



A Class of Spline Functions for Solving 2-Order Linear Differential Equations with Boundary Conditions

Chengzhi Liu^{1,*}, Xuli Han² and Juncheng Li¹

- ¹ School of Mathematics and Finance, Hunan University of Humanities, Science and Technology, Loudi 417000, China; lijuncheng@huhst.edu.cn
- ² School of Mathematics and Statistics, Central South University, Changsha 410083, China; xlhan@csu.edu.cn
- * Correspondence: 162101002@csu.edu.cn

Received: 9 August 2020; Accepted: 11 September 2020; Published: 15 September 2020



Abstract: In this paper, we exploit an numerical method for solving second order differential equations with boundary conditions. Based on the theory of the analytic solution, a series of spline functions are presented to find approximate solutions, and one of them is selected to approximate the solution automatically. Compared with the other methods, we only need to solve a tri-diagonal system, which is much easier to implement. This method has the advantages of high precision and less computational cost. The analysis of local truncation error is also discussed in this paper. At the end, some numerical examples are given to illustrate the effectiveness of the proposed method.

Keywords: spline function; truncation error; convergence; second order differential equations

1. Introduction

Ordinary differential equations (ODEs) of second order are used throughout engineering, mathematics and science. It is well known that of all differential equations, only a little of them have analytic solutions. Most of them, especially for these with variable coefficients are very difficult to solve. Very often, the solving techniques are difficult to deal with complicated equations in practice. In this paper, we consider the numerical solutions for second order linear differential equations of the form

$$y''(x) + P(x)y = Q(x), \quad x \in [a, b],$$
 (1)

with the boundary conditions

$$y(a) = A, y(b) = B,$$
(2)

where *A*, *B* are constants and P(x), Q(x) are sufficiently smooth functions. We remark that Equation (1) doesn't contain the first derivative term y'(x), because we can eliminate it by proper transformation.

How to solve the differential equations effectively has gain great attentions in recent years, researchers have been putting forth effort on obtaining stable algorithms with less float operations and higher accuracy. Numerical methods for differential equations are well developed and some of them are widely used, such as Euler's method, the linear multi-step method, Runge–Kutta Method and so on. For more detailed researches on this topic, we refer the reader to read [1].

Recently, solving differential equations by spline functions have been investigated by more and more researchers and a large of spline functions have been exploited to solve these equations. Usmani [2] derived an uniform quartic polynomial spline at mid-knots. Ramadan et al. [3,4] presented some quadratic, cubic polynomial and nonpolynomial spline functions to find the approximate solutions. They also shown that the nonpolynomial splines could yield better approximations and

2 of 11

have lower computational cost than the polynomial ones. By using the nonpolynomial quintic splines, Srivastava et al. developed a numerical algorithm [5]. In [6], a quadratic non-polynomial spline functions based method was proposed by Islam et al. Zahra and Mhlawy studied the singularly perturbed semi-linear boundary value problem with exponential spline [7]. Lodhi and Mishra exploited the numerical method to solve second order singularly perturbed two-point linear and nonlinear boundary value problems [8]. In [9], Zahra proposed the exponential spline functions consisting of a polynomial part of degree three and an exponential part to find approximation of linear and nonlinear fourth order two-point boundary value problems. By solving a algebraic system, Said and Noor proposed the optimized multigrid methods [10]. Ideon and Peeter investigated the collocation method with linear/linear rational spline for the numerical solution of two-point boundary value problems [11]. By using B-spline, Rashidinia and Ghasemi developed a numerical method which can solve the general nonlinear two-point boundary value problems up to order 6 [12].

Despite the fact that there are a large number of spline functions to approximate the solutions to the equations, most of them are not constructed based on the differential equations to be solved, and can not achieve satisfying results. To the best of our knowledge, there is still no any research about the construction of spline functions based on the differential equations. It is known to all that second order linear differential equations with constant coefficients can be solved analytically. This leads us to think that whether we can use the analytic solutions to solve these with variable coefficients. The spline function based on analytic solutions can achieve better approximation performance. Therefore, we put forth effort on constructing a spline function based on analytic solutions of equations with constant coefficients.

2. Preliminary

We first review some conclusions on the theory of solutions to linear differential equations.

Equation (1) is said to be homogeneous if $Q(x) \equiv 0$, and nonhomogeneous if not. Moreover, the equation y'' + P(x)y = 0 is said to be the associated homogeneous equation of (1). Equation (1) is said to have constant coefficients when P(x) is a constant, and have variable coefficients if not. If Equation (1) has constant coefficients, it can be written as

$$y'' + py = Q(x), \tag{3}$$

where *p* is a constant. The associated homogeneous equation of (3) is y'' + py = 0 and the characteristic equation of (3) is $\lambda^2 + p = 0$. Hence the characteristic roots of (3) are $\lambda = \pm \sqrt{-p}$.

Let y_p denote any particular solution of Equation (3) and y_h represent the general solution of its associated homogeneous equation, then the general solution of Equation (3) is $y = y_h + y_p$.

The following two lemmas are introduced for obtaining the general solution y_h and particular solution y_p of Equation (3), which will be used in the next section.

Lemma 1 ([13]). If p < 0, then λ is real, and hence the general solution is $y_h = k_1 e^{\lambda x} + k_2 e^{-\lambda x}$. If p = 0, so is the value of λ , then the general solution is $y_h = k_1 e^{0x} + k_2 x e^{0x} = k_1 + k_2 x$. If p > 0, then λ is a pair of conjugated imaginary numbers, and hence the general solution is $y_h = k_1 \sin(\lambda x) + k_2 \cos(\lambda x)$. Here k_1, k_2 are arbitrary constants.

Lemma 2 ([13]). Suppose that Q(x) is a polynomial of degree *n*, then a particular solution of Equation (2) is

$$y_p = e^{\alpha x} (k_n x^n + k_{n-1} x^{n-1} + \dots + k_1 x_1 + k_0),$$

where α is a known constant and k_i ($i = 0, 1, 2, \dots, n$) are constants to be determined.

Specially, when the degree of Q(x) equals to 0, the particular solution $y_p = c$ if $\lambda \neq 0$; and $y_p = cx^2$ if $\lambda = 0$, where c is a constant.

