Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 13, Number 12 (2017), pp. 8209-8223

http://www.ripublication.com

Comparative Analysis Between Cubic and Quartic Non-Polynomial Spline Solutions for fourth-order two-Point BVPS

H. Justine and J. Sulaiman

Department of Mathematics with Economics,
Faculty of Science and Natural Resources,
Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia.
E-mail: cherryjust20@gmail.com and jumat@ums.edu.my

Abstract

Fourth-order two-point boundary value problems (BVPs) were solved based on a discretization of cubic and quartic degree of non-polynomial spline general functions. The discretization process began with simple algebraic substitution in order to get the value of all constants and further simplified by replacing certain variables with finite difference method. At the end of this process, the general approximation equations for both degrees of splines were finally developed. To solve the problems iteratively, the single equation of fourth-order problems were first reduced to two separated equations of second-order problems by means of differentiation. Then, the general approximation equations obtained earlier were imposed into these two equations to form two systems of linear equations. After these corresponding linear systems were constructed, Quarter-Sweep Successive-Over Relaxation (QSSOR) method was implemented to solve the two linear systems at varied matrix sizes. In order to assess the performances of this proposed idea, Full-Sweep Successive-Over Relaxation (FSSOR) and Half-Sweep Successive-Over Relaxation (HSSOR) were also conducted. Finally, the performances were presented evidently in terms of iterations number, execution time and maximum absolute error. Based on the performances obtained, the quartic degree of non-polynomial spline was found to be superior compared to the cubic degree when solving the fourth-order two-point BVPs.

Keywords: cubic non-polynomial spline; quartic non-polynomial spline; fourth-order two-point boundary value problems; half-sweep successive-over relaxation.

INTRODUCTION

Numerical solution of two-point boundary BVPs can be found widely in the field of engineering, sciences, and economics. Some of the applications of these problems in the field of engineering and sciences are including modeling of heat transfer for fuel elements in nuclear reactor and

modeling of chemical in rocket thrust chamber liners [1], modeling of cantilever beams deflection under concentrated load [2-3], solving Troesch's problem of radiation pressure to confine a plasma column [4-5], distribution of temperature for radiation fin of trapezoidal profile [6], and dealing with obstacle, unilateral and contact problems [7]. Whereas, some of the related applications of solving these problems in the field of economics are modeling of several theories such as capital theory, growth theory, investment theory, labor economics, and resource economics [8].

Generally, there are three standard approaches can be used to solve these two-point BVPs which are finite differences, shooting, and projections [9]. The finite differences approach composes of central finite difference, backward finite difference, and forward finite difference. This approach usually used to replace the derivatives of the ordinary differential equation so that a linear or nonlinear case of an algebraic problem can be formed and then solved to get the two-point BVPs approximation solution in a discrete form. Shooting approach, on the other hand, is applied to reduce the original BVPs to an appropriate initial boundary problem (IVP). Then, initial value techniques are conducted to solve this IVP that in the end its solution actually converges iteratively to the exact solution of the original BVPs. As for projections approach, a piecewise polynomials function which is in a simpler form is used to approximate the solution of the BVPs and at the same time satisfy both differential equations and boundary conditions.

In this paper, only finite differences approach was applied together with the cubic non-polynomial spline discretization scheme by referring to [7] and [9-14]. There are many other numerical methods were initiated to get the general approximation equation of the two-point BVPs such as collocation method [2], Adomian decomposition method [15], non-linear shooting method [16], precise time integration method [17], polynomial spline [18], and finite element and finite volume method [19]. However, by considering [20-21], only non-polynomial spline was given focus in this study together with the finite differences approach.

Based on the previous study, there are various iterative methods were initiated to solve for a linear system and these can be found in [22-27]. In addition, there are also several families of iterative methods with a different concept called as the concept of block iteration which was introduced by [28] and was further explored by [29-30]. This present study eventually aims to solve the fourth-order two-point BVPs with non-polynomial spline at cubic and quartic degree by using FSSOR, HSSOR, and QSSOR iterative method so that a comparison in terms of their performances can be determined.

The general equation of the fourth-order two-point BVPs is in the following form

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

subject to the boundary conditions (2)

$$y(a) = \alpha_1, \quad y(b) = \beta_1, \quad y''(a) = \alpha_2, \quad y''(b) = \beta_2 \quad (2)$$

where $a_1(x)$, $a_2(x)$, $a_3(x)$, $a_4(x)$ and $f(x)$ are known functions for the domain $[a,b]$, whereas α_i and β_i are constants with $i = 1,2$. In order to solve the problems, Eq. (1) was reduced to second-order two-point BVPs which referred as a process lowering of order. This process eventually reduced the single equation of fourth-order two-point BVPs to two separate equations of second-order two-point BVPs as shown by Eq. (3) and (5) by using simple mathematical differentiation followed by substitution.

