Asian Journal of Advances in Research

4(1): 750-792, 2021

CUBIC SPLINE AND FINITE DIFFERENCE METHOD FOR SOLVING BOUNDARY VALUE PROBLEMS OF ORDINARY DIFFERENTIAL EQUATION

ABE NURA WARE[1] AND AHMED BUSERI ASHINE[2*]

[1]Department of Mathematics, College of Natural and Computational Sciences, Arsi University, Asella, Ethiopia.
[2]Department of Mathematics, College of Natural and Computational Sciences, Madda Walabu University, Bale Robe, Ethiopia.

AUTHORS' CONTRIBUTIONS

This work was carried out in collaboration between both authors. Both authors read and approved the final manuscript.

Received: 01 June 2021
Accepted: 05 August 2021
Published: 10 August 2021

Original Research Article

ABSTRACT

The paper is an over view of the theory of cubic spline interpolation and finite difference method for solving boundary value problems of ordinary differential equation. It specially focuses on cubic spline functions to develop a numerical method for the solution of second-order two-point boundary value problems. The resulting system of equations has been solved by a tri-diagonal solver. The two-point boundary value problem of differential equations, which is part of differential equations with conditions imposed at different points, has been applied in mathematics, engineering and various field of science. The rapid increasing on its applications has led to formulating and upgraded of several existing methods and new approaches. To describe a numerical method for solving the boundary value problem of ordinary differential equation with second-order by using cubic spline function. First, the polynomial and non-polynomial spline basis functions are introduced, and then we use the correlation between polynomial and non-polynomial spline basis functions to approximate the solution. Finally, we obtain the numerical solution by solving tri-diagonal systems. The results are compared with finite difference method, and cubic B- spline through examples which show that the cubic spline method is efficient and feasible. The absolute errors in test examples are estimated, and the comparison of approximate values, exact values, and absolute errors at the nodal points are shown graphically. Further, we have shown that non-polynomial spline produces accurate results in comparison with the results obtained by the B-spline method and finite difference method.

Keywords: *Cubic spline; Cubic B-spline interpolation method; Finite difference method; two-point boundary value problems; convergence analysis.*

1. INTRODUCTION

1.1 Background of the Study

Ordinary Differential Equations (ODEs) has a long history and widely applied in many fields. The numerical solution of ODE has great development in the 20th century. There have been emerged many new ideas as well as many complex methods for solving ODE, so that the numerical methods for solving ODE have been deepened.

Numerical methods have been introduced to generate approximation solution of problems as equation (1a)

**Corresponding author: Email: amebuseri@gmail.com;*

is difficult to be solved analytically. Finite difference method is used to solve linear and nonlinear boundary value problems and commonly used in explaining numerical methods in solving ordinary differential equations $[1-4]$.

Cubic spline interpolation method on approximating two-point boundary value problem, which had successfully led to many further investigations on application of spline in boundary value problems are introduced [5]. Part of them is [6, 7] that focused on the application of cubic spline interpolation on boundary value problems. A comparison on finite difference, finite element and finite volume methods applied to two-point boundary value problem had proposed [8]. Five years later, [9], had proposed to compare the performance of the techniques studied [8], with the new approach that developed as improved version of cubic spline interpolation in solving two-point boundary value problem [9]. The method is known as cubic B-spline interpolation. The word "spline" originally meant a thin wood or metal slat in east Anglian dialect. Work at General Motors resulted in a number of papers being published in the early 1960s, including some of the fundamental work B-splines [10].

In mathematics, a spline is numeric functions that is piecewise-designed by polynomials functions, and which possesses a sufficiently high degree of smoothness at the place where the polynomial piece connects (which are known as knots)[11,12].

In interpolating problems, spline interpolation is often preferred to polynomial interpolation because it yields similar results to interpolating with higher degree polynomials while avoiding instability due to Runge's phenomenon. In computer graphics, parametric curves whose coordinates are given by spline are popular because of the simplicity of their construction, their easy and accuracy of evaluation, and their capacity to approximate complex shapes through curve fitting and interactive curve design. The fundamental idea behind cubic spline interpolation is based on the engineer's tool used to draw smooth curves through a number of points. This spline consists of weight attached to a flat surface at the points to be connected. A flexible strip is then bent across each of these weights, resulting in a pleasingly smooth curve.

The mathematical spline is similar in principle. The points, in this case, are numerical data. The weights are the coefficients on the cubic polynomials used to interpolate the data. These coefficients bend the line so that it passes through each of the data points without any erratic behavior or breaks in continuity [13].

Systems of ordinary differential equation have been applied to many problems in physics, chemistry, engineering, biology and so on. The theory of spline functions is a very active field of approximation theory and boundary value problems (BVPs) when numerical aspects are considered. Piecewise cubic interpolation function is a cubic spline. The word "spline"originally meant a thin wood or metal slat in east Anglian dialect; by 1895 it had come to mean a flexible ruler used to draw curves [14]. These splines were used in the aircraft and shipbuilding industries. The physical spline minimizes potential energy subject to the interpolation constraints. The corresponding mathematical spline must have a continuous second derivative and satisfy the same interpolation constraints. The breakpoints of a spline are also referred to as its knots. The world of splines extends far beyond the basic one-dimensional, cubic, interpolators spline. There are multidimensional, high-order, variable knot, and approximating splines. A valuable expository and reference text for both the mathematics and the software is a practical Guide to spline [6]. Numerical data are usually difficult to analyze. For example, numerous data are obtained in the study of chemical reactions, and any function which would effectively correlate the data would be difficult to find. To this end, the idea of the cubic spline interpolation was developed. Using this process, a series of unique cubic polynomials is fitted between each of the data points, with the stipulation that the curve obtained be continuous and appears smooth. These cubic splines can then be used to determine rates of change and cumulative change over an interval [15].

The concept of splines originated from the mechanical drafting tool called " spline " used by designers for drawing smooth curves. These curves resemble cubic curves and hence the name "cubic spline" has been given to piecewise cubic interpolating polynomials.

In the mathematical field of numerical analysis, interpolation is a method of constructing new data points within the range of a discrete set of known data points. Interpolation provides a means of estimating of the value at the new data points within the range of parameters.

Cubic spline interpolation is a useful technique to interpolate between known data points due to its stability and smooth characteristics. Cubic splines are popular because of their ability to interpolate data with smooth curves. It is believed that a cubic polynomial spline always appears smooth to the eyes [16].

Spline interpolation is an alternative approach to data interpolation, compare to polynomial interpolation using on single formula to correlate all the data points, spline interpolation uses several formulas: each formula is a low degree polynomial to pass through all the data points. These resulting functions are called splines. Spline interpolation is preferred over polynomial interpolation because the interpolation error can be made small even when using low degree polynomials for the spline. One of subfield in mathematics, mathematical modeling is known with involvement of many problems related to boundary value problems [2]. Besides that, physical problems which involved position-dependent rather than time-dependent, are often described in differential equations with condition imposed at more than one point, which is mathematically represented as two-point boundary value problems [1]. In conjunction with the increasing on application of two-point boundary value problem, numbers of numerical methods have been introduced and upgraded, to generate a numerical method with the approximation solution. The second-order boundary value problems generally arise in the mathematical modeling of Viscoelastic flows. The second-order boundary value problems were investigated by finite difference, finite element and finite volume methods [8], and considered by means of B-spline interpolation method [9]. Moreover, the cubic spline methods were used and the obtained results produced improvements over other works [17]. However, the performance of the approaches used so far is well known in that it provides the solution at grid points only. We should point out that these approaches which were provided to solve this type of problems require a large amount of computational effort. The present work is motivated by the desire to obtain analytical and numerical solutions. The cubic spline method, cubic B-spline method, and finite difference method has been shown to solve effectively, easily and accurately a large class of linear, non- linear, stochastic or partial deterministic differential equations with approximate solutions which converge very rapidly to an accurate solution. This paper will concentrate on linear two-point boundary value problems, represented as

$$y''(x) + q(x)y'(x) + r(x)y(x) = f(x), \quad a \le x \le b \tag{1a}$$

With boundary conditions of $y(a) = \alpha$ and$y\ (b)\ = \beta$.

This type of boundary value problem is assumed to have a unique solution, $y(x)$ if

$q(x) \in C'[a,b], q > 0, and\ \ r(x), f(x) \in C[a,b]$ [9, 2, 18, 17, 8] and the parameters α and β are real constants.

1.2 Statement of the Problem

There are several different interpolation methods on accuracy, how expensive is the algorithm of implementation, smoothness of interpolation function, etc. Out of those methods, spline interpolations are chosen by the investigator. This is because of their ability to interpolate data with smooth curves. Moreover, in computer graphics, parametric whose coordinates are given by splines are popular because of simplicity of their construction, their ease and accuracy of evaluation, and their capacity to approximate complex shapes through curve fitting and interactive curve design and solving initial value problems and boundary value problems of ordinary differential equations [9, 5, 19, 18, 17, 8]. Moreover, this paper has tried to address the following basic questions.

1. How can we determine the unknown coefficients of cubic spline?
2. How can we approximate functions by splines?
3. How can we apply cubic splines to interpolate data with smooth curves?
4. Which method is better efficient for solving boundary value problems (BVPs) of linear ordinary differential equations (ODE)?
5. How can we apply splines and finite difference method for solving boundary value problems of ordinary differential equation?

1.3 Procedures and Methods

In this section, the paper methodology, secondary documents are treated. The researcher solved some boundary value problems of ordinary differential equation by iterations method in order to construct spline functions. With the comparing of the iteration and applying the developed software to the problems comes across the better approximation using cubic spline method. Finally, we considered the following computer

software which include applications in analysis of experimental data and that will do the work for us and produce the fit.

- ✓ MAT LAB
- ✓ Microsoft Excel

1.4 Procedures of the Study

After the necessary and relevant information have been collected from different sources such as books, articles and internet that are related to the topics of our research, the researcher used the following procedures to analysis.

- The cubic interpolation spline functions with their respective conditions were defined.
- The diagonal coefficient matrices were obtained.
- The coefficients were evaluated on their proper interval.
- Interpolating polynomial of degree $N-1$ was obtained.
- Description of finite difference method, B-spline method, and Cubic spline method for two-point boundary value problems.
- Using finite difference method, B-spline method, and cubic spline method the iteration was done.
- Discussing the consistency, stability, and convergence of cubic spline method.
- Discussing the scheme from the graphs and tables.
- Comparison of methods for the test equation.
- Writing the code for boundary value problems of ordinary differential equation we solved by MATLAB.

1.5 Instrumentation (Tools)

Determine coefficients of cubic spline we construct and showing graphs using MAT LAB applying cubic spline method, B-spline method, and finite difference method for solving boundary value problem of linear ordinary differential equation.

2. PRELIMINARIES AND FORMULATION OF THE METHODS

2.1 Cubic Spline Interpolation

The most common piecewise polynomial approximation uses cubic polynomials between each successive pairs of nodes is called cubic spline interpolation. A general cubic polynomial involves four constants, so there is sufficient flexibility in the cubic spline procedure to ensure that the interpolants is not only continuously differentiable on the interval, but also has a continuous second derivative. The goal of cubic spline interpolation is to get an interpolation formula that is continuous in both the first and second derivatives, both within the intervals and at the interpolating nodes. This will give us smoother interpolating functions. The continuity of first derivative means that the graph $y = S(x)$ will not sharp corners. The continuity of second derivative means that the radius of curvature is defined at each point. The construction of cubic spline does not, however, assume that the derivatives of the interpolant agree with those of the function it is approximating, even at the nodes [16]. When a function f defined on interval $[x_0, x_N]$ and a set of nodes $\{x_0, x_1, \dots, x_N\}$ such that $a = x_o, x_1, \dots, x_N = b$.

A cubic spline interpolation S for f is a function that satisfies the following conditions:

1. $S(x)$ is a cubic polynomial, denoted by $S_i(\mathrm{x})$ on subinterval $[x_i, x_{i+1}]$ for each $i = 0,1, \dots, N-1$;
2. $S(x_i) = f(x_{i+1})$for each $i = 0,1,2, \dots, N$
3. $S_{i+1}\,(\,x_{i+1}\,) = S_i\,(\,x_{i+1}\,)$ for each $i = 0,1,2, \dots, N-2; (S \in \mathrm{C}[a,b])$
4. $S_{i+1}{}'\,(\,x_{i+1}\,) = S_i{}'\,(\,x_{i+1}\,)$ for each $i = 0,1,2, \dots, N-2; (S \in \mathrm{C}^1[a,b])$
5. $S_{i+1}{}''\,(\,x_{i+1}\,) = S_i{}''\,(\,x_{i+1}\,)$ for each $i = 0,1,2, \dots, N-2;\ (S \in \mathrm{C}^2[a,b])$
6. One of the following sets of boundary conditions is satisfied:

$S''\,(\,x_0\,) = 0 = S''\,(\,x_N\,)$ for **free or natural boundary** and $S'(x_0) = f'(x_0)$, and $S'(x_N) = f'(x_N)$ for **Clamped boundary**

Remark: To construct the cubic spline interpolating$S$ for the function $f$this defined on the values [1].
Let $a = x_o < x_1 < \dots < x_N = b$.Satisfying $S''(x_o) = S''(x_N)$ and $S(x) = S_i(x) = a_i + b_i(x - x_i) + c_i(x - xi)2 + di(x - xi)3$ for $xi \le x \le xi+1$.

Picard's Theorem: If $f(x,y)$ and $\frac{\partial f}{\partial y}$ are both continuous on a closed rectangle R, then through each point (x_0, y_0) in the interior of R, then there exists a unique curve of the equation $\frac{dy}{dx} = f(x,y)$ that passes through it.

Definition: - A cubic spline $S(x)$ is a piecewise defined function that satisfies the following conditions:

1. $S(x) = S_i(\mathrm{x})$ is a cubic polynomial on each sub interval $[x_i, x_{i+1}]$ for $i = 0,1, \dots, N-1$
2. $S(x) = u_i$ for $i = 0, 1 \dots, N$ (S have to interpolate all the points)

3. $S(x), S^{'}(x)$, and $S''(x)$ are continuous on $[a,b](S$ is smooth). So, we write the m cubic polynomial pieces as:
$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad , \quad i = 0,1, \dots, N-1$
where $a_i, b_i, c_i,$ and d_i represent $4^{'}$N unknown coefficients.

2.2 Construction of Cubic Spline

How to determine the unknown coefficients a_i, $b_{i,}\, c_{i,}\ and\ d_i$ of the cubic spline $S(x)$ so that we can construct it? Given $S(x)$ is cubic spline that has all the properties as in the definition above, [14].

From cubic polynomial pieces between each data points, we have:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \ \ i = 0, 1 \dots, N-1 \tag{1}$$

The first and second derivatives:

$$S^{'}(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2 \tag{2}$$

$$S^{''}(x) \ = 2c_i + 6d_i(x - x_i) \tag{3}$$

$$\text{Let } S\,(x_i) = u_i \text{for } i = 1,2, \dots, N-1. \text{ since } x_i \in [x_i, x_{i+1}] \tag{4}$$

$$S(x_i) = S_i(x_i)$$
$$u_i = \ S_i(x_i)$$

$$u_i = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 u_i = a_i$$
$$\text{for each } i = 1, 2, \dots N-1 \tag{5}$$

From continuous properties of cubic spline method across each interval we have

$$S(x_i) = S_i(x_i)$$
$$S_i(x_i) = S_{i-1}(x_i) \text{ , for } i = 2,3, \dots, N \tag{6}$$

From (5) we have $S_i(x_i) = a_i$ and

$$S_{i-1}(x_i) = a_{i-1} + b_{i-1}(x - x_{i-1}) + c_{i-1}(x - x_{i-1})^2 + d_{i-1}(x - x_{i-1})^3$$
$$\text{So, } \ a_i = a_{i-1} + b_{i-1}(x - x_{i-1}) + c_{i-1}(x - x_{i-1})^2 + d_{i-1}(x - x_{i-1})^{\ 3}$$
$$\text{From } i = 2, 3, \dots, N-1 \text{ and } h = x - x_{i-1}$$
$$a_i = a_{i-1} + b_{i-1}h + c_{i-1}h^2 + d_{i-1}h^3 \tag{7}$$

To make a curve smooth across each interval, the derivative must be equal at the data points.

$$i.e. \ S_i{'}(x_i) = S_{i-1}{'}(x_i)$$
$$\Longrightarrow S_i{'}(x_i) = b_i \text{ and} \tag{8}$$

$$S_{i-1}{'}(x_i) = b_{i-1} + 2c_{i-1}(x_i - x_{i-1}) + 3d_{i-1}(x_i - x_{i-1})^2$$
$$b_i = b_{i-1} + 2c_{i-1}h^2 + 3d_{i-1}h^3 for\ i = 1,2, \dots, N-1 \tag{9}$$

From equation (3) $S_i{''}(x) = 6d_i(x - x_i) + 2c_i$

$$S_i{''}(x_i) = 2c_i \text{ For i= } 2,3, \dots, N-2 \tag{10}$$

Lastly, since $S_i{''}(x)$ has to be continuous across the interval,

$$S_i{''}(x_{i+1}) = 6d_i(x_{i+1} - x_i) + 2c_i$$
$$S_{i+1}{''}(x_{i+1}) = 6d_i(x_{i+1} - x_i) + 2c_i \tag{11}$$

And letting $h = x_{i+1} - x_i,$ using the conclusion from equation (10) and (11);

$$S_{i+1}{}''(x_{i+1}) = 6d_i(x_{i+1} - x_i) + 2c_i$$

$$2c_{i+1} = 6d_i h + 2c_i \tag{12}$$

The equation can be much simplified by substituting M_i for $S_i{}''(x_i)$ and expressing the above equation in terms of M_i and u_i.