3. Derivation of The Method

3.1. Definition of the Piecewise Spline Functions

In this subsection, we use the theory of the analytic solution to construct the spline functions. Firstly, the interval [a, b] is equally divided into n subintervals $[x_{i-1}, x_i](i = 1, 2, ..., n)$ such that $\triangle : a = x_0 < x_1 < \cdots < x_n = b$ with $h = x_i - x_{i-1} = (b - a)/n$, and we denote $x_{i\pm 1/2} = x_i \pm \frac{1}{2}h$.

If *h* is small, the value of *x* varies little at the interval $[x_{i-1}, x_i]$, then P(x) and Q(x) can be seem as constants at the interval $[x_{i-1}, x_i]$, therefore Equation (1) can also be seem as a second order linear differential equation with constant coefficients at the interval $[x_{i-1}, x_i]$. We assume that the value of *x* at the interval $[x_{i-1}, x_i]$ identically equals to the midpoint $x_{i-1/2}$, hence Equation (1) can be seem as

$$y'' + P_{i-1/2}y = Q_{i-1/2},$$
(4)

where $P_{i-1/2} = P(x_{i-1/2}), Q_{i-1/2} = Q(x_{i-1/2}).$

The roots of the characteristic equation of (4) is $\lambda_i = \pm \sqrt{-P_{i-1/2}}$. Therefore, the solution of (4) can be obtained in the following three cases.

case 1. If $P_{i-1/2} < 0$, the characteristic roots λ_i are nonzero real numbers, it follows from Lemma 1 that $y_h = a_i e^{\lambda_i (x-x_i)} + b_i e^{-\lambda_i (x-x_i)}$. Additionally, since the right side of Equation (4) identically equals to the constant $Q_{i-1/2}$, we can assume that $y_p = c$ according to Lemma 2. By substituting y_p into Equation (4), we have $c = \frac{Q_{i-1/2}}{P_{i-1/2}}$. Therefore, the approximate function at the interval $[x_{i-1}, x_i]$ can be defined as

$$\tilde{y}_i(x) = a_i e^{\lambda_i (x - x_{i-1})} + b_i e^{-\lambda_i (x - x_{i-1})} + \frac{Q_{i-1/2}}{P_{i-1/2}}$$

case 2. If $P_{i-1/2} > 0$, then the characteristic roots λ_i are imaginary numbers, it follows from Lemma 1 that $y_{li} = c_i \sin \lambda_i (x - x_{i-1}) + d_i \cos \lambda_i (x - x_{i-1})$. Similarly to that described in Case 1, the particular solution is $y_p = \frac{Q_{i-1/2}}{P_{i-1/2}}$. Therefore, the approximate function at the interval $[x_i, x_{i+1}]$ can be defined as

$$\tilde{y}_i(x) = c_i \sin \lambda_i (x - x_{i-1}) + d_i \cos \lambda_i (x - x_{i-1}) + \frac{Q_{i-1/2}}{P_{i-1/2}}.$$

case 3. If $P_{i-1/2} = 0$, then $\lambda_i = 0$, the general solution of the homogeneous is $y_h = e_i + f_i(x - x_{i-1})$. There are many particular solutions, here we take the simplest form $y_p = g_i(x - x_{i-1})^2$. By substituting y_p into Equation (4), we yield $g_i = \frac{1}{2}Q_{i-1/2}$. Therefore, the approximate function at the interval $[x_i, x_{i+1}]$ can be defined as

$$\tilde{y}_i(x) = \frac{1}{2}Q_{i-1/2}(x-x_{i-1})^2 + e_i(x-x_{i-1}) + f_i,$$

To summarize, the approximate function at the interval $[x_{i-1}, x_i]$ is defined as

$$\tilde{y}_{i}(x) = \begin{cases} a_{i}e^{\lambda_{i}(x-x_{i-1})} + b_{i}e^{-\lambda_{i}(x-x_{i-1})} + \frac{Q_{i-1/2}}{P_{i-1/2}}, & \text{if } P_{i-1/2} < 0; \\ c_{i}\sin\lambda_{i}(x-x_{i-1}) + d_{i}\cos\lambda_{i}(x-x_{i-1}) + \frac{Q_{i-1/2}}{P_{i-1/2}}, & \text{if } P_{i-1/2} > 0; \\ \frac{1}{2}Q_{i-1/2}(x-x_{i-1})^{2} + e_{i}(x-x_{i-1}) + f_{i}, & \text{if } P_{i-1/2} = 0, \end{cases}$$
(5)

where a_i, b_i, c_i, d_i, e_i and f_i are the coefficients to be determined.

Let the spline Function (5) satisfy the interpolation conditions

$$\tilde{y}_i(x_{i-1}) = y_{i-1}, \tilde{y}_i(x_i) = y_i.$$
 (6)

From (5) and (6), by a tedious but straightforward calculation, we can obtain the coefficients

$$a_{i} = \frac{(y_{i-1} - \frac{Q_{i-1/2}}{P_{i-1/2}})e^{\lambda_{i}h} - y_{i} + \frac{Q_{i-1/2}}{P_{i-1/2}}}{A_{i}}; \ b_{i} = \frac{(y_{i-1} - \frac{Q_{i-1/2}}{P_{i-1/2}})e^{-\lambda_{i}h} - y_{i} + \frac{Q_{i-1/2}}{P_{i-1/2}}}{B_{i}};$$

$$c_{i} = y_{i} - \frac{Q_{i-1/2}}{P_{i-1/2}}; d_{i} = \frac{y_{i} - \frac{Q_{i-1/2}}{P_{i-1/2}} - (y_{i-1} - \frac{Q_{i-1/2}}{P_{i-1/2}})\cos(\lambda_{i}h)}{\sin(\lambda_{i}h)}; e_{i} = \frac{y_{i} - y_{i-1} - \frac{1}{2}Q_{i-1/2}h^{2}}{h}; f_{i} = y_{i-1},$$

where $A_i = e^{\lambda_i h} - e^{-\lambda_i h}$, $B_i = e^{\lambda_i h} + e^{-\lambda_i h}$.