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

where

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

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

and

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

where

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

After that, the general approximation equations obtained from the discretization of cubic and quartic degree of non-polynomial spline functions were used to generate two systems of linear equations based on Eq. (3) and (5). To make the non-polynomial spline discretization scheme simpler, the solution domain $[a,b]$ was divided into a uniform separation of subintervals, m as shown in Fig. 1 below

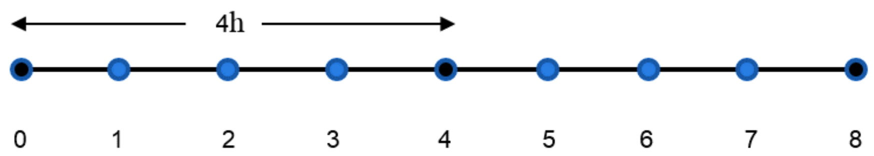

Figure 1: Separation of uniform subintervals, $m = 2$

The length, Δx of each subinterval was calculated by using following equation

$$\Delta x = \frac{b-a}{m} = 4h. \quad (7)$$

By referring to [12], the generated two systems of linear equations from Eq. (3) and (5) were solved iteratively by using FSSOR, HSSOR, and QSSOR methods and the performances of these approaches were analyzed in terms of iterations num ber, execution time and maximum absolute error at different size of matrix (128, 256, 512, 1024 and 2048).

Method of Solution

The general function of non-polynomial spline is given by Eq. (8) as follows

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

whereas, for cubic and quartic degree of non-polynomial spline functions are given by Eq. (9) and Eq. (10), respectively in the following form

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

$$Q_i(x) = a_i \cos k(x - x_i) + b_i \sin k(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i) + e_i \quad (10)$$

where a_i, b_i, c_i, d_i and e_i are constants with $i = 0, 1, \cdots, n$ and k is the frequency for the trigonometric function in Eq. (9) and Eq. (10) as $k \to 0$. The spline discretization scheme was started by defining Eq. (9) into several variables as in (11) and Eq. (10) as in (12) by mean of differentiation.

$$Q_i(x_i) = y_i, \; Q_i(x_{i+4}) = y_{i+4}, \quad Q_i''(x_i) = S_i, \; Q_i''(x_{i+4}) = S_{i+4} \quad (11)$$

$$Q_i(x_i) = y_i, \quad Q_i(x_{i+4}) = y_{i+4}, \quad Q_i''(x_i) = S_i, \quad Q_i''(x_{i+4}) = S_{i+4},$$
$$Q_i^{(iv)}(x_i) = \frac{1}{2}(F_i + F_{i+4}) \quad (12)$$

Then, all the values of constant a_i, b_i, c_i, d_i and e_i were calculated by considering the value of $y(x)$ as an exact solution, so that the non-polynomial spline equations, S_i for $y_i = y(x_i)$ which passing through the points (x_i, S_i) and (x_{i+4}, S_{i+4}) can be obtained from all the segments of $Q_i(x)$ as shown in Figure 2.

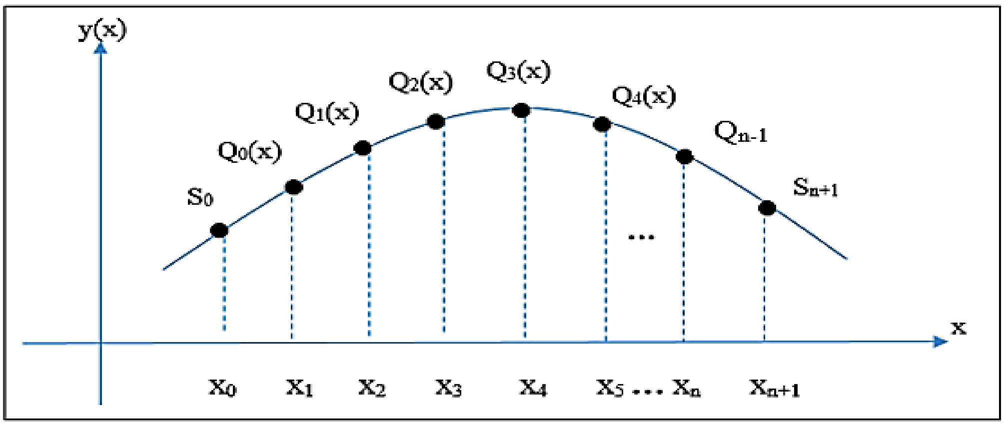

Figure 2: llustration of non-polynomial spline equations for $m=8$

After several algebraic manipulations, all the constants for the cubic degree of non-polynomial spline were finally expressed in terms of y_i, y_{i+4}, S_i and S_{i+4} as follows