This makes the determination the weights a_i, b_i, c_i, and d_i a much easier task. Each c_i can be represented by:

$$S_i{}''(x_i) = 2c_i \implies M_i = 2c_i \implies c_i = \frac{M_i}{2} \tag{13}$$

And a_i has already been determined to be $a_i = u_i$

Similarly using equation (12) d_i can be written as:

$$2c_{i+1} = 6d_i h + 2c_i \implies 6d_i h = 2c_{i+1} - 2c_i$$
$$\implies d_i = \frac{2c_{i+1} - 2c_i}{6h} = \frac{2(\frac{M_{i+1}}{2}) - 2(\frac{M_i}{2})}{6h}$$
$$d_i = \frac{M_{i+1} - M_i}{6h} \tag{14}$$

From equation (7) b_i can be written as:

$$a_{i+1} = a_i + b_i h + c_i h^2 + d_i h^3$$
$$\implies b_i h = -a_i - c_i h^2 - d_i h^3 + a_{i+1}$$
$$\implies b_i = \frac{-a_i - c_i h^2 - d_i h^3 + a_{i+1}}{h}$$
$$\implies b_i = \frac{u_{i+1} - u_i}{h} - \frac{h}{6}(M_{i+1} + 2M_i) \tag{15}$$

We now have our equation for determining the weights of our $N-1$ equations:

$$a_i = u_i\,, \quad b_i = \frac{u_{i+1} - u_i}{h} - \frac{h}{6}(M_{i+1} + 2M_i), c_i = \frac{M_i}{2}\,, d_i = \frac{M_{i+1} - M_i}{6h}, \tag{16}$$

These Systems can be handled more conveniently by putting them in Matrix form as follows

From (9), $b_{i+1} = b_i + 2c_i h + 3d_i h^2$ for $i = 1,2,\dots,N-1$
$$\implies 3d_i h^2 + 2c_i h = b_{i+1} - b_i \tag{17}$$

When we substitute the values of equation (16) in to (17) we have:

$$3(\frac{M_{i+1} - M_i}{6h})\, h^2 + 2(\frac{M_i}{2})\, h + \frac{u_{i+1} - u_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6}\right) h = \frac{u_{i+2} - u_{i+1}}{h} - \left(\frac{M_{i+2} + 2M_{i+1}}{6}\right) h$$

$$3(\frac{M_{i+1} - M_i}{6h})h^2 + 2(\frac{M_i}{2})h + \frac{u_{i+1} - u_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6}\right) h + \left(\frac{M_{i+2} + 2M_{i+1}}{6}\right) h = -\left(\frac{u_{i+1} - u_i}{h}\right) + \frac{u_{i+2} - u_{i+1}}{h}$$
$$h\,(\frac{3M_{i+1} - 3M_i}{6}) + \frac{6M_i}{6} - \left(\frac{M_{i+1} + 2M_i}{6}\right) + \left(\frac{M_{i+2} + 2M_{i+1}}{6}\right) = \frac{u_i - 2u_{i+1} + u_{i+2}}{h}$$
$$\frac{h}{6}(M_i + 4M_{i+1} + M_{i+2}) = \frac{u_i - 2u_{i+1} + u_{i+2}}{h}$$
$$M_i + 4M_{i+1} + M_{i+2} = \frac{6}{h^2}(u_i - 2u_{i+1} + u_{i+2}) \text{ for } i = 1,2,\dots,N-1 \tag{18}$$

This leads to the Matrix Equation:

$$\begin{bmatrix} 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{N-3} \\ M_{N-2} \\ M_{N-1} \\ M_N \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} u_1 - 2u_2 + u_3 \\ u_2 - 2u_3 + u_4 \\ u_3 - 2u_4 + u_5 \\ \vdots \\ u_{N-4} - 2u_{N-3} + u_{N-2} \\ u_{N-3} - 2u_{N-2} + u_{N-1} \\ u_{N-2} - 2u_{N-1} + u_N \end{bmatrix} \quad \text{for}$$

$$i = 1,2, \dots, N-1 \tag{19}$$

Note that this system has $N-2$ rows and N columns, it is under-determined. In order to construct a unique cubic spline, two other conditions must be imposed up on the system.

2.3 Cubic Spline Interpolation Types

2.3.1 Natural or free spline

This spline type includes the stipulation that the second derivative be equal to zero at the endpoints.

$$M_1 = M_N = 0 \quad (20)$$

This results in the extending as a line outside the end points. Since this provides a boundary condition that completes the system of $N-2$ equations.

This produces a so-called "natural" cubic spline and leads to a simple tri-diagonal system which can be solved easily to give the coefficients of the polynomials.
The matrix for determining the $M_1 - M_N$ values can be adapted accordingly.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{N-3} \\ M_{N-2} \\ M_{N-1} \\ 0 \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} u_1 - 2u_2 + u_3 \\ u_2 - 2u_3 + u_4 \\ u_3 - 2u_4 + u_5 \\ \vdots \\ u_{N-4} - 2u_{N-3} + u_{N-2} \\ u_{N-3} - 2u_{N-2} + u_{N-1} \\ u_{N-2} - 2u_{N-1} + u_N \end{bmatrix} \tag{21}$$

For reasons of convenience, the first and last columns of this matrix can be eliminated as they correspond to the M_1, M_N values, which are both 0.

$$\begin{bmatrix} 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & 4 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{N-3} \\ M_{N-2} \\ M_{N-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} u_1 - 2u_2 + u_3 \\ u_2 - 2u_3 + u_4 \\ u_3 - 2u_4 + u_5 \\ \vdots \\ u_{N-4} - 2u_{N-3} + u_{N-2} \\ u_{N-3} - 2u_{N-2} + u_{N-1} \\ u_{N-2} - 2u_{N-1} + u_N \end{bmatrix} \tag{22}$$

This results in an $N-2$ by $N-2$ matrix, which will determine the remaining solutions for M_2 through $M_{N-1}$the spline is now unique.

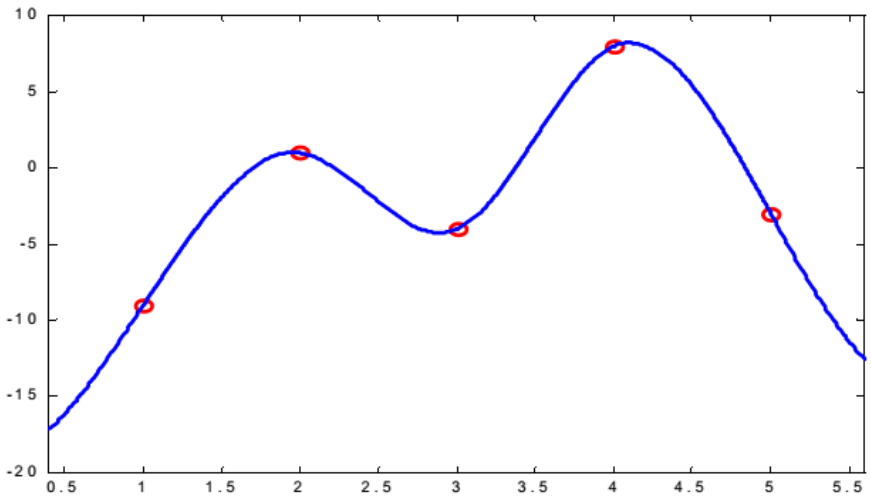


Fig. 1. Natural interpolating curves

To construct the cubic spline interpolating S for the function f, denoted at the numbers $a = x_o < x_1 < \ldots < x_N = b$ Satisfying $S''(x_o) = S''(x_N) = 0$:

INPUT N; $x_o, x_1, \ldots, x_N$; $a_o = f(x_o), a_1 = f(x_1) \ldots a_N = f(x_N)$.
OUTPUT a_i, b_i, c_i, d_i, *for* $i = 0,1, \ldots, N-1$
(*Note:* $S(x) = S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ for $x_i \le x \le x_{i+1}$.)

Step 1 for $i = 0,1, \ldots, N-1$ set $h_i = x_i \le x \le x_{i+1}$
Step 2 for $i = 1,2, \ldots, N-2$ set $\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$.
Step 3 set $l_o = 1$; $\mu_o = 0$; $z_o = 0$.
Step 4 for $i = 1,2, \ldots, N-1$ set $l_i = 2(x_{i+1} - x_i) - h_{i-1}\mu_{i-1}$; $\mu_i = \frac{h_i}{l_i}$; $z_i = \frac{\alpha_i - h_{i-1}z_{i-1}}{l_i}$.
Step 5 set $l_N = 1$; $z_N = 0$; $c_N = 0$.
Step 6 for $i = N-1, N-2, \ldots, 0$ Set $c_i = z_i - \mu_i c_{i+1}$;
$b_i = \frac{a_{i+1} - a_i}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i)$, $d_i = \frac{(c_{i+1} - c_i);}{3h_i}$.
Step 7 OUTPUTS $(a_i, b_i, c_i, d_i$, *for* $i = 0,1, \ldots, N-1)$;
STOP.

2.3.2 Clamped cubic spline

This type of spline is set end point first derivatives $S'_1(x_0)$ and $S'_N(x_N)$ to certain value, thus the slope at the beginning and end of the spline are under user's control.

For example, data points $(0, 0)$, (1,0.5), (2, 1.8), (3,1.5) and $S'_1(x_0) = 0.5$,
$S'_N(x_N) = 0.5$

Construct the unique cubic spline (namely, Clamped spline) for the data points (see Fig. 2.)

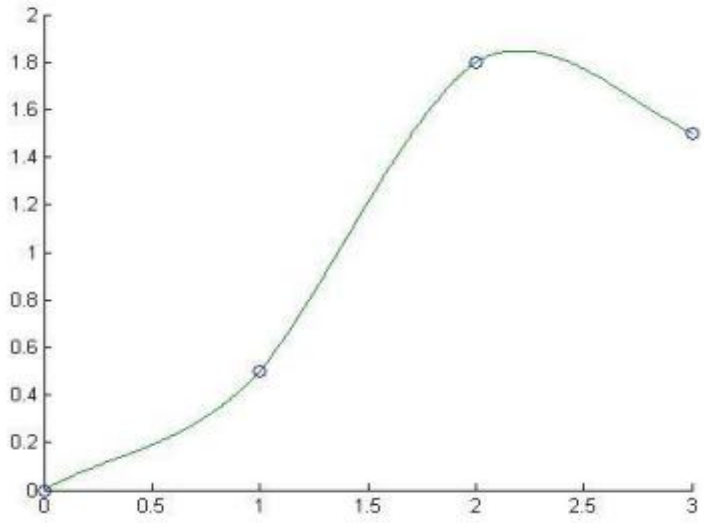


Fig. 2. Clamped cubic spline

To construct the cubic spline interpolating S for the function f, denoted at the numbers $a = x_0 < x_1 < \ldots < x_N = b$ Satisfying $S'(x_0) = f'(x_0)$:

INPUT N; $x_0, x_1, \ldots, x_N$;
$a_o = f(x_0), a_1 = f(x_1), \ldots, a_N = f(x_N)$;
$FPO = f'(x_o)$; $FPN = f'(x_N)$
OUTPUT a_i, b_i, c_i, d_i, for $i = 0,1, \ldots, N-1$
(*Note:* $S(x) = S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ for $x_i \le x \le x_{i+1}$.)

Step 1 for $i = 0,1, \ldots, N-1$ set $h_i = x_i \le x \le x_{i+1}$
Step 2 set $\alpha_o = \frac{3}{h_o}(a_1 - a_o) - 3FPO$; $\alpha_N = 3FPN - \frac{3}{h_{N-1}}(a_N - a_{N-1})$
Step 3 for $i = 1,2, \ldots, N-1$ set $\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$.
Step 4 set $l_o = 2h_o$; $\mu_o = 0.5$; $z_o = \frac{\alpha_o}{l_o}$.
Step 5 for $i = 1,2, \ldots, N-1$ set $l_i = 2(x_{i+1} - x_i) - h_{i-1}\mu_{i-1}$; $\mu_i = \frac{h_i}{l_i}$;
$z_i = \frac{\alpha_i - h_{i-1}z_{i-1}}{l_i}$.
Step 6set $l_N = h_{N-1}(2 - \mu_{N-1})$; $z_N = \frac{\alpha_N - h_{N-1}z_{N-1}}{l_N}$; $C_N = Z_N$.
Step7 for $i = N-1, N-2, \ldots, 0$ Set $c_i = z_i - \mu_i c_{i+1}$;
$b_i = \frac{a_{i+1} - a_i}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i)$;
$d_i = \frac{(c_{i+1} - 2c_i)}{3h_i}$.
Step 8 OUTPUTS (a_i, b_i, c_i, d_i, for $i = 0,1, \ldots, N-1$);
STOP.

2.3.3 Parabolic run out spline

The parabolic spline imposes the condition that the second derivative at the endpoints,

M_1 and M_N,be equal to M_2 and M_{N-1} respectively.

$$M_1 = M_2$$
$$M_N = M_{N-1} \quad (23)$$

The result of this condition is the curve becomes a parabolic at the endpoints. This type of cubic spline is useful for periodic and exponential data.

The matrix equation for this type of spline is

$$\begin{bmatrix} 5 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{N-3} \\ M_{N-2} \\ M_{N-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} u_1 - 2u_2 + u_3 \\ u_2 - 2u_3 + u_4 \\ u_3 - 2u_4 + u_5 \\ \vdots \\ u_{N-4} - 2u_{N-3} + u_{N-2} \\ u_{N-3} - 2u_{N-2} + u_{N-1} \\ u_{N-2} - 2u_{N-1} + u_N \end{bmatrix} \tag{24}$$

We can now determine the values of M_2 through M_{N-1}, with the values for M_1 *and* M_N already determined.

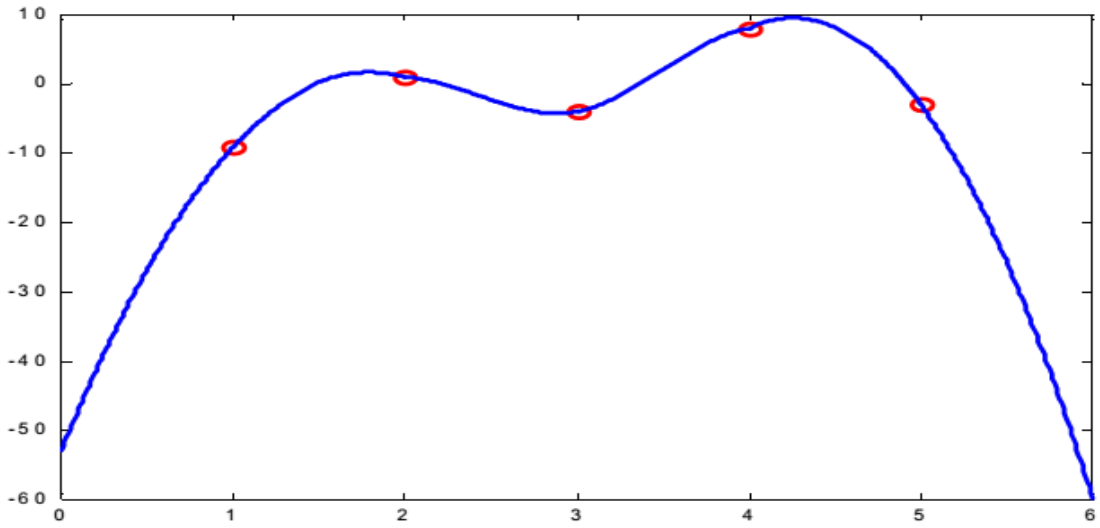


Fig. 3. Parabolic Runout curves

Note that the endpoint behavior is a bit more extreme than with the natural spline option.

2.3.4 Cubic run out spline

This type of spline has the most extreme endpoint behavior. It assigns M_1 to be

$2M_2 - M_3$ and M_N to be $2M_{N-1} - M_{N-2}$. This causes the curve to degrade to single cubic curve over the last two intervals, rather than two separate functions.

The matrix equation for this type is

$$\begin{bmatrix} 6 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} M_2 \\ M_3 \\ M_4 \\ \vdots \\ M_{N-3} \\ M_{N-2} \\ M_{N-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} u_1 - 2u_2 + u_3 \\ u_2 - 2u_3 + u_4 \\ u_3 - 2u_4 + u_5 \\ \vdots \\ u_{N-4} - 2u_{N-3} + u_{N-2} \\ u_{N-3} - 2u_{N-2} + u_{N-1} \\ u_{N-2} - 2u_{N-1} + u_N \end{bmatrix} \tag{25}$$

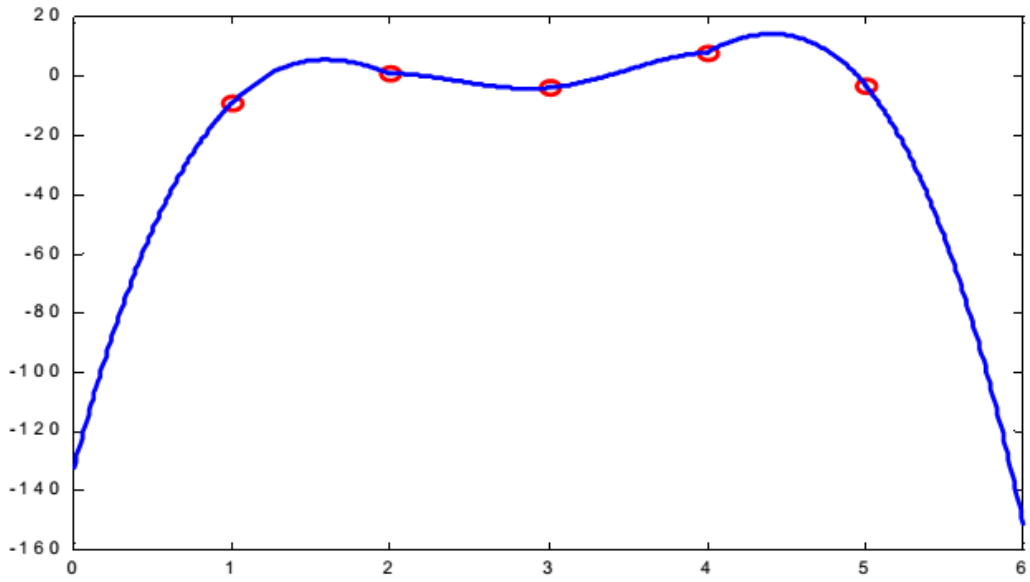


Fig. 4. Cubic Runout curve

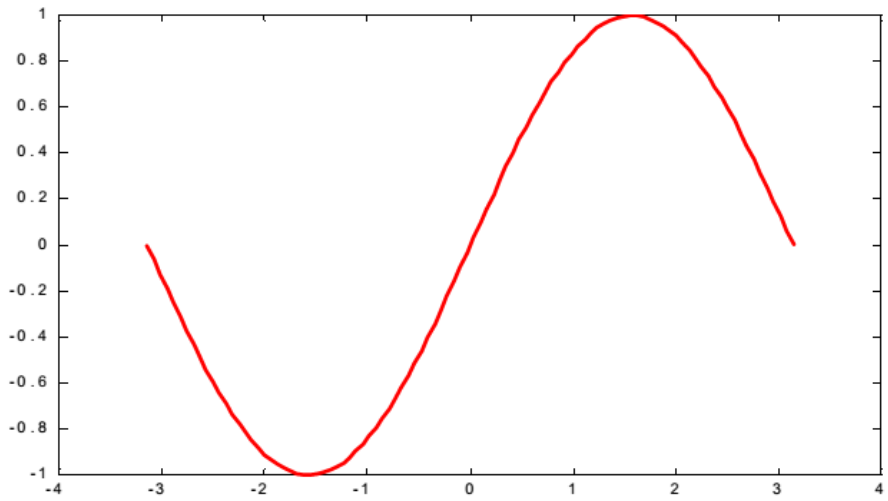


Fig. 5. The graph of $y = \sin(x)$

Note the Pronounced curvature at the endpoints.

