Sains Malaysiana 40(11)(2011): 1285–1290

Extended Cubic B-spline Method for Linear Two-Point Boundary Value Problems

(Kaedah Splin-B Kubik Lanjutan untuk Masalah Nilai Sempadan Dua Titik Linear)

NUR NADIAH ABD HAMID*, AHMAD ABD. MAJID & AHMAD IZANI MD. ISMAIL

ABSTRACT

Second order linear two-point boundary value problems were solved using extended cubic B-spline interpolation method. Extended cubic B-spline is an extension of cubic B-spline consisting of one shape parameter, called λ. The resulting approximated analytical solution for the problems would be a function of λ. Optimization of λ was carried out to find the best value of λ that generates the closest fit to the differential equations in the problems. This method approximated the solutions for the problems much more accurately compared to finite difference, finite element, finite volume and cubic B-spline interpolation methods.

Keyword: Cubic B-spline; extended cubic B-spline; spline interpolation; two-point boundary value problem

ABSTRAK

Masalah nilai sempadan dua titik linear peringkat kedua diselesaikan menggunakan kaedah interpolasi Splin-B kubik lanjutan. Splin-B kubik lanjutan ialah satu perlanjutan daripada Splin-B kubik yang mengandungi satu parameter bentuk, iaitu λ. Penyelesaian analitikal anggaran yang terhasil kepada masalah tersebut merupakan fungsi λ. Pengoptimuman λ dijalankan untuk mencari nilai λ yang terbaik yang menghasilkan penyesuaian terdekat kepada persamaan pembezaan dalam masalah tersebut. Kaedah ini menganggarkan penyelesaian untuk masalah tersebut dengan lebih tepat berbanding dengan kaedah-kaedah beza terhingga, unsur terhingga, isipadu terhingga dan interpolasi Splin-B kubik.

Kata kunci: Interpolasi splin; masalah nilai sempadan dua titik; Splin-B kubik; Splin-B kubik lanjutan

INTRODUCTION

Boundary value problems are abundant in the field of physics, chemistry and engineering. Generally, these problems are difficult to solve analytically. Hence, numerous methods had been developed over the years to approximate the solutions for the problems. Some of the already established methods are shooting, finite difference and Rayleigh-Ritz while some of the more recent methods are variational iteration, extended Adomian decomposition and homotopy perturbation (Burden & Faires 2005; Chun & Sakthivel 2010; Jang 2008; Lu 2007). As of now, homotopy perturbation method was claimed to produce the most accurate results out of all (Chun & Sakthivel 2010).

This paper considers the simplest form of boundary value problems, which is second order linear two-point boundary value problems and focuses on the application of extended cubic B-spline interpolation in approximating the solutions. The general form of these problems is where the accompanying continuity and negativity conditions are necessary for the existence and uniqueness of the solutions (Burden & Faires 2005):

$$u''(x) + p(x)u'(x) + q(x)u(x) = r(x),\ x \in [a, b],$$
$$u(a) = \alpha,\ u(b) = \beta,\ p, q, r \in C^1,\ q(x) < 0. \qquad (1)$$

The use of the most basic cubic spline in solving these problems was first explored by Bickley in 1968. His work was immediately analyzed and modified in Albasiny and Hoskins (1969) and Fyfe (1969). Following these developments, many other analysis and improvements were made throughout the years as in Al-Said (1998) and Khan (2004), and the references there in. However, only in 2006, the idea of replacing cubic spline with cubic B-spline, which is another representation of cubic spline that is easier to compute, was proposed by Caglar et al. (2006). This method was called cubic B-spline interpolation method (CBIM). CBIM was tested on the more simplified version of second order linear two-point boundary value problems,

$$-(p(x)u'(x))' = r(x),\quad x \in [a, b],\ u(a) = u(b) = 0, \quad (2)$$

which was proven to have unique solutions if $p, r \in C^1$ and $p(x) > 0$. The results were found to be promising (Caglar et al. 2006). Nevertheless, CBIM is also applicable to the general problem stated in . Therefore, continuing with this work, we applied the same procedure in CBIM but using an extended version of cubic B-spline.