$$a_i = \frac{-h^2}{\theta^2} S_i , \quad b_i = \frac{h^2 \cos 4\theta}{\theta^2 \sin 4\theta} S_i - \frac{h^2}{\theta^2 \sin 4\theta} S_{i+4} , \qquad 12)$$

$$c_i = \frac{1}{4h} y_{i+4} + \frac{h}{4\theta^2} S_{i+4} - \frac{1}{4h} y_i - \frac{h}{4\theta^2} S_i , \quad d_i = y_i + \frac{h^2}{\theta^2} S_i ,$$

whereas for quartic degree, all the constants were obtained in terms of $F_{i,} F_{i+4,} S_{i,} S_{i+4,}$ y_i, and y_{i+4} as given by (13)

$$a_i = \frac{h^4}{2\theta^4} F_i + \frac{h^4}{2\theta^4} F_{i+4} - \frac{h^2}{2\theta^2} S_i + \frac{h^2}{2\theta^2} S_{i+4} , \qquad (13)$$

$$b_i = \frac{\mathrm{h}^4 - \mathrm{h}^4 \cos 4\theta}{2\theta^4 \sin 4\theta} F_i + \frac{\mathrm{h}^4 - \mathrm{h}^4 \cos 4\theta}{2\theta^4 \sin 4\theta} F_{i+4} + \frac{h^2 + h^2 \cos 4\theta}{2\theta^2 \sin 4\theta} S_i - \frac{h^2 - h^2 \cos 4\theta}{2\theta^2 \sin 4\theta} S_{i+4} ,$$

$$c_i = \frac{h^2}{4h^2} F_i + \frac{h^2}{4h^2} F_{i+4} + \frac{1}{4} S_i + \frac{1}{4} S_{i+4} ,$$

$$d_i = \frac{1}{4h} y_{i+4} - \frac{1}{4h} y_i - \frac{h^3}{4\theta^2} F_i - \frac{h^3}{4\theta^2} F_{i+4} - \frac{2 - 8h^2 k^2}{8hk^2} S_i + \frac{2 - 8h^2 k^2}{8hk^2} S_{i+4} ,$$

$$e_i = y_i - \frac{h^4}{2\theta^4} F_i - \frac{h^4}{2\theta^4} F_{i+4} + \frac{h^2}{2\theta^2} S_i - \frac{h^2}{2\theta^2} S_{i+4} ,$$

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

In order to form the general approximation equation, all these constants were used to satisfy the continuity condition $Q_{i-1}^{m}(x)=Q_{i}^{m}(x)$ where $m=0,1$. By applying this continuity condition at first derivative, the general approximation equation for cubic degree was obtained as below

$$-\left(\frac{1}{4h}\right)y_{i-4}+\left(\frac{1}{2h}\right)y_{i}-\left(\frac{1}{4h}\right)y_{i+4}+\alpha hS_{i-4}+2\beta hS_{i}+\alpha hS_{i+4}=0$$

(14)

where $\alpha=\left(\frac{1}{\theta\sin 4\theta}-\frac{1}{4\theta^{2}}\right)$, $\beta=\left(\frac{1}{4\theta^{2}}-\frac{4\theta\cos 4\theta}{\theta\sin 4\theta}\right)$ and $i=0,1,2,..,n.$ As for the quartic degree, the continuity conditions were satisfied at its first and third derivatives which yield to the following general approximation equation

$$\lambda y_{\text{i-4}}-\gamma y_{\text{i}}+\lambda y_{\text{i+4}}+\alpha S_{i\text{-}4}+\beta S_{i}+\alpha S_{i+4}=0 \qquad (15)$$

Then, Eq. (14) and Eq. (15) were further simplified by substituting the finite differences method composing of central, backward and forward finite differences into the respective variables of S, given that all the finite differences are as follows

$S_{i-4}=-f_{i-4}y'_{i}-q_{i-4}y_{i-4}+g_{i-4}$, $S_{i}=-f_{i}y'_{i}-q_{i}y_{i}+g_{i}$,
$S_{i+4}=-f_{i+4}y'_{i+4}-q_{i+4}y_{i+4}+g_{i+4}$ (16)

where, $y'_{i-4}=\frac{-y_{i+4}+4y_{i}-3y_{i-4}}{8h}$, $y'_{i}=\frac{y_{i+4}-y_{i-4}}{8h}$ and $y'_{i+4}=\frac{3y_{i+4}-4y_{i}+y_{i-4}}{8h}$.

