5-2014

Cubic B-spline: Derivation with two applications

Maria Mungia
University of Texas-Pan American

CUBIC B-SPLINE:

DERIVATION WITH TWO APPLICATIONS

A Thesis

by

MARIA MUNGUIA

Submitted to the Graduate School of
The University of Texas-Pan American
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2014

Major Subject: Mathematics

CUBIC B-SPLINE:

DERIVATION WITH TWO APPLICATIONS

A Thesis
by
MARIA MUNGUIA

COMMITTEE MEMBERS

Dr. Dambaru Bhatta
Chair of Committee

Dr. Andras Balogh
Committee Member

Dr. Paul Bracken
Committee Member

Dr. Bao-Feng Feng
Committee Member

May 2014

ABSTRACT

Munguia, Maria, Cubic B-Spline: Derivation with Two Applications. Master of Science (MS), May, 2014, 58 pp., 12 tables, 20 figures, 13 references, 14 titles.

We investigate the application of cubic B-spline functions in this work. First, we derive the cubic B-spline functions. Then we use these functions for two applications: one for interpolation and the other for solving boundary value problems. For interpolation, we consider two types of boundary conditions, namely natural and clamped. For boundary value problems, we consider constant coefficient and variable coefficient cases. Examples with numerical results are presented for both applications. Also we compare the results obtained using cubic B-spline functions with the results obtained by other methods. For this investigation, it is observed that cubic B-spline gives better results.

DEDICATION

This thesis is dedicated to my lovely family especially my parents and my husband, Daniel. It was their unconditional love and support that motivated me to continue my education. I also dedicate this thesis to my uncle, Antonio. He always made sure I had everything I needed to succeed in college. Thank you all.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to the committee members especially my advisor Dr. Dambaru Bhatta. I would like to thank him for his motivation, support, and infinite patience.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

We study the uses of cubic B-spline functions for two applications: one for interpolation and the other for solving second-order linear boundary value problems. The use of B-splines have become very popular among many areas of mathematics, engineering, and computer science in recent years. Originally B-splines were used for approximation purposes, but their popularity have extended their applications and one of them is to solve differential equations. The most popular one is the cubic B-spline (Cheng & Zizhi, 1990).

We have studied interpolatory methods such as Lagrange interpolation, Neville's method, and divided differences. These methods use a single polynomial to approximate a function over a finite interval. For these methods, when more accuracy is needed, we increase the degree of the interpolant. This may lead to oscillations near the end points. Cubic B-spline interpolation avoids this kind of instability by using low-degree polynomials joined together such that they fit well.

The first mathematician who introduced the concept of splines was Isaac Jacob Schoenberg in 1946. This Romanian-American mathematician was known as the father of splines. His work was a motivation to other mathematicians such Carl de Boor who worked directly with Schoenberg. De Boor, Birkhoff and Hall studied the error bound and convergence of spline interpolation (Birkhoff & De Boor, 1964), (Hall, 1968). In the early 1970's de Boor also introduced a recursive definition for splines. Now splines, especially B-splines play an important role in the areas of mathematics and engineering. Splines are popular in computer graphing due to their smoothness, flexibility, and accuracy.

Approximated solutions to second-order linear boundary value problems have been obtained using different types of methods. For example, Raghavarao and Sanyasiraju had shown solutions of such equations using cubic splines (Raghavarao, Sanyasiraju, & Suresh, 1993). Fang, Tsuchiya, and Yamamoto have presented solutions to second-order boundary value problems using different methods such as the finite difference and the finite element methods (Fang, Tsuchiya, & Yamamoto, 2001). Bhatta, Bhatti and Bracken showed solutions of differential equations using Bernstein polynomial basis (Bhatti & Bracken, 2006), (Bhatta & Bhatti 2006), (Bhatta, 2008). This last method and the B-spline method share similar properties. In our case, we seek to approximate solutions to second-order linear boundary value problems with nonhomogeneous boundary conditions using cubic B-splines.

Chapter II introduces the cubic B-spline and its notation. Derivations of the cubic B-spline interpolation formula are presented using two types of boundary conditions and corresponding results and discussions are presented in Chapters III and IV. Chapters V and VI present the general solution to a second-order linear boundary value problem with nonhomogeneous boundary conditions using the cubic B-spline method, as well as, results and discussions to demonstrate the performance of our method.

CHAPTER II

MATHEMATICAL FORMULATION FOR CUBIC B-SPLINES

In this chapter, we introduce the cubic B-spline and some of its notations, properties and definitions. Let us start the section by defining a partition $\Delta_N : a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$ on a given interval $[a, b]$ where N is the number of intervals and let $h = \frac{b-a}{N}$ be the mesh size of the partition. Then Definition 1 defines a spline function of degree k.

Definition 1: Given a partition Δ_N, a piecewise polynomial function s on the interval $[a, b]$ is called a spline of degree k if $s \in C^{k-1}[a, b]$ and s is a polynomial of degree at most k on each subinterval $[x_i, x_{i+1}]$.

Let $S_k(\Delta_N)$ denote the set of all polynomials of degree k associated with Δ_N. This set is a linear space with respect to Δ_N of dimension $N + k$.

Now that we have defined spline functions, we will introduce a special kind of spline functions called B-spline of degree 3. These spline functions form a basis for the set of all cubic splines. Definition 2 defines the recursive relation introduced by Carl de Boor in the early 1970's. This definition will help us construct the cubic B-spline we will be working with from this point.

Definition 2: The B-splines of degree zero are defined by

$$B_i^0(x) = \begin{cases} 1 & \text{if } x_i \leq x < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and those of degree $k \in \mathbb{Z}^+$, are defined recursively in terms of B-splines of degree $k - 1$ by

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

for $i = 0, \pm 1, \pm 2, \pm 3, \ldots$ (Phillips 2003). The basis functions B_i^k in (1) are called B-splines of degree k.

Using the recurrence relation (1) and assuming the partition Δ_N, we can derive the non-uniform cubic B-spline as

$$B_i^3(x) = \begin{cases} \frac{(x-x_i)^3}{(x_{i+3}-x_i)(x_{i+2}-x_i)(x_{i+1}-x_i)} & \text{if } x_i \le x < x_{i+1} \\ \frac{(x-x_i)^2(x_{i+2}-x)}{(x_{i+3}-x_i)(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} + \frac{(x-x_i)(x_{i+3}-x)(x-x_{i+1})}{(x_{i+3}-x_i)(x_{i+3}-x_{i+1})(x_{i+2}-x_{i+1})} & \\ \quad + \frac{(x_{i+4}-x)(x-x_{i+1})^2}{(x_{i+4}-x_{i+1})(x_{i+3}-x_{i+1})(x_{i+2}-x_{i+1})} & \text{if } x_{i+1} \le x < x_{i+2} \\ \frac{(x-x_i)(x_{i+3}-x)^2}{(x_{i+3}-x_i)(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})} + \frac{(x_{i+4}-x)(x-x_{i+1})(x_{i+3}-x)}{(x_{i+4}-x_{i+1})(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})} & \\ \quad + \frac{(x_{i+4}-x)^2(x-x_{i+2})}{(x_{i+4}-x_{i+1})(x_{i+4}-x_{i+2})(x_{i+3}-x_{i+2})} & \text{if } x_{i+2} \le x < x_{i+3} \\ \frac{(x_{i+4}-x)^3}{(x_{i+4}-x_{i+1})(x_{i+4}-x_{i+2})(x_{i+4}-x_{i+3})} & \text{if } x_{i+3} \le x < x_{i+4} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

This is a cubic spline with knots $x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}$. Note that the cubic B-spline is zero except on the interval $[x_i, x_{i+4})$. This is true for all B-splines. In fact, $B_i^k(x) = 0$ if $x \notin [x_i, x_{i+k+1})$ otherwise $B_i^k(x) > 0$ if $x \in (x_i, x_{i+k+1})$.

Since we will only be referring to B-splines of degree 3, we will write B_i instead of B_i^3. In our case, we restrict our attention to equally-spaced knots. Therefore, after including four

additional knots, we will assume that

$\Delta : x_{-2} < x_{-1} < x_0 < x_1 < \cdots < x_{N-1} < x_N < x_{N+1} < x_{N+2}$ is a uniform partition. Using (2) and letting $h = x_{i+1} - x_i$ for any $0 \leq i \leq N$, we define the uniform cubic B-spline, $B_i(x)$, as

$$B_i(x) = \frac{1}{6h^3} \begin{cases} (x - x_{i-2})^3 & \text{if } x_{i-2} \leq x < x_{i-1} \\ -3(x - x_{i-1})^3 + 3h(x - x_{i-1})^2 + 3h^2(x - x_{i-1}) + h^3 & \text{if } x_{i-1} \leq x < x_i \\ -3(x_{i+1} - x)^3 + 3h(x_{i+1} - x)^2 + 3h^2(x_{i+1} - x) + h^3 & \text{if } x_i \leq x < x_{i+1} \\ (x_{i+2} - x)^3 & \text{if } x_{i+1} \leq x < x_{i+2} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and its graph is shown in Figure 1

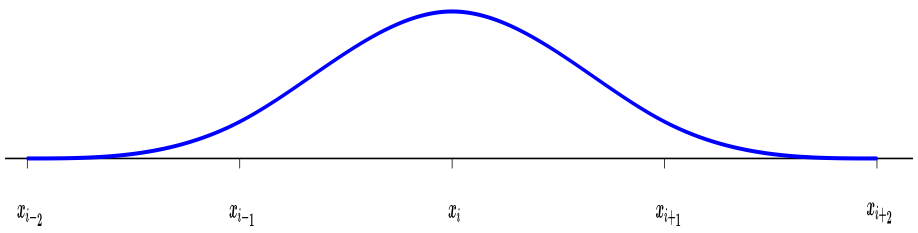


Figure 1: Graph of the Cubic B-spline

The basis functions $B_i(x)$ are twice continuously differentiable on $[a, b]$. This method has error of order $O(h^4)$. Next we will derive the cubic B-spline interpolation formula and give some results. Then in later chapters we will derive the cubic B-spline method for approximating the solution to second-order linear boundary value problems and give some results.

CHAPTER III

CUBIC B-SPLINES TO INTERPOLATE

In this section, we will construct a cubic B-spline function which interpolates the function $f(x)$ on the interval $[a, b]$. Let $S_3(x)$ be the cubic spline with knots Δ. Then $S_3(x)$ can be written as linear combinations of the basis functions satisfying

$$S_3(x_j) = f(x_j) \text{ for all } j = 0, \cdots, N. \tag{4}$$

This cubic B-spline function is defined below

$$S_3(x) \quad = \quad \sum_{i=-1}^{N+1} a_i B_i(x) \tag{5}$$

where the constants a_i are to be determined and the functions $B_i(x)$ are basis functions such that $S_3(x) \in S_3(\Delta)$. These basis functions are defined in (3).

We know define the constants a_i. Since we have N subintervals $[x_i, x_{i+1}]$ and every cubic spline has 4 unknowns, it follows that there is a total of $4N$ degrees of freedom. When fitting the data points in (4), we have $N + 1$ interpolatory conditions. From definition 1, we know that B, B', and B'' are continuous at the interior points, therefore we have $3(N - 1)$ continuity conditions. This gives a total of

$$(N + 1) + 3(N - 1) = 4N - 2$$

constrains. Since two more constrains need to be satisfied, we require boundary conditions. In this case, we will be working with two types of boundary conditions. These conditions are as follow:

Case I: $S_3''(x_0) = 0$ and $S_3''(x_N) = 0$ is called natural spline or free spline;

Case II: $S_3'(x_0) = f'(x_0)$ and $S_3'(x_N) = f'(x_N)$ is called complete spline or clamped spline.

Some advantages of Case I are that it is easier to work with these boundary conditions and these conditions do not require any knowledge of either values of the derivative at the end points. On the other hand even if we are have no knowledge of the derivatives at the end points, in order to apply Case II, we can use numerical approximations to obtain an accurate approximation for those values. Case II leads to more accurate approximations than Case I especially at the end points since Case II includes more information about the function (Zarowski 2004).

To determine the coefficients a_i, we need to evaluate $S_3(x)$ at $x = x_i$. Note that the interval $[x_i, x_{i+1})$ has nonzero contributions from B_{i-1}, B_i, B_{i+1} and B_{i+2}. We have a better understanding of this from Figure 2.

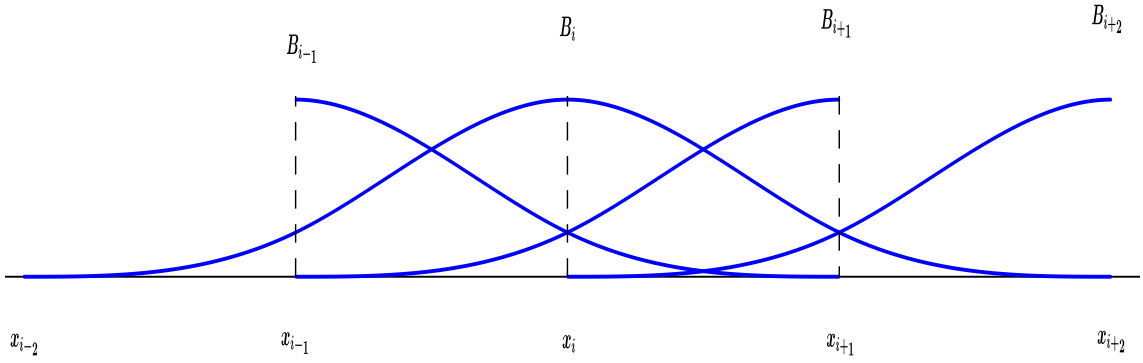


Figure 2: Graphs of Basis Functions Affecting the Interval $[x_i, x_{i+1})$

Equations (4) and (5) and definition (3) of B-splines yield

$$S_3(x_i) = a_{i-1}B_{i-1}(x_i) + a_iB_i(x_i) + a_{i+1}B_{i+1}(x_i) + a_{i+2}B_{i+2}(x_i) = f(x_i)$$

where

$$\begin{aligned} B_{i-1}(x_i) &= \frac{1}{6} \\ B_i(x_i) &= \frac{2}{3} \\ B_{i+1}(x_i) &= \frac{1}{6} \\ B_{i+2}(x_i) &= 0 \end{aligned}$$

and therefore, for any $i = 0, \ldots, N$, we obtain

$$a_{i-1} + 4a_i + a_{i+1} = 6\, f(x_i). \tag{6}$$

We now need to apply the boundary conditions. Let us start with Case I which is the natural spline.