EXTENDED CUBIC B-SPLINE

Essentially, cubic B-spline function is a piecewise polynomial function of degree 3, constructed from a linear combination of some recursive functions, called cubic B-spline basis. The derivation of B-spline basis and the construction of B-spline function are discussed in many curves and surfaces books such as (Agoston 2005; Patrikalakis & Maekawa 2002; Prautzsch et al. 2002; Salomon 2006). As the name suggests, extended cubic B-spline is an extension of cubic B-spline. Its basis is constructed in such a way that one free parameter, λ, is included and the degree of the polynomial is increased. In Xu and Wang (2008), three extended cubic B-spline of degree 4, 5 and 6 were presented. However, for a start, extended cubic B-spline basis of degree 4 was selected to replace cubic B-spline in CBIM.

Suppose that:

$$x_1 = a + ih, \quad h = \frac{b-a}{n}, \quad n \in \mathbb{Z}^+, \quad i \in \mathbb{Z}$$

Extended cubic B-spline basis of degree 4, $E_i^4(x)$, is defined by the following equation:

$$\frac{1}{24h^4}\begin{cases} -4h(\lambda-1)(x-x_i)^3 + 3\lambda(x-x_i)^4, & x \in [x_i, x_{i+1}] \\ (4-\lambda)h^4 + 12h^3(x-x_{i+1}) & \\ \quad +6h^2(2+\lambda)(x-x_{i+1})^2 - 12h(x-x_{i+1})^3 - 3\lambda(x-x_{i+1})^4, & x \in [x_{i+1}, x_{i+2}], \\ (16+2\lambda)h^4 - 12h^2(2+\lambda)(x-x_{i+2})^2 & \\ \quad +12h(1+\lambda)(x-x_{i+2})^3 - 3\lambda(x-x_{i+2})^4, & x \in [x_{i+2}, x_{i+3}], \\ -(h+x_{i+3}-x)^3\left[h(\lambda-4) + 3\lambda(x-x_{i+3})\right], & x \in [x_{i+3}, x_{i+4}], \\ 0, & \text{elsewhere.} \end{cases}$$

$E_i^4(x)$ degenerates into cubic B-spline basis when $\lambda = 0$. Figure 1 shows a family of extended cubic B-spline bases, $E_i^4(x)$ when λ is varied.

Analogous to B-spline function, extended cubic B-spline function, $S(x)$ is a linear combination of the extended cubic B-spline basis, as in (3).

$$S(x) = \sum_{i=-3}^{n-1} C_i E_i^4(x), \quad x \in [x_0, x_n], \quad C_i \in \mathbb{R} \quad n \geq 1. \tag{3}$$

As a result, $S(x)$ is a piecewise polynomial function of degree 4. The properties and behaviors of this function are discussed further in (Xu & Wang 2008).

Evaluating (3) at x_i, it can be verified from the basis function definition thar for $i = 0, 1, \ldots, n$,

$$\begin{aligned} S(x_i) &= C_{-3}E_{i-3}^4(x_i) + C_{i-2}E_{i-2}^4(x_i) + C_{i-1}E_{i-1}^4(x_i), \\ &= C_{i-3}\left(\frac{4-\lambda}{24}\right) + C_{i-2}\left(\frac{8+\lambda}{12}\right) + C_{i-1}\left(\frac{4-\lambda}{24}\right). \end{aligned} \tag{4}$$

Similarly, the first and second derivatives of $S(x_i)$ can be simplified into expressions involving C_{i-3}, C_{i-2} and C_{i-1} only, as in (5) and (6).

$$\begin{aligned} S'(x_i) &= C_{-3}E_{i-3}^4{}'(x_i) + C_{i-2}E_{i-2}^4{}'(x_i) + C_{i-1}E_{i-1}^4{}'(x_i), \\ &= C_{i-3}\left(-\frac{1}{2h}\right) + C_{i-2}(0) + C_{i-1}\left(\frac{1}{2h}\right). \end{aligned} \tag{5}$$

$$\begin{aligned} S''(x_i) &= C_{-3}E_{i-3}^4{}''(x_i) + C_{i-2}E_{i-2}^4{}''(x_i) + C_{i-1}E_{i-1}^4{}''(x_i), \\ &= C_{i-3}\left(\frac{2+\lambda}{2h^2}\right) + C_{i-2}\left(-\frac{2+\lambda}{2h^2}\right) + C_{i-1}\left(\frac{2+\lambda}{2h^2}\right). \end{aligned} \tag{6}$$

These simplifications are very useful in solving two-point boundary value problems by extended cubic B-spline interpolation method.