Finally, the general approximation equation for cubic degree was obtained in the following form

$$\sigma_{i}y_{i-4}+\delta_{i}y_{i}+\lambda_{i}y_{i+4}=F_{i},\quad i=0,1,2,...,n \qquad (17)$$

where,

$\sigma_{i}=\frac{-\mu_{0}}{4}+\gamma\left(\frac{3p_{i-4}}{8h}-q_{i-4}\right)+\varphi\frac{p_{i}}{8h}-\gamma\frac{p_{i+4}}{8h}$, $\delta_{i}=\frac{\mu_{0}}{2}-\gamma\frac{p_{i-4}}{2h}-\varphi q_{i}+\gamma\frac{p_{i+4}}{2h}$,

$\lambda_{i}=\frac{-\mu_{0}}{4}+\gamma\frac{p_{i-4}}{8h}-\varphi\frac{p_{i}}{8h}+\gamma\left(\frac{3p_{i+4}}{8h}-q_{i+4}\right)$, $F_{i}=-\gamma f_{i-4}-\varphi f_{i}-\gamma f_{i+4}$,

$\gamma=\mu_{1}h^{2}$, $\varphi=2\mu_{2}h^{2}$, $\mu_{0}=4\theta^{2}\sin 4\theta$, $\mu_{1}=4\theta-\sin 4\theta$,
$\mu_{2}=\sin 4\theta-4\theta\cos 4\theta$.

On the other hand, the general approximation equation for quartic degree was obtained as follows

$$\sigma_{i}y_{i-4}+\delta_{i}y_{i}+\lambda_{i}y_{i+4}=F_{i},\ i=0,1,2,...,n \qquad (18)$$

where

$$\sigma_i = \phi + \alpha\left(\frac{3p_{i-4}}{8h} - q_{i-4}\right) + \beta\frac{p_i}{8h} - \alpha\frac{p_{i+4}}{8h}, \quad \delta_i = -\gamma - \alpha\frac{p_{i-4}}{2h} - \beta q_i + \alpha\frac{p_{i+4}}{2h},$$

$$\lambda_i = \phi + \alpha\frac{p_{i-4}}{8h} - \beta\frac{p_i}{8h} + \alpha\left(\frac{3p_{i+4}}{8h} - q_{i+4}\right), \quad F_i = -\alpha f_{i-4} - \beta f_i - \alpha f_{i+4}$$

$$\alpha = \frac{\mu_1}{\mu_0} - \omega, \quad \beta = \frac{\mu_2}{\mu_0} + 2\omega, \quad \phi = \frac{\mu_4}{2\mu_3}, \quad \gamma = \frac{\mu_4}{\mu_3}, \quad \omega = \frac{\theta^2(1+\cos 4\theta)}{h^2 - h^2\cos 4\theta},$$

$$\mu_0 = -2h^2 + 2h^2\cos 4\theta + 4\theta h^2\sin 4\theta, \quad \mu_1 = -2\theta^2 - 2\theta^2\cos 4\theta + \theta\sin 4\theta - 4\theta^3\sin 4\theta,$$

$$\mu_2 = 4\theta^2 + 4\theta^2\cos 4\theta - 2\theta\sin 4\theta - 8\theta^3\sin 4\theta, \quad \mu_3 = -h^4 + h^4\cos 4\theta + 2\theta h^4\sin 4\theta,$$

$$\mu_4 = 4\theta^3\sin\theta, \quad \omega = \frac{\theta^2(1+\cos 4\theta)}{h^2 - h^2\cos 4\theta}.$$

By using Eq. (17) and (18), the single fourth-order two-point BVPs equation that was reduced earlier to two separated equations of second-order problems, were developed as a system of linear equations in a matrix form

$$A\underset{\sim}{y} = \underset{\sim}{F} \qquad (19)$$

where,

$$A = \begin{bmatrix} \delta_1 & \lambda_1 & & & & \\ \sigma_2 & \delta_2 & \lambda_2 & & & \\ & \sigma_3 & \delta_3 & \lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \sigma_{n-1} & \delta_{n-1} & \lambda_{n-1} \\ & & & & \sigma_n & \delta_n \end{bmatrix}_{(n\times n)}$$

$$\underset{\sim}{y} = [y_1 \quad y_2 \quad y_3 \cdots \quad y_{n-1} \quad y_n]^T$$

.