3.2. Continuity Condition

Very often, the approximate function on adjacent subintervals is usually required to be C^1 continuous, i.e.,

$$\tilde{y}_{i-1}'(x_i^-) = \tilde{y}_i'(x_i^+), i = 1, 2, \cdots, n-1.$$
(7)

From (5), we have

$$\tilde{y}_{i}'(x) = \begin{cases} a_{i}\lambda_{i}e^{\lambda_{i}(x-x_{i-1})} - \lambda_{i}b_{i}e^{-\lambda_{i}(x-x_{i-1})}, & \text{if } P_{i-1/2} < 0; \\ c_{i}\lambda_{i}\cos\lambda_{i}(x-x_{i-1}) - \lambda_{i}d_{i}\sin\lambda_{i}(x-x_{i-1}), & \text{if } P_{i-1/2} > 0; \\ Q_{i-1/2}(x-x_{i-1}) + f_{i}, & \text{if } P_{i-1/2} = 0. \end{cases}$$
(8)

The derivative functions in (7) are selected according to the values of $P_{i-1/2}$ and $P_{i+1/2}$, and hence the C^1 continuity condition (7) can be arranged as one of the following situations:

1. If $P_{i-1/2} < 0$ and $P_{i+1/2} < 0$, then from (7) we have

$$\alpha_i^1 \tilde{y}_{i-1} + \beta_i^1 \tilde{y}_i + \gamma_i^1 \tilde{y}_{i+1} = \delta_i^1, \tag{9}$$

where $\alpha_i^1 = -2\lambda_{i-1}A_i$, $\beta_i^1 = \lambda_{i-1}A_iB_{i-1} + \lambda_iA_{i-1}B_i$, $\gamma_i^1 = -2\lambda_iA_{i-1}$, $\delta_i^1 = \lambda_{i-1}A_i\frac{Q_{i-1/2}}{P_{i-1/2}}(B_{i-1} - A_i)$ 2) + $\lambda_i A_{i-1} \frac{Q_{i+1/2}}{P_{i+1/2}} (B_i - 2)$. 2. If $P_{i-1/2} > 0$ and $P_{i+1/2} > 0$, then from (7) we have

$$\alpha_i^2 \tilde{y}_{i-1} + \beta_i^2 \tilde{y}_i + \gamma_i^2 \tilde{y}_{i+1} = \delta_i^2, \tag{10}$$

where $\alpha_i^2 = -\lambda_{i-1} \sin(\lambda_i h), \beta_i^2 = \lambda_{i-1} \sin(\lambda_i h) \cos(\lambda_{i-1} h) + \lambda_i \sin(\lambda_{i-1} h) \cos(\lambda_i h), \gamma_i^2 = -\lambda_i \sin(\lambda_{i-1} h), \delta_i^2 = \lambda_{i-1} \sin(\lambda_i h) [\cos(\lambda_{i-1} h) - 1] \frac{Q_{i-1/2}}{P_{i-1/2}} + \lambda_i \sin(\lambda_{i-1} h) [\cos(\lambda_i h) - 1] \frac{Q_{i+1/2}}{P_{i+1/2}}.$ 3. If $P_{i-1/2} = 0$ and $P_{i+1/2} = 0$, then from (7) we have

$$\alpha_i^3 \tilde{y}_{i-1} + \beta_i^3 \tilde{y}_i + \gamma_i^3 \tilde{y}_{i+1} = \delta_i^3,$$
(11)

where $\alpha_i^3 = -1$, $\beta_i^3 = 2$, $\gamma_i^3 = -1$, $\delta_i^3 = -\frac{1}{2}[Q_{i-1/2} + Q_{i+1/2}]h^2$. 4. If $P_{i-1/2} < 0$ and $P_{i+1/2} > 0$, then from (7) we have

$$\alpha_i^4 \tilde{y}_{i-1} + \beta_i^4 \tilde{y}_i + \gamma_i^4 \tilde{y}_{i+1} = \delta_i^4, \tag{12}$$

where $\alpha_i^4 = -2\lambda_{i-1}\sin(\lambda_i h), \beta_i^4 = \lambda_{i-1}B_{i-1}\sin(\lambda_i h) + \lambda_i A_{i-1}\cos(\lambda_i h), \gamma_i^4 = -\lambda_i A_{i-1}, \delta_i^4 = \lambda_{i-1}\sin(\lambda_i h)\frac{Q_{i-1/2}}{P_{i-1/2}}(B_{i-1}-2) + \lambda_i A_{i-1}[\cos(\lambda_i h)-1]\frac{Q_{i+1/2}}{P_{i+1/2}}.$ 5. If $P_{i-1/2} < 0$ and $P_{i+1/2} = 0$, then from (7) we have

$$\alpha_i^5 \tilde{y}_{i-1} + \beta_i^5 \tilde{y}_i + \gamma_i^5 \tilde{y}_{i+1} = \delta_i^5,$$
(13)

where $\alpha_i^5 = -2\lambda_{i-1}h$, $\beta_i^5 = A_{i-1} + \lambda_{i-1}B_{i-1}h$, $\gamma_i^5 = -A_{i-1}$, $\delta_i^5 = \lambda_{i-1}h\frac{Q_{i-1/2}}{P_{i-1/2}}(B_{i-1}-2) - 2\lambda_{i-1}h$ $\frac{1}{2}Q_{i+1/2}A_{i-1}h^2$.