Case I: Natural Cubic B-Spline

Using (5) and Case I, the second derivative of $S_3(x)$ is given by

$$S_3''(x) = \sum_{i=-1}^{N+1} a_i B_i''(x)$$

where

$$B_i''(x) = \frac{1}{h^3} \begin{cases} x - x_{i-2} & \text{if } x_{i-2} \le x < x_{i-1} \\ -3(x - x_{i-1}) + h & \text{if } x_{i-1} \le x < x_i \\ -3(x_{i+1} - x) + h & \text{if } x_i \le x < x_{i+1} \\ x_{i+2} - x & \text{if } x_{i+1} \le x < x_{i+2} \\ 0 & \text{otherwise} \end{cases}$$

The natural boundary conditions yield the following equations

$$S_3''(x_0) = a_{-1}B_{-1}''(x_0) + a_0 B_0''(x_0) + a_1 B_1''(x_0) + a_2 B_2''(x_0) = 0$$
$$S_3''(x_N) = a_{N-1}B_{N-1}''(x_N) + a_N B_N''(x_N) + a_{N+1} B_{N+1}''(x_N) + a_{N+2} B_{N+2}''(x_N) = 0$$

for $x = x_0$ and $x = x_N$, we obtain

$$B_{-1}''(x_0) = 1 = B_{N-1}''(x_N)$$
$$B_0''(x_0) = -2 = B_N''(x_N)$$
$$B_1''(x_0) = 1 = B_{N+1}''(x_N)$$
$$B_2''(x_0) = 0 = B_{N+2}''(x_N)$$

therefore the end conditions for Case I are given below

$$a_{-1} - 2a_0 + a_1 = 0 \tag{7}$$
$$a_{N-1} - 2a_N + a_{N+1} = 0. \tag{8}$$

Now using (6), (7), and (8), we obtain a linear matrix representation where the coefficient matrix is order $(N+3) \times (N+3)$.

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{-1} \\ a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{N-3} \\ a_{N-2} \\ a_{N-1} \\ a_N \\ a_{N+1} \end{pmatrix} = 6 \begin{pmatrix} \frac{1}{2}f(x_0) \\ f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{N-3}) \\ f(x_{N-2}) \\ f(x_{N-1}) \\ f(x_N) \\ \frac{1}{2}f(x_N) \end{pmatrix} \tag{9}$$

or $M\overline{a} = 6\,\overline{b}$.

The matrix M is a strictly diagonally dominant matrix, and therefore M is nonsingular. The system has a unique solution. Thus, the natural cubic B-spline interpolating f at the knots Δ is obtained with (5) where the constant coefficients satisfy the tridiagonal system defined in (9).

Case II: Complete Cubic B-Spline

Now let us look at Case II. Using (5), the first derivative of $S_3(x)$ is given by

$$S_3'(x) = \sum_{i=-1}^{N+1} a_i B_i'(x)$$

where

$$B_i'(x) = \frac{1}{2h^3} \begin{cases} (x - x_{i-2})^2 & \text{if } x_{i-2} \le x < x_{i-1} \\ -3(x - x_{i-1})^2 + 2h(x - x_{i-1}) + h^2 & \text{if } x_{i-1} \le x < x_i \\ 3(x_{i+1} - x)^2 - 2h(x_{i+1} - x) - h^2 & \text{if } x_i \le x < x_{i+1} \\ -(x_{i+2} - x)^2 & \text{if } x_{i+1} \le x < x_{i+2} \\ 0 & \text{otherwise} \end{cases}$$

The complete boundary conditions yield the following equations

$$S_3'(x_0) = a_{-1}B_{-1}'(x_0) + a_0 B_0'(x_0) + a_1 B_1'(x_0) + a_2 B_2'(x_0) = f'(x_0)$$
$$S_3'(x_N) = a_{N-1}B_{N-1}'(x_N) + a_N B_N'(x_N) + a_{N+1} B_{N+1}'(x_N) + a_{N+2} B_{N+2}'(x_N) = f'(x_N)$$

for $x = x_0$ and $x = x_N$, we obtain

$$\begin{aligned} B_{-1}'(x_0) &= -\frac{1}{2h} = B_{N-1}'(x_N) \\ B_0'(x_0) &= \quad 0 \quad = B_N'(x_N) \\ B_1'(x_0) &= \frac{1}{2h} = B_{N+1}'(x_N) \\ B_2'(x_0) &= \quad 0 \quad = B_{N+2}'(x_N) \end{aligned}$$

therefore the end conditions for Case II are

$$-a_{-1} + a_1 = 2hf'(x_0) \tag{10}$$

$$-a_{N-1} + a_{N+1} = 2hf'(x_N). \tag{11}$$

In a similar manner as in Case I, using (6), (10), and (11), we obtain a linear matrix representation (12) where the coefficient matrix is order $(N+3) \times (N+3)$.

$$\begin{pmatrix} -1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{-1} \\ a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{N-3} \\ a_{N-2} \\ a_{N-1} \\ a_N \\ a_{N+1} \end{pmatrix} = 6 \begin{pmatrix} \frac{h}{3} f'(x_0) \\ f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{N-3}) \\ f(x_{N-2}) \\ f(x_{N-1}) \\ f(x_N) \\ \frac{h}{3} f'(x_N) \end{pmatrix} \tag{12}$$

or $M\overline{a} = 6\,\overline{b}$.

The system has a unique solution. Thus, the complete cubic B-spline function which interpolates the function f at the values Δ is obtained with (5) where the constant coefficients satisfy the tridiagonal system defined in (12).

Lagrange Interpolation Definition

In the next Chapter, we will compare our cubic B-spline interpolation results with the Lagrange interpolation results. For this reason it is important to give a brief definition of the Lagrange interpolation.

Definition 3: Given $N+1$ values of the function f at the distinct knots $x_0 < x_1 < \cdots < x_{N-1} < x_N$, there is a unique polynomial $P(x)$ of degree at most N passing through the $N+1$ points with

$$f(x_k) = P(x_k) \text{ for each } k = 0, 1, \ldots, N.$$

This polynomial is given by

$$P(x) = \sum_{k=0}^{N} f(x_k) L_{N,k}, \tag{13}$$

where

$$L_{N,k} = \prod_{i=0, i\neq k}^{N} \frac{(x - x_i)}{(x_k - x_i)}.$$

The polynomial (13) is called the Lagrange interpolating polynomial (Burden & Faires, 2011).

CHAPTER IV

RESULTS AND DISCUSSIONS FOR INTERPOLATION

In this chapter, we present computational results to the methods presented in chapter III. The program used to implement these results was coded in Java, the cubic polynomials were created in Mathematica, and the figures were created in Geogebra and CoPlot. We tested the methods on certain functions, and in this chapter, we present a few of them.

In the first example, we interpolate a function $f(x)$ on the interval $[a, b]$ using both the natural and the complete cubic B-spline interpolants. Then we compare the results and determine which one of the interpolants gives a better approximation. In the second example, we again interpolate a function $f(x)$ on the interval $[a, b]$ using both the natural and complete cubic B-splines. In this case, we compare the results with the Lagrange interpolation results. Finally, in the third example, given a set of data points obtained from an unknown function, we estimate values in between the knots at different values of h using the natural cubic B-spline interpolation. We then compare the results with the Lagrange interpolation and determine which method gives a better approximation to the unknown function.

Example 1a: Consider the function $f(x) = e^{-x}$ on the interval $[0, 2]$. We want to interpolate $f(x)$ using the natural cubic B-spline interpolation at 11 knots. Since there are 11 knots, we have 10 subintervals and $h = 0.2$. In this case, we first need to find the constant coefficients a_i for $i = -1, 0, 1, \ldots, 10, 11$. Using the system of linear equations (9) where M is a 13×13 matrix, we obtain

$$\begin{array}{lllll} a_{-1} = 1.188489670 & a_0 = 1.000000000 & a_1 = 0.811510330 & a_2 = 0.666343197 & a_3 = 0.545037157 \\ a_4 = 0.446377993 & a_5 = 0.365424656 & a_6 = 0.299200029 & a_7 = 0.244940499 & a_8 = 0.200619759 \\ a_9 = 0.163959572 & a_{10} = 0.135335283 & a_{11} = 0.106710995 & & \end{array}$$

Then the natural cubic B-spline interpolation polynomials are as follow

$$S_3(x) = \begin{cases} 0.90255x^3 - 0.000000000000004x^2 - 0.94245x + 1 & \text{for } x \in [0, 0.2) \\ -0.40545x^3 + 0.78480x^2 - 1.09941x + 1.01046 & \text{for } x \in [0.2, 0.4) \\ -0.02530x^3 + 0.32862x^2 - 0.91694x + 0.98613 & \text{for } x \in [0.4, 0.6) \\ -0.10294x^3 + 0.46838x^2 - 1.00079x + 1.00291 & \text{for } x \in [0.6, 0.8) \\ -0.06202x^3 + 0.37018x^2 - 0.92223x + 0.98196 & \text{for } x \in [0.8, 1.0) \\ -0.05758x^3 + 0.35683x^2 - 0.90889x + 0.97751 & \text{for } x \in [1.0, 1.2) \\ -0.04221x^3 + 0.30154x^2 - 0.84253x + 0.95097 & \text{for } x \in [1.2, 1.4) \\ -0.04746x^3 + 0.32358x^2 - 0.87339x + 0.96537 & \text{for } x \in [1.4, 1.6) \\ 0.00782x^3 + 0.05822x^2 - 0.44882x + 0.73893 & \text{for } x \in [1.6, 1.8) \\ -0.16741x^3 + 1.00449x^2 - 2.15210x + 1.76089 & \text{for } x \in [1.8, 2.0) \end{cases}$$

and Table 1 shows the results obtained from these polynomials.

i	x_i	Natural Cubic B-Spline	Exact	$\mid f(x) - S_3(x) \mid$
0	0.00	1.000000000	1.000000000	0.000000000
	0.05	0.952990402	0.951229425	0.001760977
	0.10	0.906657718	0.904837418	0.001820300
	0.15	0.861678864	0.860707976	0.000970887
1	0.20	0.818730753	0.818730753	0.000000000
	0.25	0.778326801	0.778800783	0.000473982
	0.30	0.740326423	0.740818221	0.000491798
	0.35	0.704425533	0.704688090	0.000262557
2	0.40	0.670320046	0.670320046	0.000000000
	0.45	0.637753396	0.637628152	0.000125245
	0.50	0.606659093	0.606530660	0.000128433
	0.55	0.577018164	0.576949810	0.000068353
3	0.60	0.548811636	0.548811636	0.000000000
	0.65	0.522010833	0.522045777	0.000034944
	0.70	0.496548256	0.496585304	0.000037048
	0.75	0.472346701	0.472366553	0.000019852
4	0.80	0.449328964	0.449328964	0.000000000
	0.85	0.427422956	0.427414932	0.000008024
	0.90	0.406577044	0.406569660	0.000007384
	0.95	0.386744712	0.386741023	0.000003688
5	1.00	0.367879441	0.367879441	0.000000000
	1.05	0.349935271	0.349937749	0.000002478
	1.10	0.332868464	0.332871084	0.000002620
	1.15	0.316635838	0.316636769	0.000000932
6	1.20	0.301194212	0.301194212	0.000000000
	1.25	0.286502325	0.286504797	0.000002472
	1.30	0.272526595	0.272531793	0.000005198
	1.35	0.259235362	0.259240261	0.000004899
7	1.40	0.246596964	0.246596964	0.000000000
	1.45	0.234579085	0.234570288	0.000008796
	1.50	0.223146782	0.223130160	0.000016622
	1.55	0.212264459	0.212247974	0.000016485
8	1.60	0.201896518	0.201896518	0.000000000
	1.65	0.192014272	0.192049909	0.000035637
	1.70	0.182616675	0.182683524	0.000066849
	1.75	0.173709592	0.173773943	0.000064351
9	1.80	0.165298888	0.165298888	0.000000000
	1.85	0.157368524	0.157237166	0.000131357
	1.90	0.149814842	0.149568619	0.000246223
	1.95	0.142512282	0.142274072	0.000238211
10	2.00	0.135335283	0.135335283	0.000000000

Table 1: Natural Cubic B-spline Approximation for Ex 1a

Figure 3 shows the graph of the natural cubic B-spline absolute error,

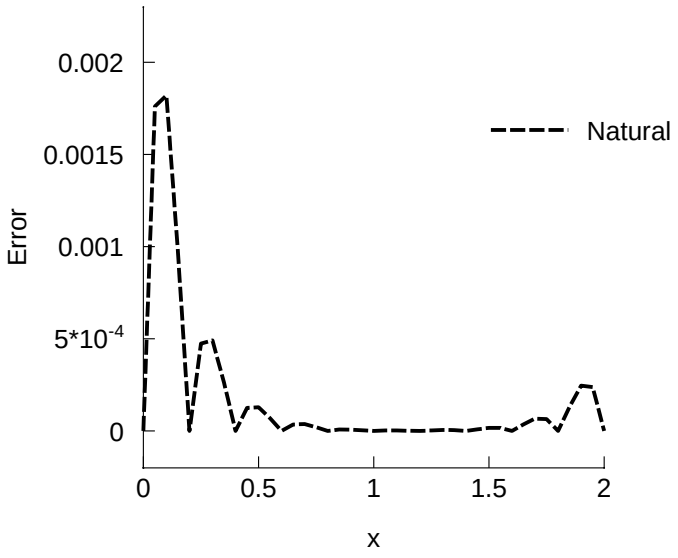


Figure 3: Natural Cubic B-spline Error for Ex 1a

by definition, the natural condition is less accurate near the end points. We see that in this graph.