EXTENDED CUBIC B-SPLINE INTERPOLATION METHOD

This section addresses the main purpose of this paper, that is, to introduce extended cubic B-spline interpolation method (ECBIM) for solving second order linear two-point

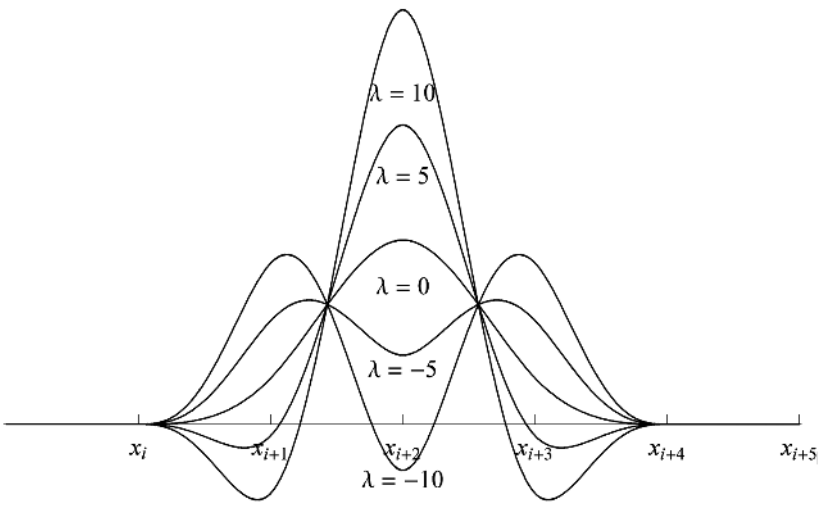

FIGURE 1. Extended cubic B-spline basis, $E_i^4(x)$, when λ = -10, -5, 0, 5, 10

boundary value problems. First of all, extended cubic B-spline function in (3) is presupposed to be the solution for the problems. Hence, becomes:

$$S''(x) + p(x)S'(x) + q(x)S(x) = r(x),$$
$$x \in [a,b],\ S(a) = \alpha,\ S(b) = \beta. \tag{7}$$

Evaluating (7) at x_i, for $i = 0,1,\ldots,n$, results:

$$S''(x) + p(x_i)\, S'(x_i) + 1(x_i)S(x_i),$$
$$x \in [a,b],\ S(a) = \alpha,\ S(b) = \beta. \tag{8}$$

$S(x_i)$, $S'(x_i)$ and $S''(x_i)$ are already simplified in previous section. From there, (4), (5) and (6) are substituted into (8) resulting in:

$$C_{i-3}\left(\frac{2+\lambda}{2h^2}\right) + C_{i-2}\left(-\frac{2+\lambda}{h^2}\right) + C_{i-1}\left(\frac{2+\lambda}{2h^2}\right) +$$
$$p(x_i)\left[C_{i-3}\left(-\frac{1}{2h}\right) + C_{i-2}(0) + C_{i-1}\left(\frac{1}{2h}\right)\right] +$$
$$q(x_i)\left[C_{i-3}\left(\frac{4-\lambda}{24}\right) + C_{i-2}\left(\frac{8+\lambda}{12}\right) + C_{i-1}\left(\frac{4-\lambda}{24}\right)\right] = r(x_i). \tag{9}$$

Similarly, the boundary conditions in (8) are simplified into (10) and (11):

$$S(a) = S(x_0) = C_{-3}\left(\frac{4-\lambda}{24}\right) + C_{-2}\left(\frac{8+\lambda}{12}\right) + C_{-1}\left(\frac{4-\lambda}{24}\right) = \alpha. \tag{10}$$

$$S(b) = S(x_n) = C_{n-3}\left(\frac{4-\lambda}{24}\right) + C_{n-2}\left(\frac{8+\lambda}{12}\right) + C_{n-1}\left(\frac{4-\lambda}{24}\right) = \beta. \tag{11}$$

(9), (10) and (11) can be arranged into a system of linear equation of dimension $(n+3)\times(n+3)$ and can be written in a matrix equation as in with C being the unknown vector.