6. If $P_{i-1/2} > 0$ and $P_{i+1/2} < 0$, then from (7) we have

$$\alpha_i^6 \tilde{y}_{i-1} + \beta_i^6 \tilde{y}_i + \gamma_i^6 \tilde{y}_{i+1} = \delta_i^6, \tag{14}$$

where $\alpha_{i}^{6} = -\lambda_{i-1}A_{i}, \beta_{i}^{6} = \lambda_{i-1}A_{i}\cos(\lambda_{i-1}h) + \lambda_{i}\sin(\lambda_{i-1}h)B_{i}, \gamma_{i}^{6} = -2\lambda_{i}\sin(\lambda_{i-1}h), \delta_{i}^{6} = \lambda_{i}\sin(\lambda_{i-1}h)(B_{i}-2)\frac{Q_{i+1/2}}{P_{i+1/2}} + \lambda_{i-1}A_{i}[\cos(\lambda_{i-1}h)-1]\frac{Q_{i-1/2}}{P_{i-1/2}}.$ 7. If $P_{i-1/2} > 0$ and $P_{i+1/2} = 0$, then from (7) we have

$$\alpha_{i}^{7}\tilde{y}_{i-1} + \beta_{i}^{7}\tilde{y}_{i} + \gamma_{i}^{7}\tilde{y}_{i+1} = \delta_{i}^{7},$$
(15)

where $\alpha_i^7 = -\lambda_{i-1}h, \beta_i^7 = \lambda_{i-1}\cos(\lambda_{i-1}h)h + \sin(\lambda_{i-1}h), \gamma_i^7 = -\sin(\lambda_{i-1}h), \delta_i^7 = -\frac{1}{2}Q_{i+1/2}\sin(\lambda_{i-1}h)h^2 + \lambda_{i-1}h[\cos(\lambda_{i-1}h) - 1]\frac{Q_{i-1/2}}{P_{i-1/2}}.$ 8. If $P_{i-1/2} = 0$ and $P_{i+1/2} < 0$, then from (7) we have

$$\alpha_i^8 \tilde{y}_{i-1} + \beta_i^8 \tilde{y}_i + \gamma_i^8 \tilde{y}_{i+1} = \delta_i^8, \tag{16}$$

where $\alpha_i^8 = -A_i$, $\beta_i^8 = A_i + \lambda_i B_i h$, $\gamma_i^8 = -2\lambda_i h$, $\delta_i^8 = -\frac{1}{2}A_i Q_{i-1/2}h^2 + \lambda_i h \frac{Q_{i+1/2}}{P_{i+1/2}}(B_i - 2)$. 9. If $P_{i-1/2} = 0$ and $P_{i+1/2} > 0$, then from (7) we have

$$\alpha_{i}^{9}\tilde{y}_{i-1} + \beta_{i}^{9}\tilde{y}_{i} + \gamma_{i}^{9}\tilde{y}_{i+1} = \delta_{i}^{9},$$
(17)

$$\begin{aligned} \alpha_i^9 &= -\sin(\lambda_i h), \beta_i^9 &= \sin(\lambda_i h) + \lambda_i \cos(\lambda_i h)h, \gamma_i^9 &= -\lambda_i h, \delta_i^9 &= -\frac{1}{2}Q_{i-1/2}\sin(\lambda_i h)h^2 + \lambda_i h [\cos(\lambda_i h) - 1] \frac{Q_{i+1/2}}{P_{i+1/2}}. \end{aligned}$$

We remark that there are n-1 linear equations including n+1 unknowns. In order to solve the system uniquely, we need two additional conditions, which can be obtained from the boundary conditions (2). The continuity conditions (7) combined with the boundary conditions (2) form a system of linear equations with n + 1 unknowns, which can be rewritten in the matrix form as

$$\begin{pmatrix} 1 & 0 & & & & \\ \alpha_{1}^{j_{1}} & \beta_{1}^{j_{1}} & \gamma_{1}^{j_{1}} & & & \\ & \alpha_{2}^{j_{2}} & \beta_{2}^{j_{2}} & \gamma_{2}^{j_{2}} & & \\ & \ddots & \ddots & \ddots & & \\ & & \alpha_{n-2}^{j_{n-2}} & \beta_{n-2}^{j_{n-2}} & \gamma_{n-2}^{j_{n-2}} & \\ & & & \alpha_{n-1}^{j_{n-1}} & \beta_{n-1}^{j_{n-1}} & \gamma_{n-1}^{j_{n-1}} \\ & & & & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{y}_{0} \\ \tilde{y}_{1} \\ \tilde{y}_{2} \\ \vdots \\ \tilde{y}_{n-2} \\ \tilde{y}_{n-1} \\ \tilde{y}_{n} \end{pmatrix} = \begin{pmatrix} A \\ \delta_{1}^{j_{1}} \\ \delta_{2}^{j_{2}} \\ \vdots \\ \delta_{n-2}^{j_{n-2}} \\ \delta_{n-1}^{j_{n-1}} \\ B \end{pmatrix}$$
(18)

Here the value of the superscript j_k (k = 1, 2, ..., n - 1) may be one of 1–9, which is determined by $P_{i-1/2}$ and $P_{i+1/2}$.

The coefficient matrix of the linear system (18) is tridiagonal, which can be solved easily by Thomas algorithm.

4. Truncation Error Estimation

Suppose y(x) is sufficiently smooth at [a, b]. By substituting $x_{i-1/2}$ into (1), we have

$$M'_{i-1/2} + P_{i-1/2}y_{i-1/2} = Q_{i-1/2},$$
(19)

where $M'_{i-1/2}, y_{i-1/2}$ denote $y''(x_{i-1/2}), y(x_{i-1/2})$ respectively.

To estimate the truncation error, we have to consider all these nine cases in Section 2, here we only analysis parts of these cases due to space limitations. Without loss of generality, we discuss the truncation errors of case 1 and 2 in detail, the results of the others are also listed. For convenience, the local truncation error is denoted by t_i^j ($i = 1, 2, \dots, n-1; j = 1, 2, \dots, 9$).

According to (9), the local truncation error can be written as

$$t_i^1 = \alpha_i^1 \tilde{y}_{i-1} + \beta_i^1 \tilde{y}_i + \gamma_i^1 \tilde{y}_{i+1} - \delta_i^1.$$