Example 1b: Now consider the same function $f(x) = e^{-x}$ on the interval $[0, 2]$. We want to interpolate $f(x)$ at 11 knots, but this time using the complete cubic B-spline interpolation. Again, since there are 11 knots, we have 10 subintervals and $h = 0.2$. Using the system of linear equations (12) where M is a 13×13 matrix, we obtain

$$a_{-1} = 1.213290991 \quad a_0 = 0.993354504 \quad a_1 = 0.813290991 \quad a_2 = 0.665866048 \quad a_3 = 0.545165091$$
$$a_4 = 0.446343404 \quad a_5 = 0.365435077 \quad a_6 = 0.299192935 \quad a_7 = 0.244958454 \quad a_8 = 0.200555032$$
$$a_9 = 0.164200527 \quad a_{10} = 0.134436190 \quad a_{11} = 0.110066414$$

Then the complete cubic B-spline interpolation functions are as follow

$$S_3(x) = \begin{cases} -0.15072x^3 + 0.49841x^2 - 1x + 1 & \text{for } x \in [0, 0.2) \\ -0.12322x^3 + 0.48191x^2 - 0.99670x + 0.99978 & \text{for } x \in [0.2, 0.4) \\ -0.10093x^3 + 0.45517x^2 - 0.98600x + 0.99835 & \text{for } x \in [0.4, 0.6) \\ -0.08262x^3 + 0.42221x^2 - 0.96623x + 0.99440 & \text{for } x \in [0.6, 0.8) \\ -0.06765x^3 + 0.38628x^2 - 0.93748x + 0.98673 & \text{for } x \in [0.8, 1.0) \\ -0.05539x^3 + 0.34949x^2 - 0.90069x + 0.97447 & \text{for } x \in [1.0, 1.2) \\ -0.04535x^3 + 0.31334x^2 - 0.85732x + 0.95712 & \text{for } x \in [1.2, 1.4) \\ -0.03713x^3 + 0.27883x^2 - 0.80899x + 0.93457 & \text{for } x \in [1.4, 1.6) \\ -0.03039x^3 + 0.24649x^2 - 0.75725x + 0.90697 & \text{for } x \in [1.6, 1.8) \\ -0.02491x^3 + 0.21688x^2 - 0.70397x + 0.87500 & \text{for } x \in [1.8, 2.0) \end{cases}$$

and Table 2 shows the results obtained from these polynomials.

i	x_i	Complete Cubic B-Spline	Exact	$\mid f(x) - S_3(x) \mid$
0	0.00	1.000000000	1.000000000	0.000000000
	0.05	0.951227191	0.951229425	0.000002234
	0.10	0.904833405	0.904837418	0.000004013
	0.15	0.860705605	0.860707976	0.000002371
1	0.20	0.818730753	0.818730753	0.000000000
	0.25	0.778799249	0.778800783	0.000001534
	0.30	0.740815240	0.740818221	0.000002981
	0.35	0.704686311	0.704688090	0.000001779
2	0.40	0.670320046	0.670320046	0.000000000
	0.45	0.637626817	0.637628152	0.000001335
	0.50	0.606528138	0.606530660	0.000002522
	0.55	0.576948310	0.576949810	0.000001500
3	0.60	0.548811636	0.548811636	0.000000000
	0.65	0.522044705	0.522045777	0.000001072
	0.70	0.496583261	0.496585304	0.000002043
	0.75	0.472365336	0.472366553	0.000001216
4	0.80	0.449328964	0.449328964	0.000000000
	0.85	0.427414049	0.427414932	0.000000883
	0.90	0.406567981	0.406569660	0.000001679
	0.95	0.386740024	0.386741023	0.000000999
5	1.00	0.367879441	0.367879441	0.000000000
	1.05	0.349937028	0.349937749	0.000000722
	1.10	0.332869711	0.332871084	0.000001373
	1.15	0.316635952	0.316636769	0.000000817
6	1.20	0.301194212	0.301194212	0.000000000
	1.25	0.286504205	0.286504797	0.000000592
	1.30	0.272530668	0.272531793	0.000001125
	1.35	0.259239591	0.259240261	0.000000670
7	1.40	0.246596964	0.246596964	0.000000000
	1.45	0.234569806	0.234570288	0.000000482
	1.50	0.223129242	0.223130160	0.000000918
	1.55	0.212247429	0.212247974	0.000000545
8	1.60	0.201896518	0.201896518	0.000000000
	1.65	0.192049507	0.192049909	0.000000402
	1.70	0.182682760	0.182683524	0.000000764
	1.75	0.173773485	0.173773943	0.000000458
9	1.80	0.165298888	0.165298888	0.000000000
	1.85	0.157236862	0.157237166	0.000000304
	1.90	0.149568040	0.149568619	0.000000579
	1.95	0.142273741	0.142274072	0.000000331
10	2.00	0.135335283	0.135335283	0.000000000

Table 2: Complete Cubic B-spline Approximation for Ex 1b

Figure 4 shows the absolute error between the exact solution and the complete cubic B-spline solution.

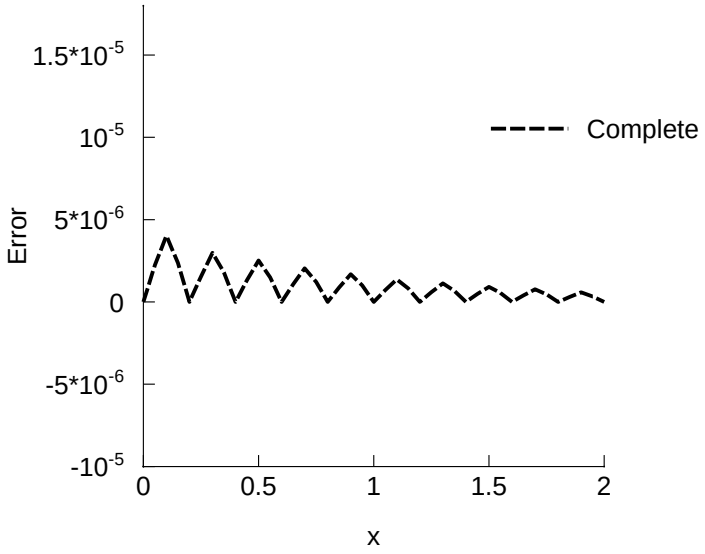


Figure 4: Complete Cubic B-spline Error for Ex 1b

it is observed that, the complete cubic B-spline interpolation gives a better approximation of the function than the natural cubic B-spline interpolation. We see this in the following graph.

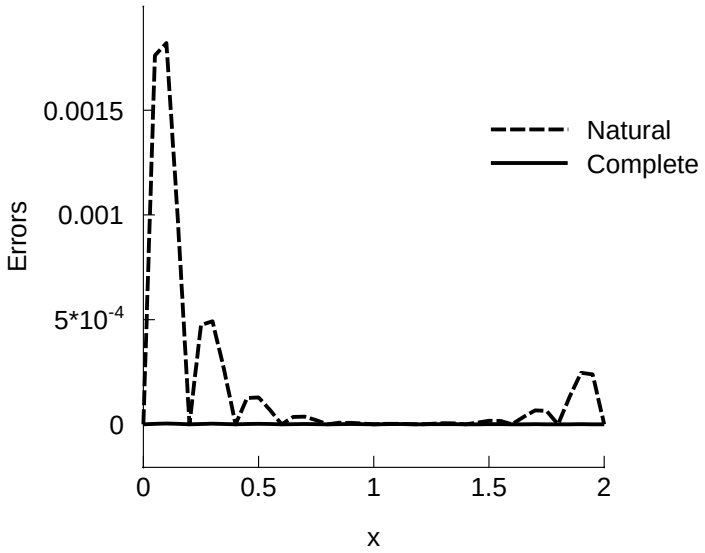


Figure 5: Comparison of the Natural and Complete Cubic B-spline Errors

Example 2: Consider the function

$$f(x) = \frac{sin(\frac{x}{2})}{4 + x^2}$$

on the interval $[-6.5, 6.5]$. We want to interpolate this function using the natural and complete cubic B-spline interpolations at 27 knots, and compare our results with the results from the Lagrange interpolation.

We have 26 subintervals with $h = 0.5$. Three extra points in between each subinterval are being interpolated. Table 3 compares the results of the three methods and the exact solution of $f(x)$. This shows how the Lagrange interpolation method oscillates at the end points while the cubic B-spline methods are more accurate to the exact solution. Table 4 shows their absolute error.

i	x_i	Natural Cubic B-Spline	Complete Cubic B-Spline	Lagrange Interpolation	Exact
	-6.375	0.000957632	0.001027989	1.737595016	0.001028015
	-6.250	-0.000458137	-0.000385342	1.100624386	-0.000385298
	-6.125	-0.001941999	-0.001903163	0.369407582	-0.001903144
1	-6.000	-0.003528000	-0.003528000	-0.003528000	-0.003528000
	-5.875	-0.005243454	-0.005262306	-0.102967438	-0.005262247
	-5.750	-0.007088740	-0.007108245	-0.082358197	-0.007108155
	-5.625	-0.009057502	-0.009067908	-0.039044335	-0.009067869
2	-5.500	-0.011143387	-0.011143387	-0.011143387	-0.011143387
	-5.375	-0.013341694	-0.013336642	-0.002935187	-0.013336539
	-5.250	-0.015654344	-0.015649117	-0.006632335	-0.015648960
	-5.125	-0.018084911	-0.018082123	-0.014078951	-0.018082057
3	-5.000	-0.020636970	-0.020636970	-0.020636970	-0.020636970
	-4.875	-0.023313377	-0.023314730	-0.025000595	-0.023314531
	-4.750	-0.026114099	-0.026115500	-0.027710884	-0.026115201
	-4.625	-0.029038388	-0.029039135	-0.029808644	-0.029039014
4	-4.500	-0.032085493	-0.032085493	-0.032085493	-0.032085493
	-4.375	-0.035254315	-0.035253953	-0.034875623	-0.035253556
	-4.250	-0.038542372	-0.038541997	-0.038157177	-0.038541407
	-4.125	-0.041946835	-0.041946634	-0.041747892	-0.041946399
5	-4.000	-0.045464871	-0.045464871	-0.045464871	-0.045464871
	-3.875	-0.049092666	-0.049092763	-0.049202973	-0.049091959
	-3.750	-0.052822451	-0.052822552	-0.052941416	-0.052821358
	-3.625	-0.056645474	-0.056645528	-0.056710909	-0.056645052
6	-3.500	-0.060552981	-0.060552981	-0.060552981	-0.060552981
	-3.375	-0.064534306	-0.064534280	-0.064491339	-0.064532652
	-3.250	-0.068571134	-0.068571107	-0.068521459	-0.068568676
	-3.125	-0.072643236	-0.072643221	-0.072614887	-0.072642222
7	-3.000	-0.076730384	-0.076730384	-0.076730384	-0.076730384
	-2.875	-0.080808584	-0.080808591	-0.080824467	-0.080805437
	-2.750	-0.084838780	-0.084838787	-0.084856866	-0.084833994
	-2.625	-0.088778148	-0.088778151	-0.088789950	-0.088776040
8	-2.500	-0.092583865	-0.092583865	-0.092583865	-0.092583865
	-2.375	-0.096206393	-0.096206391	-0.096190259	-0.096200912
	-2.250	-0.099569320	-0.099569318	-0.099547154	-0.099560562
	-2.125	-0.102589523	-0.102589522	-0.102576417	-0.102584947
9	-2.000	-0.105183873	-0.105183873	-0.105183873	-0.105183873
	-1.875	-0.107260214	-0.107260214	-0.107261156	-0.107254035
	-1.750	-0.108690260	-0.108690261	-0.108688066	-0.108678726
	-1.625	-0.109336695	-0.109336695	-0.109334544	-0.109328362
10	-1.500	-0.109062202	-0.109062202	-0.109062202	-0.109062202
	-1.375	-0.107727534	-0.107727534	-0.107726146	-0.107731706
	-1.250	-0.105185726	-0.105185725	-0.105178480	-0.105186027
	-1.125	-0.101287882	-0.101287882	-0.101274875	-0.101280033
11	-1.000	-0.095885108	-0.095885108	-0.095885108	-0.095885108
	-0.875	-0.088859225	-0.088859225	-0.088907468	-0.088902559
	-0.750	-0.080214923	-0.080214923	-0.080285736	-0.080278910
	-0.625	-0.069987611	-0.069987611	-0.070026349	-0.070021583
12	-0.500	-0.058212696	-0.058212696	-0.058212696	-0.058212696
	-0.375	-0.044982693	-0.044982693	-0.045013462	-0.045018155
	-0.250	-0.030618543	-0.030618543	-0.030682557	-0.030689165
	-0.125	-0.015498296	-0.015498296	-0.015549409	-0.015554071