Keep in mind that there are many other types of interpolation spline curves, such as the Curvature-adjusted cubic spline, End slope cubic spline and Periodic spline. The four discussed in this work are simply the ones which have chosen to examine; they are not intrinsically superior, or more widely used than these other types of splines

2.3.5 Curve fitting with splines

The main application of cubic spline interpolation techniques is, of course, curve fitting. To this end, the consistency and efficiency of the spline as a data correlation tool will be demonstrated.

2.4 Function Imitation

Cubic splines would not be necessary were it simply to determine a well-behaved function to fit any data set. This is, however, usually not the case. Thus, the cubic spline technique is used to generate a function to fit the data. Moreover, it can be shown that data generated by a particular function is interpolated by a spline which behaves more or less like the original function. This is testimony to the consistency of splines.

For Example, the following figure was generated using the function $y = \sin(x)$.

This next figure was generated by connecting six data points along the above line with a cubic spline function.

By superimposing **Fig. 6** on **Fig. 5**, we can see the degree to which the cubic curve imitates the original function $y = \sin(x)$.

Clearly, the cubic curve closely imitates the sine curve. It has no extreme behavior between data points, and it effectively correlates the points. This characteristic also works with erratic functions. Take for instance the function below.

While the fit is not perfect, it does closely approximate the function without a great degree of divergence.

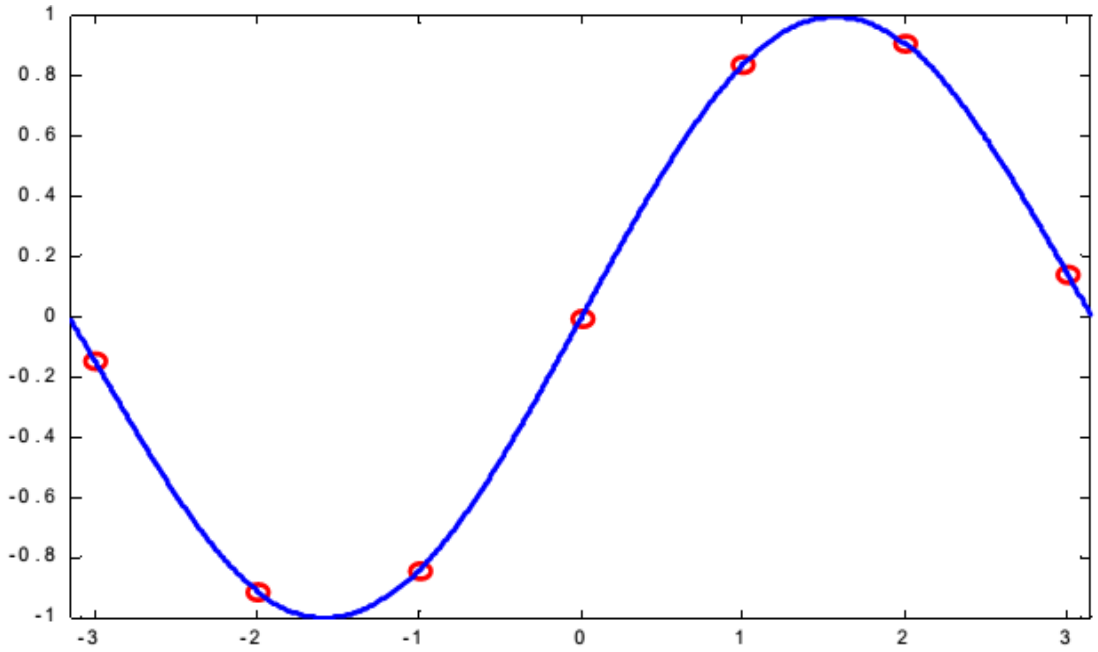


Fig. 6. Spline through six points on a sine curve

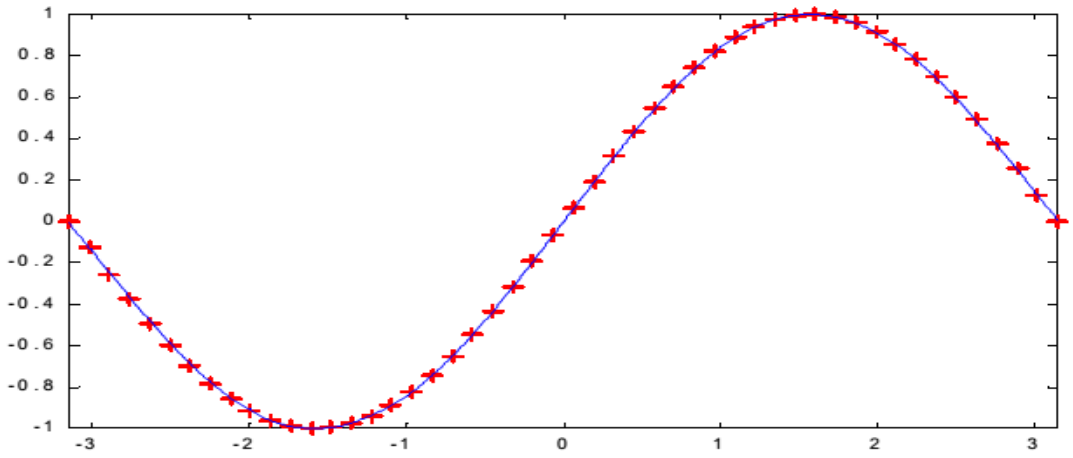


Fig. 7. Superimposition of a spline curve on a sine function

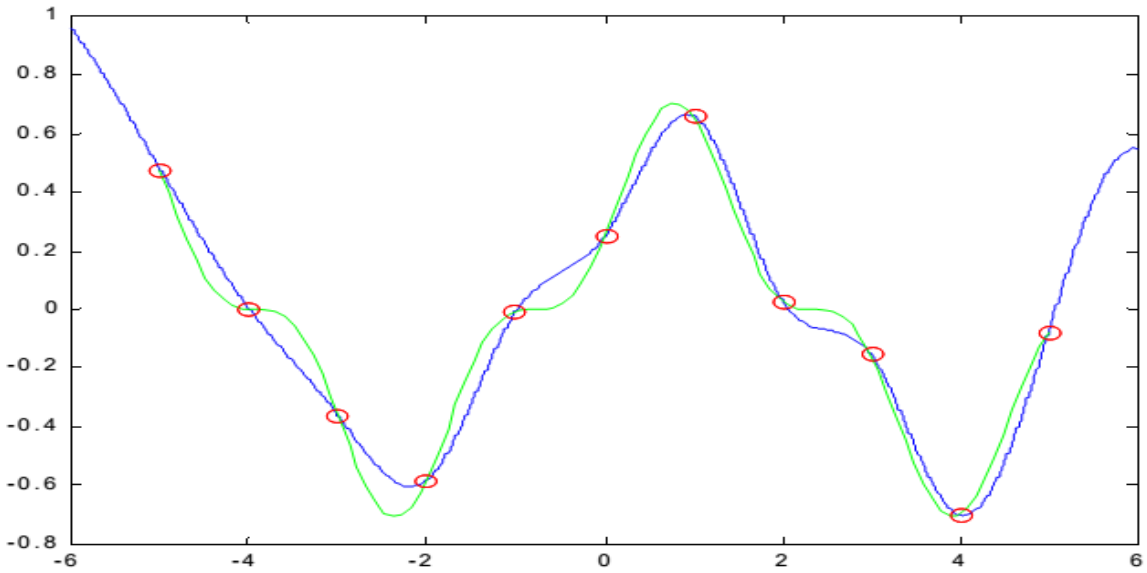


Fig. 8. A cubic spline approximation of $[\sin(x) + \cos(x)]^{\frac{3}{4}}$.

2.5 Data Correlation and Interpolation

Data interpolation is widely used in real world, especially with large volume of irregular data. Find polynomial functions guide these data within certain interval, can greatly help with data analysis and data predicting. Cubic splines could be used for finding an interpolation to correlate data. The other spline strength lies in its ability to correlate data which doesn't follow any specific pattern without a single polynomial's extreme behavior. Take, for instance, the 100 random points below.

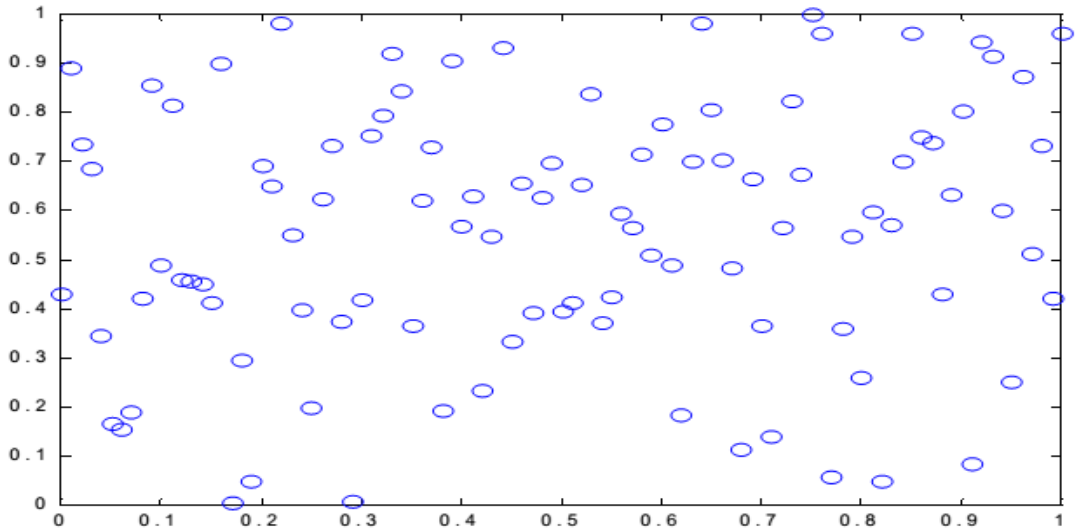


Fig. 9. Random data points

Clearly there is no relation between these data points. A spline, however, can interpolate all 100 points without the drastic behavior that the necessary 99^{th} degree polynomial would exhibit.

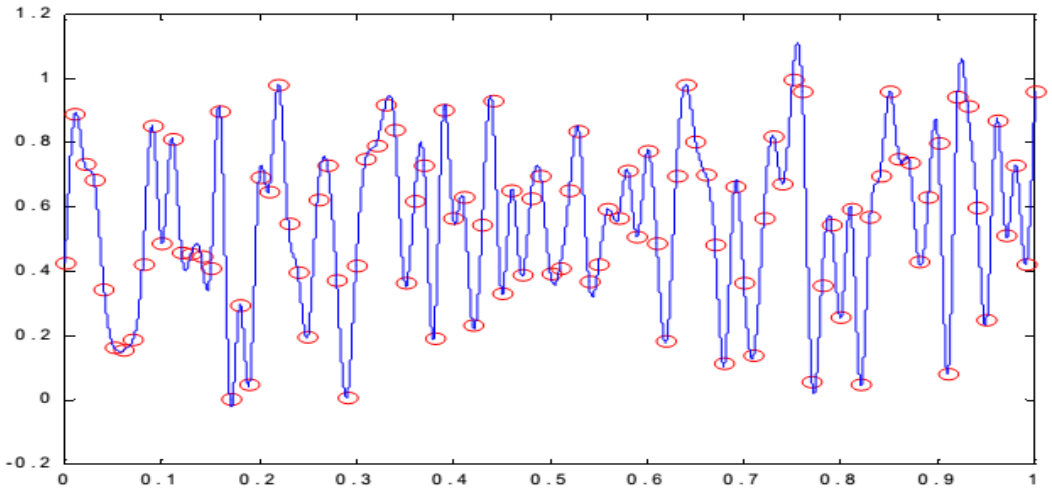


Fig. 10. Cubic curves through random data

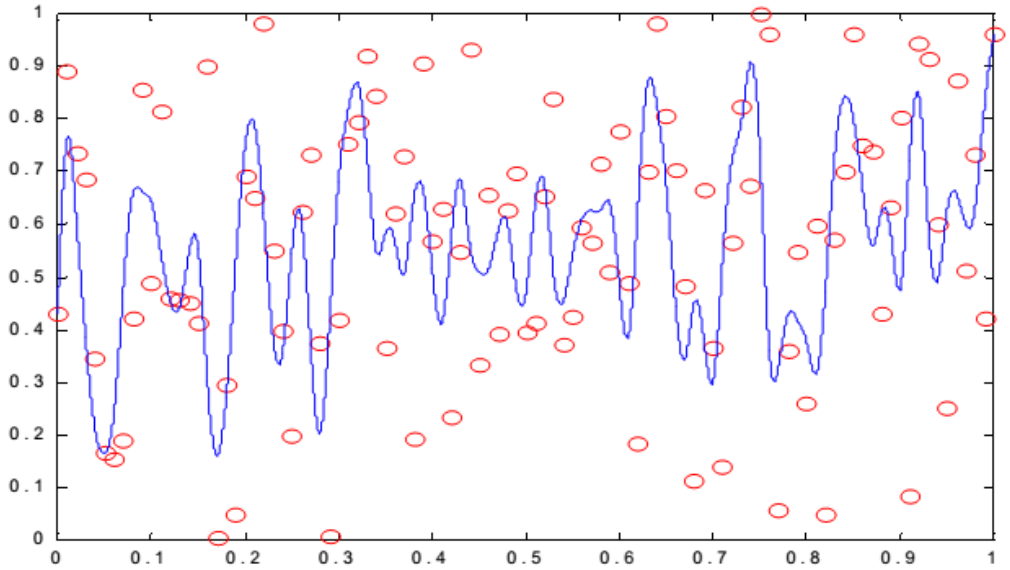


Fig. 11. Approximating cubic curves

The above figure demonstrates the general trends in the data (of which there are none) without necessarily connecting all data points. Its curvature is much severe that of **Fig. 10**

For example, in the chemical experiment, the following data was obtained (see arrays t and G below and **Fig.12**):

$$t = [0\ \ 0.1\ \ 0.499\ \ 0.5\ \ 0.6\ \ 1.0\ \ 1.4\ \ 1.5\ \ 1.899\ \ 1.9\ \ 2.0]G$$
$$= [0\ \ 0.06\ \ 0.17\ \ 0.19\ \ 0.21\ \ 0.26\ \ 0.29\ \ 0.29\ \ 0.30\ \ 0.31\ \ 0.31]$$

We need to estimate the value of G when $t = 1.2$. Using the natural Cubic spline interpolation (the formula $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ on $[x_i, x_{i+1}]$), we found $G = 0.27527649$.

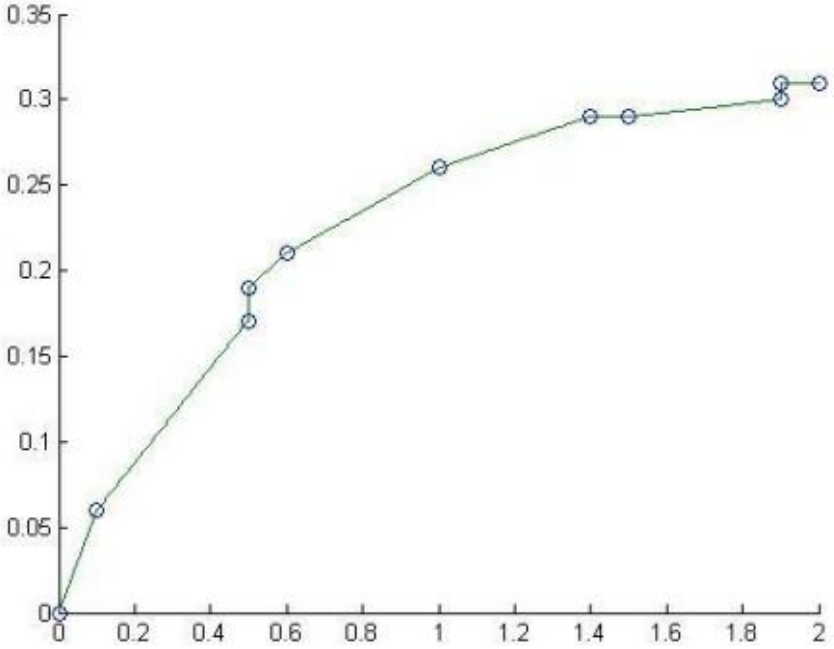


Fig. 12. The Chemical Experiment Data. $G = f(t)$

Example: Consider the polynomial function. $g(x) = \frac{\sin(x)}{x^2}$On the interval $[\frac{\pi}{4}, \frac{3\pi}{2}]$ we can take few coupled data points: $(\frac{\pi}{4}, 1.1463)$, $(\frac{\pi}{2}, 0.4053)$, $(\frac{3\pi}{4}, 0.1274)$, $(\pi, 0)$, $(\frac{5\pi}{4}, -0.0459)$, $(\frac{3\pi}{4}, -0.0450)$. With natural Cubic spline, the interpolating function is as a solid line in **Fig. 13** below.