$$[\mathbf{A}]_{(n+3)\times(n+3)}[\mathbf{C}]_{(n+3)\times 1} = [\mathbf{R}]_{1\times(n+3)}, \tag{12}$$

where:

$$\mathbf{A} = \begin{pmatrix} \frac{4-\lambda}{24} & \frac{8+\lambda}{12} & \frac{4-\lambda}{24} & 0 & 0 & \cdots & \cdots & 0 \\ A_1(x_0) & A_2(x_0) & A_3(x_0) & 0 & 0 & \cdots & \cdots & 0 \\ 0 & A_1(x_1) & A_2(x_1) & A_3(x_1) & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \cdots & 0 & 0 & A_1(x_n) & A_2(x_n) & A_3(x_n) \\ 0 & \cdots & \cdots & 0 & 0 & \frac{4-\lambda}{24} & \frac{8+\lambda}{12} & \frac{4-\lambda}{24} \end{pmatrix},$$

$$A_1(x_i) = \frac{2+\lambda}{2h^2} - p(x_i)\frac{1}{2h} + q(x_i)\frac{4-\lambda}{24},$$
$$A_2(x_i) = -\frac{2+\lambda}{h^2} + q(x_i)\frac{8+\lambda}{12},$$
$$A_3(x_i) = \frac{2+\lambda}{2h^2} + p(x_i)\frac{1}{2h} + q(x_i)\frac{4-\lambda}{24},$$

$$\mathbf{C} = \begin{pmatrix} C_{-3} \\ C_{-2} \\ \vdots \\ C_{n-1} \end{pmatrix}, \qquad \mathbf{R} = \begin{pmatrix} \alpha \\ r(x_0) \\ r(x_1) \\ \vdots \\ r(x_n) \\ \beta \end{pmatrix}.$$

The first and the last lines of **A** are the boundary conditions from and , whereas the rest are from . Hence, **C** can be solved by taking $\mathbf{C} = \mathbf{A}^{-1}\mathbf{R}$. The obtained values of C_i, for $i = -3,-2,\ldots, n-1$, are substituted in , which is the approximated analytical solution to the problems. However, this solution contains two free parameters, which are x and λ. Thus, in order to get the right value of λ, an optimization of that value is needed. For the sake of clarity, starting here, $S(x)$ is referred to $S(x, \lambda)$. $S(x, \lambda)$ is a piecewise polynomial with n intervals, as in , where each $S_i(x, \lambda)$, for $i = 1,2,\ldots,n$, is a polynomial of degree 4:

$$S(x,\lambda)\begin{cases} S_1(x,\lambda), & x \in [x_0, x_1], \\ S_2(x,\lambda), & x \in [x_1, x_2], \\ \vdots & \vdots \\ S_n(x,\lambda), & x \in [x_{n-1}, x_n]. \end{cases} \tag{13}$$

The main idea of obtaining optimized value of λ is by referring back to the general form of the problem:

$$u''(x) + p(x)\, u'(x) + q(x)\, u(x) = r(x). \tag{14}$$

By moving $r(x)$ to the left side of (14) and substituting the approximated solution, $S(x, \lambda)$ as well as its derivatives, we have:

$$S''(x,\lambda) + p(x)S'(x,\lambda) + q(x)S(x,\lambda) - r(x) \approx 0. \tag{15}$$

Interestingly, (15) represents a version of error formula, where the exact expression, $r(x)$, is subtracted from the approximated expression. Therefore, the left side of (15) can be manipulated to optimize λ. For clarity, the expression is called $D(x, \lambda)$,

$$D(x,\lambda) = S''(x,\lambda) + p(x)S'(x,\lambda) + q(x)S(x,\lambda) - r(x),$$
$$x \in [x_0, x_n],$$

which can be expanded into:

$$\begin{cases} S_1''(x,\lambda) + p(x)S_1'(x,\lambda) + q(x)S_1(x,\lambda) - r(x), & x \in [x_0, x_1], \\ S_2''(x,\lambda) + p(x)S_2'(x,\lambda) + q(x)S_2(x,\lambda) - r(x), & x \in [x_1, x_2], \\ \vdots & \vdots \\ S_n''(x,\lambda) + p(x)S_n'(x,\lambda) + q(x)S_n(x,\lambda) - r(x), & x \in [x_{n-1}, x_n]. \end{cases}$$

Since $D(x, \lambda)$ is a piecewise function with n equations, it is wise to have some representatives from every sub-interval. Suppose we have a sequence of $\{x*_i\}_{i=0}^{m}$, where

