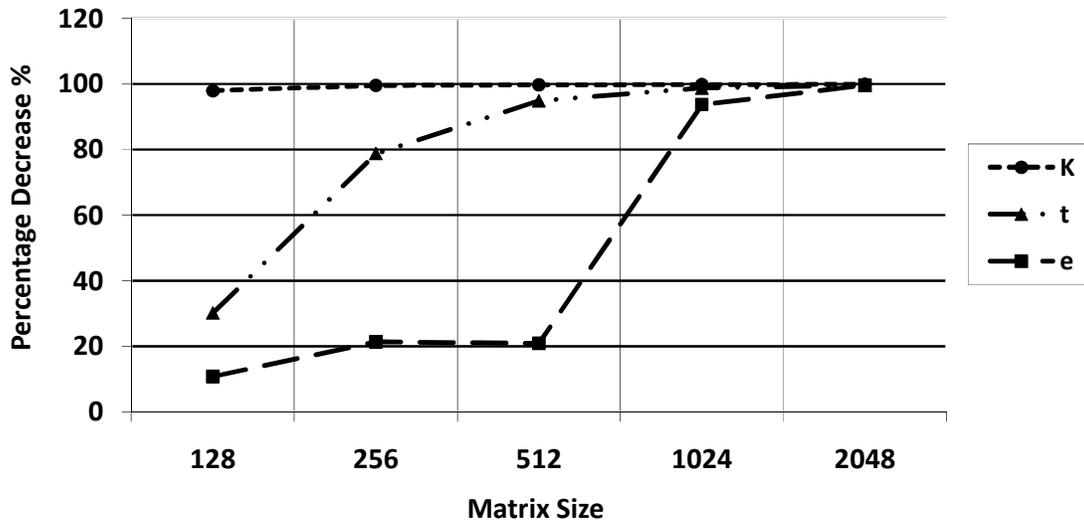


Fig.2. Percentage decrease for Example 2

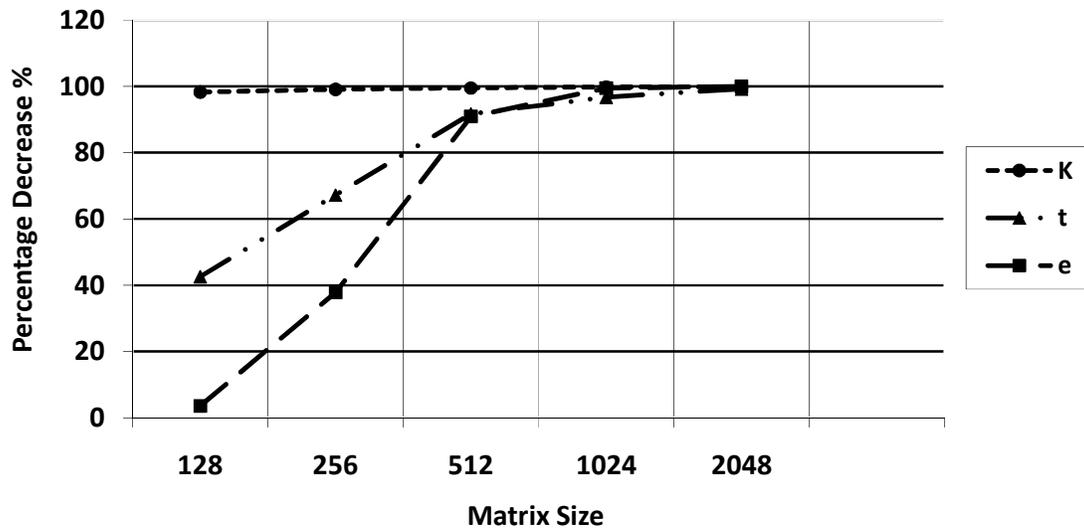


Fig. 3. Percentage decrease for Example 3

By observing these graphs of percentage decrease from Fig.1, Fig.2 and Fig.3, the performances of SOR method can be seen to have a notable improvement in terms of iterations number followed by the execution time and maximum absolute error. As the matrix size increasing, the performances of SOR method are improving over GS method for every increment of mesh sizes. This indicates that the non-polynomial spline scheme together with the SOR method can cope with the accumulated round-off error better than GS method. Thus,

it can be stated that SOR method gives better performance especially in term of iterations number.

For future work, it is highly recommended to extend this study for solving high order two-point BVPs with different degree of spline such as quartic, quintic and so on by referring to [1-5], [14-15, 22].