By using Taylor series, the terms y_{i-1} , $y_{i+1/2}$, y_{i+1} , $y_{i+1/2}$ and $M'_{i-1/2}$, $M'_{i+1/2}$ can be expanded around the point x_i , hence we have

$$t_i^1 = \sum_{j=0}^6 k_j^1 y_i^{(j)} + O(h^7),$$

where

$$\begin{split} k_0^1 &= 0; k_1^1 = h(\lambda_{i-1}A_i - \lambda_iA_{i-1}) + \frac{h}{2}(\lambda_{i-1}A_iB_{i-1} - \lambda_iA_{i-1}B_i); \\ k_2^1 &= -\frac{3h^2}{4}(\lambda_{i-1}A_i + \lambda_iA_{i-1}) - \frac{h^2}{8}(\lambda_{i-1}A_iB_{i-1} + \lambda_iA_{i-1}B_i) + \frac{A_i(2 - B_{i-1})}{\lambda_{i-1}} + \frac{A_{i-1}(2 - B_i)}{\lambda_i}; \\ k_3^1 &= \frac{7h^3}{24}(\lambda_{i-1}A_i - \lambda_iA_{i-1}) + \frac{h^3}{48}(\lambda_{i-1}A_iB_{i-1} - \lambda_iA_{i-1}B_i) - \frac{hA_i(2 - B_{i-1})}{2\lambda_{i-1}} + \frac{hA_{i-1}(2 - B_i)}{2\lambda_i}; \\ k_4^1 &= -\frac{15h^4}{192}(\lambda_{i-1}A_i + \lambda_iA_{i-1}) - \frac{h^4}{384}(\lambda_{i-1}A_iB_{i-1} + \lambda_iA_{i-1}B_i) + \frac{h^2A_i(2 - B_{i-1})}{8\lambda_{i-1}} + \frac{h^2A_{i-1}(2 - B_i)}{8\lambda_i}; \\ k_5^1 &= \frac{31h^5}{1920}(\lambda_{i-1}A_i - \lambda_iA_{i-1}) + \frac{h^5}{3840}(\lambda_{i-1}A_iB_{i-1} - \lambda_iA_{i-1}B_i) - \frac{h^3A_i(2 - B_{i-1})}{48\lambda_{i-1}} + \frac{h^3A_{i-1}(2 - B_i)}{48\lambda_i}; \\ k_6^1 &= -\frac{63h^6}{23040}(\lambda_{i-1}A_i + \lambda_iA_{i-1}) - \frac{h^6}{46080}(\lambda_{i-1}A_iB_{i-1} + \lambda_iA_{i-1}B_i) + \frac{h^4A_i(2 - B_{i-1})}{384\lambda_{i-1}} + \frac{h^4A_{i-1}(2 - B_i)}{384\lambda_{i-1}}; \\ \end{split}$$

Note that
$$A_i = 2 \sinh(\lambda_i h)$$
, $B_i = 2 \cosh(\lambda_i h)$, since the hyperbolic functions can be expanded in

$$\sinh(x) = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots; \cosh(x) = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots.$$

By expanding the terms A_{i-1} , A_i , B_{i-1} and B_i in series, and by a tedious but straightforward calculation, k_i^1 (j = 1, 2, 3, 4, 5, 6) can be written as

$$\begin{aligned} k_1^1 &= \frac{h^4}{3}\lambda_{i-1}\lambda_i(\lambda_{i-1}^2 - \lambda_i^2) - \frac{h^6}{20}\lambda_{i-1}\lambda_i(\lambda_i^4 - \lambda_{i-1}^4) + O(h^7); \\ k_2^1 &= -\frac{h^5}{12}\lambda_{i-1}\lambda_i(\lambda_{i-1}^2 + \lambda_i^2) + O(h^7); \\ k_3^1 &= -\frac{h^6}{72}\lambda_{i-1}\lambda_i(\lambda_i^2 - \lambda_{i-1}^2) + O(h^7); \\ k_4^1 &= \frac{h^5}{6}\lambda_{i-1}\lambda_i + O(h^7); \\ k_5^1 &= k_6^1 = 0 + O(h^7). \end{aligned}$$

Similarly, the local truncation error of (10) can be written as

$$t_i^2 = \alpha_i^2 \tilde{y}_{i-1} + \beta_i^2 \tilde{y}_i + \gamma_i^2 \tilde{y}_{i+1} - \delta_i^2 = \sum_{j=0}^6 k_j^2 y_i^{(j)} + O(h^7),$$

where

$$\begin{split} k_0^2 &= 0; k_1^2 = \frac{h}{2} (\lambda_{i-1} \sin(\lambda_i h) [1 + \cos(\lambda_{i-1} h)] - \frac{h}{2} (\lambda_i \sin(\lambda_{i-1} h) [1 + \cos(\lambda_i h)]; \\ k_2^2 &= -\frac{3h^2}{8} [\lambda_{i-1} \sin(\lambda_i h) + \lambda_i \sin(\lambda_{i-1} h)] - \frac{h^2}{8} [\lambda_{i-1} \sin(\lambda_i h) \cos(\lambda_{i-1} h) + \lambda_i \sin(\lambda_{i-1} h) \cos(\lambda_i h)] \\ &+ \frac{\sin(\lambda_i h) [1 - \cos(\lambda_{i-1} h)]}{\lambda_{i-1}} + \frac{\sin(\lambda_{i-1} h) [1 - \cos(\lambda_i h)]}{\lambda_i}; \\ k_3^2 &= \frac{7h^3}{48} [\lambda_{i-1} \sin(\lambda_i h) - \lambda_i \sin(\lambda_{i-1} h)] + \frac{h^3}{48} [\lambda_{i-1} \sin(\lambda_i h) \cos(\lambda_{i-1} h) - \lambda_i \sin(\lambda_{i-1} h) \cos(\lambda_i h)] \\ &- \frac{h \sin(\lambda_i h) [1 - \cos(\lambda_{i-1} h)]}{2\lambda_{i-1}} + \frac{h \sin(\lambda_{i-1} h) [1 - \cos(\lambda_i h)]}{2\lambda_i}; \end{split}$$