13	0.000	0.000000000	0.000000000	0.000000000	0.000000000
	0.125	0.015498296	0.015498296	0.015549409	0.015554071
	0.250	0.030618543	0.030618543	0.030682557	0.030689165
	0.375	0.044982693	0.044982693	0.045013462	0.045018155
14	0.500	0.058212696	0.058212696	0.058212696	0.058212696
	0.625	0.069987611	0.069987611	0.070026349	0.070021583
	0.750	0.080214923	0.080214923	0.080285736	0.080278910
	0.875	0.088859225	0.088859225	0.088907468	0.088902559
15	1.000	0.095885108	0.095885108	0.095885108	0.095885108
	1.125	0.101287882	0.101287882	0.101274875	0.101280033
	1.250	0.105185726	0.105185725	0.105178480	0.105186027
	1.375	0.107727534	0.107727534	0.107726146	0.107731706
16	1.500	0.109062202	0.109062202	0.109062202	0.109062202
	1.625	0.109336695	0.109336695	0.109334544	0.109328362
	1.750	0.108690260	0.108690261	0.108688066	0.108678726
	1.875	0.107260214	0.107260214	0.107261156	0.107254035
17	2.000	0.105183873	0.105183873	0.105183873	0.105183873
	2.125	0.102589523	0.102589522	0.102576417	0.102584947
	2.250	0.099569320	0.099569318	0.099547154	0.099560562
	2.375	0.096206393	0.096206391	0.096190259	0.096200912
18	2.500	0.092583865	0.092583865	0.092583865	0.092583865
	2.625	0.088778148	0.088778151	0.088789950	0.088776040
	2.750	0.084838780	0.084838787	0.084856866	0.084833994
	2.875	0.080808584	0.080808591	0.080824467	0.080805437
19	3.000	0.076730384	0.076730384	0.076730384	0.076730384
	3.125	0.072643236	0.072643221	0.072614887	0.072642222
	3.250	0.068571134	0.068571107	0.068521459	0.068568676
	3.375	0.064534306	0.064534280	0.064491339	0.064532652
20	3.500	0.060552981	0.060552981	0.060552981	0.060552981
	3.625	0.056645474	0.056645528	0.056710909	0.056645052
	3.750	0.052822451	0.052822552	0.052941416	0.052821358
	3.875	0.049092666	0.049092763	0.049202973	0.049091959
21	4.000	0.045464871	0.045464871	0.045464871	0.045464871
	4.125	0.041946835	0.041946634	0.041747892	0.041946399
	4.250	0.038542372	0.038541997	0.038157177	0.038541407
	4.375	0.035254315	0.035253953	0.034875623	0.035253556
22	4.500	0.032085493	0.032085493	0.032085493	0.032085493
	4.625	0.029038388	0.029039135	0.029808644	0.029039014
	4.750	0.026114099	0.026115500	0.027710884	0.026115201
	4.875	0.023313377	0.023314730	0.025000595	0.023314531
23	5.000	0.020636970	0.020636970	0.020636970	0.020636970
	5.125	0.018084911	0.018082123	0.014078951	0.018082057
	5.250	0.015654344	0.015649117	0.006632335	0.015648960
	5.375	0.013341694	0.013336642	0.002935187	0.013336539
24	5.500	0.011143387	0.011143387	0.011143387	0.011143387
	5.625	0.009057502	0.009067908	0.039044335	0.009067869
	5.750	0.007088740	0.007108245	0.082358197	0.007108155
	5.875	0.005243454	0.005262306	0.102967438	0.005262247
25	6.000	0.003528000	0.003528000	0.003528000	0.003528000
	6.125	0.001941999	0.001903163	−0.369407582	0.001903144
	6.250	0.000458137	0.000385342	−1.100624386	0.000385298
	6.375	−0.000957632	−0.001027989	−1.737595016	−0.001028015

Table 3: Approximations Compared to the Exact Solution for Ex 2

i	x_i	Natural Cubic B-Spline Error	Complete Cubic B-Spline Error	Lagrange Interpolation Error
	-6.375	0.000070383	0.000000026	1.736567001
	-6.250	0.000072839	0.000000044	1.101009684
	-6.125	0.000038855	0.000000019	0.371310726
1	-6.000	0.000000000	0.000000000	0.000000000
	-5.875	0.000018793	0.000000059	0.097705192
	-5.750	0.000019415	0.000000090	0.075250042
	-5.625	0.000010367	0.000000039	0.029976466
2	-5.500	0.000000000	0.000000000	0.000000000
	-5.375	0.000005155	0.000000104	0.010401352
	-5.250	0.000005384	0.000000157	0.009016625
	-5.125	0.000002854	0.000000066	0.004003106
3	-5.000	0.000000000	0.000000000	0.000000000
	-4.875	0.000001154	0.000000199	0.001686065
	-4.750	0.000001102	0.000000299	0.001595683
	-4.625	0.000000626	0.000000121	0.000769629
4	-4.500	0.000000000	0.000000000	0.000000000
	-4.375	0.000000759	0.000000397	0.000377932
	-4.250	0.000000966	0.000000590	0.000384230
	-4.125	0.000000436	0.000000236	0.000198507
5	-4.000	0.000000000	0.000000000	0.000000000
	-3.875	0.000000707	0.000000804	0.000111014
	-3.750	0.000001093	0.000001194	0.000120058
	-3.625	0.000000422	0.000000476	0.000065857
6	-3.500	0.000000000	0.000000000	0.000000000
	-3.375	0.000001654	0.000001628	0.000041314
	-3.250	0.000002458	0.000002431	0.000047216
	-3.125	0.000001014	0.000000999	0.000027334
7	-3.000	0.000000000	0.000000000	0.000000000
	-2.875	0.000003147	0.000003154	0.000019030
	-2.750	0.000004785	0.000004793	0.000022872
	-2.625	0.000002108	0.000002112	0.000013910
8	-2.500	0.000000000	0.000000000	0.000000000
	-2.375	0.000005481	0.000005479	0.000010653
	-2.250	0.000008758	0.000008756	0.000013408
	-2.125	0.000004576	0.000004575	0.000008529
9	-2.000	0.000000000	0.000000000	0.000000000
	-1.875	0.000006179	0.000006179	0.000007121
	-1.750	0.000011534	0.000011535	0.000009340
	-1.625	0.000008333	0.000008333	0.000006182
10	-1.500	0.000000000	0.000000000	0.000000000
	-1.375	0.000004172	0.000004172	0.000005560
	-1.250	0.000000301	0.000000301	0.000007546
	-1.125	0.000007849	0.000007849	0.000005158
11	-1.000	0.000000000	0.000000000	0.000000000
	-0.875	0.000043334	0.000043334	0.000004909
	-0.750	0.000063987	0.000063987	0.000006826
	-0.625	0.000033972	0.000033972	0.000004766
12	-0.500	0.000000000	0.000000000	0.000000000
	-0.375	0.000035462	0.000035462	0.000004692
	-0.250	0.000070622	0.000070622	0.000006609
	-0.125	0.000055775	0.000055775	0.000004662

13	0.000	0.000000000	0.000000000	0.000000000
	0.125	0.000055775	0.000055775	0.000004662
	0.250	0.000070622	0.000070622	0.000006609
	0.375	0.000035462	0.000035462	0.000004692
14	0.500	0.000000000	0.000000000	0.000000000
	0.625	0.000033972	0.000033972	0.000004766
	0.750	0.000063987	0.000063987	0.000006826
	0.875	0.000043334	0.000043334	0.000004909
15	1.000	0.000000000	0.000000000	0.000000000
	1.125	0.000007849	0.000007849	0.000005158
	1.250	0.000000301	0.000000301	0.000007546
	1.375	0.000004172	0.000004172	0.000005560
16	1.500	0.000000000	0.000000000	0.000000000
	1.625	0.000008333	0.000008333	0.000006182
	1.750	0.000011534	0.000011535	0.000009340
	1.875	0.000006179	0.000006179	0.000007121
17	2.000	0.000000000	0.000000000	0.000000000
	2.125	0.000004576	0.000004575	0.000008529
	2.250	0.000008758	0.000008756	0.000013408
	2.375	0.000005481	0.000005479	0.000010653
18	2.500	0.000000000	0.000000000	0.000000000
	2.625	0.000002108	0.000002112	0.000013910
	2.750	0.000002108	0.000004793	0.000022872
	2.875	0.000003147	0.000003154	0.000019030
19	3.000	0.000000000	0.000000000	0.000000000
	3.125	0.000001014	0.000000999	0.000027334
	3.250	0.000002458	0.000002431	0.000047216
	3.375	0.000001654	0.000001628	0.000041314
20	3.500	0.000000000	0.000000000	0.000000000
	3.625	0.000000422	0.000000476	0.000065857
	3.750	0.000001093	0.000001194	0.000120058
	3.875	0.000000707	0.000000804	0.000111014
21	4.000	0.000000000	0.000000000	0.000000000
	4.125	0.000000436	0.000000236	0.000198507
	4.250	0.000000966	0.000000590	0.000384230
	4.375	0.000000759	0.000000397	0.000377932
22	4.500	0.000000000	0.000000000	0.000000000
	4.625	0.000000626	0.000000121	0.000769629
	4.750	0.000001102	0.000000299	0.001595683
	4.875	0.000001154	0.000000199	0.001686065
23	5.000	0.000000000	0.000000000	0.000000000
	5.125	0.000002854	0.000000066	0.004003106
	5.250	0.000005384	0.000000157	0.009016625
	5.375	0.000005155	0.000000104	0.010401352
24	5.500	0.000000000	0.000000000	0.000000000
	5.625	0.000010367	0.000000039	0.029976466
	5.750	0.000019415	0.000000090	0.075250042
	5.875	0.000018793	0.000000059	0.097705192
25	6.000	0.000000000	0.000000000	0.000000000
	6.125	0.000038855	0.000000019	0.371310726
	6.250	0.000072839	0.000000044	1.101009684
	6.375	0.000070383	0.000000026	1.736567001

Table 4: Natural, Complete and Lagrange Interpolation Errors for Ex 2

We see in Table 4 that the complete cubic B-spline error is smaller than the natural cubic B-spline error near both end points. The error graph in Figure 6 compares the natural cubic B-spline error and the Lagrange interpolation error.

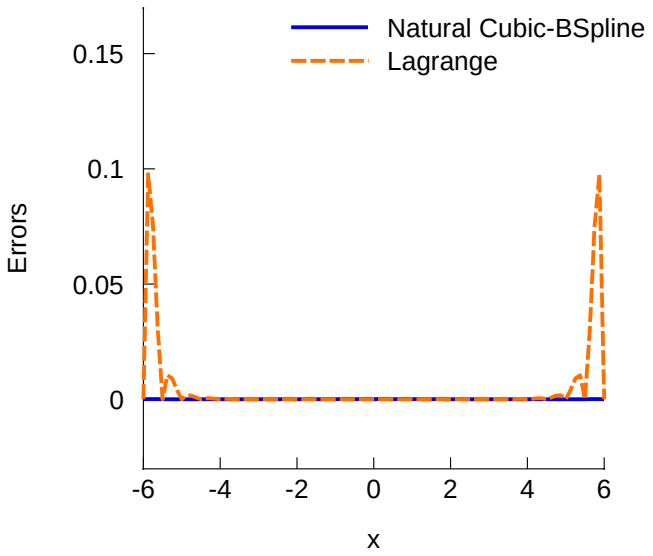


Figure 6: Natural Cubic B-spline and Lagrange Interpolation Errors for Ex 2

Example 3a: Consider the picture below. Suppose we want to interpolate the upper portion of this camel on the interval $[0, 10]$.