$x*_i \in [x_0, x_n]$ and $m \in \mathbb{Z}^+$. Evaluating $D(x, \lambda)$ at $\{x*_i\}_{i=0}^{m}$ would produce a sequence of $(m+1)$ elements containing only one free parameter, λ:

$$\{D(x*_0, \lambda), D(x*_1, \lambda), \ldots, D(x*_m, \lambda)\}. \tag{16}$$

By treating like the error at collocation points, the expressions are combined using the L^2-norm or sum-squared formula resulting (17). Thus,(17) measures the accuracy of the approximated solution, $S(x, \lambda)$:

$$\sqrt{\sum_{i=0}^{m}\left[D\left(x*_i, \lambda\right)\right]^2}. \tag{17}$$

In order to optimize λ, minimization can be applied to . But, minimizing also implies minimizing $d(\lambda)$ in (18), which is easier to calculate than the former:

$$d(\lambda) = \sum_{i=0}^{m}\left[D\left(x*_i, \lambda\right)\right]^2. \tag{18}$$

Lastly, after the optimized value of λ is obtained, it can be substituted back in and hence the approximated analytical solution for the problems.

NUMERICAL EXPERIMENTS AND CONCLUSIONS

ECBIM was implemented on four problems with different nature. This was done in order to check the versatility of the method. The problems and their respective exact solutions are as the following:

Problem 4.1 (Caglar et al. 2006; Fang et al. 2002)

$u''(x) - u'(x) = -e^{x-1} - 1,\ x \in [0,1],\ u(0) = 0,\ u(1) = 0.$

Exact solution: $u(x) = x(1 - e^{x-1})$.

Problem 4.2 (Asaithambi 1995)

$u''(x) + (x+1)u'(x) - 2u(x) = (1-x^2)e^{-x},\ x \in [0,1],$

$u(0) = -1,\ u(1) = 0.$

Exact solution: $(x-1)e^{-x}$.

Problem 4.3 (Burden & Faires 2005)

$u''(x) - \pi^2 u(x) = -2\pi^2 \sin(\pi x),\ x \in [0,1],$

$u(0) = u(1) = 0.$

Exact solution: $u(x) = \sin(\pi x)$.

Problem 4.4 (Asaithambi 1995)

$u''(x) - u(x) = 0,\ x \in [0,1],\ u(0) = 0,\ u(1) = \sinh(1).$

Exact solution: $u(x) = \sinh(x)$.

The values of $x*_i$ were set to be the midpoint of each interval,

$$x*_i = \frac{x_i + x_{i+1}}{2}, \quad i = 0,1,\ldots,n.$$

This implies that only one representation was taken from each interval. This was done to avoid having a very long expression of $d(\lambda)$, as the equation itself is already complicated. Furthermore, taking the collocation point, x_i, for all i, as $x*_i$ was avoided because $r(x_i)$ was used when solving for C_i Thus, in theory,

$$D(x_i, \lambda) = 0, \quad i = 0,1,\ldots,n.$$

For all the problems, the value of n was set to be 10, thus, $h = 0.1$. Hence, $D(x,\lambda)$ was a piecewise polynomial with 10 intervals. All the calculations were performed using MATLAB 7.6.0. The minimization of $d(\lambda)$ was done using Newton's method and the built-in function in MATLAB, *fminsearch*, with 0 as the initial guess.

The norms of the approximated solutions were compared with those of finite difference (FDM), finite element (FEM), finite volume (FVM) and cubic B-spline interpolation methods (CBIM). The results from these methods were generated by solving Problems 4.1 to 4.4 using the methods explained in (Burden & Faires 2005; Caglar et al. 2006; Fang et al. 2002). The results are shown in Table 1 and Table 2. ECBIM(N) shows the results when minimizing was done using Newton's method while ECBIM(B) shows for the built-in function. The norms are defined as the following:

$$\text{Max-norm} = \max_{i=1}^{n-1}\left|S(x_i) - u(x_i)\right|.$$

$$L^2 - \text{norm} = \sqrt{\sum_{i=1}^{n-1}\left[S(x_i) - u(x_i)\right]^2}.$$

For all the problems, ECBIM solved the problems much more accurately than FDM, FEM, FVM and CBIM. When using Newton's method to minimize λ, only five or six iterations needed for the method to converge. The approximated analytical solution for the most improved result, Problem 4.4, using ECBIM(N) is shown in . The analytical form of the error can be obtained by subtracting the corresponding exact solution for Problem 4.4 from . The plot of this error is presented in Figure 2.