$$\begin{split} k_4^2 &= -\frac{15h^4}{384} [\lambda_{i-1}\sin(\lambda_i h) + \lambda_i \sin(\lambda_{i-1} h)] - \frac{h^4}{384} [\lambda_{i-1}\sin(\lambda_i h)\cos(\lambda_{i-1} h) + \lambda_i \sin(\lambda_{i-1} h)\cos(\lambda_i h)] \\ &+ \frac{h^2 \sin(\lambda_i h)[1 - \cos(\lambda_{i-1} h)]}{8\lambda_{i-1}} + \frac{h^2 \sin(\lambda_{i-1} h)[1 - \cos(\lambda_i h)]}{8\lambda_i}; \\ k_5^2 &= \frac{31h^5}{3840} [\lambda_{i-1}\sin(\lambda_i h) - \lambda_i \sin(\lambda_{i-1} h)] + \frac{h^5}{3840} [\lambda_{i-1}\sin(\lambda_i h)\cos(\lambda_{i-1} h) - \lambda_i \sin(\lambda_{i-1} h)\cos(\lambda_i h)] \\ &- \frac{h^3 \sin(\lambda_i h)[1 - \cos(\lambda_{i-1} h)]}{48\lambda_{i-1}} + \frac{h^3 \sin(\lambda_{i-1} h)[1 - \cos(\lambda_i h)]}{48\lambda_i}; \\ k_6^2 &= -\frac{63h^6}{46080} [\lambda_{i-1}\sin(\lambda_i h) + \lambda_i \sin(\lambda_{i-1} h)] - \frac{h^6}{46080} [\lambda_{i-1}\sin(\lambda_i h)\cos(\lambda_{i-1} h) + \lambda_i \sin(\lambda_{i-1} h)\cos(\lambda_i h)] \\ &+ \frac{h^4 \sin(\lambda_i h)[1 - \cos(\lambda_{i-1} h)]}{384\lambda_{i-1}} + \frac{h^4 \sin(\lambda_{i-1} h)[1 - \cos(\lambda_i h)]}{384\lambda_i}. \end{split}$$

Since the trigonometric functions can be expressed in

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots; \cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots.$$

By expanding the trigonometric functions of k_j^2 ($j = 1, 2, \dots, 6$) in series, then k_j^2 ($j = 1, 2, \dots, 6$) can be written as

$$\begin{split} k_1^2 &= \frac{h^4}{12} \lambda_{i-1} \lambda_i (\lambda_i^2 - \lambda_{i-1}^2) - \frac{h^6}{80} \lambda_{i-1} \lambda_i (\lambda_i^4 - \lambda_{i-1}^4) + O(h^7); \\ k_2^2 &= \frac{h^6}{288} \lambda_{i-1} \lambda_i (\lambda_i^2 - \lambda_{i-1}^2) + \frac{h^6}{720} \lambda_{i-1} \lambda_i (\lambda_i^4 - \lambda_{i-1}^4) + O(h^7); \\ k_2^2 &= \frac{h^6}{24} \lambda_{i-1} \lambda_i (\lambda_i^2 - \lambda_{i-1}^2) + \frac{h^6}{720} \lambda_{i-1} \lambda_i (\lambda_i^4 - \lambda_{i-1}^4) + O(h^7); \\ k_2^2 &= k_6^2 = 0 + O(h^7). \end{split}$$

Then the local truncation error of (10) is $O(h^4)$. Next, we only give conclusions of the other cases. The truncation error of (11) is

$$t_i^3 = \sum_{j=0}^6 k_i^3 y_i^{(j)} + O(h^7),$$

where $k_0^3 = k_1^3 = k_2^3 = k_3^3 = 0$; $k_4^3 = \frac{h^4}{24}$; $k_5^3 = 0$; $k_6^3 = -\frac{h^6}{5760}$. The truncation error of (12) is

$$t_i^4 = \sum_{j=0}^6 k_j^4 y_i^{(j)} + O(h^7),$$

where

$$\begin{split} k_0^4 &= 0; k_1^4 = \frac{h^4}{6} \lambda_{i-1} \lambda_i (\lambda_{i-1}^2 + \lambda_i^2) - \frac{h^6}{40} \lambda_{i-1} \lambda_i (\lambda_i^4 - \lambda_{i-1}^4) + O(h^7); k_2^4 = \frac{h^5}{24} \lambda_{i-1} \lambda_i (\lambda_i^2 - \lambda_{i-1}^2) + O(h^7); k_3^4 = \frac{h^6}{144} \lambda_{i-1} \lambda_i (\lambda_{i-1}^2 + \lambda_i^2) + O(h^7); k_4^4 = \frac{h^5}{12} \lambda_{i-1} \lambda_i + O(h^7); k_5^4 = k_6^4 = 0 + O(h^7). \end{split}$$

The truncation error of (13) is

$$t_i^5 = \sum_{j=0}^6 k_j^5 y_i^{(j)} + O(h^7),$$

where $k_0^5 = 0$; $k_1^5 = \frac{h^4}{6}\lambda_{i-1}^3 + \frac{h^6}{40}\lambda_{i-1}^5 + O(h^7)$; $k_2^5 = -\frac{h^5}{24}\lambda_{i-1}^3 + O(h^7)$; $k_3^5 = -\frac{11h^6}{144}\lambda_{i-1}^3 + O(h^7)$; $k_4^5 = -\frac{h^6}{144}\lambda_{i-1}^3 + O(h^7)$; $k_5^5 = -\frac{h^6}{144}\lambda_{i-1}^3 +$ $\frac{h^5}{12}\lambda_{i-1} + O(h^7); k_5^5 = k_6^5 = 0 + O(h^7).$

The truncation error of (14) is

$$t_i^6 = \sum_{j=0}^6 k_6^6 y_i^{(j)} + O(h^7),$$

where

$$\begin{aligned} k_0^6 &= 0; k_1^6 = -\frac{h^4}{6}\lambda_{i-1}\lambda_i(\lambda_{i-1}^2 + \lambda_i^2) + \frac{h^6}{40}\lambda_{i-1}\lambda_i(\lambda_{i-1}^4 - \lambda_i^4) + O(h^7); k_2^6 &= \frac{h^5}{24}\lambda_{i-1}\lambda_i(\lambda_{i-1}^2 - \lambda_i^2) + O(h^7); \\ k_3^6 &= -\frac{h^6}{6}\lambda_{i-1}\lambda_i(\lambda_{i-1}^2 + \lambda_i^2) + O(h^7); k_4^6 &= \frac{h^5}{12}\lambda_{i-1}\lambda_i + O(h^7); k_5^6 &= k_6^6 = 0 + O(h^7). \end{aligned}$$

The truncation error of (15) is

$$t_i^7 = \sum_{j=0}^6 k_j^7 y_i^{(j)} + O(h^7),$$

where $k_0^7 = 0; k_1^7 = -\frac{h^4}{12}\lambda_{i-1}^3 + \frac{h^6}{80}\lambda_{i-1}^5 + O(h^7); k_2^7 = \frac{h^5}{48}\lambda_{i-1}^3 + O(h^7); k_3^7 = \frac{h^6}{288}\lambda_{i-1}^3 + O(h^7); k_4^7 = 0$ $\frac{h^5}{24}\lambda_{i-1} + O(h^7); k_5^7 = k_6^7 = 0 + O(h^7).$