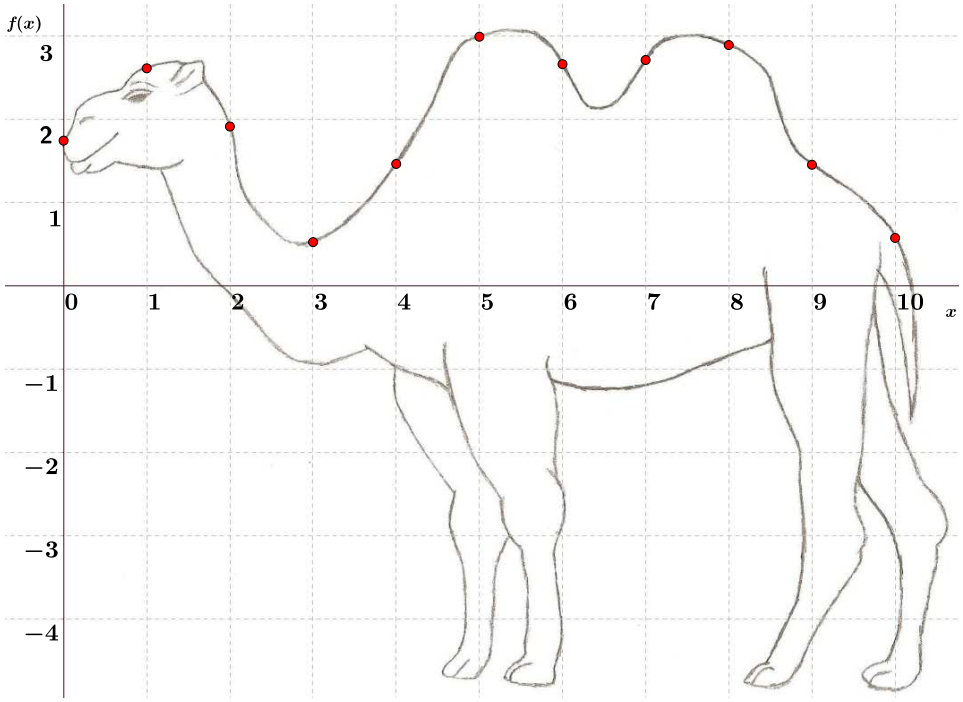


Figure 7: Original Data for Ex 3a

We assume this upper portion is an unknown function $f(x)$ on the interval $[0, 10]$. We will interpolate $f(x)$ using $N = 10$ and then $N = 20$ to find the best approximation. Notice that we have no information about the derivatives of the function at the end points. Therefore it is best to interpolate $f(x)$ using the natural cubic B-spline interpolation. As in the previous example, we will compare our method with the Lagrange interpolation.

Let us first interpolate $f(x)$ on $[0, 10]$ using $N = 10$ which gives $h = 1$. There are 11 points $(x_i, f(x_i))$ for $i = 0, 1, \ldots, 10$ given in Table 5.

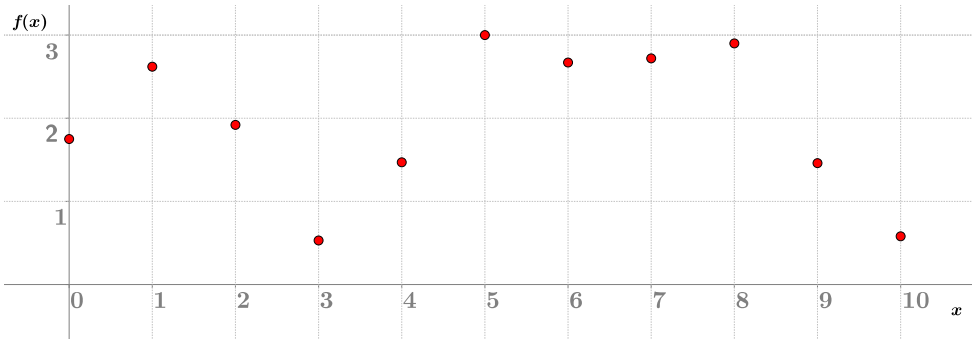


Figure 8: Input Data for Ex 3a

x	0	1	2	3	4	5	6	7	8	9	10
$f(x)$	1.75	2.62	1.92	0.53	1.47	3.00	2.67	2.72	2.90	1.46	0.58

Table 5: Data for Ex 3a

We want to estimate three values in between each subinterval. This gives us a total of 30 estimated new points. We first interpolate $f(x)$ using the Lagrange interpolation. Secondly, we interpolate the function using the natural cubic B-spline method. Finally we give a comparison of the results from both methods.

Using Lagrange interpolation, we fit a polynomial of degree at most 10 to the 11 data points given in Table 5, and the result is given below

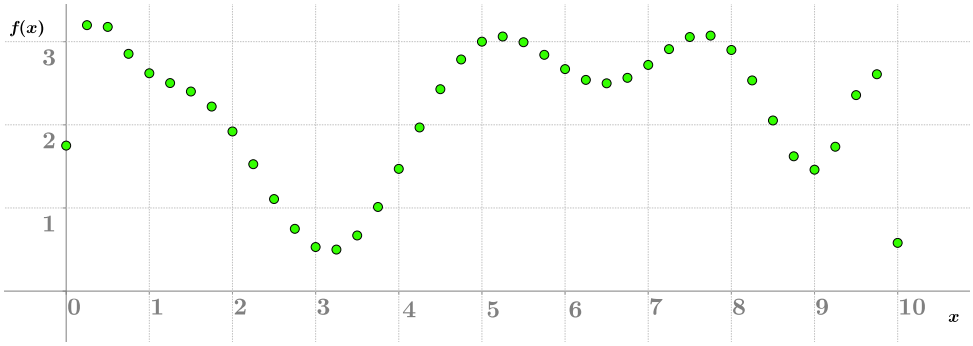


Figure 9: Lagrange Interpolation Results for Ex 3a

We know that the more data points used to interpolate a function with the Lagrange interpolation, the higher the degree of the interpolatory polynomial. This gives us a problem since the interpolation faces greater oscillation in between each interval. In this case, the degree of the interpolant is at most 10. We observe from Figure 9 that the interpolation faces greater oscillation on the intervals near the end points of the data range.

Using the natural cubic B-spline interpolant, each subinterval will be represented with a unique polynomial of degree 3. The result is given below,

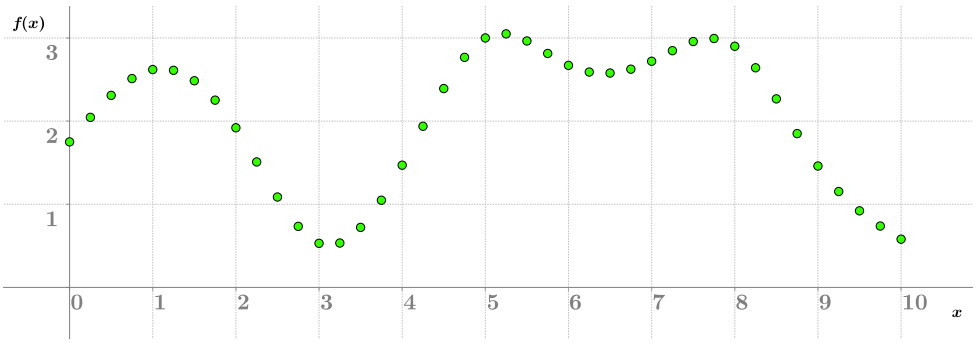


Figure 10: Cubic B-spline Interpolation Results for Ex 3a

and Figure 11 combines Figures 7 and 10 to demonstrate the accuracy of the natural cubic B-spline for $h = 1$.

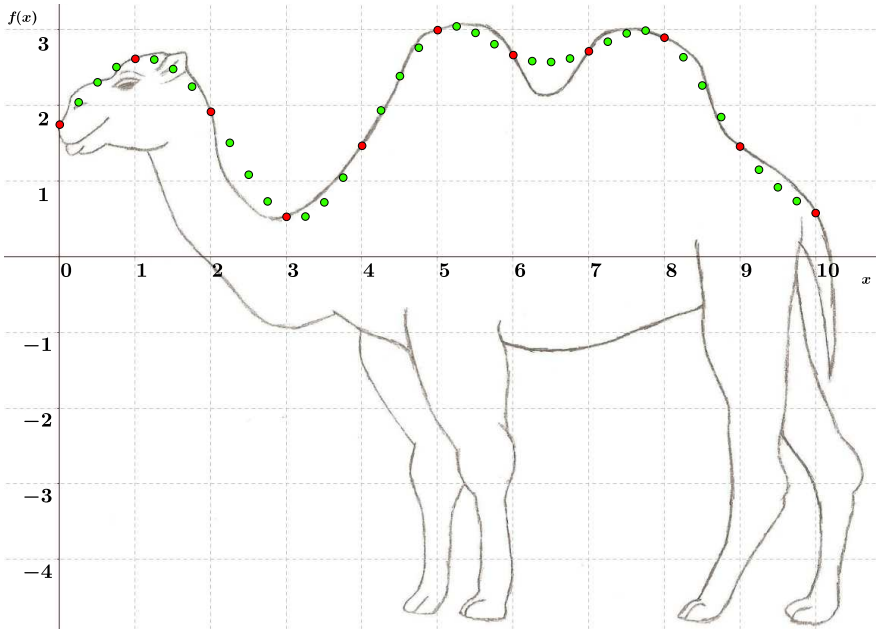
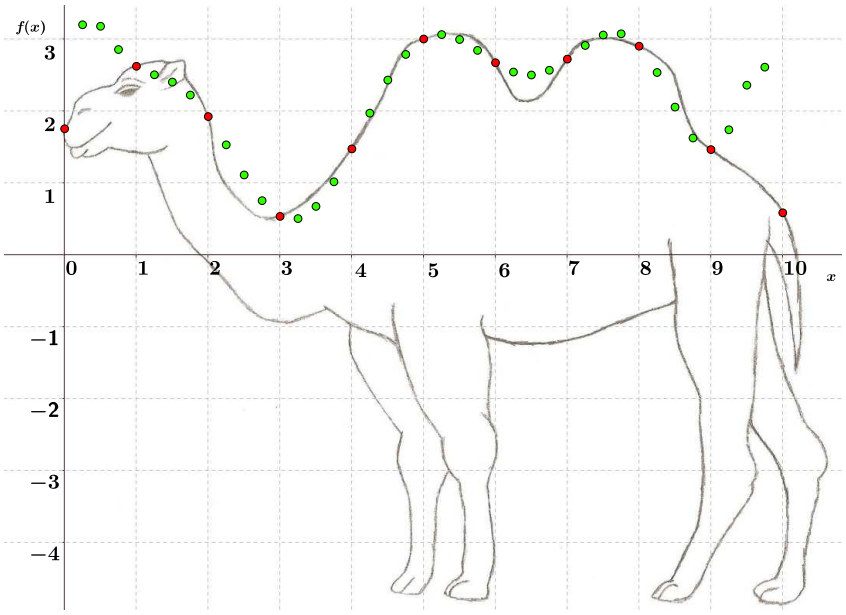
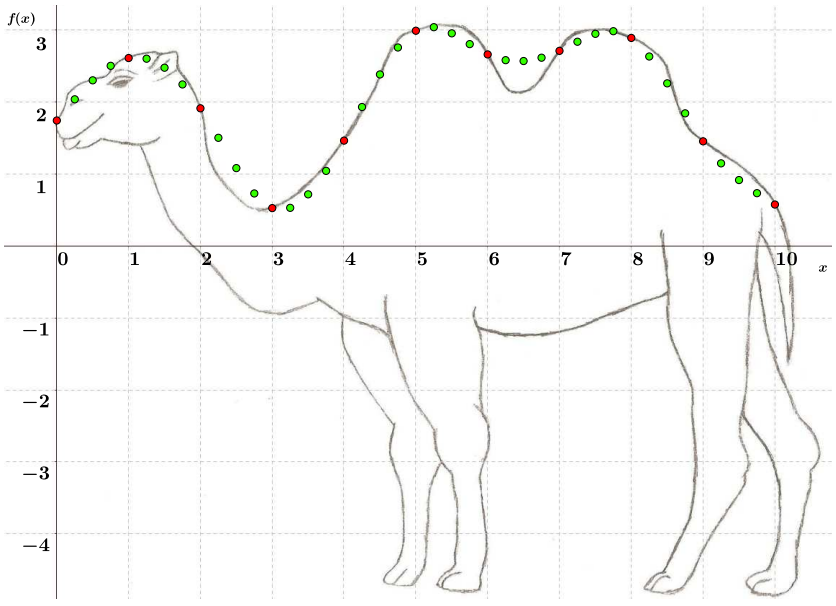


Figure 11: Original Data with the Cubic B-spline Interpolation Results for Ex 3a

The results for the Lagrange interpolation and the natural cubic B-spline interpolation using $N = 10$ are combined in Figure 12. We can see that $N = 10$ is sufficiently large for the Lagrange interpolation to oscillate at the end points, while the natural cubic B-spline is smoothly stable.



(a) Lagrange Interpolation Results for Ex 3a



(b) Cubic B-spline Interpolation Results for Ex 3a

Figure 12: Lagrange and Cubic B-spline Interpolation Results for Ex 3a with $h = 1$

Example 3b: Now, suppose that we want to interpolate the upper portion of the camel on the same interval $[0, 10]$ using $N = 20$. This means that there are 21 knots and $h = 0.5$. The set of points is given in Table 6.

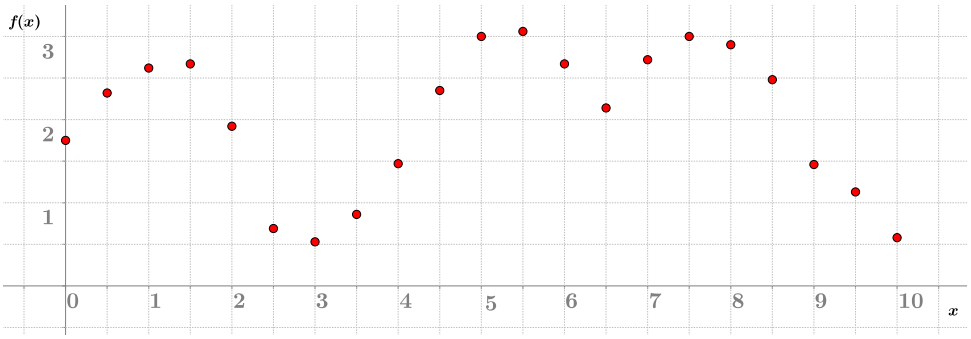


Figure 13: Input Data for Ex 3b

x	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	8.5	9.0	9.5	10.0
$f(x)$	1.75	2.32	2.62	2.67	1.92	0.69	0.53	0.86	1.47	2.35	3.00	3.06	2.67	2.14	2.72	3.00	2.90	2.48	1.46	1.13	0.58

Table 6: Data for Ex 3b

Again we want to interpolate three values in between each subinterval. In this case, the total of new estimated points is 60. First, we present the Lagrange interpolation results then the natural cubic B-spline results.