The exact values of the polynomial function $g(x)$ (*) are also shown in **Fig. 13**. Comparing the results, we can see that the cubic spline interpolating function approximates the exact function with a good accuracy.

Example: Use the data points $(0, 1), (1, e), (2, e^2)$, and $(3, e^3)$ to form a natural spline $S(x)$ that approximate $f(x) = e^x$ and evaluate $f(2.5)$.

Solution: -We have $n = 3, h_o = h_1 = h_2 = 1, a_o = 1, a_1 = e, a_2 = e^2, a_3 = e^3$.So the matrix A and Vectors b and x given below

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 3(e^2 - 2e + 1) \\ 3(e^3 - 2e^2 + 1) \\ 0 \end{bmatrix} \qquad \text{and } x = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

The Vector equation $Ax = b$ is equivalent to the system of equations

$$\begin{aligned} c_0 &= 0 \\ c_0 + 4c_1 + c_2 &= 3(e^2 - 2e + 1), \\ c_1 + 4c_2 + c_3 &= 3(e^3 - 2e^2 + e), \\ c_3 &= 0 \end{aligned}$$

This system has the solution $c_o = c_3 = 0$, and to decimal places

$c_1 = \frac{1}{5}(-e^3 + 6e^2 - 9e + 4) \approx 0.75685$, and
$c_2 = \frac{1}{5}(4e^3 - 9e^2 + 6e - 1) \approx 5.83007$,

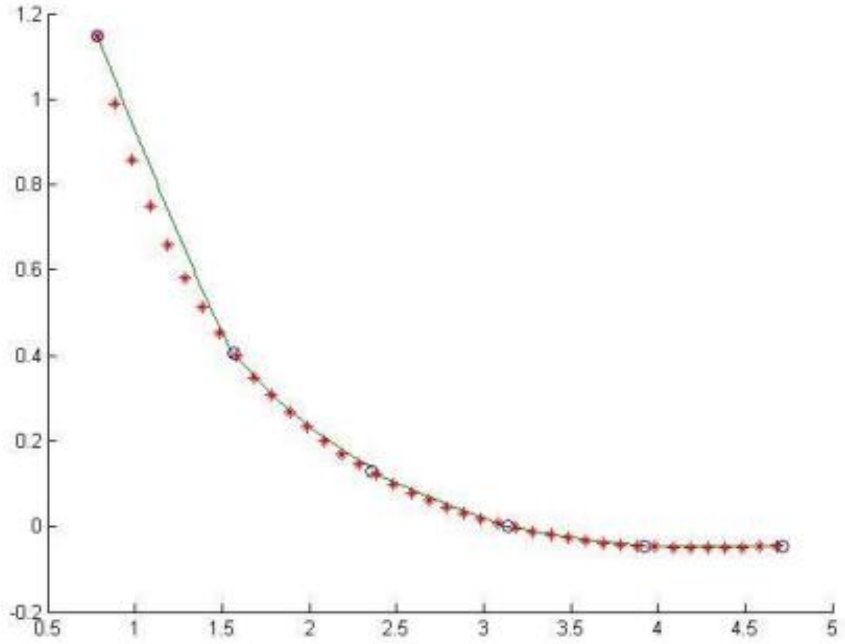


Fig. 13. Comparing the natural cubic spline interpolating function and the exact solution

Solving for the remaining constants gives

$$b_o = \frac{1}{h_o}(a_1 - a_o) - \frac{h_o}{3}(c_1 + 2c_o) = (e-1) - \frac{1}{15}(-e^3 + 6e^2 - 9e + 4) \approx 1.46600,$$
$$b_1 = \frac{1}{h_1}(a_2 - a_1) - \frac{h_1}{3}(c_2 + 2c_1) = (e^2 - e) - \frac{1}{15}(2e^3 + 3e^2 - 12e + 7) \approx 2.22285,$$
$$b_2 = \frac{1}{h_2}(a_3 - a_2) - \frac{h_2}{3}(c_3 + 2c_2)$$
$$= (e^3 - e^2) - \frac{1}{15}(-8e^3 + 18e^2 + 12e - 2) \approx 8.80977,$$
$$d_o = \frac{1}{3h_o}(c_1 - c_0) = \frac{1}{15}(-e^3 + 6e^2 - 9e + 4) \approx 0.25228,$$
$$d_1 = \frac{1}{3h_1}(c_2 - c_1) = \frac{1}{3}(e^3 - 3e^2 + 3e + 1) \approx 1.69007,$$

And $d_2 = \frac{1}{3h_2}(c_3 - c_2) = \frac{1}{15}(-4e^3 + 9e^2 - 6e + 1) \approx -1.94336.$

The natural cubic spline is described piecewise by

$$S(x) = \begin{cases} 1 + 1.46600x + 0.25228x^3, & for\ x \in [0,1] \\ 2.71828 + 2.22285(x-1) + 0.75685(x-1)^2 + 1.69107(x-1)^3, & for x \in [1,2] \\ 7.38906 + 8.80977(x-2) + 5.83007(x-2)^2 - 1.94336(x-2)^3, & for x \in [2,3] \end{cases}$$

since $x = 2.5$ is in [16,21],

$$S(2.5) = 7.38906 + 8.80977(\mathbf{2.5} - 2) + \mathbf{5.83007}(\mathbf{2.5} - \mathbf{2})^{\mathbf{2}} - \mathbf{1.94336}(\mathbf{2.5} - \mathbf{2})^{\mathbf{3}} \approx \mathbf{13.0085}\textbf{(True} \approx \mathbf{12.1825)}$$

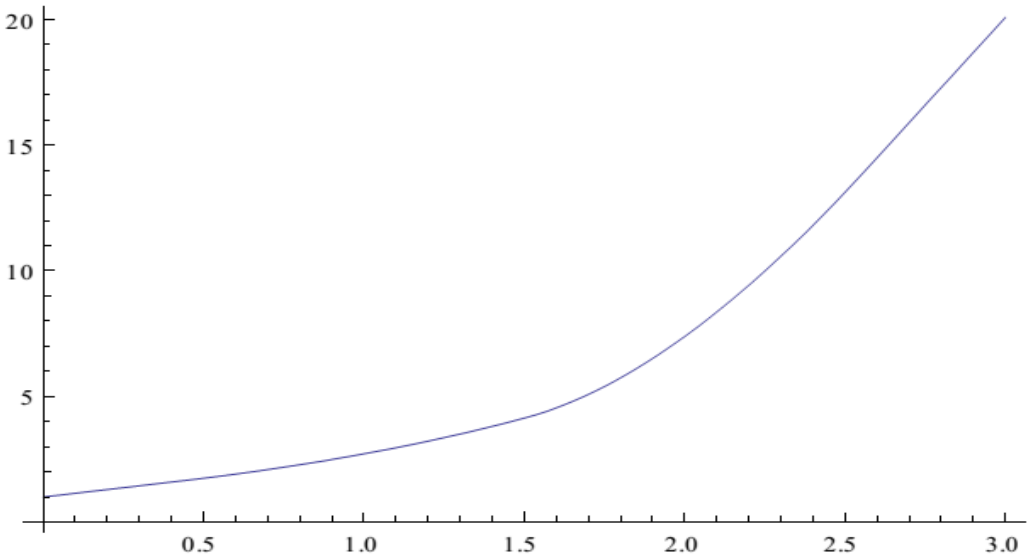


Fig. 14. Graphical representation of $y = e^x$

Example: use the data points $(-\pi, -1)$, $(\frac{\pi}{2}, 0)$, $(0, 1)$, $(\frac{\pi}{2}, 0)$ to form a natural spline that approximation $f(x)=\cos(x)$.

Solution: We have N=3, $h_o = h_1 = h_2 = \frac{\pi}{2} a_o = -1 a_1 = 0 a_2 = 1 a_3 = 0$.So the matrix A and the vectors b and x given below.

$$A = \begin{bmatrix} 1.57 & 0 & 0 & 0 \\ 1.57 & 6.28 & 1.57 & 0 \\ 0 & 1.57 & 6.28 & 1.57 \\ 0 & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 0 \\ \frac{3}{\frac{\pi}{2}}(1-0) - \frac{3}{\frac{\pi}{2}}(0-(-1)) \\ \frac{3}{\frac{\pi}{2}}(0-1) - \frac{3}{\frac{\pi}{2}}(1-0) \\ 0 \end{bmatrix} \text{ and } x = \begin{bmatrix} c_o \\ c_1 \\ c_2 \\ c_2 \end{bmatrix}$$

The vector equation $Ax = b$ is equivalent to the system of equations

$$c_0 = 0, \quad \frac{\pi}{2} c_0 + 2\pi c_1 + \frac{\pi}{2} c_2 = 0, \quad \frac{\pi}{2} c_1 + 2\pi c_2 + \frac{\pi}{2} c_3 = -3.81971$$

This system has the solution $c_0 = c_3 = 0$, and to decimal places,

$$c_1 \approx 0.16211, \text{ And } c_2 = 0.64845$$

Solving for the remaining constants gives

$$b_o = \frac{1}{h_0}(a_1 - a_0) - \frac{h_o}{3}(c_1 + 2c_0) \approx 0.55173$$
$$b_1 = \frac{1}{h_1}(a_2 - a_1) - (c_2 + 2c_1) \approx 0.29709$$
$$b_2 = \frac{1}{h_2}(a_3 - a_2) - \frac{h_2}{3}(c_3 + 2c_2) \approx -0.04243$$
$$d_o = \frac{1}{3h_o}(c_2 - c_1) \approx 1.01859$$
$$d_2 = \frac{1}{3h_2}(c_3 - c_2) \approx -1.35811$$

The natural cubic spline is described piecewise by

$$S(x) = \begin{bmatrix} 0.33952(x+\pi)^3 + 0.555173(x+\pi) - 1, & for x \in \left[-\pi, \frac{\pi}{2}\right], \\ 1.01859(x+\frac{\pi}{2})^3 + 0.16211(x+\frac{\pi}{2})^2 + 0.29709\left(x+\frac{\pi}{2}\right), & for x \in \left[-\frac{\pi}{2}, 0\right], \\ -1.35811x^3 + 0.64845x^2 - 0.04243x + 1, & for x \in [0, \frac{\pi}{2}], \end{bmatrix}$$

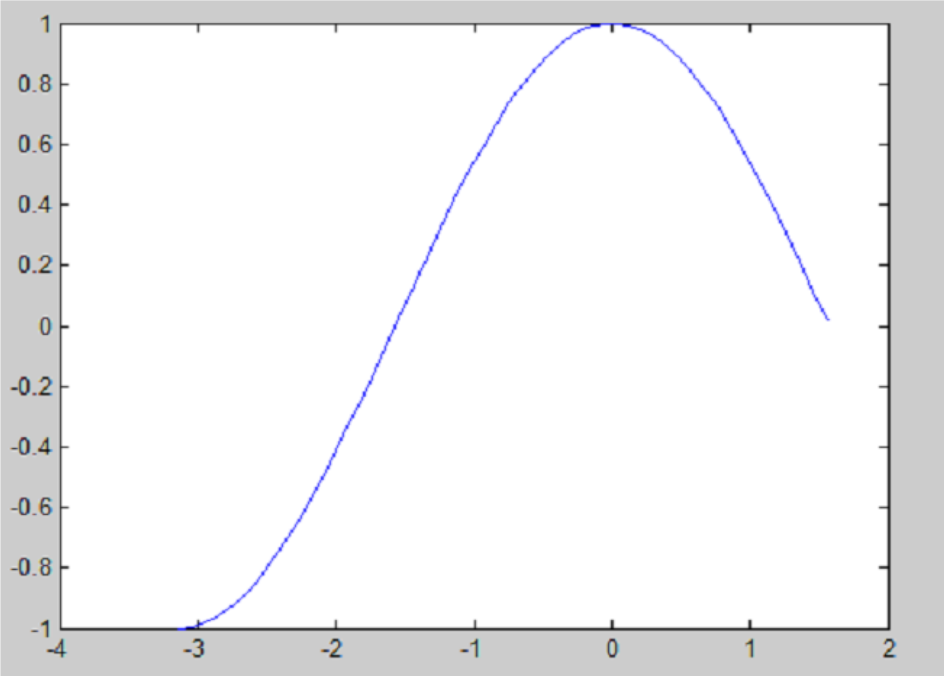


Fig. 15. The graph of $y = \cos(x)$.

3. DESCRIPTION OF THE METHODS

3.1 Finite Difference Approximations

3.1.1 General Principle

The principle of finite difference method is close to the numerical schemes used to solve ordinary differential equations. It consists an approximating the differential operator by replacing the derivatives in the equation using differential quotients. The domain is partitioned in space and in time and approximations of the solution are computed at the space or time points. The error between the numerical solution and the exact solution is determined by the error that is committed by going from a differential operator to a difference operator. This error is called the discretization error or truncation error. The term truncation error reflects the fact that a finite part of a Taylor series is used in the approximation. For the sake of simplicity, we shall consider the one-dimensional case only. The main concept behind any finite difference scheme is related to the definition of a smooth function u at a point $x \in \mathbb{R}$:
We have:

$$u'(x) = \lim_{h\to 0} \frac{u(x+h)-u(x)}{h}, \qquad (26)$$

And to the fact that when h tends to 0 (without vanishing), the quotient on the right- hand side provides a "good" approximation of the derivative. In other words, h should be sufficiently small to get a good approximation. It remains to indicate what exactly a good approximation, in what sense. Actually, the approximation is good when the error committed in this approximation (i.e., when replacing the derivative by the differential quotient) tends towards zero when h tends to zero. If the function sufficiently smooth in the neighborhood of x, it is possible to quantify this error using a Taylor expansion:

3.2 Taylor Series

Suppose the function u is C^2 continuous in the neighborhood of x. For any $h > 0$

$$U(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x+h_1) \qquad (27)$$

Where h_1 is a number between 0 and h ($i.e.\ x + h_1$ is a point of] $x, x + h$[. For the treatment of the problems, it is convenient to retain only the first two terms of the previous expression:

$$U(x + h) = u(x) + hu'(x) + O(h^2) \tag{28}$$

Where the term $O(h^2)$ indicates that the error of the approximation is proportional to h^2. From the equation (27), we deduce that there exists a constant $C > 0$, such that for $h > 0$ sufficiently small we have:

$$|u(x + h) - u(x)| \leq Ch, \quad C = \sup_{y \in [x, x + h_o]} \left|\frac{u''(y)}{2}\right| \tag{29}$$

For $h \leq h_0$ ($h_0 > 0$) given. The error committed by replacing the derivative $u'(x)$ by the differential quotient is of order h. The approximant of u'at point x is said to be Consistent at the first order. This approximation is known as the forward difference approximant of u'. Moregenerally, we define an approximation at order p of the derivative.

Definition: The approximation of the derivative u' at point x is of order p ($p > 0$) if there exists a constant $C > 0$, independent of h, such that the error between the derivative and its approximation is bounded by Ch^p(i.e.is exactly $O(h^p)$).
Likewise, we can define the first order backward difference approximation of u'at point x as:

$$U(x - h) = u(x) - hu'(x) + O(h^2) \tag{30}$$

Obviously, other approximations can be considered. In order to improve the accuracy of the approximation, we define a consistant approximation, called the central difference approximation, by taking the points $x - h$ and $x + h$ into account. Suppose that the function u is trice differentiable in the vicinity of x:

$$u(x + h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u^{(3)}(\xi^+) \tag{31}$$

$$u(x - h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u^{(3)}(\xi^-) \tag{32}$$

Where $\xi^+ \in]x, x + h[$ and $\xi^- \in]x - h, x[$, by subtracting these two expressions we obtain, thanks to the intermediate value theorem;

$\frac{u(x+h)-u(x-h)}{2h} = u'(x) + \frac{h^2}{6}u^{(3)}(\xi)$, where ξ is a point of $]x - h, x + h[$ (33)

Hence, for every $h \in]0, h_0[$, we have the following bound on the approximation error:

$$\left|\frac{u(x+h)-u(x-h)}{2h} - u'(x)\right| \leq Ch^2, \qquad C = \sup_{y \in [x - h_o, x + h_o]} \frac{|h^{(3)}(y)|}{6} \tag{34}$$

This defines a second order consistent approximation to u'.

Remark: The order of the approximation is related to the regularity of the function u. If u is C^2 continuous, then the approximation is consistant at the order one only.

3.3 Approximation of the Second Order Derivatives

Lemma: Supposes u is a C^4 continuous function on an interval $[x - h_o, x + h_o]$,

$h_o > 0$. Then, there exists a constant $C > 0$ such that for every $h \in]0, h_o[$ we have;

$$\left|\frac{u(x+h)-2u(x)+u(x+h)}{h^2} - u''(x)\right| \leq Ch^2 \tag{35}$$

The differential quotient $\frac{u(x+h)-2u(x)+u(x-h)}{h^2}$ is a consistant second-order approximation of the second derivative u'' of u at point x.

Proof: We use Taylor expansions up to the fourth order to achieve the result:

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u^{(2)} + \frac{h^3}{6}u^{(3)} + \frac{h^4}{24}h^{(4)}(\xi^+)$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u^{(3)} - \frac{h^3}{6}u^{(3)} + \frac{h^4}{24}h^{(4)}(\xi^-)$$

Where $\xi^+ \in]x, x+h[$ and$\xi^- \in]x-h, x[$. Like previously, the intermediate value theorem allows us to write;

$\frac{u(x+h)-2u(x)+u(x-h)}{h^2} = u''(x) + \frac{h^2}{12}u^{(4)}(\xi)$, where $\xi \in]x-h, x+h[$.