Then, the matrix was used to solve for three example of fourth-order two-point BVPs by using FSSOR, HSSOR and QSSOR iterative methods at varied matrix sizes.

Algorithm

By referring to [22], an algorithm for the QSSOR iterative method was tabulated in Table 1 as below

TABLE 1: Algorithm of QSSOR iterative method

i.	Initialize $U_i^{(0)} \leftarrow 0, \varepsilon \leftarrow 10^{-10}$
ii.	Assign the optimal value of ω
iii.	Calculate $U_i^{(k+1)}$ using $\underset{\sim}{S}^{(k+1)} = (1-\omega)\underset{\sim}{S}^{(k)} + \omega(D+L)^{-1}\left(-U\underset{\sim}{S}^{(k)} + \underset{\sim}{F}\right)$ for $i = 4,8,12,...,m-4$.
iv.	Check the convergence test, $\left\|U_i^{(k+1)} - U_i^{(k)}\right\| \leq \varepsilon = 10^{-10}$. If yes, go to step (v). Otherwise go back to step (iii).
v.	Display the approximate solutions.

The method can be seen as a decomposition of the coefficient matrix, A from Eq. (19) as follows

$$A = D + L + U \tag{20}$$

where *D*, *L*, and *U* are diagonal, lower triangular and upper triangular matrices respectively. By imposing Eq. (20) into equation (19), the general equation of the SOR iterative method can be written as

$$\underset{\sim}{y}^{(k+1)} = (1\text{-}\omega)\underset{\sim}{y}^{(k)} + \omega(D+L)^{-1}\left((\text{-}U\underset{\sim}{y}^{(k)} + \underset{\sim}{F})\right). \tag{21}$$

However, the optimum value of the parameter ω has to be determined first before solving the linear system. Several computer programs were conducted to get the best approximate value of ω and the value of the parameter ranges from $1 \leq \omega < 2$ in practice.

Numerical Performance Analysis

There are three problems were solved by using the proposed approach, which is to solve the second-order two-point BVPs together with the cubic and quartic degree of non-polynomial spline by using QSSOR iterative method.

Problem 1 [31]

$y'' - 4y = 4\cosh(1)$, $x \in [0,1]$

given that the exact solution for the problem is

$y(x) = \cosh(2x-1) - \cosh(1)$.

Problem 2 [31]

$-\dfrac{d^2y}{dx^2} = 9\sin(3x)$, $x \in [0,1]$

with its exact solution given by

$y(x) = \sin(3x)$.

Problem 3 [32]

$\varepsilon y'' = y + \cos^2(\pi x) + 2\varepsilon\pi^2 \cos(2\pi x)$, $x \in [0,1]$

with its exact solution

$y(x) = \exp(-(1-x)/\sqrt{\varepsilon}) + \exp(-x/\sqrt{\varepsilon})/[1+\exp(-1/\sqrt{\varepsilon})] - \cos^2(\pi x)$.

RESULTS AND DISCUSSION

The performance results in terms of iterations number (K), execution time (t) in seconds and maximum absolute error (ε) were successfully tabulated in Table 2, 3, 4 when solving all the problems with the cubic non-polynomial spline and, Table 5, 6, 7 for the quartic non-polynomial spline. By observing the tables for every matrix size, the increment of the size has caused the iterations number to increase and this happened due to the accumulated round-off error which occurred at every iteration. Nevertheless, when analysing the performances between these two spline schemes at the cubic and quartic degree of non-polynomial, the fourth-order two-point BVPs were found to give the best performances when solved with QSSOR iterative method which evidently shown through the value of iterations number.