In this case, the Lagrange interpolation fit a polynomial of degree 20 or less to the 21 data points

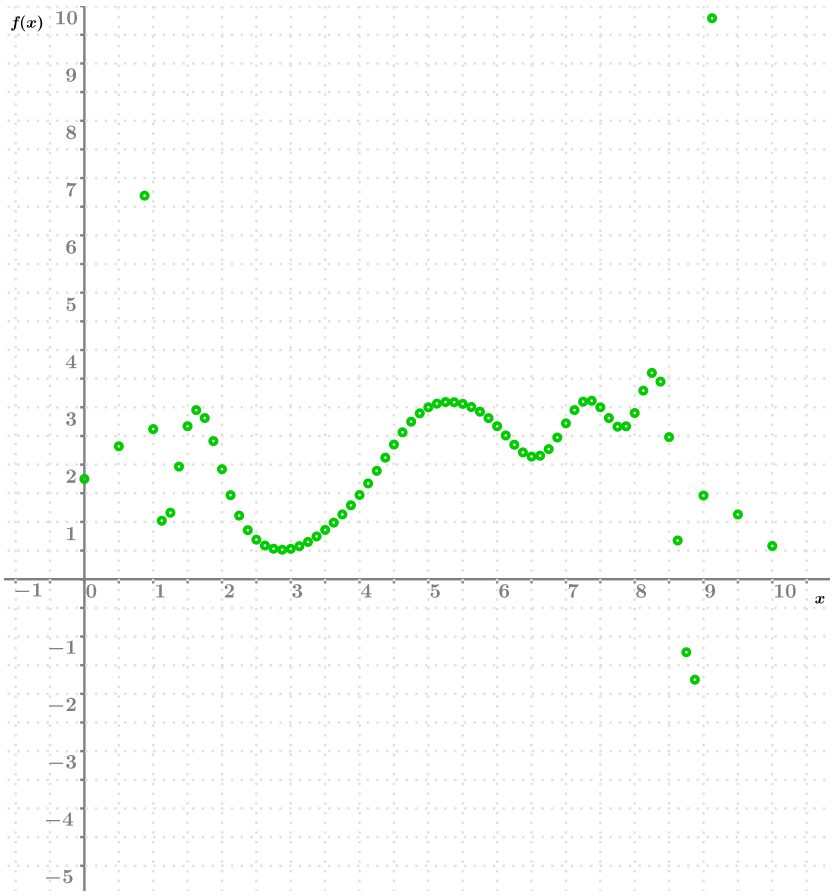


Figure 14: Lagrange Interpolation Results for Ex 3b

The interpolating polynomial of degree 10 experienced some oscillation near the end points. In this case, the interpolating polynomial is of degree 20, and as we can see, it oscillates wildly on the intervals near the end points. Therefore, the Lagrange interpolation does not produce better approximation.

Using the natural cubic B-spline interpolant, each of the 20 subinterval will be represented with a unique polynomial of degree 3. The result is shown in Figure 15,

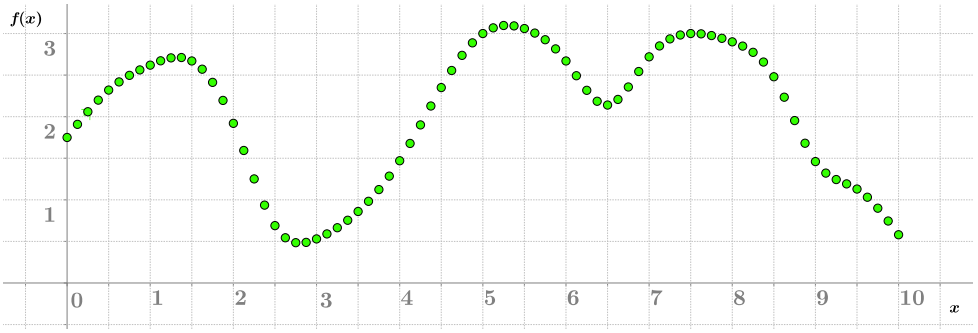


Figure 15: Cubic B-spline Interpolation Results for Ex 3b

and Figure 16 combines Figure 15 and the camel profile to demonstrate the accuracy and smoothness of the natural cubic B-spline for $h = 0.5$.

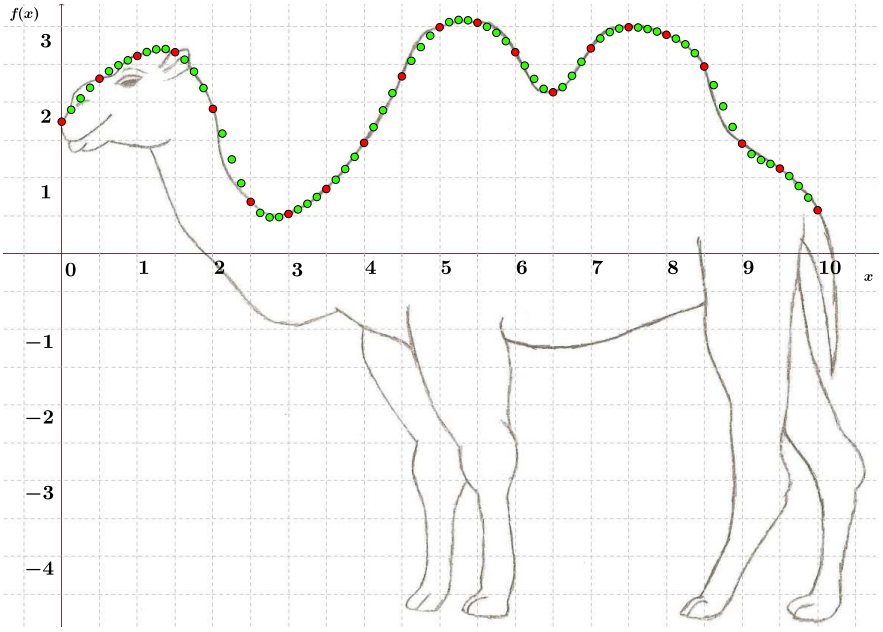
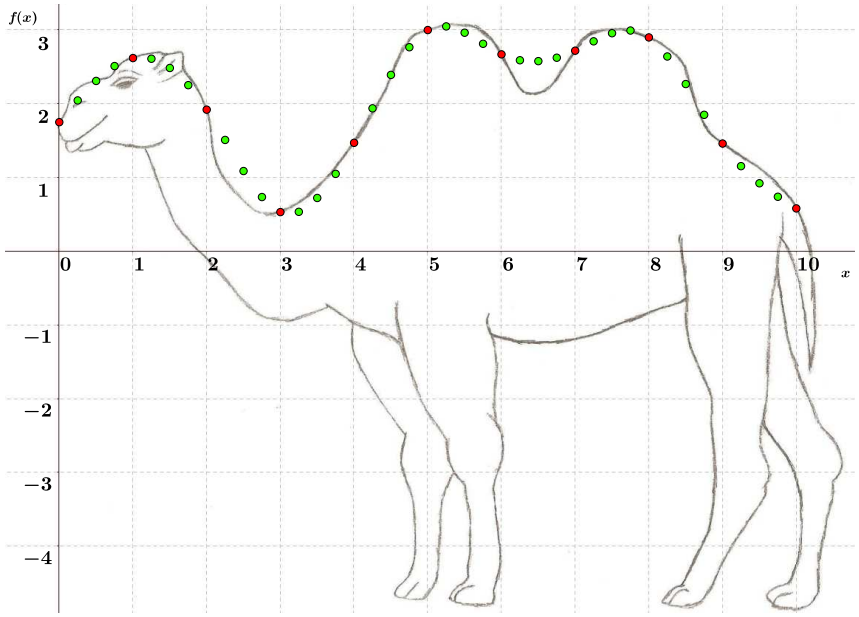
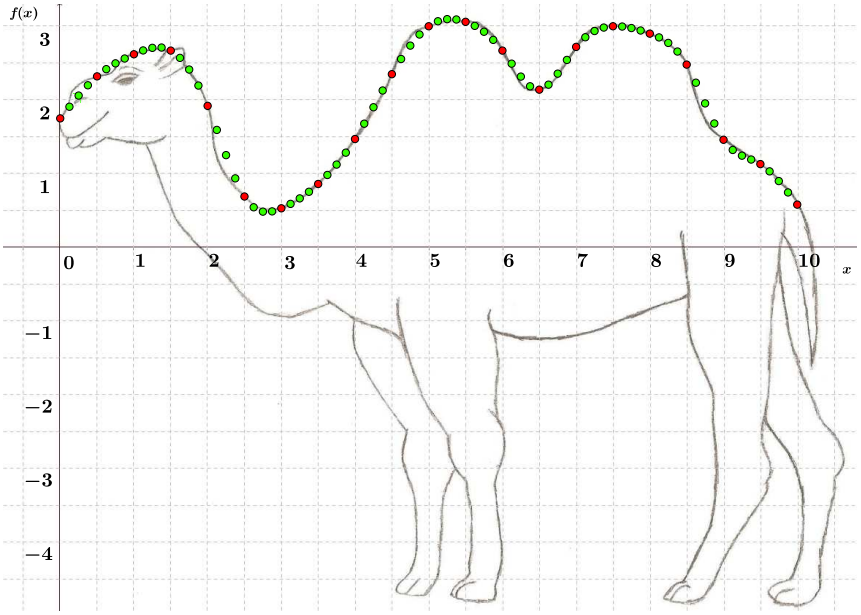


Figure 16: Original Data with the Cubic B-spline Interpolation Results for Ex 3b

The results for the natural cubic B-spline interpolation using $N = 10$ and $N = 20$ are combined in Figure 17 respectively. When $N = 20$, the natural cubic B-spline yields a satisfactory approximation to the function $f(x)$ on the interval $[0, 10]$.



(a) Cubic B-spline Interpolation Results for $h = 1$



(b) Cubic B-spline Interpolation Results for $h = 0.5$

Figure 17: Cubic B-spline Interpolation Results for Ex 3 with $h = 1$ and $h = 0.5$

In all cases, the cubic B-spline interpolation has demonstrated that it is superior to the Lagrange interpolation. The cubic B-spline interpolation creates unique cubic polynomials at each subinterval, regardless of the value of N. This leads to smooth results. In addition, we have shown that the complete cubic B-spline interpolation gives better results than the natural Cubic B-spline interpolation near the end points. In contrast, the Lagrange interpolation tends to oscillate near the end points when using large values for N.

CHAPTER V

CUBIC B-SPLINES TO SOLVE BVP

In this section, we study the use of cubic B-splines to solve second-order linear boundary value problems. We are going to apply the same idea we learned previously to approximate solutions to boundary value problems (BVP) of the form

$$a_1(x)y'' + a_2(x)y' + a_3(x)y = f(x) \tag{14}$$

with boundary conditions

$$\begin{aligned} y(a) &= \alpha \\ y(b) &= \beta \end{aligned} \tag{15}$$

where $a_1(x) \neq 0$, $a_2(x)$, $a_3(x)$ and $f(x)$ are continuous real-valued functions on the interval $[a, b]$.

We are going to use cubic B-splines to approximate the solution of the BVP. As before, we need to find a matrix representation to solve for the coefficients c_i in (16). After obtaining our numerical results, we will compare these with the results obtained from the central finite difference method.