The truncation error of (16) is

$$t_i^8 = \sum_{j=0}^6 k_j^8 y_i^{(j)} + O(h^7),$$

where $k_0^8 = 0; k_1^8 = -\frac{h^4}{6}\lambda_i^3 - \frac{h^6}{40}\lambda_i^5 + O(h^7); k_2^8 = -\frac{h^5}{24}\lambda_i^3 + O(h^7); k_3^8 = \frac{h^4}{18}\lambda_i^3 - \frac{h^6}{16}\lambda_i^3 + \frac{h^6}{360}\lambda_i^5 + O(h^7); k_4^8 = \frac{h^5}{12}\lambda_i + O(h^7); k_5^8 = k_6^8 = 0 + O(h^7).$ The truncation error of (17) is

$$t_i^9 = \sum_{j=0}^6 k_j^9 y_i^{(j)} + O(h^7),$$

where $k_0^9 = 0; k_1^9 = \frac{h^4}{12}\lambda_i^3 - \frac{h^6}{80}\lambda_i^5 + O(h^7); k_2^9 = \frac{h^5}{48}\lambda_i^3 + O(h^7); k_3^9 = \frac{h^6}{288}\lambda_i^3 + O(h^7); k_4^9 = \frac{h^5}{24}\lambda_i + O(h^7); k_5^9 = k_6^9 = 0 + O(h^7).$

To summarize, the local truncation errors of the proposed method is $O(h^4)$.

5. Numerical Examples

In this section, some numerical examples are given to illustrate the effectiveness of the proposed method. We begin with two definitions to measure the effectiveness of the proposed method.

The maximum absolute error [14] is computed by

$$e^{\infty} = \max_{0 \le i \le n} |\tilde{y}(x_i) - y(x_i)|,$$

where $\tilde{y}(x_i)$ and $y(x_i)$ represent the numerical and exact solution at the node x_i , respectively.

The rate of convergence [14,15] is computed by

$$\rho = \frac{\ln(e_1^\infty/e_2^\infty)}{\ln(n_2/n_1)},$$

where e_1^{∞} and e_2^{∞} are the maximum absolute errors estimated for two different grids with $n_1 + 1$ and $n_2 + 1$ points.

Example 1. Consider the linear differential equation with variable coefficients:

$$y'' + xy = (x^2 + 1)e^x, -2 \le x \le -1.$$
 (20)

with boundary conditions $y(-2) = -3e^{-2}$, $y(-1) = -2e^{-1}$.

Example 2. Consider the linear differential equation with variable coefficients:

$$y'' + (\sin x)y = 2\cos x - x\sin x + x\sin^2 x, -\frac{\pi}{6} \le x \le \frac{\pi}{6}.$$
 (21)

with boundary conditions $y(-\frac{\pi}{6}) = y(\frac{\pi}{6}) = \frac{\pi}{12}$.

Example 3. Consider the linear differential equation with constant coefficients:

$$y'' + y = -1, 0 \le x \le 1.$$
⁽²²⁾

with boundary conditions $y(0) = 0, y(1) = \cos 1 - 1$.

The analytic solutions of (20)–(22) are $y = (x - 1)e^x$, $y = x \sin x$ and $y = \cos x - 1$, respectively. In Table 1, we list the approximate and exact solutions to (20). In Table 2, we show the maximum errors in Examples 1–3 and the rate of convergence in Examples 1 and 2. From Table 2, we can observe that the order of convergence is 2. It is evident from Tables 1 and 2 that the proposed method produces high precision numerical solutions. Especially in Example 3, the errors of numerical solutions are close to the machine accuracy. This is because the approximate Function (5) is constructed based on the analytic solutions to the linear differential equation with constant coefficients, and Equation (22) happens to be the one with constant coefficients. It is not surprising that we can yield the nearly exact solutions in Example 3.

Table 1. Comparison of the numerical solutions with the exact solutions in Example 1 ($h = 2^{-3}$).

x	$\tilde{y}(x)$	y(x)	x	$\tilde{y}(x)$	y(x)	x	$\tilde{y}(x)$	y(x)
-2.000	-0.406006	-0.406006	-1.625	-0.516883	-0.516893	-1.250	-0.644628	-0.644636
-1.875	-0.440891	-0.440896	-1.500	-0.557815	-0.557825	-1.125	-0.689882	-0.689886
-1.750	-0.477870	-0.477878	-1.375	-0.600484	-0.600494	-1.000	-0.735759	-0.735759

Table 2. Numerical results in Examples 1–3.

Example 1				Example 2			Example 3		
h	e∞	ρ	n	e∞	ρ	n	e^{∞}		
2^{-2}	4.327×10^{-5}	_	8	$3.785 imes 10^{-4}$	_	4	2.012×10^{-16}		
2^{-4}	$2.673 imes 10^{-6}$	2.0084	32	$2.390 imes 10^{-5}$	1.9926	16	$2.317 imes 10^{-15}$		
2^{-6}	$1.672 imes10^{-7}$	2.0039	128	$1.494 imes10^{-6}$	1.9962	64	$2.185 imes10^{-13}$		
2^{-8}	$1.045 imes10^{-8}$	2.0026	512	$9.343 imes10^{-8}$	1.9974	256	4.247×10^{-13}		

In Figure 1a, we show the exact solution and the numerical results obtained by the proposed method. The corresponding error curve are also shown in Figure 1b. These results coincide with the results listed in Table 2.



Figure 1. Numerical results in Example 2 (n = 32).



$$y'' + xy = (3 - x - x^2 + x^3)\sin x + 4x\cos x, 0 \le x \le 1.$$
(23)

with boundary conditions y'(0) = -1, $y'(1) = 2 \sin 1$.