TABLE 2: Comparison of iterations number (K), execution time (t), and maximum absolute error (ε) for Cubic Non-Polynomial Scheme (Problem 1)

Method		**Size of Matrix**				
		128	**256**	**512**	**1024**	**2048**
FSSOR	K	546	1024	1948	3963	7682
	t	0.37	1.19	1.25	3.10	6.87
	ε	1.1471e-05 ω =1.9574	2.8723e-06 ω =1.9772	7.2508e-07 ω =1.9875	1.8881e-07 ω =1.9943	6.3598e-08 ω =1.9969
HSSOR	K	266	510	984	1948	3963
	t	0.19	0.23	0.28	1.02	1.9943
	ε	4.2267e-04 ω =1.9111	1.0811e-04 ω =1.9523	2.7338e-05 ω =1.9751	6.8736e-06 ω =1.9875	1.7233e-06 ω =1.9943
QSSOR	K	139	265	510	1024	1990
	t	0.45	1.00	0.33	0.55	0.96
	ε	1.6149e-03 ω =1.8167	4.2267e-04 ω =1.9112	1.0811e-04 ω =1.9523	2.7338e-05 ω =1.9772	6.8736e-06 ω =1.9879

TABLE 3. Comparison of iterations number (K), execution time (t), and maximum absolute

error (ε) for Cubic Non-Polynomial Scheme (Problem 2)

Method		Size of Matrix				
		128	**256**	**512**	**1024**	**2048**
FSSOR	K	513	1025	2049	4097	8193
	t	0.38	0.55	1.83	2.19	6.23
	ε	1.9543e-06 ω =1.9497	4.9024e-07 ω =1.9744	1.2433e-07 ω =1.9871	3.4412e-08 ω =1.9935	2.3722e-08 ω =1.9967
HSSOR	K	257	513	1025	2049	4097
	t	0.19	1.38	1.27	1.15	2.23
	ε	1.0832e-04 ω =1.9036	2.7364e-05 ω =1.9497	6.8768e-06 ω =1.9744	1.7237e-06 ω =1.9871	4.3149e-07 ω =1.9935
QSSOR	K	128	257	513	1025	2049
	t	0.26	0.67	0.97	1.25	1.91
	ε	4.2439e-04 ω =1.8144	1.0832e-04 ω =1.9101	2.7364e-05 ω =1.9497	6.8768e-06 ω =1.9744	1.7237e-06 ω =1.9871

TABLE 4. Comparison of iterations number (K), execution time (t), and maximum absolute

error (ε) for Cubic Non-Polynomial Scheme (Problem 3)

Method		Size of Matrix				
		128	**256**	**512**	**1024**	**2048**
FSSOR	K	512	996	1951	3763	7406
	t	0.84	1.34	1.56	3.13	5.98
	ε	4.8161e-06 ω =1.9551	1.2057e-07 ω =1.9772	2.9898e-07 ω =1.9886	6.9839e-08 ω =1.9943	1.6101e-08 ω =1.9972
HSSOR	K	258	512	1068	1951	3763
	t	0.20	0.19	1.33	1.57	1.98
	ε	1.3429e-04 ω =1.9114	4.2024e-05 ω =1.9561	1.0508e-05 ω =1.9802	2.6257e-06 ω =1.9886	6.5548e-07 ω =1.9943
QSSOR	K	139	256	512	996	1951
	t	0.67	0.94	1.15	1.45	2.07
	ε	6.7202e-04 ω =1.8409	1.6807e-04 ω =1.9772	4.2023e-05 ω =1.9551	1.0506e-05 ω =1.9772	2.6257e-06 ω =1.9886

TABLE 5. Comparison of iterations number (K), execution time (t), and maximum absolute

error (ε) for Quartic Non-Polynomial Scheme (Problem 1)

Method		**Size of Matrix**				
		128	**256**	**512**	**1024**	**2048**
FSSOR	K	510	1018	1950	3812	7369
	t	0.22	0.29	0.53	1.24	3.75
	ε	1.7097e-09 ω =1.9523	5.7282e-09 ω =1.9761	1.3114e-08 ω =1.9875	7.1954e-09 ω =1.9937	3.0627e-08 ω =1.9970
HSSOR	K	265	510	984	1949	4156
	t	0.12	0.13	0.21	0.51	1.32
	ε	3.1833e-04 ω =1.9113	8.1253e-05 ω =1.9525	2.0525e-05 ω =1.9751	5.1579e-06 ω =1.9875	1.2928e-06 ω =1.9948
QSSOR	K	135	265	510	984	2107
	t	0.80	1.39	1.03	2.66	3.60
	ε	1.2214e-03 ω =1.8211	3.1833e-04 ω =1.9112	8.1253e-05 ω =1.9523	2.0525e-05 ω =1.9751	5.1579e-06 ω =1.9894