We assume equally spaced knots, i.e., $h = x_{i+1} - x_i$. Let

$$Y(x) = \sum_{i=-1}^{N+1} c_i B_i(x) \tag{16}$$

be the approximating function where the constants c_i are to be determined and the $B_i(x)$ are cubic B-spline functions defined in (3). It is required that (16) satisfies our BVP (14-15) at $x = x_i$ where x_i is an interior point. That is

$$a_1(x_i)Y''(x_i) + a_2(x_i)Y'(x_i) + a_3(x_i)Y(x_i) = f(x_i)$$

and the boundary conditions

$$Y(x_0) = \alpha \text{ for } x_0 = a$$

$$Y(x_N) = \beta \text{ for } x_N = b$$

The approximating function (16) gives

$$\begin{aligned}
Y(x_i) &= c_{i-1}B_{i-1}(x_i) + c_iB_i(x_i) + c_{i+1}B_{i+1}(x_i) + c_{i+2}B_{i+2}(x_i) \\
Y'(x_i) &= c_{i-1}B'_{i-1}(x_i) + c_iB'_i(x_i) + c_{i+1}B'_{i+1}(x_i) + c_{i+2}B'_{i+2}(x_i) \\
Y''(x_i) &= c_{i-1}B''_{i-1}(x_i) + c_iB''_i(x_i) + c_{i+1}B''_{i+1}(x_i) + c_{i+2}B''_{i+2}(x_i)
\end{aligned}$$

and these yield

$$\begin{aligned}
&c_{i-1}[a_1(x_i)B''_{i-1}(x_i) + a_2(x_i)B'_{i-1}(x_i) + a_3(x_i)B_{i-1}(x_i)] \\
&+ c_i[a_1(x_i)B''_i(x_i) + a_2(x_i)B'_i(x_i) + a_3(x_i)B_i(x_i)] \\
&+ c_{i+1}[a_1(x_i)B''_{i+1}(x_i) + a_2(x_i)B'_{i+1}(x_i) + a_3(x_i)B_{i+1}(x_i)] \\
&+ c_{i+2}[a_1(x_i)B''_{i+2}(x_i) + a_2(x_i)B'_{i+2}(x_i) + a_3(x_i)B_{i+2}(x_i)] = f(x_i)
\end{aligned} \tag{17}$$

also by the properties of cubic B-spline functions, we obtain the following

$$\begin{array}{lll}
B''_{i-1}(x_i) = \dfrac{1}{h^2} & B'_{i-1}(x_i) = -\dfrac{1}{2h} & B_{i-1}(x_i) = \dfrac{1}{6} \\
B''_i(x_i) = -\dfrac{2}{h^2} & B'_i(x_i) = 0 & B_i(x_i) = \dfrac{2}{3} \\
B''_{i+1}(x_i) = \dfrac{1}{h^2} & B'_{i+1}(x_i) = \dfrac{1}{2h} & B_{i+1}(x_i) = \dfrac{1}{6} \\
B''_{i+2}(x_i) = 0 & B'_{i+2}(x_i) = 0 & B_{i+2}(x_i) = 0
\end{array} \tag{18}$$

If we combine (17) and (18), we obtain

$$c_{i-1}[6a_1(x_i) - 3a_2(x_i)h + a_3(x_i)h^2] + c_i[-12a_1(x_i) + 4a_3(x_i)h^2] \tag{19}$$
$$+ c_{i+1}[6a_1(x_i) + 3a_2(x_i)h + a_3(x_i)h^2] = 6h^2 f(x_i)$$

Now we apply the boundary conditions:

$$Y(x_0) = c_{-1}B_{-1}(x_0) + c_0B_0(x_0) + c_1B_1(x_0) + c_2B_2(x_0) = \alpha$$
$$Y(x_N) = c_{N-1}B_{N-1}(x_N) + c_NB_N(x_N) + c_{N+1}B_{N+1}(x_N) + c_{N+2}B_{N+2}(x_N) = \beta$$

where the value of $B_i(x)$ at $x = x_0$ and $x = x_N$ are given below

$$B_{-1}(x_0) = \frac{1}{6} = B_{N-1}(x_N)$$
$$B_0(x_0) = \frac{4}{6} = B_N(x_N)$$
$$B_1(x_0) = \frac{1}{6} = B_{N+1}(x_N)$$
$$B_2(x_0) = 0 = B_{N+2}(x_N).$$

Therefore,

$$c_{-1} + 4c_0 + c_1 = 6\alpha \tag{20}$$

$$c_{N-1} + 4c_N + c_{N+1} = 6\beta. \tag{21}$$

Now that we have found all the constant coefficients (19), (20) and (21), we can write a system of $N + 3$ linear equations in $N + 3$ unknowns. This system is represented in (22) where M is an $(N + 3) \times (N + 3)$ matrix.

The cubic B-spline approximation for the boundary value problem (14-15) is obtained using (16), where the constant coefficients c_i satisfy the tridiagonal system defined in (22)

$$\begin{pmatrix} 1 & 4 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ p_0 & q_0 & r_0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & p_1 & q_1 & r_1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_2 & q_2 & r_2 & \cdots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & p_{N-2} & q_{N-2} & r_{N-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & p_{N-1} & q_{N-1} & r_{N-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & p_N & q_N & r_N \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 4 & 1 \end{pmatrix} \begin{pmatrix} c_{-1} \\ c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-2} \\ c_{N-1} \\ c_N \\ c_{N+1} \end{pmatrix} = 6 \begin{pmatrix} \alpha \\ h^2 f(x_0) \\ h^2 f(x_1) \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{N-2}) \\ h^2 f(x_{N-1}) \\ h^2 f(x_N) \\ \beta \end{pmatrix}, \quad (22)$$

or $M\bar{c} = 6 \cdot \bar{b}$.

This system has a unique solution and p_i, q_i, and r_i are defined below

$$p_i = 6a_1(x_i) - 3a_2(x_i)h + a_3(x_i)h^2$$

$$q_i = -12a_1(x_i) + 4a_3(x_i)h^2$$

$$r_i = 6a_1(x_i) + 3a_2(x_i)h + a_3(x_i)h^2$$

Central Finite Difference Method

Since we will compare the cubic B-spline method with the central finite difference method (FDM), it is convenient to mention the finite difference method. We use the following definition

$$y' = \frac{y_{i+1} - y_{i-1}}{2h} \text{ and } y'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

The system (23) is obtained by approximating the boundary value problem (14) with boundary conditions (15) using the above equations. The diagonal matrix is an $(N \times N)$ matrix. This system of equations has a unique solution, and the method has error of order $O(h^2)$.

$$\begin{pmatrix} Q_1 & R_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ P_2 & Q_2 & R_2 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & P_3 & Q_3 & R_3 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & P_4 & Q_4 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & Q_{N-3} & R_{N-3} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & P_{N-2} & Q_{N-2} & R_{N-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & P_{N-1} & Q_{N-1} & R_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & P_N & Q_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{N-3} \\ y_{N-2} \\ y_{N-1} \\ y_N \end{pmatrix} = 2 \begin{pmatrix} h^2 f(x_1) - \frac{P_1}{2}\alpha \\ h^2 f(x_2) \\ h^2 f(x_3) \\ h^2 f(x_4) \\ \vdots \\ h^2 f(x_{N-3}) \\ h^2 f(x_{N-2}) \\ h^2 f(x_{N-1}) \\ h^2 f(x_N) - \frac{R_N}{2}\beta \end{pmatrix}, \tag{23}$$

where P_i, Q_i, and R_i are defined below

$$\begin{aligned} P_i &= 2a_1(x_i) - a_2(x_i)h \\ Q_i &= 2a_3(x_i)h^2 - 4a_1(x_i) \\ R_i &= 2a_1(x_i) + a_2(x_i)h. \end{aligned}$$

Now consider the following boundary value problem with variable coefficients

$$y'' + (1 - x)y' + xy = x \text{ for } 0 < x < 1$$

with boundary conditions

$$y(0) = 0$$

$$y(1) = 2.$$

We obtained this example from the book Differential Equations with Boundary Value Problems by Zill and Wright. The solution for this BVP was obtained using the central finite difference method with $N = 10$. Results are shown below.

	Book Result	Our Result
y_1	0.2660	0.2659676365
y_2	0.5097	0.5097313927
y_3	0.7357	0.7356869150
y_4	0.9471	0.9471265461
y_5	1.1465	1.1464546393
y_6	1.3353	1.3353449981
y_7	1.5149	1.5148552741
y_8	1.6855	1.6855091017
y_9	1.8474	1.8473538705

We compared our results with their results to check the effectiveness of our linear system of equations (23) (Zill & Wright, 2012).

CHAPTER VI

RESULTS AND DISCUSSIONS FOR SOLVING BVP

This chapter demonstrates some numerical results of the solution to the boundary value problem presented in chapter V. The program used to implement these results was coded in Java, the cubic polynomials were created in Mathematica and the figures were created using CoPlot.

Example 4: We consider a linear boundary value problem with constant coefficients

$$y'' + 2y' + 5y = 6\cos(2x) - 7\sin(2x) \text{ for } 0 < x < \frac{\pi}{4} \tag{24}$$

with boundary conditions

$$\begin{aligned} y(0) &= 4 \\ y(\frac{\pi}{4}) &= 1. \end{aligned} \tag{25}$$

The exact solution to boundary value problem is

$$y(x) = 2(1 + e^{-x})\cos(2x) + \sin(2x). \tag{26}$$

We approximate the solution of (24) with the boundary conditions (25) using the cubic B-spline method with $N = 15$. Then we compare our results with the central finite difference method.

In order to use (16), we first need to find the constant coefficients c_i for $i = -1, 0, 1, \ldots, 16$ using the system of linear equations (22) where M is an 18×18 matrix. These coefficients are given below

$$c_{-1} = 3.987262950 \quad c_0 = 4.006396007 \quad c_1 = 3.987153023 \quad c_2 = 3.931607119 \quad c_3 = 3.841948301$$
$$c_4 = 3.720466810 \quad c_5 = 3.569538179 \quad c_6 = 3.391609786 \quad c_7 = 3.189188622 \quad c_8 = 2.964830030$$
$$c_9 = 2.721127165 \quad c_{10} = 2.460700940 \quad c_{11} = 2.186190232 \quad c_{12} = 1.900242176 \quad c_{13} = 1.605502342$$
$$c_{14} = 1.304604683 \quad c_{15} = 1.000161134 \quad c_{16} = 0.694750780$$

and therefore the cubic B-spline polynomials are as follow

$$Y(x) = \begin{cases} 2.40700x^3 - 6.99895x^2 - 0.00105x + 4 & \text{for } x \in [0, \frac{\pi}{60}) \\ 2.54272x^3 - 7.02027x^2 + 0.00007x + 3.99998 & \text{for } x \in [\frac{\pi}{60}, \frac{\pi}{30}) \\ 2.65909x^3 - 7.05683x^2 + 0.00389x + 3.99985 & \text{for } x \in [\frac{\pi}{30}, \frac{\pi}{20}) \\ 2.75813x^3 - 7.10350x^2 + 0.01123x + 3.99946 & \text{for } x \in [\frac{\pi}{20}, \frac{\pi}{15}) \\ 2.84154x^3 - 7.15591x^2 + 0.02220x + 3.99870 & \text{for } x \in [\frac{\pi}{15}, \frac{\pi}{12}) \\ 2.91075x^3 - 7.21027x^2 + 0.03643x + 3.99745 & \text{for } x \in [\frac{\pi}{12}, \frac{\pi}{10}) \\ 2.96689x^3 - 7.26318x^2 + 0.05306x + 3.99571 & \text{for } x \in [\frac{\pi}{10}, \frac{7\pi}{60}) \\ 3.01080x^3 - 7.31145x^2 + 0.07075x + 3.99355 & \text{for } x \in [\frac{7\pi}{60}, \frac{2\pi}{15}) \\ 3.04302x^3 - 7.35195x^2 + 0.08771x + 3.99118 & \text{for } x \in [\frac{2\pi}{15}, \frac{3\pi}{20}) \\ 3.06389x^3 - 7.38144x^2 + 0.10161x + 3.98900 & \text{for } x \in [\frac{3\pi}{20}, \frac{\pi}{6}) \\ 3.07347x^3 - 7.39649x^2 + 0.10949x + 3.98763 & \text{for } x \in [\frac{\pi}{6}, \frac{11\pi}{60}) \\ 3.07165x^3 - 7.39336x^2 + 0.10769x + 3.98797 & \text{for } x \in [\frac{11\pi}{60}, \frac{\pi}{5}) \\ 3.05817x^3 - 7.36794x^2 + 0.09171x + 3.99132 & \text{for } x \in [\frac{\pi}{5}, \frac{13\pi}{60}) \\ 3.03260x^3 - 7.31573x^2 + 0.05618x + 3.99938 & \text{for } x \in [\frac{13\pi}{60}, \frac{7\pi}{30}) \\ 2.99446x^3 - 7.23185x^2 - 0.00531x + 4.01440 & \text{for } x \in [\frac{7\pi}{30}, \frac{\pi}{4}) \end{cases}$$

We now obtain the cubic B-spline results using the above polynomials, as well as the

results for the central finite difference method using the system linear of equations (23). Table 7 shows the comparison of these results with the exact solution (26).

x_i	Cubic B-Spline Approximation	Finite Difference Method	Exact
0	4.0000000000	4.0000000000	4.0000000000
$\frac{\pi}{60}$	3.9811025365	3.9817079768	3.9811495230
$\frac{\pi}{30}$	3.9259216332	3.9270105161	3.9260011002
$\frac{\pi}{20}$	3.8366445222	3.8381015310	3.8367442991
$\frac{\pi}{15}$	3.7155589531	3.7172762777	3.7156689945
$\frac{\pi}{12}$	3.5650382186	3.5669162139	3.5651504068
$\frac{\pi}{10}$	3.3875276578	3.3894753481	3.3876356206
$\frac{7\pi}{60}$	3.1855323844	3.1874678274	3.1856313280
$\frac{2\pi}{15}$	2.9616059850	2.9634565066	2.9616925424
$\frac{3\pi}{20}$	2.7183399385	2.7200422515	2.7184120337
$\frac{\pi}{6}$	2.4583535256	2.4598537418	2.4584102510
$\frac{11\pi}{60}$	2.1842840072	2.1855375547	2.1843255138
$\frac{\pi}{5}$	1.8987768798	1.8997483341	1.8988042779
$\frac{13\pi}{60}$	1.6044760379	1.6051388754	1.6044913061
$\frac{7\pi}{30}$	1.3040137014	1.3043499846	1.3040196024
$\frac{\pi}{4}$	1.0000000000	1.0000000000	1.0000000000

Table 7: Cubic B-spline and FDM Comparison to the Exact Solution for Ex 4

Table 8 gives the absolute error for both methods. We see that the cubic B-spline method gives a better approximation to this boundary value problem.