## 6. CONCLUSION

The cubic non-polynomial spline approximation equations were successfully derived from the single function of cubic non-polynomial spline general function. The performances in term of iterations number, execution time and maximum absolute error were analyzed by solving several examples of fourth-order two-point BVPs together with its exact solution by using SOR and GS iterative methods. Then, it was found that the SOR iterative method is superior compared to GS iterative method for different mesh sizes (128, 256, 512, 1024, 2048) and can be seen from the improvement through the percentage decrease of its respective iterations number, execution time and maximum absolute error. Hence, it can be concluded that the fourth-order two-point BVPs together with the cubic non-polynomial spline scheme are best solved by using SOR iterative method compared to GS iterative method.

## 7. REFERENCES

- [1] Noor M A, Tirmizi I A, Khan M A. Quadratic non-polynomial spline approach to the solution of a system of second-order boundary-value problems. *Applied Mathematics and Computation*, 2006, 179(1):153-160
- [2] Al-Said E A. The use of cubic splines in the numerical solution of a system of second-order boundary value problems. *Computers and Mathematics with Applications*, 2001, 42(6-7):861-869
- [3] Tirmizi I A, Haq F. Quartic non-polynomial splines approach to the solution of a system of second-order boundary-value problems. *International Journal of High Performance Computing Applications*, 2007, 21(1):42-49
- [4] Al-Said E A. Cubic spline method for solving two-point boundary-value problems. *Korean Journal of Computational and Applied Mathematics*, 1998, 5(3):669-680

- 
- [5] Ramadan MA, Lashien IF, Zahra WK. Quinticnonpolynomial spline solutions for fourth order two-point boundary value problem. *Communications in Nonlinear Science and Numerical Simulation*, 2007, 14(4):1105-1114
- [6] Na T. Y. *Computational methods in engineering boundary value problems*. New York: Academic Press, 1979
- [7] Tatari M, Dehghan M. The use of the Adomian decomposition method for solving multipoint boundary value problems. *PhysicaScripta*, 2006, 73(6):672-676
- [8] Fang Q, Tsuchiya T, Yamamoto T. Finite difference, finite element and finite volume methods applied to two-point boundary value problems. *Journal of Computational and Applied Mathematics*, 2002, 139(1):9-19
- [9] Ha SN. A nonlinear shooting method for two-point boundary value problems. *Computers and Mathematics with Applications*, 2001, 42(10):1411-1420
- [10] Chen B, Tong L, Gu Y. Precise time integration for linear two-point boundary value problems. *Applied Mathematics and Computation*, 2006, 175(1):182-211
- [11] Aziz A. K. *Numerical solutions of boundary value problems for ordinary differential equations*. New York: Academic Press, 2014
- [12] Young D. M. *Iterative solution of large linear systems*. Amsterdam: Elsevier, 2014
- [13] Sunarto A, Sulaiman J, Saudi A. Full-sweep SOR method for solving space fractional diffusion equations. *Australian Journal of Basic and Applied Science*, 2014, 8:153-8
- [14] Sunarto A, Sulaiman J, Saudi A. SOR method for implicit finite difference solution of time-fractional diffusion equation. In *11th Seminar on Science and Technology*, 2013, pp. 132-139
- [15] Justine H, Sulaiman J. Cubic non-polynomial solution for solving two-point boundary value problems using SOR iterative method. *Transactions on Science and Technology*, 2016, 3(3):469-475
- [16] Justine H, Sulaiman J. Cubic non-polynomial spline solution for solving two-point boundary value problems using half-sweep successive over-relaxation method. In *Innovation and Analytics Conference and Exhibition*, 2016, pp. 1-6
- [17] Ozisik M. N. *Boundary value problems of heat conduction*. New York: Dover Publications Inc., 2013

- 
- [18] Bisshopp KE, Drucker DC. Large deflection of cantilever beams. *Quarterly of Applied Mathematics*, 1945, 3(3):272-275
- [19] Goffe WL. A user's guide to the numerical solution of two-point boundary value problems arising in continuous time dynamic economic models. *Computational Economics*, 1993, 6(3):249-255
- [20] Weibel ES. Confinement of a plasma column by radiation pressure. *The Plasma in a Magnetic Field*, 1958, pp. 60-76
- [21] Keller HH, Holdredge ES. Radiation heat transfer for annular fins of trapezoidal profile. *Journal of Heat Transfer*, 1970, 92(1):113-116
- [22] Taiwo OA, Ogunlaran OM. A non-polynomial spline method for solving linear fourth-order boundary-value problems. *International Journal of Physical Sciences*, 2011, 6(13):3246-3254

**How to cite this article:**

Justine H, Sulaiman J. Solution of fourth-order two-point BVPs with cubic non-polynomial spline and sor iterative method. *J. Fundam. Appl. Sci.*, 2017, 9(5S), 579-593.