TABLE 6. Comparison of iterations number (K), execution time (t), and maximum absolute

error (ε) for Quartic Non-Polynomial Scheme (Problem 2)

Method		**Size of Matrix**				
		128	**256**	**512**	**1024**	**2048**
FSSOR	K	513	1025	2049	4097	8193
	t	0.23	0.31	0.55	1.38	4.18
	ε	1.7039e-09 ω =1.9497	3.2312e-09 ω =1.9744	3.7206e-09 ω =1.9871	5.8889e-09 ω =1.9935	1.7727e-08 ω =1.9967
HSSOR	K	257	513	1025	2049	4097
	t	0.11	0.15	0.25	0.52	1.40
	ε	8.1259e-05 ω =1.9023	2.0559e-05 ω =1.9497	5.1622e-06 ω =1.9744	1.2934e-06 ω =1.9870	3.2369e-07 ω =1.9935
QSSOR	K	128	257	513	1025	2049
	t	0.48	0.85	2.33	2.25	4.12
	ε	3.2055e-04 ω =1.8144	8.1529e-05 ω =1.9101	2.0559e-05 ω =1.9497	5.1622e-06 ω =1.9744	1.2934e-06 ω =1.9871

TABLE 7. Comparison of iterations number (K), execution time (t), and maximum absolute

error (ε) for Quartic Non-Polynomial Scheme (Problem 3)

Method		Size of Matrix				
		128	**256**	**512**	**1024**	**2048**
FSSOR	K	512	996	1951	3764	7406
	t	0.18	0.26	0.48	1.02	3.02
	ε	1.5096e-09 ω =1.9549	2.8455e-09 ω =1.9772	3.1125e-09 ω =1.9886	7.3787e-09 ω =1.9943	7.9896e-09 ω =1.9972
HSSOR	K	256	512	996	1951	3763
	t	1.21	1.44	3.05	1.00	1.94
	ε	1.2601e-04 ω =1.9143	3.1507e-05 ω =1.9581	7.8769e-06 ω =1.9772	1.9684e-06 ω =1.9886	4.9116e-07 ω =1.9943
QSSOR	K	129	256	512	996	1951
	t	0.37	1.39	1.22	2.58	3.78
	ε	5.0391e-04 ω =1.8363	1.2601e-04 ω =1.9121	3.1507e-05 ω =1.9551	7.8769e-06 ω =1.9772	1.9684e-06 ω =1.9886

CONCLUSION

The general function of cubic and quartic non-polynomial spline schemes were successfully discretized to get the approximation equation for solving the fourth-order two-point BVPs. Then, a numerical experiment was conducted in order to get the performance of these schemes in respect of its iterations number, execution time and maximum absolute error when solved by using FSSOR, HSSOR and QSSOR iterative methods. According to the numerical analysis results, it can be pointed out that the implementation of QSSOR iterative method together with the cubic and quartic non-polynomial spline schemes are superior compared to FSSOR and HSSOR iterative methods at different grid sizes (128, 256, 512, 1024, 2048).

ACKNOWLEDGEMENT

The authors are grateful for the fund received from Universiti Malaysia Sabah upon publication of this paper.

REFERENCE

[1] Ozisik, M. N., 1989, Boundary value problems of heat conduction, Courier Corporation.

[2] Na, T. Y., 1979, Computational Methods in Engineering Boundary Value Problems Academic, New York.

[3] Bisshopp, K. E., and D. C. Drucker., 1945, "Large deflection of cantilever beams," Quarterly App. Math., 3(3), pp. 272-275.

[4] Robert, S., and Shipman, J., 1972, "Solution of Troesch's two-point boundary value problems by shooting techniques," J. Comput. Phy., 10, pp. 232-41.

[5] Weibel, E. S., 1958, "Confinement of a plasma column by radiation pressure. The plasma in a magnetic field" 1958; 60-76.

[6] Keller, H. H., and Holdredge, E. S., 1970, "Radiation heat transfer for annular fins of trapezoidal profilem," J. Heat Trans., 92(1), pp. 113-116.

[7] Noor, M. A., Tirmizi, I. A., and Khan, M. A., 2006, "Quadratic non-polynomial spline approach to the solution of a system of second-order boundary-value problems," Appl. Math. Comput., 179(1), pp. 153-160.

[8] Goffe, W. L., 1993, "A user's guide to the numerical solution of two-point boundary value problems arising in continuous time dynamic economic models," Comput. Eco., 6(3-4), pp. 249-55.