Hence, we deduce the relation (35) with the constant

$$C = \sup\left|\frac{u^{(4)}(y)}{12}\right|. \qquad y \in [x-h_o, x+h_o]$$

Remark: Likewise, the error estimate depends on the regularity of the function u. If u is C^3 continuous, then the error is of order h only.

3.4 Finite Difference Method (FDM)

The idea of the finite difference method is to replace each derivative involved in boundary value problem in terms of central difference approximations, which then leads to system of equations. The solution of the system of equations which later turned to a problem of solving a system of tri-diagonal matrix, yields approximations to the solution of the original differential equations at discrete points[2]. By considering Taylor polynomial expansion about

x_i which evaluated at $x_{i+1}$and x_{i-1}, we have

$$y(x_{i+1}) = y(x_{i+h}) = y(x_i) + \frac{h}{1!}y'(x_i) + \frac{h^2}{2!}y''(x_i) + \frac{h^3}{3!}y'''(x_i) + \ldots \quad (36)$$

$$y(x_{i-1}) = y(x_{i-h}) = y(x_i) - \frac{h}{1!}y'(x_i) + \frac{h^2}{2!}y''(x_i) - \frac{h^3}{3!}y'''(x_i) + \ldots \quad (37)$$

By adding and subtracting (36) and (37) respectively and rearrange them in terms of y' and y'', resulting to central difference approximations for derivatives $y'$and y''. We reject the truncation error and establish the differential equations. Then, we can obtain the numerical solution by combining the boundary conditions.
Consider the linear boundary value problem

$$y'' = w_1(x)y' + w_2(x)y + w_3(x), \ x \in [a,b] \quad (38)$$
$$y(a) = \alpha\,, y(b) = \beta$$

Collocation points are knot averages in interval $[a, b]$,

Let $x_i = a + ih \ \ (i = 0,1,2, \ldots, N)$,are grid points in the interval $[a, b]$, so that

$x_0 = a, \ x_N = b,$ We use fist-order and second-order centered difference instead of the first and second derivative at the internal knots, and $y_k$substitute in $y(x_k)$

$$y'(x_k) = \frac{y(x_{k+1})-y(x_{k-1})}{2h} + O(h^2)$$
$$y''(x_k) = \frac{y(x_{k+1}) - 2y(x_k) + y(x_{k-1})}{h^2} + O(h^2)$$

Then, we get a differential equation which truncation error is $O(h^2)$ it follows that

$$\frac{1}{h^2}[y_{k-1} - 2y_k + y_{k+1}] + \frac{w_1(x_k)}{2h}[y_{k-1} - y_{k+1}] - w_2(x_k)y_k = w_3(x_k)$$

Also, combined with the boundary value problem

$$y(a) = \alpha, y(b) = \beta$$

And, the linear equations as follow:

$$\begin{cases} -[2 + h^2 w_2(x_1)]y_1 + \left[1 - \frac{w_1(x_1)}{2}h\right] y_2 & = h^2 w_3(x_1) - [1 + \frac{w_1(x_1)h}{2}]\alpha \\ \left[1 + \frac{w_1(x_k)h}{2}\right] y_{k-1} - [2 + h^2 w_2(x_k)]y_k + \left[1 - \frac{w_1(x_k)}{2}h\right] y_{k+1} & = h^2 w_3(x_k) \quad (k = 2,3, \dots, n-2) \\ \left[1 + \frac{w_1(x_{N-1})h}{2}\right] y_{N-2} - [2 + h^2 w_2(x_k)]y_{N-1} & = h^2 w_3(x_{N-1}) - [1 - \frac{w_1(x_{N-1})}{2}h]\beta \end{cases} \quad (39)$$

In matrix notation the system becomes:

$$AY = U$$

Where $A = \begin{bmatrix} -(2 + h^2 w_2(x_1) & 1 - \frac{w_1(x_1)h}{2} & & \\ & (2 + h^2 w_2(x_2) & 1 - \frac{w_1(x_2)h}{2} & \\ 1 + \frac{w_1(x_2)h}{2} & \ddots & \ddots & \\ & \ddots & & \\ & 1 + \frac{w_1(x_{N-2})h}{2} & 2 + h^2 w_2(x_1 & 1 - \frac{w_1(x_1)h}{2} \\ & & 1 - \frac{w_1(x_1)h}{2} & 2 + h^2 w_2(x_1 \end{bmatrix}$

$$Y = \begin{bmatrix} y_{(x_1)} \\ y(x_2) \\ \vdots \\ y(x_{N-2}) \\ y(x_{N-1}) \end{bmatrix}, \qquad U = \begin{bmatrix} h^2 w_3(x_1) - [1 + \frac{w_1(x_1)}{2}h]\alpha \\ h^2 w_3(x_2) \\ \vdots \\ h^2 w_3(x_{N-2}) \\ h^2 w_3(x_{N-1}) - [1 - \frac{w_1(x_{N-1})}{2}h]\beta \end{bmatrix}$$

From a system of equations, with tri-diagonal NxN matrix can be formed, which later is used for numerical result.

Theorem of uniqueness: If $w_1(x), w_2(x)\ and\ w_3(x)$ are continuous and $w_2(x) > 0$

On $[a, b]$ the problem $AY = U$ has a unique solution provided $h < \frac{2}{L}$ were

$$L = max\ a \le x \le b |w_1(x)|.$$

3.5 Cubic B-Spline Interpolation Method (BSI)

B-spline for curves and surfaces were first proposed in 1940s and were seriously developed in the 1970s by several researchers. The term 'B' stands for basis, which explained the full name of this method as basis spline [19**]**. B-spline interpolation is a piecewise polynomial approximation, in which the interval is divided into a collection of subintervals and construct approximating polynomials on each subinterval. The details explanation on connection of B-spline and piecewise polynomial functions can be referred to **[**20**]**. This study is concentrated on B-spline with uniform Partition. The interval of $[a, b]$, is subdivided into equal distant by a step size, $h = \frac{b-a}{N}$, which$N$ as number of subintervals and mesh points, $x_i = a + ih\ at\ i = 0,1,2.3, \dots, N$

B-spline basis functions: The basis function of B-spline is constructed using piecewise polynomial function, with C^2continuity. The first order B-spline basis function or also known as zero-degree B-spline basis function (Step function), is defined as

$$B^1{}_i(\mathrm{x})=\begin{cases}1, & x \in [x_i, x_{i+1}]\\ 0, & otherwise\end{cases} \tag{40}$$

For k^{th} order or k−1-degree B-spline k> 1,the B-spline basis function is defined by equation (41)

$$B_i{}^k(x)=\frac{x-x_i}{x_{i+1-k}-x_i}(x)+B_i{}^{k-1}(x)\frac{x_{i+k}-x}{x_{i+k}-x_{i+1}}B_{i+1}{}^{k-1}(x) \tag{41}$$

Therefore, combining definition (40) and (41) and rearranging the equation will produce a cubic B-spline basis function which represented by:

$$B_i{}^4(\mathrm{x})=\begin{cases}\frac{(x-x_i)^3}{6h^3}, & x \in [x_i, x_{i+1}]\\ \frac{(x-x_i)^2(x_{i+2}-x)}{6h^3}+\frac{(x-x_i)(x_{i+3}-x)(x-x_{i+1})}{6h^3}+\frac{(x-x_{i+1})^2(x_{i+4}-x)}{6h^3}, & x \in [x_{i+1}, x_{i+2}]\\ \frac{(x-x_i)(x_{i+3}-x)^2}{6h^3}+\frac{(x-x_{i+1})(x_{i+3}-x)(x_{i+4}-x)}{6h^3}+\frac{(x-x_{i+2})(x_{i+4}-x)^2}{6h^3}, & x \in [x_{i+2}, x_{i+3}]\\ \frac{(x_{i+4}-x)^3}{6h^3}, & x \in [x_{i+3}, x_{i+4}]\\ 0, & \text{otherwise}\end{cases} \tag{42}$$

Almost directly from this definition, we can conclude some fundamental properties of the B-splines.

The B-splines are:

1. Nonnegative, $B_i{}^k(\mathrm{x}) \geq 0$
2. Compactly supported, $B_i{}^k(\mathrm{x})= 0, x \notin [x_i, x_{i+k+1}]$
3. Piecewise polynomials of degree at most k, $B_i{}^k(\mathrm{x})/ (x_l, x_{l+1}) \in \pi_k$.

It is very easy to prove these properties simultaneously by induction on k: first, they are immediately verified for $k = 0$ and then the inductive step consists of noting that the recurrence (41) implies that

$B_i{}^k(\mathrm{x})= 0$, $x \notin ([x_i, x_{i+k}] \cup [x_{i+1}, x_{i+k+1}])= [x_i, x_{i+k+1}]$ and for $x \in [x_{i+1}, x_{i+k}]$ all the terms in (41) are nonnegative while for $x \in [x_i, x_{i+k}] \cup [x_{i+k}, x_{i+k+1}]$ the negative linear term is rendered irrelevant by a zero spline function.

Cubic B-spline interpolation method: The approximation of BSI, $S(x)$ is defined as linear combination of cubic B-spline basis functions, represented as equation (43), were

$B^4{}_i$are cubic B-spline functions and $c_i$are the unknowns' real coefficients [18].

$$S(x) = \sum_{i=-3}^{n-1} c_i\, B_i{}^4(x) \;\; with\; x \in [x_0, x_N] \tag{43}$$

Evaluation of equation (43), into equations (44), (45) and (46), in which

$x_i \in [x_0, x_N]$ And combining all the evaluation will produce a linear system of equations.

$$\begin{aligned}S(x_i)= {} & c_{-3}B_{-3}{}^4(x_i)+c_{-2}B_{-2}{}^4(x_i) + c_{-1}B_{-1}{}^4(x_i) +\\ & c_0B_0{}^4(x_i) + c_1B_1{}^4(x_i) +\ldots+ c_{n-1}B_{n-1}{}^4(x_i)\end{aligned} \tag{44}$$

$$\begin{aligned}S'(x_i)= {} & c_{i-3}B_{i-3}{}^{4\,\prime}(x_i)+c_{i-2}B_{i-2}{}^{4\prime}(x_i) + c_{i-1}B_{i-1}{}^{4\,\prime}(x_i) +\\ & c_iB_i{}^{4\;\prime}(x_i) + \cdots+c_{n-1}B_{n-1}{}^{4\prime}(x_i)\end{aligned} \tag{45}$$

$$S''(x_i) = c_{i-3}B_{i-3}^{4}\,''(x_i) + c_{i-2}B_{i-2}^{4}\,''(x_i) + c_{i-1}B_{i-1}^{4}\,''(x_i) + c_iB_i^{4\prime\prime}(x_i) + \ldots + c_{n-1}B_{n-1}^{4\prime\prime}(x_i) \quad (46)$$

Then, B-spline curve (which is just an arbitrary curve with some unknowns) can be presumed to be the solution of the problem and the unknown constants (c_i) can be made known by solving a system of linear equations. This B-spline curve is interpolated as the approximation of analytical solution for the problem. By returning to (1a) and taking $S(x_i)$ as approximation of cubic B-spline interpolation method (BSI) for $y(x)$ and substituting equations $(44), (45)$ and (46) into $(1a)$, the equation becomes

$$S''^{(x_i)} + q(x)S'^{(x_i)} + r(x)S(x_i) = f(x), S(a) = \alpha\,, S(b) = \beta \quad (47)$$

Then, by solving (47) at $i = 0, 1, 2, \ldots, N$, will result into linear system of equations of order $(N+1)x(N+3)$. Therefore, two equations of boundary conditions in equation (47) should be added to the system, which caused the system to be of N+3 linear equations in the N+3 unknowns $c_{-3}, c_{-2} \ldots, c_{N-1}$ are obtained, using (43) can obtain the numerical solution.

This system can be written in the matrix-vector form as follows:

$AE = F$, where E= $[c_{-3}, c_{-2}, \ldots, c_{N-1}]^T$

F= $[0, f(x_o), f(x_1), \ldots f(x_N), 0]^T$ and A is an $(N+3)x\,(N+3)$ dimensional

Tri-diagonal matrix given by

$$A = \begin{bmatrix} 1 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \alpha_O(x_o) & \beta_o(x_o) & \gamma_o(x_o) & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \alpha_1(x_1) & \beta_1(x_1) & \gamma_1(x_1) & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & 0 & \cdots & \alpha_N(x_N) & \beta_N(x_N) & \gamma_N(x_N) \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 \end{bmatrix}$$

Also, the coefficients in the matrix A have the following form

$$\begin{aligned} \alpha_i(x_i) &= \frac{6}{h^2} + q(x_i)\frac{-3}{h} + r(x_i), \qquad (i = 0,1, \ldots, N) \\ \beta_i(x_i) &= \frac{-12}{h^2} + 4r(x_i), (i = 0,1, \ldots, N) \\ \gamma_i(x_i) &= \frac{6}{h^2} + q(x_i)\frac{3}{h} + r(x_i), (i = 0,1, \ldots, N) \end{aligned} \quad (48)$$

Then, a system of linear equation can be built as shown below:

$$\begin{bmatrix} 1 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \alpha_O(x_o) & \beta_o(x_o) & \gamma_o(x_o) & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \alpha_1(x_1) & \beta_1(x_1) & \gamma_1(x_1) & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & 0 & \cdots & \alpha_N(x_N) & \beta_N(x_N) & \gamma_N(x_N) \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} c_{-3} \\ c_{-2} \\ \vdots \\ c_{N-2} \\ c_{N-1} \end{bmatrix} = 6 \begin{bmatrix} 0 \\ f(x_o) \\ \vdots \\ f(x_N) \\ 0 \end{bmatrix} \quad (49)$$

This system can be used to solve the problem for coefficient c_i.The $c_i's$ is then substituted into (43), to represent an approximated of analytical solution by BSI to ($1a$). To solve second order boundary value problems, $B_i(x), B_i'(x), and B_i''(x)$ evaluated at nodal points are needed, which are summarized in **Table 1**

Table 1. Values of $B_i(x)$,$B_i'(x)$ and $B_i''(x)$ at nodal points

	$B_i(x)$,		$B_i'(x)$	$B_i''(x)$
x_i	0	0		0
x_{i+1}	1		$\frac{-3}{h}$	$\frac{6}{h^2}$
x_{i+2}	4	0		$\frac{-12}{h^2}$
x_{i+3}	1		$\frac{3}{h}$	$\frac{6}{h^2}$
x_{i+4}	0	0		0

3.6 Cubic Spline Method for Two-point Boundary Value Problems

We consider a uniform mesh Δ with nodal points x_i on $[a,b]$ such that

$$\Delta:\ a = x_o < x_1 < x_2 < \cdots < x_{N-1} < x_N = b, \qquad (50)$$
$$x_i = a + ih, \quad i = 0,1,2,\dots,N, \quad \text{where } h = \tfrac{b-a}{N}.$$

A non-polynomial function $s_\Delta(x)$ of class $C^2[a,b]$ which interpolates $y(x)$ at the mesh points x_i , for $i = 0,1,2,\dots,N$, depends on a parameter τ, and reduces to ordinary cubic spline in $[a,b]$as $\tau \to 0$. The spline function we propose has the following form:

$T_3 = Span\{1, x, cos\tau x, sin\tau x\}$, where τ is the frequency of the trigonometric part of the spline function which can be real or pure imaginary and which will be used to raise the accuracy of the method.

When correlation between polynomial and non-polynomial spline basis functions are investigated in the following manner:

$$T_3 = Span\{1, x, cos\tau x, sin\tau x\}$$
$$= Span\left\{1, x, \left(\tfrac{2}{\tau^2}\right)(1 - cos\tau x), (\tfrac{6}{\tau^3})(\tau x - sin\tau x)\right\}. \qquad (51)$$

It follows that $\lim_{\tau\to 0} T_3 = \{1, x, x^2, x^3\}$. (52)

Thus, in each subinterval $x_i \le x \le x_{i+1}$, we have

$$Span\{1, x, cos\tau x, sin\tau x\}\ or\ Span\{1, x, x^2, x^3\}(\text{When } \tau \to 0) \qquad (53)$$

For each segment $[x_i, x_{i+1}]$, $i = 0,1,2,\dots,N-1$ the non-polynomial $s_\Delta(x)$ has the following form:

$$s_\Delta(x) = a_i + b_i(x - x_i) + c_i sin\tau x(x - x_i) + d_i cos\tau x(x - x_i), \qquad (54)$$
$$i = 0,1,2,\dots,N,$$

Where $a_i, b_i, c_i, and\ d_i$ are constants and τ is a free parameter.

Let y_i be an approximation to $y(x_i)$, obtained by the segment $s_\Delta(x)$ of mixed spline function passing through the points (x_i, y_i) and (x_{i+1}, y_{i+1}). To obtain the necessary conditions for the coefficients introduced in (54), we not only require $s_\Delta(x)$ to satisfies interpolate conditions at x_i and x_{i+1}, but also the continuity of first derivative at the common nodes (x_i, y_i) to be fulfilled.

To derive an expression for the coefficients of (54) in terms of $y_i, y_{i+1}, M_i, M_{i+1}$, we first denote:

$$s_\Delta(x_i) = y_i, \quad s_\Delta(x_{i+1}) = y_{i+1}$$
$$s''_\Delta(x_i) = M_i, \quad s''_\Delta(x_{i+1}) = M_{i+1}. \qquad (55)$$

From algebraic simplification we get the following expression:

$$a_i = y_i + \frac{M_i}{\tau^2}, b_i = \frac{y_{i+1}-y_i}{h} + \frac{M_{i+1}-M_i}{\tau\theta} \quad (56)$$
$$c_i = \frac{M_i cos\theta - M_{i+1}}{\tau^2 sin\theta},\ d_i = -\frac{M_i}{\tau^2},$$

Where $\theta = \tau h$ and $i = 0,1,2,\dots,N-1$ using continuity of the first derivative at (x_i, y_i); that is,

$$s'_{\Delta i-1}(x_i) = s'_{\Delta i}(x_i).$$

We obtain the following relations for$i = 0,1,2,\dots,N-1$

$$\alpha M_{i+1} + 2\beta M_i + \alpha M_{i-1} = \frac{1}{h^2}(y_{i+1} - 2y_i + y_{i-1}), \quad (57)$$
where $\alpha = \frac{1}{h^2}(\theta csc\theta - 1),\ \beta = \frac{1}{h^2}(1 - \theta cot\theta),$

and $\theta = \tau h$. whenever $\tau \to 0$, then $\alpha \to \frac{1}{6}$ and $\beta \to \frac{1}{3}$.