$$S(x) = \begin{cases} 1.000x + 0.1665x^3 + 0.002085x^4, & x \in [0, 0.1], \\ 4.201\times10^{-7} + 1.000x + 0.0002517x^2 + 0.1649x^3 + 0.006275x^4, & x \in [0.1, 0.2], \\ 7.242\times10^{-6} + 0.9998x + 0.001274x^2 + 0.1614x^3 + 0.01053x^4, & x \in [0.2, 0.3], \\ 0.00004263 + 0.9994x + 0.003630x^2 + 0.1562x^3 + 0.01489x^4, & x \in [0.3, 0.4], \\ 0.0001583 + 0.9982x + 0.007962x^2 + 0.1490x^3 + 0.01939x^4, & x \in [0.4, 0.5], \\ 0.0004528 + 0.9959x + 0.01502x^2 + 0.1396x^3 + 0.02409x^4, & x \in [0.5, 0.6], \\ 0.001095 + 0.9916x + 0.02571x^2 + 0.1277x^3 + 0.02904x^4, & x \in [0.6, 0.7], \\ 0.002354 + 0.9844x + 0.04111x^2 + 0.1131x^3 + 0.03427x^4, & x \in [0.7, 0.8], \\ 0.004643 + 0.9730x + 0.06255x^2 + 0.09521x^3 + 0.03985x^4, & x \in [0.8, 0.9], \\ 0.008571 + 0.9555x + 0.09161x^2 + 0.07369x^3 + 0.04582x^4, & x \in [0.9, 1.0]. \end{cases} \tag{19}$$

TABLE 1. Norms for Problem 4.1

Method	Max-Norm	L^2-Norm
FDM (Fang et al. 2002)	8.2396×10^{-5}	1.9095×10^{-4}
FEM (Fang et al. 2002)	6.3520×10^{-5}	1.4530×10^{-4}
FVM (Fang et al. 2002)	3.1767×10^{-5}	7.2668E-05
FDM (Burden & Faires 2005)	1.6265×10^{-4}	3.6971×10^{-4}
CBIM (Caglar et al. 2006)	2.8996×10^{-4}	6.6089×10^{-4}
ECBIM(N) ($\lambda = 2.9097 \times 10^{-3}$)	7.9187×10^{-6}	1.6711×10^{-5}
ECBIM(B) ($\lambda = 2.9375 \times 10^{-3}$)	5.7388×10^{-6}	1.1479×10^{-5}

TABLE 2. Norms for Problems 4.2, 4.3 and 4.4

Problem	Method	Max-Norm	L^2-Norm
4.2	FDM (Burden & Faires 2005)	2.8758×10^{-4}	6.6113×10^{-4}
	CBIM (Caglar et al. 2006)	2.3108×10^{-4}	5.2220×10^{-4}
	ECBIM(N) ($\lambda = 2.9100 \times 10^{-3}$)	6.6128×10^{-6}	1.3810×10^{-5}
	ECBIM(B) ($\lambda = 2.9375 \times 10^{-3}$)	4.9685×10^{-6}	9.9122×10^{-6}
4.3	FDM (Burden & Faires 2005)	4.1157×10^{-3}	9.2030×10^{-3}
	CBIM (Caglar et al. 2006)	4.1088×10^{-3}	9.1875×10^{-3}
	ECBIM(N) ($\lambda = -1.6483 \times 10^{-2}$)	5.1503×10^{-6}	1.1516×10^{-5}
	ECBIM(B) ($\lambda = -1.6500 \times 10^{-2}$)	8.9601×10^{-7}	2.0035×10^{-6}
4.2	FDM (Burden & Faires 2005)	5.1880×10^{-5}	1.1764×10^{-4}
	CBIM (Caglar et al. 2006)	5.2011×10^{-5}	1.1794×10^{-4}
	ECBIM(N) ($\lambda = 1.6663 \times 10^{-3}$)	6.4967×10^{-9}	1.4732×10^{-8}
	ECBIM(B) ($\lambda = 1.6875 \times 10^{-3}$)	6.6718×10^{-7}	1.5129×10^{-6}