The analytic solution of (23) is $y = (x^2 - 1) \sin x$. In Table 3, we list the maximum errors and the elapsed CPU time. The numerical results obtained by implementing the methods [3,5] are also listed for comparison. Note that the elapsed CPU time for the same routine in Matlab may be different every time. In order to count the time accurately, we took the average of all runtimes of 100 independent runs as the elapsed CPU time.

On one hand, our method is more accurate than Ramadan's method [3] and comparable to Srivastava's method [5]. On the other hand, the runtime of our method is less than that of Srivastava's method and comparable to Ramadan's method. This indicates that Srivastava's method can achieve the best precision in our experiment, while our method is superior in the term of elapsed CPU time. This is because we only need to solve a tridiagonal linear system instead of solving a penta-diagonal system in [5], which is much easier to implement.

n	Ramadan et al. [3]		Srivastava	a et al. [5]	Our Method		
	e^{∞}	Time(s)	e∞	Time(s)	e^{∞}	Time(s)	
4	$4.946 imes 10^{-2}$	$1.78 imes 10^{-4}$	$5.927 imes 10^{-3}$	$5.33 imes 10^{-3}$	$2.710 imes 10^{-3}$	$2.48 imes 10^{-4}$	
8	$1.231 imes 10^{-2}$	$1.89 imes10^{-4}$	$6.173 imes10^{-4}$	$5.34 imes10^{-3}$	$6.700 imes10^{-4}$	$2.62 imes10^{-4}$	
16	$3.081 imes 10^{-3}$	$2.61 imes 10^{-4}$	$1.797 imes10^{-4}$	$5.54 imes10^{-3}$	$1.670 imes10^{-4}$	$3.68 imes10^{-4}$	
32	$7.704 imes10^{-4}$	$3.43 imes 10^{-4}$	$3.495 imes 10^{-5}$	$5.72 imes 10^{-3}$	$4.183 imes10^{-5}$	$5.94 imes10^{-4}$	
64	$1.926 imes 10^{-4}$	$5.75 imes10^{-4}$	$6.702 imes 10^{-6}$	$6.28 imes 10^{-3}$	$1.046 imes 10^{-5}$	$1.10 imes 10^{-3}$	
128	$4.800 imes 10^{-5}$	$1.62 imes 10^{-3}$	$1.179 imes10^{-6}$	$7.36 imes 10^{-3}$	$2.615 imes 10^{-6}$	$2.25 imes 10^{-3}$	

Table 3. Maximum errors and elapsed CPU times in Example 4.

Clearly, the proposed method is efficient, therefore it provides a new approach to deal with second order linear differential equations.

6. Conclusions

Based on the analytic solutions to the linear differential equation with constant coefficients, we propose an numerical method for second order differential equations in this paper. The analytic solutions are used to construct the approximation function and the local truncation error is analyzed. Numerical examples have shown the effectiveness of the proposed method. Compared with the other methods, we only need to solve a tridiagonal system, which is much easier to implement.

Author Contributions: Methodology, X.H.; Resources, C.L.; Writing—original draft, C.L.; Writing—review and editing, J.L. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 11771453, Natural Science Foundation of Hunan Province, grant number 2020JJ5267 and Scientific Research Funds of Hunan Provincial Education Department, grant number 18C877.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- 1. Griffiths, D.F.; Higham, D.J. Numerical Methods for Ordinary Differential Equations; Springer: London, UK, 2010.
- 2. Usmani, R.A. The use of quartic splines in the numerical solution of a fourth-order boundary value problem. *J. Comput. Appl. Math.* **1992**, *44*, 187–200. [CrossRef]
- Ramadan, M.A.; Lashien, I.F.; Zahra, W.K. Polynomial and nonpolynomial spline approaches to the numerical solution of second order boundary value problems. *Appl. Math. Comput.* 2007, 184, 476–484. [CrossRef]
- 4. Ramadan, M.A.; Lashien, I.F.; Zahra, W.K. High order accuracy nonpolynomial spline solutions for 2μth order two point boundary value problems. *Appl. Math. Comput.* **2008**, 204, 920–927. [CrossRef]
- Srivastava, P.K.; Kumar, M.; Mohapatra, R.N. Quintic nonpolynomial spline method for the solution of a second order boundary-value problem with engineering applications. *Comput. Math. Appl.* 2011, 62, 1707–1714. [CrossRef]
- 6. 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. *Appl. Math. Comput.* **2007**, *179*, 153–160.
- 7. Zahra, W.K.; Mhlawy, A.M.E. Numerical solution of two-parameter singularly perturbed boundary value problems via exponential spline. *J. King Saud Univ. Sci.* **2013**, *25*, 201–208. [CrossRef]
- 8. Lodhi, R.K.; Mishra, H.K. Quintic B-spline method for solving second order linear and nonlinear singularly perturbed two-point boundary value problems. *J. Comput. Appl. Math.* **2017**, *319*, 170–187. [CrossRef]
- 9. Zahra, W.K. A smooth approximation based on exponential spline solutions for nonlinear fourth order two point boundary value problems. *Appl. Math. Comput.* **2011**, 217, 8447–8457. [CrossRef]
- 10. Al-Said, E.A.; Noor, M.A. Quartic spline method for solving fourth order obstacle boundary value problems. *J. Comput. Appl. Math.* **2002**, *143*, 107–116. [CrossRef]
- 11. Ideon, E.; Oja, P. Linear/linear rational spline collocation for linear boundary value problems. *J. Comput. Appl. Math.* **2014**, *263*, 32–44. [CrossRef]
- 12. Rashidinia, J.; Ghasemi, M. B-spline collocation for solution of two-point boundary value problems. *J. Comput. Appl. Math.* **2011**, 235, 2325–2342. [CrossRef]
- 13. Wanner, G. Solving Ordinary Differential Equations II; Springer: Berlin/Heidelberg, Germany, 1991.
- 14. Ahmad, G.I.; Shelly, A.; Kukreja, V.K. Cubic Hermite Collocation Method for Solving Boundary Value Problems with Dirichlet, Neumann, and Robin Conditions. *Int. J. Eng. Math.* **2014**, 2014, 365209.
- 15. Ge, Y.; Cao, F. Multigrid method based on the transformation-free HOC scheme on nonuniform grids for 2D convection diffusion problems. *J. Comput. Phys.* **2011**, 230, 4051–4070. [CrossRef]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).