Therefore (57) reduces to the consistency relation of cubic splines:

$$M_{i+1} + 4M_i + M_{i-1} = \frac{6}{h^2}(y_{i+1} - 2y_i + y_{i-1}), \quad (58)$$

From Equation (1a)

$$\frac{d^2y}{dx^2} + q(x)\frac{dy}{dx} + r(x)y = f(x), \quad (59)$$
$$y(a) = y(b) = 0$$

The proposed differential equation (59) can be discretized at the nodal point x_i by

$$y''_i + q_i y'_i + r_i y_i = f_i, \quad (60)$$
where $q_i = q(x_i), r_i = r(x_i), f_i = f(x_i).$

By using moment of the spline in (60) we obtain

$$M_i + q_i y'_i + r_i y_i = f_i. \quad (61)$$

Taking approximations for the first derivative of y we have

$$y'_i = \frac{y_{i+1}-y_{i-1}}{2h},\ y'_{i+1} = \frac{3y_{i+1}-4y_i+y_{i-1}}{2h},\ \ y'_{i-1} = \frac{-y_{i+1}+4y_i-3y_{i-1}}{2h} \quad (62)$$

Substituting (61), (62) in (57) and simplifying, we get the following tri-diagonal system which gives the approximations

$y_1, y_2, y_3 \dots, y_{N-1}$of the solution $y(x)$at $x_1, x_2, x_3 \dots, x_{N-1}$:

$$\left(\frac{3h}{2}\alpha q_{i-1} + h\beta q_i - \frac{h}{2}\alpha q_{i+1} - h^2\alpha r_{i-1} - 1\right)y_{i-1} + (-2\mathrm{h}\alpha q_{i-1} + 2\mathrm{h}\alpha q_{i+1} - h^2 2\beta r_i + 2)y_i +$$
$$\left(\frac{h}{2}\alpha q_{i-1} - \mathrm{h}\beta q_i - \frac{3h}{2}\alpha q_{i+1} - h^2\alpha r_{i+1} - 1\right)y_{i+1} = -h^2(\alpha f_{i-1} + 2\beta f_i + \alpha f_i),\ \text{with}\ y(a) = y(b) = 0,$$
i=1(1a)$N-1$. (63)

4. CONVERGENCE ANALYSIS

The tri-diagonal linear system (63) can be written in the following matrix form,

$$AY + h^2 DF = G, \quad (64)$$

where $A = P + hBQ - h^2\text{BR}$.

A is tri-diagonally dominant matrix of order $(N-1)$.

$P = (P_{ij})$isatri-diagonal matrix defined by

$$(P_{ij}) = \begin{cases} 2 & i = j = 1,2,\dots,N-1, \\ -1 & |i-j| = 1, \\ 0 & \text{otherwise} \end{cases} \tag{65}$$

$BQ = \left(Z_{ij}\right),\ BR = (R_{ij})$ are tri-diagonal matrices defined as

$$\left(Z_{ij}\right) = \begin{cases} 2\alpha(-q_0 + q_2), & i = j = 1, \\ \frac{3}{2}\alpha q_{i-1} + \beta q_i - \frac{1}{2}\alpha q_{i+1}, & i > j, \\ 2\alpha(-q_{i-1} + q_{i+1}), & i = j, \\ \frac{1}{2}\alpha q_{i-1} - \beta q_i - \frac{3}{2}\alpha q_{i+1}, & i < j, \\ 2\alpha(-q_{N-2} + q_N), & i = j = N-1, \end{cases} \tag{66}$$

$$\left(R_{ij}\right) = \begin{cases} 2\beta r_i & i = j = 1,2,\dots,N-1, \\ \alpha r_{i-1,} & i > j, \\ \alpha r_{i+1,} & i < j, \end{cases}$$

$$F = (f_1, f_2, f_3, \dots, f_{N-1})^T.$$

$$Y = (y_1, y_2, y_3, \dots, y_{N-1})^T. \tag{67}$$

The tri-diagonal matrix D is defined by

$$D = \begin{bmatrix} 2\beta & \alpha & 0 & 0 & 0 & \cdot & \cdot & 0 \\ \alpha & 2\beta & \alpha & 0 & 0 & \cdot & \cdot & 0 \\ 0 & \alpha & 2\beta & \alpha & 0 & \cdot & \cdot & 0 \\ 0 & 0 & 0 & \alpha & 2\beta & \alpha & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \alpha & 2\beta & \alpha & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \alpha & 2\beta & \alpha \\ 0 & 0 & 0 & 0 & 0 & \cdot & \alpha & 2\beta \end{bmatrix}, \tag{68}$$

and $G = (g_1\ 0,0,0,\dots,0,\ g_{N-1})^T$

where$g_1 = -h^2\alpha f_0, \quad g_i = 0,$

$for\ i = 2,3,\dots,N-2, g_{N-1} = -h^2\alpha f_N.$

We assume that $\bar{Y} = (y(x_1), y(x_2), \dots, y(x_{N-1}))^T$ be the exact solution of the given boundary value problem (1a) at nodal points $x_i, i = 0,1,\dots,N-1,$

And then, we have

$$A\,\bar{Y} + h^2 DF = T(h) + G, \tag{69}$$

where $T = (T(x_1), T(x_2), \dots, T(x_{N-1}))^T$ is truncation error vector

By expanding (63) in Taylor series about x_i, and using (60) we obtain the following local truncation error.

$$T_i(h) = [-1 + 2(\alpha+\beta)]h^2 y''(\xi_i) + \frac{1}{3}(\alpha q_{i+1} - \beta q_i + \alpha q_{i-1})h^4 y'''(\xi_i) + \frac{1}{12}[(-1 + 12\alpha) + \alpha h(q_{i+1} - q_{i-1})]h^4 y^{(4)}(\xi_i) + O(h)^5, \tag{70}$$

$$x_{i-1} < \xi_i < x_{i+1},$$

From (64) and (69) we have

$$A(\bar{Y} - Y) = AE = T(h), \tag{71}$$

where $E = \bar{Y} - Y = (e_i, e_2, \dots, e_{N-1})^T$. (72)

The main purpose is to drive a bound on$\|E\|$. For this the following lemma is needed.

Lemma: If W is a matrix of order N and$\|W\| < 1$, then $(I + W)^{-1}$ exists and

$$\|(I + W)^{-1}\| < \frac{1}{1-\|W\|}.$$

From (71) we have $E = A^{-1} = [P + hBQ]^{-1}T$

$= [I + hP^{-1}BQ]^{-1}P^{-1}T$, then we have $\|E\| \leq [I + hP^{-1}BQ]^{-1}\|P^{-1}\|\|T\|$,

Then we obtain $\|E\| \leq \frac{\|P^{-1}\|\|T\|}{1-h\|P^{-1}\|\|BQ\|}$ (73)

Provided that $h\|P^{-1}\|\|BQ\| \leq 1$

Following [22] we have $\|P^{-1}\| \leq \frac{(b-a)^2}{8h^2}$, (74)

And then we have $\|BQ\| \leq q(8\alpha + 2\beta)$, (75)

where $q = Max|q(x_i)|$, $a < x_i < b$.

Note for $\alpha + \beta = \frac{1}{2}$ and $\alpha \neq \frac{1}{12}$, we have$\|T\| \leq \eta_1 h^4 M_4$,

Where $M_4 = \max_{a\leq\xi\leq b}\left|y^{(4)}(\xi)\right|$, η_1 constant independent of h then we have

$$\|E\| \leq \frac{\eta_1(b-a)^2h^2M_4}{1-\omega q} \equiv O(h^2). \tag{76}$$

But for the choice of parameters $\alpha = \frac{1}{12}$ and $\beta = \frac{5}{12}$ we have:

$T_i(h) = \frac{1}{360}(\alpha q_{i+1} - \beta q_i + \alpha q_{i-1})h^4 y'''(\xi_i) + \frac{1}{144}(q_{i+1} - q_{i-1})h^4 y^{(4)}(\xi_i) + O(h)^5$,
$x_{i-1} < \xi_i < x_{i+1}$,

Thus, in this case we have optimal second-order method.

Remark: for$\alpha + \beta = \frac{1}{2}$, our method is a second order method.

Remark: for $\alpha = \frac{1}{12}$ and $\beta = \frac{5}{12}$, our method is optimal second-order method.

5. NUMERICAL ILLUSTRATIONS AND RESULTS

In present work two linear boundary value problems with $h = 0.1$, have been solved, whose exact solutions are known. The approximate solution, exact solutions, and absolute errors at the nodal points are tabulated in **Table 2-7** and comparisons are shown in **Fig.. 16-20**. The results of the present work are compared with exact solution of the two problems and with finite difference method, and B-spline method of Example 2.

Test Equations and Results

Example 1: Solve the following boundary value problem using B-spline method, Finite difference method, and Cubic spline method.

$$\begin{cases} y''(x) - y'(x) = -(e^{x-1} + 1), \quad 0 < x < 1 \\ \qquad\qquad y(0) = 0, \quad y(1) = 0 \end{cases} \tag{77}$$

The analytical solution of this problem is:

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

Respectively, the observed maximum absolute errors for various value of N and a specific choice of parameters α and β are given in **Table (2), (3)** and **(4)**. The numerical results are illustrated in **Figs. (16)** and **(.17)**.

Method1: Using Cubic B-Spline Interpolation Method (BSI) we can get the coefficient A by using (48) f or $N = 10$

$$A = \begin{bmatrix} 1 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{6+3h}{h^2} & \frac{-12}{h^2} & \frac{6-3h}{h^2} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{6+3h}{h^2} & \frac{-12}{h^2} & \frac{6-3h}{h^2} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \frac{6+3h}{h^2} & \frac{-12}{h^2} & \frac{6-3h}{h^2} \\ 0 & 0 & 0 & 0 & 0 & & 1 & 4 & 1 \end{bmatrix}$$

and $F = [0, -8.2073, -8.4394, \ldots, -11.4290, -12.0000, 0]^T$

Then if $h = 0.1$, we can find B as follows;

$$E = [-0.0657, 0.0012, 0.0608, \ldots, 0.0050, -0.1102]^T$$

and we can get the function

$$\mathrm{S(x)} = \sum_{\mathrm{i=-3}}^{\mathrm{N-1}} \mathrm{c_i B^4_i},$$

For example

$$S(x) = \begin{cases} -0.2x^3 - 0.365x^2 + 0.6325x - 0.0000166667, & x \in [0,0.1) \\ -0.233x^3 - 0.355x^2 + 0.6315x + 0.000016, & x \in [0.1,0.2) \\ -0.25x^3 - 0.3449x^2 + 0.6294x + 0.00015, & x \in [0.2,0.3) \\ -0.3x^3 - 0.3x^2 + 0.616x + 0.0015, & x \in [0.3,0.4) \\ -0.32x^3 - 0.28x^2 + 0.608x + 0.002483, & x \in [0.4,0.5) \\ -0.4x^3 - 0.155x^2 + 0.5455x + 0.0129833, & x \in [0.5,0.6) \\ -0.417x^3 - 0.125x^2 + 0.5275x + 0.016583, & x \in [0.6,0.7) \\ -0.467x^3 - 0.02x^2 + 0.454x + 0.033733, & x \in [0.7,0.8) \\ -0.6x^3 + 0.3x^2 + 0.198x + 0.102, & x \in [0.8,0.9) \\ -0.6x^3 + 0.3x^2 + 0.1979x + 0.102, & x \in [0.9,1] \end{cases}$$

Therefore, numerical solutions are obtained by the B-spline method, they follow that

$$\begin{aligned} s_1 &= 0.059383000000000, \\ s_2 &= 0.110234000000000, \\ s_3 &= 0.151200000000000, \\ s_4 &= 0.180403000000000, \end{aligned}$$

$s_5 = 0.196983300000000,$
$s_6 = 0.198011000000000,$
$s_7 = 0.181655230000000,$
$s_8 = 0.145200000000000,$
$s_9 = 0.085710000000000$

The analytical solution is given by

$y_1(x_1) = 0.059343034025940,$
$y_2(x_2) = 0.110134207176556,$
$y_3(x_3) = 0.151024408862577,$
$y_4(x_4) = 0.180475345562389,$
$y_5(x_5) = 0.196734670143683,$
$y_6(x_6) = 0.197807972378616,$
$y_7(x_7) = 0.181427245522798,$
$y_8(x_8) = 0.145015397537615,$
$y_9(x_9) = 0.085646323767636,$
and
$y_1(x_1) - s_1 = -0.00003696597405990510,$
$y_2(x_2) - s_2 = -0.00009979282344428631,$
$y_3(x_3) - s_3 = -0.00017755911374228536,$
$y_4(x_4) - s_4 = 0.00007234556238944201,$
$y_5(x_5) - s_5 = -0.00024686298563167122,$
$y_6(x_6) - s_6 = -0.00020302762138357 80,$
$y_7(x_7) - s_7 = -0.00022799844825657843,$
$y_8(x_8) - s_8 = -0.00018460246238544 14,$
$y_9(x_9) - s_9 = -0.00036367623236354369,$

Thus, the max-absolute error is given by

$\omega = 0.00024686298563167122$

Method2: Finite Difference Method (FDM): At first, the interval of solution is divided into many small regions and gets the set of interval node. In these nodes, we use difference coefficient instead of differential. We reject the discretization error and establish the differential equations. Then, we can obtain the numerical solution by combining the boundary conditions. From (38) and (77) we have

$$w_1(x) = 1, \quad w_2(x) = 0, \quad w_3(x) = -(e^{x-1} + 1) \tag{78}$$

Hence, numerical solutions are obtained by the Finite difference method, they follow that

$s_1 = 0.0595,$
$s_2 = 0.1103,$
$s_3 = 0.1513,$
$s_4 = 0.1808,$
$s_5 = 0.1971$
$s_6 = 0.1981,$
$s_7 = 0.1817,$
$s_8 = 0.1452,$
$s_9 = 0.0857,$

Also, the max-absolute error is given by

$\omega = 0.0004$

Table 2. Shows the max-absolute errors for the methods with respect to the true solution for example 1

Methods	h	**Max-absolute errors**
Cubic B-spline Interpolation method	0.1	0.0002486298563167122
Finite difference method	0.1	0.0004

Table 3. The maximum absolute errors in solution of problem

1 h **Methods**	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
Our method (cubic) $\alpha=\frac{1}{4},\beta=\frac{1}{4}$	0.0000550	0.0000137	0.00000344
Our method (cubic) $\alpha=\frac{1}{6},\beta=\frac{1}{3}$	0.0000451	0.0000112	0.00000282
Our method (cubic) $\alpha=\frac{1}{14},\beta=\frac{3}{7}$	0.0000338	0.0000844	0.00000211
Our method (cubic) $\alpha=\frac{1}{18},\beta=\frac{4}{9}$	0.0000319	0.00000796	0.00000199
Our method (cubic) $\alpha=\frac{1}{24},\beta=\frac{11}{24}$	0.0000302	0.00000755	0.00000189
Our optimal method $\alpha=\frac{1}{12},\beta=\frac{5}{12}$	0.0000352	0.000000879	0.000000220

From the table we can say for different values of α and β the error is rapidly reduced when step size increases.

Table 4. Comparison of maximum absolute error in our methods with other methods

h **Methods**	$\frac{1}{10}$	$\frac{1}{100}$
Problem 1 finite difference	0.00824	0.00831
B-spline interpolation	0.00290	0.00289
Our method (cubic) $\alpha=\frac{1}{18},\beta=\frac{8}{18}$	0.00188	0.000187
$\alpha=\frac{1}{12},\beta=\frac{5}{12}$	0.000226	0.000225

From the above table for choosing different values of α and β we can say that the error is reduced more rapidly than using finite difference method, and B-spline method.

Example: 2. Solve the following boundary problem using B-spline method, Finite difference method, and Cubic spline method.

$$\begin{cases} y''(x) = 2(y'(x) + y(x) - 1), \quad 0 \le x \le 1 \\ y(0) = 0\,, \quad y(1) = 0 \end{cases} \tag{79}$$

The analytical solution of this problem is:

$$y(x) = \left(\frac{1-e^{1+\sqrt{3}}}{e^{1+\sqrt{3}}-e^{1-\sqrt{3}}}\right) e^{(1-\sqrt{3})x} + \left(\frac{e^{1-\sqrt{3}}-1}{e^{1+\sqrt{3}}-e^{1-\sqrt{3}}}\right) e^{(1+\sqrt{3})x} + 1.$$

Respectively, the observed maximum absolute errors for various values of *N* are given in **Table (5).** The numerical results are illustrated in **Fig (18)** and **(19).** As is evident from the numerical results, the B-spline approximates the exact solution very well than finite difference method.