[9] Ascher, U. M., Mattheij, R. M., and Russell, R. D. "Numerical solution of boundary value problems for ordinary differential equations," SIAM.

[10] Al-Said, E. A., 2001, "The use of cubic splines in the numerical solution of a system of second-order boundary value problems," Comput. Math. Appl., 42(6), pp. 861-869.

[11] Tirmizi, I. A., and Haq, F., 2007, "Quartic non-polynomial splines approach to the solution of a system of second-order boundary-value problems," Int. J. High Perform. Comput. Appl., 21(1), pp. 42-49.

[12] Justine, H. and Sulaiman, J., 2016, "Cubic non-polynomial solution for solving two-point boundary value problems using SOR iterative method," Trans. Sci. Tech., 3(3), pp. 469-475.

[13] Justine, H. and Sulaiman, J., 2016, "Application of conjugate gradient method with cubic non-polynomial spline scheme for two-point boundary value problems," Global J. Eng. Tech. Review, 1(1), pp. 85-92.

[14] Justine, H. and Sulaiman, J., 2016, "Cubic non-polynomial spline solution for solving two-point boundary value problems using half-sweep successive over-relaxation method," Innovation and Analytics Conference and Exhibition

(IACE), Sintok, Kedah, Malaysia: 31 Oct– 1 Nov, pp. 1-6.

[15] Tatari, M., and Dehghan, M., 2006, "The use of the adomian decomposition method for solving multipoint boundary value problems," Physica Scripta, 73(6), pp. 672.

[16] Ha, S. N., 2001, "A nonlinear shooting method for two-point boundary value problems," Comput. Math. Appl., 42(10), pp. 1411-1420.

[17] Chen, B., Tong, L., and Gu, Y., 2006., "Precise time integration for linear two-point boundary value problems,". Appl. Math. Comput., 175(1), pp. 182-211.

[18] Li, B. N., Huan, W. L., and Yanping C. 2011, "Polynomial spline approach for solving second-order boundary-value problems with Neumann conditions," Appl. Math. Comput., 217, pp. 6872-6882.

[19] Fang, Q., Tsuchiya, T., and Yamamoto, T., 2002, "Finite difference, finite element and finite volume methods applied to two-point boundary value problems," J. Comput. Appl. Math., 139(1), pp. 9-19.

[20] Srivastava, P. K. 2014. "Study of differential equations with their polynomial and nonpolynomial spline based approximation," Acta Technica Corviniensis-Bulletin of Eng. 7(3), pp. 139.

[21] Ramadan, M. A., Lashien, I. F., and Zahra, W. K. 2007, "Polynomial and nonpolynomial spline approaches to the numerical solution of second order boundary value problems," Appl. Math. Comput., 184(2), pp. 476-484.

[22] Young, D. M., 2014, Iterative solution of large linear systems, Elsevier.

[23] Hackbusch, W., 2012, Iterative solution of large sparse systems of equations (Vol. 95), Springer Science and Business Media.

[24] Saad, Y., 1996, Iterative methods for sparse linear systems, PWS, Boston.

[25] Hestenes, M. R., and Stiefel, E., 1952, Methods of conjugate gradients for solving linear systems (Vol. 49), NBS.

[26] Kelley, C. T., 1995, Iterative Methods for Linear and Nonlinear Equations (Frontiers in Applied Mathematics Vol. 16), Philadelphia, SIAM.

[27] Burgerscentrum, J. M., 2011, "Iterative solution methods," Appl. Numerical Math., 51(4), 437-450.

[28] Evans, D. J., 1985, "Group explicit iterative methods for solving large linear systems," Int. J. Comput. Math., 17(1), pp. 81-108.

[29] Yousif, W. S., and Evans, D. J., 1995, "Explicit de-coupled group iterative methods and their parallel implementations," Parallel Algorithms Appl., *7*(1-2), pp. 53-71.

[30] Abdullah, A. R., and Ibrahim, A. 1995, "Solving the two-dimensional diffusion-convection equation by the four point explicit decoupled group iterative method," Int. J. Comput. Math., 58(1-2), pp. 61-71.

[31] Taiwo, O. A., and Ogunlaran, O. M., 2011, "A nonpolynomial spline method for solving linear fourth order boundary value problems," Int. J. Phy. Sci., 6(13), pp. 3246-3254.

[32] Aziz, T., and Khan, A. 2002, "A spline method for second-order singularly perturbed boundary-value problems". *J. Comput. Appl. Math., 147*(2), pp. 445-452.