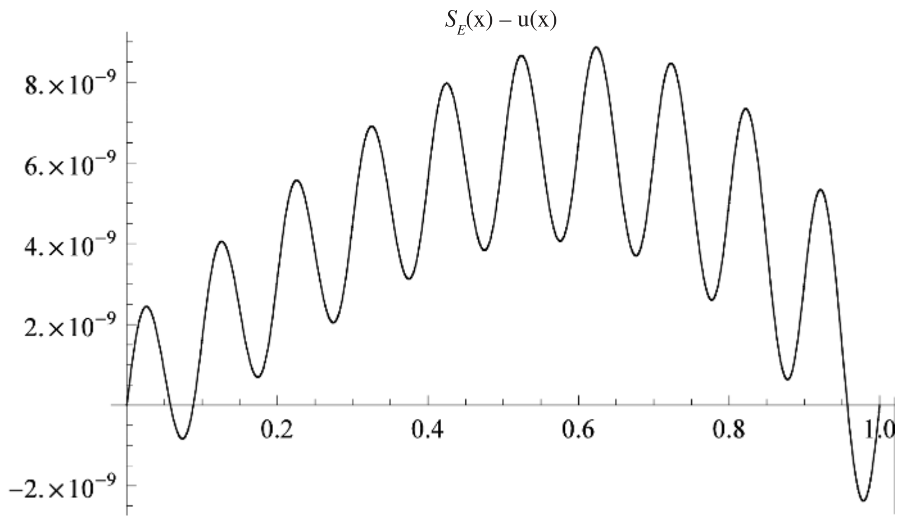

FIGURE 2. Error plot for Problem 4.4 using ECBIM(N)

ACKNOWLEDGEMENT

The authors thank the referee for the helpful comments.

REFERENCES

Agoston, M.K. 2005. Computer graphics and geometric modeling *Implementation and Algorithms* 387-389: 404-445.

Al-Said, E.A. 1998. Cubic spline method for solving two-point boundary-value problems. *Journal of Applied Mathematics and Computing* 5(3): 669-680.

Albasiny, E.L. & Hoskins, W.D. 1969. Cubic spline solutions to two-point boundary value problems. *The Computer Journal* 12(2): 151-153.

Asaithambi, N.S. 1995. *Numerical Analysis Theory and Practice*. Orlando: Saunders College Publishing.

Bickley, W.G. 1968. Piecewise Cubic Interpolation and Two-Point Boundary Problems. *The Computer Journal* 11(2): 206-208.

Burden, R.L. & Faires, J.D. 2005. *Numerical Analysis*. 8th ed. Belmont: Brooks/Cole, Cengage Learning.

Caglar, H., Caglar, N. & Elfaituri, K. 2006. B-spline interpolation compared with finite difference, finite element and finite volume methods which applied to two-point boundary value problems. *Applied Mathematics and Computation* 175(1): 72-79.

Chun, C. & Sakthivel, R. 2010. Homotopy perturbation technique for solving two-point boundary value problems - comparison with other methods. *Computer Physics Communications* 181: 1021-1024.

Fang, Q., Tsuchiya, T. & Yamamoto, T. 2002. Finite difference, finite element and finite volume methods applied to two-point boundary value problems. *Journal of Computational and Applied Mathematics* 139(1): 9-19.

Fyfe, D.J. 1969. The use of cubic splines in the solution of two-point boundary value problems. *The Computer Journal 12*(2): 188-192.

Jang, B. 2008. Two-point boundary value problems by the extended Adomian decomposition method. *Journal of Computational and Applied Mathematics* 219(1): 253-262.

Khan, A. 2004. Parametric cubic spline solution of two point boundary value problems. *Applied Mathematics and Computation* 154(1): 175-182.

Lu, J. 2007. Variational iteration method for solving two-point boundary value problems. *Journal of Computational and Applied Mathematics* 207(1): 92-95.

Patrikalakis, N.M. & Maekawa, T. 2002. *Shape Interrogation for Computer Aided Design and Manufacturing*. New York: Springer-Verlag.

Prautzsch, H., Boehm, W. & Paluszny, M. 2002. *Bezier and B-Spline Techniques*. New York: Springer.

Salomon, D. 2006. *Curves and Surfaces for Computer Graphics*. New York: Springer Science+Business Media, Inc.

Xu, G. & Wang, G.-Z. 2008. Extended Cubic Uniform B-spline and [alpha]-B-spline. *Acta Automatica Sinica* 34(8): 980-984.

School of Mathematical Sciences
Universiti Sains Malaysia
11800 USM, Penang
Malaysia

*Corresponding author; email: nurnadiah_abdhamid@yahoo.com

Received: 20 September 2010
Accepted: 30 March 2011