Method1: Using Cubic B-Spline Interpolation Method (BSI). We can get the coefficient matrix A by using (43)

$$\mathrm{A}=\begin{bmatrix} 1 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{6}{h^2}+\frac{6}{h-2} & \frac{-12}{\mathrm{h}^2}-8 & \frac{6}{\mathrm{h}^2}-\frac{6}{h-2} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{6}{\mathrm{h}^2}+\frac{6}{\mathrm{h}-2} & \frac{-12}{\mathrm{h}^2}-8 & \frac{6}{\mathrm{h}^2}-\frac{6}{h-2} & 0 & \cdots & 0 & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & 0 & \cdots & \frac{6}{\mathrm{h}^2}+\frac{6}{\mathrm{h}-2} & \frac{-12}{\mathrm{h}^2}-8 & \frac{6}{\mathrm{h}^2}-6 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 4 & \mathrm{h}-2 \\ & & & & & & & & 1 \end{bmatrix}$$

and $F = [0, -12, -12, \ldots, -12, -12, 0]^T$

Then, if $h = 0.1$, we can find B as follows: $E = [-0.0638, 0.0013, \ldots, 0.0071, -0.1273]^T$
and we can get the function

$$S(x) = \sum_{i=-3}^{N-1} c_i B^4{}_i,$$

For example,

$$S(x) = \begin{cases} -0.1x^3 - 0.385x^2 + 0.6125x + 0.0000166667, & x \in [0,0.1); \\ -0.83x^3 - 0.39x^2 + 0.613x, & x \in [0.1,0.2); \\ -0.217x^3 - 0.31x^2 + 0.597x - 0.001066666, & x \in [0.2,0.3); \\ -0.25x^3 + 29.255x^2 - 8.2725x + 0.0020, & x \in [0.3,0.4); \\ -0.35x^3 - 0.16x^2 + 0.54x + 0.0084, & x \in [0.4,0.5); \\ -0.5x^3 + 0.065x^2 + 0.4275x + 0.027, & x \in [0.5,0.6); \\ -0.67x^3 + 0.365x^2 + 0.2475x + 0.063, & x \in [0.6,0.7); \\ -0.9x^3 + 0.855x^2 - 0.0955x + 0.14315, & x \in [0.7,0.8) \\ -1.183x^3 + 1.535x^2 - 0.6395x + 0.289, & x \in [0.8,0.9); \\ -1.57x^3 + 2.57x^2 - 1.571x + 0.568, & x \in [0.9,1] \end{cases}$$

Therefore, numerical solutions are obtained by the B-spline method, they follow that

$$\begin{aligned} s_1 &= 0.056570000000000, \\ s_2 &= 0.104197334600000, \\ s_3 &= 0.146450000000000, \\ s_4 &= 0.176400000000000, \\ s_5 &= 0.194500000000000, \\ s_6 &= 0.198180000000000, \\ s_7 &= 0.186550000000000, \\ s_8 &= 0.154104000000000, \end{aligned}$$

$$s_9 = 0.091270000000000,$$

The analytical solution is given by

$$y_1(x_1) = 0.057170833644836,$$
$$y_2(x_2) = 0.106090158916539,$$
$$y_3(x_3) = 0.146015961602584,$$
$$y_4(x_4) = 0.175842492561400,$$
$$y_5(x_5) = 0.193995213412198,$$
$$y_6(x_6) = 0.198292086489174,$$
$$y_7(x_7) = 0.185760681390258,$$
$$y_8(x_8) = 0.152397259732250,$$
$$y_9(x_9) = 0.092849649107364,$$

and

$$y_1(x_1) - s_1 = 0.0006008336448357701,$$
$$y_2(x_2) - s_2 = 0.001862824916539,$$
$$y_3(x_3) - s_3 = -0.0004340383974152651,$$
$$y_4(x_4) - s_4 = -0.0005575074385995305,$$
$$y_5(x_5) - s_5 = 0.0005047865878023417,$$
$$y_6(x_6) - s_6 = 0.0001120864891739348,$$
$$y_7(x_7) - s_7 = -0.0007893186097420735,$$
$$y_8(x_8) - s_8 = -0.001706740267750,$$
$$y_9(x_9) - s_9 = 0.001579649107364,$$

Thus, the max-absolute error is given by

$$\omega = 0.001828624916539,$$

Method 2: Finite Difference Method (FDM). From (38) and (79) we have

$$w_1(x) = 2$$
$$w_2(x) = 2$$
$$w_3(x) = -2$$

Hence, numerical solutions are obtained by the finite difference method (FDM), they follow that

$s_1 = 0.0399$, $s_2 = 0.0897$,
$s_3 = 0.1302$, $s_4 = 0.1604$,
$s_5 = 0.1787$, $s_6 = 0.1827$,
$s_7 = 0.1695$, $s_8 = 0.1350$,
$s_9 = 0.0735$, also the max-absolute error is given by $\omega = 0.00193$

Table 5. Shows the max-absolute errors for the methods with respect to the true solution for example 2

Methods	***h***	**Max-absolute errors**
Cubic B-spline interpolation method	0.1	0.001828624916539
Finite difference method	0.1	0.00193

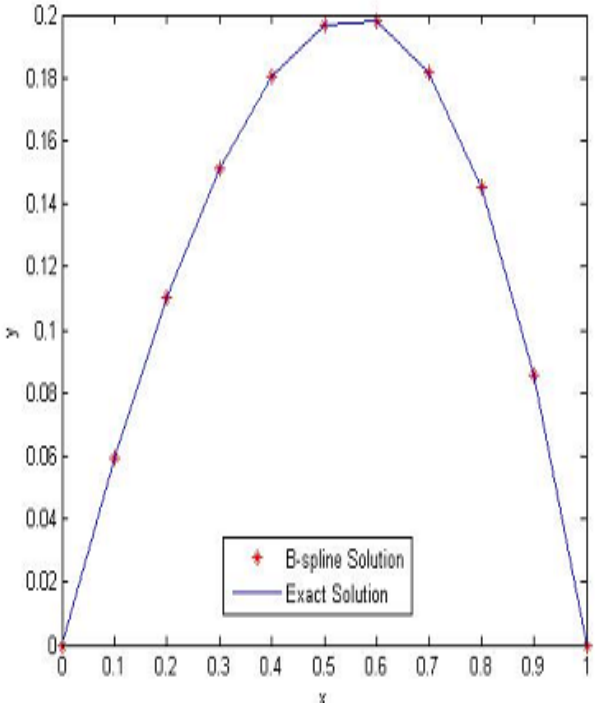


Fig. 16. B-spline method solution and exact solution

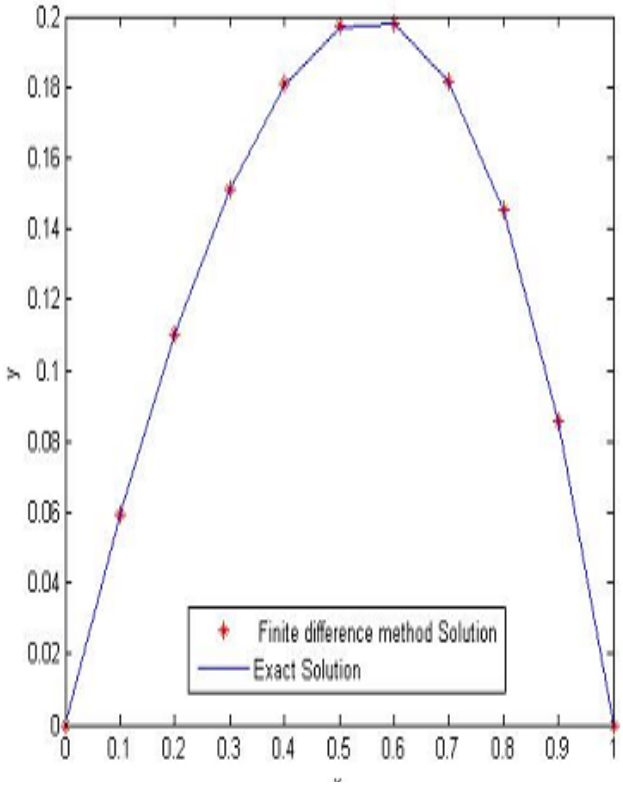


Fig. 17. Finite difference method solution and exact solution

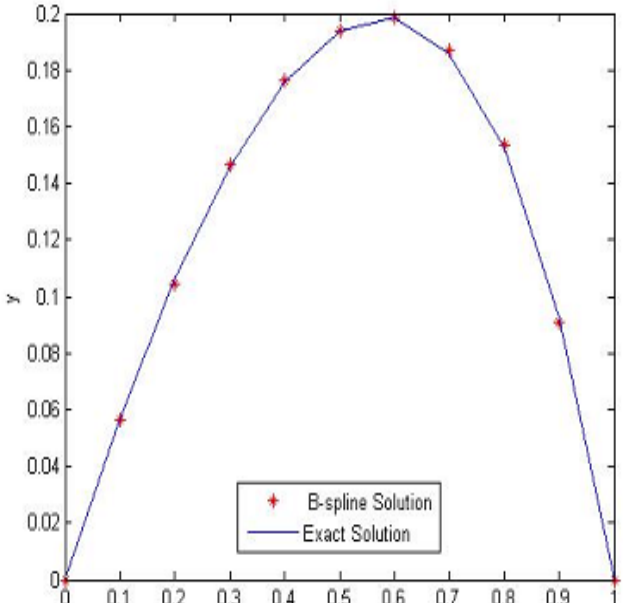


Fig. 18. B-spline method solution and exact solution

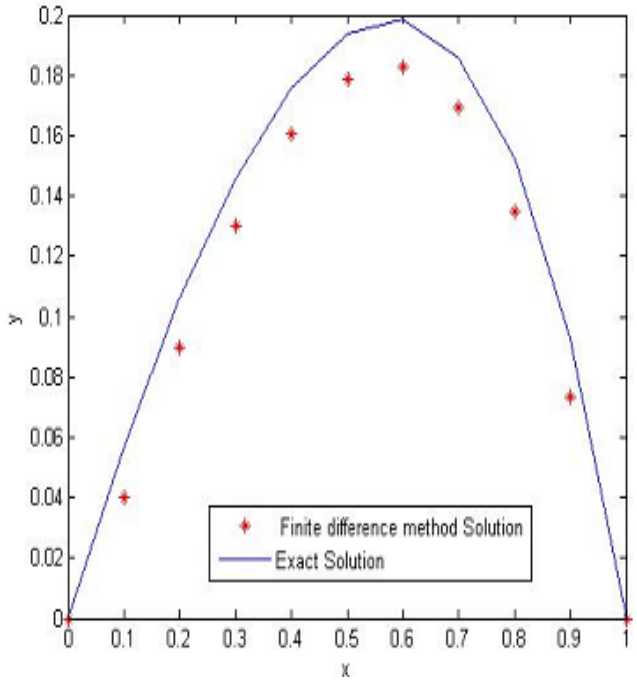
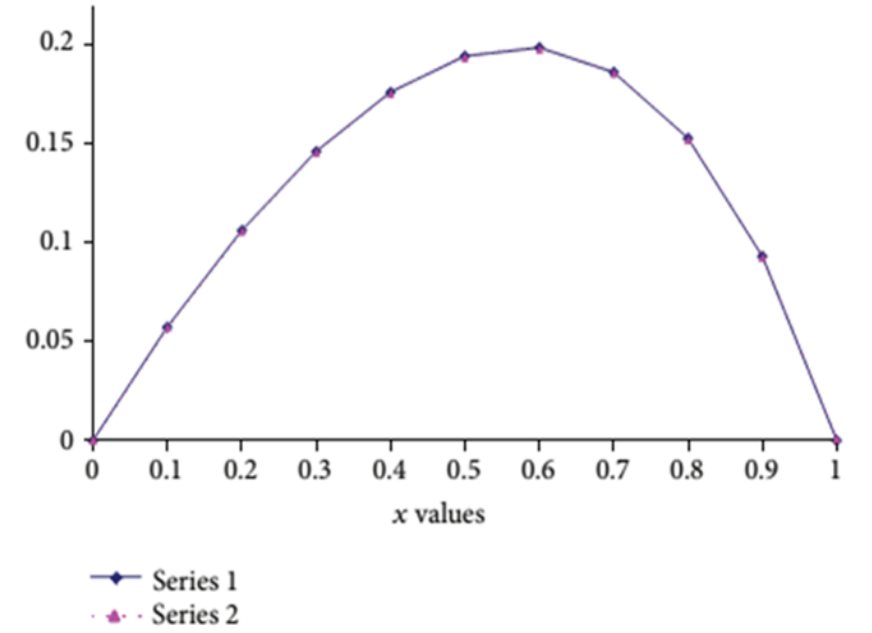
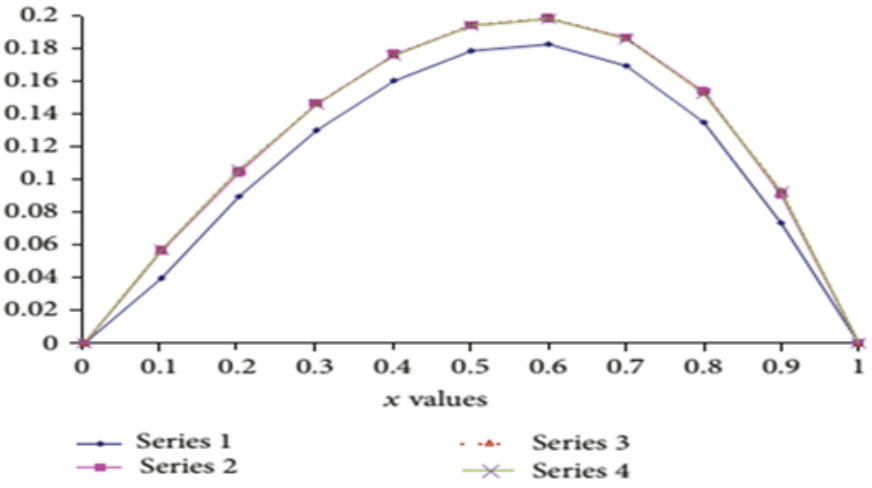


Fig. 19. Finite difference method solution and exact solution



(a)



(b)

Fig. 20. (a) Comparison of approximate values and exact values for Example 2.
(b) Comparison of results obtained by cubic spline method of Example 2with other values obtained by other methods (Table 7)

Method 3: Using Cubic Spline Method

Comparing the given equation with (59) we have

$$q_i = q(x_i) = -2, r(x) = -2, f_i = f(x_i)$$

Here $h = 0.1$, so the nodal points are $x_0, x_1, x_1, \dots, x_{10}$. Solution of (79) is given by (64). Here $G = (g_1, 0,0,0,0,0,0,0, g_9)^T$, where $g_1 = 0.00333333, g_9 = 0.003333333$ and the function values at the nodal points is -2. That is,

$$F = (f_1, f_2, f_3, \dots, f_{N-1})^T, \text{Where } f_i = -2 \text{ for } i = 1,2, \dots, 9 \quad (80)$$

Substituting the values of p_i, q_{i+1}, and$r_i$at $i = 0,1,2,\dots,10$ in (65), (66), and (67) we get the values of (P_{ij}) , (Z_{ij}), and (R_{ij}), and substituting these values in (64) we get the tri-diagonal matrix A.From (80) we have G and F. So, we obtain a system of linear equations. This system in the matrix form is

$AY + h^2DF = G$; Solving this we get the solution matrix Y.

Approximation solution, the exact solution, and the absolute errors at the nodal points of $y(x)$ are given in **Table (6)** and the comparison is given in **Fig (19)**. Comparative results obtained by finite difference method and B-spline method are given **Table (7)** and shown in **Fig (19)**

From the results, we will see the difference between them and conclude that the cubic spline method is the better to interpolate any smooth functions than others. The numerical results for our example are shown in **Table (6)** and **(7)**, which shows that there is a big difference for the errors between cubic spline method, B-spline method and the Finite difference method unless there is no remarkable difference among the accuracy of the other method in the case where f is sufficiently smooth. The above two examples are the deformation singular perturbation problem [23]. The singularly-perturbed differential equation is that

$$-\epsilon y''(x) + [q(x)y(x)]' + r(x)y(x) = f(x) \tag{81}$$

Subject to $y(0) = \theta$, and $y(1) = \mu$ where $0 < \epsilon \leq 1$,

ϵ is a positive parameter,θ, μ are known constants and$q(x), r(x)$ and $f(x)$ are assumed to be sufficiently differentiable function in [0, 1] and $p(x) \geq M > 0$ throughout the interval [0, 1], where M is some positive Constant. It is so attractive to mathematicians due to the fact that the solution exhibits a multi-scale character, that is, regions of rapid change in the solution near the end-points or the solution experiences the global phenomenon of rapid oscillation throughout the entire interval. Typically, these problems arise very frequently in fluid dynamics, elasticity, quantum mechanics, chemical reactor theory and many other allied areas. In recent years, there are wide classes of special purpose methods available for solving the above type problems. But this will be one of our future research works.

Table 6. Approximate solution (Series1), exact solution (Series2), and absolute error of Example 2

x values	Approximate solution (Cubic)	Exact solution	Absolute error
0.1	0.057301	0.0572	0.000101
0.2	0.0106357	0.1061	0.000257
0.3	0,146421	0.1460	0.000421
0.4	0.176383	0.1758	0.000583
0.5	0.194657	0.1940	0.00065
0.6	0.199044	0.1983	0.000744
0.7	0.186544	0.1858	0.000744
0.8	0.153113	0.1524	0.00071
0.9	0.093335	0.0928	0.000535

Table 7. Comparison of present method (Series3) with B-spline (Series2), finite difference (Series1) and with exact solution (Series4) for Example2

x values	Finite difference method	B-spline method	Cubic method	Exact solution
0.1	0.0399	0.05657	0.05730	0.0572
0.2	0.0897	0.104297	0.106357	0.1061
0.3	0,1302	0.1464167	0.146421	0.1460
0.4	0.1604	0.1763667	0.176383	0.1758
0.5	0.1787	0.193999	0.194657	0.1940
0.6	0.1827	0.1982966	0.199044	0.1983
0.7	0.1695	0.18655	0.186544	0.1858
0.8	0.1350	0.153771	0.15311	0.1524
0.9	0.0735	0.909366	0.093335	0.0928

5.1 The Minimum-norm Property of Splines

The draughtsman's spline assumes a shape which minimizes the potential energy due to the bending strain. The strain energy is approximately proportional to the integral of the square of the second derivatives along the path of the spline; and therefore, the minimization of the potential energy leads to a property of the minimum curvature. We can demonstrate that the Cubic spline has a similar property; which justifies us in likening it to the draughtsman's spline. Let $f(x) \in \mathrm{C}^2$ be any function defined over the interval $[x_o, x_N]$ which has a continuous Second-derivative. Then as a measure of the curvature the function, we may use the squared norm

$$\|f\|^2 = \int_{x_o}^{x_n} \{f''\}^2 \, dx \tag{82}$$

This differs from the ideal measure of curvature which would be the line integral of $\{ f''(x)\}^2$ along the path of the function. Thus, the squared norm provides only a rough approximation to the potential energy of the draughtsman's spline. Our objective is to show that, amongst all functions $f(x) \in \mathrm{C}^2$, which pass through the points (x_i, y_i)

: $i = 0, \dots, N$, it precisely the spline function which minimizes the squared norm. Let us denote the spline by $S(x)$,where $x \in [x_o, x_N]$, and let us continue to express the

$$i^{th}\text{Segment as} S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \tag{83}$$

Where $x \in [x_i, x_{i+1}]$ the derivatives of this function are

$$\begin{aligned} S_i'(x) &= 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \,, \\ S_i''(x) &= 6a_i(\mathrm{x} - x_i) + 2b_i \quad (84) \\ S_i'''(x) &= 6a_i \end{aligned} \tag{85}$$

We can establish the minimum-norm property of the cubic spline using the following result:
Let $f(x) \in C^2$ be a function defined on the interval $[x_o, x_N]$ whichpasses through the points $(x_i, y_i); i = 0, \dots, N$ which are the knots of the spline function $S(x)$,then

$$//f - S//^2 = //f//^2 - //S//^2 - 2\,[S''(x)\{f'(x) - S'(x)\}]$$

Proof: By definition, we have

$$\begin{aligned} ||f - S||^2 &= ||f||^2 - 2\int_{x_o}^{x_n} f''(x)\, S''(\mathrm{x})\, dx + ||S||^2 \\ &= \quad ||f||^2 - 2\int_{x_o}^{x_n} S''(\mathrm{x})\, \{f''(x)\} dx - ||S||^2 \end{aligned} \tag{86}$$

within this expression, we find, through integrating by parts, that

$$\int_{x_o}^{x_n} S''(\mathrm{x})\, \{f''(x) - S''(x)\} dx = [S''(x)\{f'(x) - S'(x)\}] - \int_{x_o}^{x_n} S''\{f'(x) - S'(x)\}\, dx \tag{87}$$

Since $S(x)$ consists of the cubic segments $S_i(\mathrm{x})$; $i = 0, \dots, N-1$ it follows that the third derivative $S'''(\mathrm{x})$ is constant in each open interval (x_i, x_{i+1}) with a value of

$S_i(x) = 6a_i$. Therefore

$$\begin{aligned} &\int_{x_o}^{x_n} S'''(x)\{f'(x) - S'(x)\}\, dx = \sum_{i=0}^{n-1} . \int_{x_i}^{x_{i+1}} 6a_i\, \{f'(x) - S'(x)\} dx \\ &= \sum_{i=0}^{n-1} 6a_i\, [\, f(x) - S(x)] = 0, \end{aligned} \tag{88}$$

Since $f(x) = S(x)$ at x_i and x_{i+1}; and hence (87) becomes

$$\int_{x_o}^{x_n} S''(x)\{f''(x) - S''(x)\} dx = [S''(x)\{f'(x) - S'(x)\}] \tag{89}$$

Putting this into (86) gives the result which we to prove,

Now consider the case of the natural spline which satisfies the conditions

$S''(x_0) = 0$ and $S''(x_N) = 0$. Putting these into the equality of (85) reduces it to

$$||f - S||^2 = ||f||^2 - ||S||^2, \quad (90)$$

This demonstrates that $\left||f|\right|^2 \geq \left||S|\right|^2$. In the case of a Clamped spline with

$S'(x_0) = f'(x_0)$ and $S'(x_N) = f'(x_N)$, the equality of (85) is also reduced to that of (90). Thus, we see that, in either case, the cubic spline has the minimum-norm property.

6. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this study, we have presented an overview of the methods of interpolation specifically the cubic spline interpolation, which is widely used in numerous real-world applications.

A cubic spline yields a third- degree polynomial connecting each pair of data points. The slopes (i.e., first derivatives) and curvature (i.e., second derivatives) of the cubic splines can be forced to be continuous at each data point. In fact, these requirements are necessary to obtain the additional conditions required to fit a cubic polynomial to two data points. Higher-degree splines can be defined in a similar manner. However, cubic splines have proven to be a good compromise between accuracy and complexity. Cubic spline interpolation is a powerful data analysis tool. Splines Correlate data efficiently and effectively, no matter how random the data may seem. Once the algorithm for spline generation is produced, interpolating data with a spline becomes an easy task.

Cubic spline method has been considered for numerical solution of boundary value problems of differential equations. The cubic spline has been tested on a problem. For example, the graph (Fig 20(b)) is plotted by comparing with the results obtained by other two methods (B-spline, and finite difference method). This shows that the accuracy of the cubic spline method is better than the accuracy of finite difference and B-spline method. All tables and figures clearly indicate that our numerical solution converges to the exact solution. We conclude that the cubic spline method is an applicable technique and approximates the solution very well, and the numerical solutions are in very good agreement with the exact solution. This method gives comparable results and is easy to compute. The cubic B-spline method produced an approximation of analytical solution of the problems with respect to the selected number of subintervals, While Finite difference method approximate the discrete solution of the problems with respect to the selected number of subintervals. Cubic B-spline method has potential of giving good approximation solution for two-point boundary value problem. This method is easily tractable and can readily be applied to other problems of differential equations. Finite difference method incorporates the boundary conditions given in the finite set of equations. The finite difference method derived based on finite difference approximations with truncation errors of $O(h^2)$. The finite difference method can be made better in approximations by using higher approximation for y'', y' (for example fifth order Taylor series) . Unfortunately, it will involve more computation to achieve higher accuracy approximation and possibility of having non-tri-diagonal system of equations. Although Finite difference method is accurate, it requires a way of solving systems of equations, which is a time-Consuming operation especially when the subinterval is larger. Several references given in this paper are of great practical importance but space constraints did not allow their discussion here. As conclusion, these three methods are acceptable as part of approximation solutions to two-point boundary value problem. Each method is structured and approximated differently which caused the different level of accuracy on each problem tested. If the two-point boundary value problems desired for approximation of analytical solution, cubic spline method has the prospective. Moreover, non-polynomial spline method has less computational cost over other polynomial spline methods. The implementation of the cubic spline method is a very easy, acceptable, and valid scheme.

6.2 Future Scope

Besides, the singular-perturbation differential equation and higher order of boundary value problems should be considered in the future work and the approximations of these methods should also be compared with other numerical methods in approximating to boundary value problems.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Burden RL, Faiers JD. Numerical analysis. 8th edition. United States: Thomson Brooks/Cole; 2005.
2. Roberts SM, Shipman JS. Two-point boundary value problems; Shooting Method. United States: American Elsevier; 1972.
3. Nagle RK, Staff EB. Fundamentals of Differential equations and boundary value problems. 2nd Edition, United States: Addison-Wesley; 1996.
4. Zill DG, Cullen MR. Differential equations with boundary-value problems.7th Edn. United States: Brooks/Cole; 2009.
5. Fyfe DJ. The use of cubic splines in the solution of two-point boundary Value problems. The Computer Journal. 1968;2011;12(2):188-192.
6. Al-Said EA. Cubic spline method for solving two-point boundary value problems. Korean Computational and Applied Mathematics. 1998.5(3):669-680.
7. Khan A. Parametric cubic spline solution of two-point boundary value problems. Applied Mathematics and Computation. 2004;175(1): 175-182.
8. Fang Q, Tsuchiya T, Yamamota T. Finite difference, finite element and finite volume methods applied to two-point boundary value problems. J. Comput. Appl. Math. 2002;139:9-19.
9. Caglar HN, Caglar, Elfaiture K. 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. 2006; 175:72-79.
10. Davis. B-splines and Geometric design. SIAM News. 1997;29(5).
11. Fan, Jiaguing, Yao, Qiwei. "Spline Methods". Nonlinear time series: non-parametric and parametric methods.Springes. 2005; 247.ISBN 978-0-387-26142-3.
12. Jaud, Kenneth L. Numerical methods in economics MT press. 1998;225. ISBN 978-0-262-10071-7.
13. Sky McKinley and Megan Levine, Cubic Spline Interpolation. Math 45: Linear Algebra.
14. Bickley WG. ′Piecewise cubic interpolation and two-point boundary problems. Computer Journal. 1968;11(2):206-208.
15. Balagurusamy E. Numerical methods, Tata Mc Grow –Hill Publishing Company, New Delhi. 1999; 302.
16. Richard L. Burden, Douglas Fairies J. Numerical Analysis (9th edition). International Edition. 1993; 147-157.
17. Rashidinia JR, Mohammadi, Jalilian R. Cubic spline method for two-point boundary value problems. Int. J. Eng. Sci, 2008;19(5-2):39-43.
18. Chang J, Yang Q, Zhao. Comparison of B-spline method and finite difference method to solve BVP of linear ODEs. J. Comput. 2011;6(10):2149-2155.
19. Salomon D. Curves and surfaces for computer graphics. United States: Springer; 2006.
20. Boor CD. A practical Guide to Splines. In Applied Mathematical Sciences 27(revised edition), United States: Springer-Verlag. 2001;87-107.
21. "Spline" Oxford English Dictionary (3rd edition.) Oxford University Press; September 2005.
22. Henrico P. Discrete variable methods in ordinary differential equations. Wiley, New York; 1962.
23. Manoj Kumar. A difference method for singular two-point boundary value problems. Applied Mathematics and Computation. 2003;879-884.

APPENDICES

MATLAB codes for the graphs and formulas used

- Computer programming (MATLAB) statement to plot the graph of $y = \cos(x)$.

$x = -pi: .1: \frac{pi}{2};$
$y = \cos(x)\,;$
$xi = linspace\ (pi: \frac{pi}{2});$
$yi = spline(x, y, xi);$
Plot $(x, y, 'o', xi, yi)$

- Computer programming (MATLAB) statement to plot the graph of $y = \exp$ (x).

$x = .5: 3;$
$y = \exp(x)\,;$
$xi = linspace\ (.5: 3);$
$yi = spline(x, y, xi);$
Plot $(x, y, 'o\ ', xi, yi)$

- Computer programming (MATLAB) statement to plot the graph of

$y = x(1 - \exp(x - 1))$
$x = 0: .1: 1;$
$y = x - x.* \exp(x - 1)\,;$
$xi = linspace\ (0: 1);$
$yi = spline(x, y, xi);$
Plot $(x, y, 'o\ ', xi, yi)$

- Computer programming (MATLAB) statement to plot the graph of

$y(x) = \left(\frac{1-e^{1+\sqrt{3}}}{e^{1+\sqrt{3}}-e^{1-\sqrt{3}}}\right)e^{(1-\sqrt{3})x} + \left(\frac{e^{1-\sqrt{3}}-1}{e^{1+\sqrt{3}}-e^{1-\sqrt{3}}}\right)e^{(1+\sqrt{3})x}\ +1$
$x = 0: .1: 1;$
$y = [\frac{1-\exp(1+sqrt(3))}{\exp(1+sqrt(3))-\exp(1-sqrt(3))}\ ].* \exp(1 - sqrt(3)) .* x\ +[\ \frac{\exp(1-sqrt(3))-1}{\exp(1+sqrt(3))-\exp(1-sqrt(3))}\ ].* exp(1 +$
$sqrt3.*x+1$
$xi = linspace(0: 1);$
$yi = spline(x, y, xi);$
Plot $(x, y, 'o\ ', xi, yi)$

- Computer programming (MATLAB) statement to plot the graph of $y = \sin(x)$

$x = 0: .1: 3 * \frac{pi}{2};$
$y = \sin(x)\,;$
$xi = linspace(0: \frac{pi}{2};);$
$yi = spline(x, y, xi);$
Plot $(x, y, 'o\ ', xi, yi)$

- Computer programming (MATLAB) statement to plot the graph of natural cubic spline interpolation.

Plot [List Interpolation [{1, E, E^2, E^3},{{0,3}} Interpolation Order→ 3][x], {x, 0,3}]

Clear;

$L = 1$

$N = 10;$

$h = \frac{L}{N};$

$y(0) = y0 = 0$

$y(10) = y10 = 0$

for$i = 1{:}N;$

$x(xi + 1) = xi + h;$

$yexact(i) = x - x.* \exp\,(x - 1)(i));$

$$y(i+1) = \left(\frac{3h}{2}q_{i-1} + hq_i - \frac{h}{2}q_{i+1} - h^2 r_{i-1} - 1\right)y_{i-1} + (-2\mathrm{h}q_{i-1} + 2\mathrm{h}q_{i+1} - h^2 2r_i + 2)y_i + \left(\frac{h}{2}q_{i-1} - \mathrm{h}q_i - \frac{3h}{2}q_{i+1} - h^2 r_{i+1} - 1\right)y_{i+1} = -h^2(f_{i-1} + 2f_i + f_i),$$

$E(i) = abs(yexact(i) - y(i);$

$x = 0{:}.1{:}\,1;$

$y = x - x.* \exp(x - 1)\,;$

Plot $(x, y, x, yexact)$

end

Clear;

$L = 1$

$N = 10;$

$h = \frac{L}{N};$

$y(0) = yo = 0$

$y(10) = y10 = 0$

for$i = 1{:}N;$

$x(xi + 1) = xi + h;$

$yexact(i) =$

$[\frac{1-\exp(1+sqrt(3))}{\exp(1+sqrt(3))-\exp(1-sqrt(3))}\ ].*\ \exp\big(1 - sqrt(3)\big).* x(i)\ +[\ \frac{\exp(1-sqrt(3))-1}{\exp(1+sqrt(3))-\exp(1-sqrt(3))}\ ].*\ exp\big(1 + sqrt3.*x(i)+1$

$$y(i+1) = y(i+1) = \left(\frac{3h}{2}q_{i-1} + hq_i - \frac{h}{2}q_{i+1} - h^2 r_{i-1} - 1\right)y_{i-1} + (-2\mathrm{h}q_{i-1} + 2\mathrm{h}q_{i+1} - h^2 2r_i + 2)y_i + \left(\frac{h}{2}q_{i-1} - \mathrm{h}q_i - \frac{3h}{2}q_{i+1} - h^2 r_{i+1} - 1\right)y_{i+1} = -h^2(f_{i-1} + 2f_i + f_i),$$

$E(i) = abs(yexact(i) - y(i);$

$x = 0{:}.1{:}\,1;$

$$y = [\frac{1 - \exp\big(1 + sqrt(3)\big)}{\exp\big(1 + sqrt(3)\big) - \exp\big(1 - sqrt(3)\big)}].* \exp\big(1 - sqrt(3)\big).* x + [\frac{\exp\big(1 - sqrt(3)\big) - 1}{\exp\big(1 + sqrt(3)\big) - \exp\big(1 - sqrt(3)\big)}].* exp\big(1 + sqrt(3)\big).* x + 1;$$

Plot $(x, y, x, yexact)$

end

- M-file to determine intermediate values and derivatives with a natural spline function$(yy, dy, d2) = natspline(x, y, xx)$

% natspline: natural spline with differentiation
% $(yy, dy, d2) = natspline(x, y, xx)$: uses a natural cubic spline
%interpolation to find yy, the values of the underlying function
%$y$at the points in the vector xx. The vector x specifies the points at which % the data y is given.
% Input:
%x =vector of independent variables%
%y =vector of dependent variables
% xx =vector of desired values of dependent variables

% yy =interpolated values at xx
% dy =first derivatives at xx
% $d2$ =second derivative at xx
N =length(x);
If length$(y) = N$, error (‘and$y$ must be same length ‘) ; end
If any $(diff(x) <= 0)$, error (‘x not strictly ascending ‘), end
$m = length(xx)$;
$b = zeros\ (n, n)$;
$aa(1,1) = 1; aa(n, n) = 1;$ % Set up eq.
$bb(1) = 0; bb(n) = 0;$
$for\ i = 2: n - 1$
$aa(i, i - 1) = h(x, i - 1);$
$aa(i, i) = 2 * \big(h(x, i - 1) + h(x, i)\big);$
$aa(i, i + 1) = h(x, i);$
$bb(i) = 3 * \big(fd(i + 1.i, x, y) - fd(i, i - 1, x, y)\big);$
end
$c = \frac{aa}{bb'}$; %solve for c coefficients
for$i = 1: n - 1$ % solve for $a, b\ and\ d$ coefficients
$a(i) = y(i);$
$b(i) = fd(i + 1, i, x, y) - \frac{h(x, i)}{3} * \big(2 * c(i) + c(i + 1)\big);$
$d(i) = (c(i + 1) - \frac{\frac{c(i)}{3}}{h(x, i)};$
end
for$i = 1: m$ % perform interpolation at desired values
$(yy(i), dy(i), d2(i) = splineinterp\big(x, n, a, b, c, d, xx(i)\big);$
end
end
function $(hh = h(x, i)$
$(hh = x(i + 1) - x(i);$
end
$function\ fdd = fd(i, j, x, y)$
$fdd = \frac{y(i) - y(j)}{x(i) - x(j)};$
end
$function\ [yyy, dyy, d2y] = splineIterp(x, n; a, b, c, xi)$
$for\ ii = 1: n - 1$
If $xi >= x(ij) - 0{,}000001\ \&xi <= x(ii + 1) + 0.000001$
$yyy = a(ii) + b(ii) + (xi - x(ii) + c(ii) * \big(xi - x(ii)\big)^2 + d(ii) * \big(xi - x(ii)\big)^3;$
$dyy = b(ii) + 2 * c(ii) * (xi - x(ii))) + 3 * d(ii) * (xi - x(ii)))$^2;
$d2y = 2 * c(ii) + 6 * d(ii) * \big(xi - x(ii)\big);$
$break$
end
end
end

$t = [0\ \ 0.1\ \ 0.499\ \ 0.5\ \ 0.6\ \ 1.0\ \ 1.4\ \ 1.5\ \ 1.899\ \ 1.9\ \ 2.0];$
$G = [0\ \ 0.06\ \ 0.17\ \ 0.19\ \ 0.21\ \ 0.26\ \ 0.29\ \ 0.29\ \ 0.30\ \ 0.31\ \ 0.31];$
$tt = lispace\left(t(.1), t\left(length(t)\right)\right);$
$(GG, dG, dG2) = naspline\ (t, G, tt);$
$plot(G, t,' o', GG, tt)$