x_i	Cubic B-Spline Error	Finite Difference Error
0	0.0000000000	0.0000000000
$\frac{\pi}{60}$	0.0000469865	0.0005584538
$\frac{\pi}{30}$	0.0000794670	0.0010094159
$\frac{\pi}{20}$	0.0000997769	0.0013572319
$\frac{\pi}{15}$	0.0001100413	0.0016072832
$\frac{\pi}{12}$	0.0001121882	0.0017658070
$\frac{\pi}{10}$	0.0001079629	0.0018397274
$\frac{7\pi}{60}$	0.0000989436	0.0018364994
$\frac{2\pi}{15}$	0.0000865574	0.0017639642
$\frac{3\pi}{20}$	0.0000720951	0.0016302178
$\frac{\pi}{6}$	0.0000567253	0.0014434908
$\frac{11\pi}{60}$	0.0000415066	0.0012120409
$\frac{\pi}{5}$	0.0000273981	0.0009440562
$\frac{13\pi}{60}$	0.0000152683	0.0006475693
$\frac{7\pi}{30}$	0.0000059010	0.0003303822
$\frac{\pi}{4}$	0.0000000000	0.0000000000

Table 8: Error of the Cubic B-spline and Finite Difference Method for Ex 4

A comparison of errors is presented in Figure 18

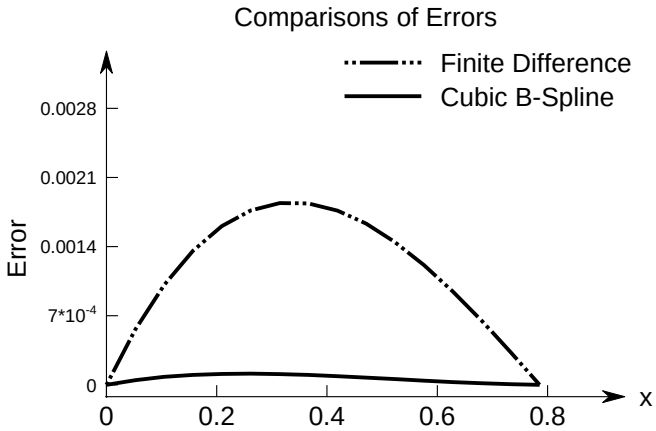


Figure 18: The Cubic B-spline and Finite Difference Errors for Ex 4

It is observed that the maximum-absolute error for the cubic B-spline is much smaller compared with the maximum-absolute error for the central finite difference method. In this sense, the cubic B-spline method is founded to be superior than the central finite difference method.

Example 5: Here we consider the following nonhomogeneous linear boundary value problem with constant coefficients

$$y'' + y' - 6y = x \text{ for } 0 < x < 1 \tag{27}$$

with boundary conditions

$$y(0) = 0 \tag{28}$$
$$y(1) = 1.$$

The exact solution to this boundary value problem is

$$y = \frac{(43 - e^2)e^{-3x} - (43 - e^{-3})e^{2x}}{36(e^{-3} - e^2)} - \frac{1}{6}x - \frac{1}{36}. \tag{29}$$

We approximate the solution of this BVP using the cubic B-spline method with $N = 20$. Again, we compare our numerical results with results obtained from the finite difference method.

We apply the same steps as we did in example 4. After finding the set of cubic polynomials, we obtain the results shown in Table 9. These results compare the solutions obtained using the cubic B-spline method and the central finite difference method with the exact solution.

x_i	Cubic B-Spline Approximation	Finite Difference Method	Exact
0.00	0.0000000000	0.0000000000	0.0000000000
0.05	0.0275351538	0.0275720526	0.0275370031
0.10	0.0542500333	0.0543245716	0.0542570003
0.15	0.0806989046	0.0808109859	0.0807133503
0.20	0.1074050277	0.1075538334	0.1074285617
0.25	0.1348698493	0.1350539152	0.1349034523
0.30	0.1635814257	0.1637986845	0.1636255435
0.35	0.1940222368	0.1942700311	0.1940768511
0.40	0.2266765374	0.2269516052	0.2267412146
0.45	0.2620373739	0.2623358089	0.2621112965
0.50	0.3006133878	0.3009305754	0.3006953693
0.55	0.3429355142	0.3432660448	0.3430239998
0.60	0.3895636812	0.3899012382	0.3896567348
0.65	0.4410936054	0.4414308305	0.4411888843
0.70	0.4981637819	0.4984921134	0.4982584988
0.75	0.5614627582	0.5617722428	0.5615536314
0.80	0.6317367884	0.6320158620	0.6318199790
0.85	0.7097979583	0.7100331954	0.7098689951
0.90	0.7965328796	0.7967087056	0.7965865702
0.95	0.8929120525	0.8930104158	0.8929423791
1.00	1.0000000000	1.0000000000	1.0000000000

Table 9: Cubic B-spline and FDM Comparison to the Exact Solution for Ex 5

Then Table 10 presents the absolute error for both approximations. Figure 19 displays the errors.

x_i	Cubic B-Spline Error	FiniteDifferenceError
0.00	0.0000000000	0.0000000000
0.05	0.0000018493	0.0000350495
0.10	0.0000069670	0.0000675713
0.15	0.0000144457	0.0000976356
0.20	0.0000235340	0.0001252717
0.25	0.0000336029	0.0001504630
0.30	0.0000441179	0.0001731409
0.35	0.0000546144	0.0001931800
0.40	0.0000646773	0.0002103905
0.45	0.0000739226	0.0002245123
0.50	0.0000819815	0.0002352062
0.55	0.0000884855	0.0002420450
0.60	0.0000930536	0.0002445034
0.65	0.0000952789	0.0002419462
0.70	0.0000947169	0.0002336146
0.75	0.0000908731	0.0002186114
0.80	0.0000831905	0.0001958830
0.85	0.0000710368	0.0001642003
0.90	0.0000536906	0.0001221354
0.95	0.0000303266	0.0000680368
1.00	0.0000000000	0.0000000000

Table 10: Errors of the Cubic B-spline and Finite Difference Method for Ex 5

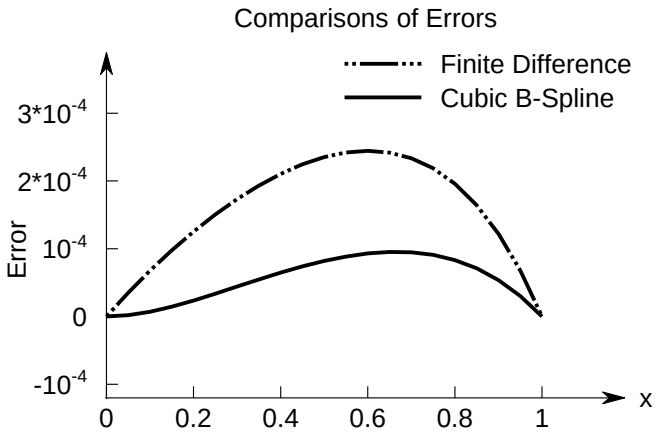


Figure 19: The Cubic B-spline and Finite Difference Errors for Ex 5

In this case, the maximum-absolute error for the cubic B-spline is founded to be much less than the maximum-absolute error for the central finite difference method.

Example 6: We consider the following linear boundary value problem with variable coefficients

$$x^2y'' + 3xy' + 3y = 0 \text{ for } 1 < x < 2 \tag{30}$$

with boundary conditions

$$y(1) = 5 \tag{31}$$
$$y(2) = 0.$$

The exact solution to this boundary value problem is

$$y(x) = \frac{5}{x}[\cos(\sqrt{2}\ln x) - \cot(\sqrt{2}\ln 2)\sin(\sqrt{2}\ln x)]. \tag{32}$$

We want to approximate the solution of this BVP using $N = 20$. In this case, the constant coefficients are computed as

$c_{-1} = 5.498922492$	$c_0 = 4.994073274$	$c_1 = 4.524784412$	$c_2 = 4.089302834$	$c_3 = 3.685654832$
$c_4 = 3.311784751$	$c_5 = 2.965642493$	$c_6 = 2.645237403$	$c_7 = 2.348670027$	$c_8 = 2.074149336$
$c_9 = 1.820000442$	$c_{10} = 1.584666183$	$c_{11} = 1.366704832$	$c_{12} = 1.164785437$	$c_{13} = 0.977681805$
$c_{14} = 0.804265806$	$c_{15} = 0.643500436$	$c_{16} = 0.494432924$	$c_{17} = 0.356188065$	$c_{18} = 0.227961883$
$c_{19} = 0.109015687$	$c_{20} = -0.001329460$	$c_{21} = -0.103697849$		

the following cubic polynomials are derived:

$$Y(x) = \begin{cases} -2.33743x^3 + 14.12436x^2 - 30.97781x + 24.19088 & \text{for } x \in [1.00, 1.05) \\ -2.63161x^3 + 15.05103x^2 - 31.95081x + 24.53143 & \text{for } x \in [1.05, 1.10) \\ -2.74087x^3 + 15.41159x^2 - 32.34744x + 24.67686 & \text{for } x \in [1.10, 1.15) \\ -2.73346x^3 + 15.38604x^2 - 32.31805x + 24.66559 & \text{for } x \in [1.15, 1.20) \\ -2.65421x^3 + 15.10071x^2 - 31.97566x + 24.52864 & \text{for } x \in [1.20, 1.25) \\ -2.53260x^3 + 14.64470x^2 - 31.40564x + 24.29113 & \text{for } x \in [1.25, 1.30) \\ -2.38804x^3 + 14.08090x^2 - 30.67270x + 23.97352 & \text{for } x \in [1.30, 1.35) \\ -2.23318x^3 + 13.45373x^2 - 29.82604x + 23.59252 & \text{for } x \in [1.35, 1.40) \\ -2.07621x^3 + 12.79446x^2 - 28.90305x + 23.16180 & \text{for } x \in [1.40, 1.45) \\ -1.92230x^3 + 12.12495x^2 - 27.93225x + 22.69258 & \text{for } x \in [1.45, 1.50) \\ -1.77460x^3 + 11.46030x^2 - 26.93528x + 22.19409 & \text{for } x \in [1.50, 1.55) \\ -1.63492x^3 + 10.81079x^2 - 25.92853x + 21.67394 & \text{for } x \in [1.55, 1.60) \\ -1.50417x^3 + 10.18318x^2 - 24.92437x + 21.13839 & \text{for } x \in [1.60, 1.65) \\ -1.38267x^3 + 9.58175x^2 - 23.93200x + 20.59259 & \text{for } x \in [1.65, 1.70) \\ -1.27036x^3 + 9.00897x^2 - 22.95827x + 20.04081 & \text{for } x \in [1.70, 1.75) \\ -1.16694x^3 + 8.46601x^2 - 22.00809x + 19.48653 & \text{for } x \in [1.75, 1.80) \\ -1.07197x^3 + 7.95315x^2 - 21.08496x + 18.93265 & \text{for } x \in [1.80, 1.85) \\ -0.98492x^3 + 7.47005x^2 - 20.19121x + 18.38151 & \text{for } x \in [1.85, 1.90) \\ -0.90525x^3 + 7.01592x^2 - 19.32836x + 17.83504 & \text{for } x \in [1.90, 1.95) \\ -0.83239x^3 + 6.58969x^2 - 18.49721x + 17.29479 & \text{for } x \in [1.95, 2.00) \end{cases}$$

Table 11 compares the cubic B-spline approximation using the above polynomials and the central finite difference approximation. Table 12 presents the absolute error for both methods.

x_i	Cubic B-Spline Approximation	Finite Difference Method	Exact
1.00	5.0000000000	5.0000000000	5.0000000000
1.05	4.5304189596	4.5301144218	4.5304962322
1.10	4.0946084301	4.0941176228	4.0947693502
1.15	3.6906178188	3.6900230208	3.6908571967
1.20	3.3164060550	3.3157641925	3.3167126115
1.25	2.9699320212	2.9692820685	2.9702917258
1.30	2.6492103552	2.6485785304	2.6496084276
1.35	2.3523444743	2.3517479159	2.3527665477
1.40	2.0775446353	2.0769940213	2.0779773959
1.45	1.8231362142	1.8226376390	1.8235677149
1.50	1.5875616676	1.5871179919	1.5879814418
1.55	1.3693784917	1.3689903176	1.3697775482
1.60	1.1672547313	1.1669211075	1.1676254805
1.65	0.9799630775	0.9796820102	0.9802992219
1.70	0.8063742444	0.8061430663	0.8066706529
1.75	0.6454500792	0.6452657156	0.6457026575
1.80	0.4962366996	0.4960958598	0.4964422651
1.85	0.3578578444	0.3577571572	0.3580140078
1.90	0.2295085473	0.2294446564	0.2296136048
1.95	0.1104491952	0.1104188253	0.1105020313
2.00	0.0000000000	0.0000000000	0.0000000000

Table 11: Cubic B-spline and FDM Comparison to the Exact Solution for Ex 6

x_i	Cubic B-Spline Error	Finite Difference Error
1.00	0.0000000000	0.0000000000
1.05	0.0000772726	0.0003818104
1.10	0.0001609202	0.0006517275
1.15	0.0002393779	0.0008341759
1.20	0.0003065565	0.0009484190
1.25	0.0003597046	0.0010096573
1.30	0.0003980724	0.0010298972
1.35	0.0004220734	0.0010186319
1.40	0.0004327606	0.0009833746
1.45	0.0004315007	0.0009300759
1.50	0.0004197742	0.0008634500
1.55	0.0003990565	0.0007872306
1.60	0.0003707492	0.0007043730
1.65	0.0003361444	0.0006172117
1.70	0.0002964084	0.0005275866
1.75	0.0002525783	0.0004369420
1.80	0.0002055654	0.0003464053
1.85	0.0001561633	0.0002568506
1.90	0.0001050575	0.0001689484
1.95	0.0000528360	0.0000832060
2.00	0.0000000000	0.0000000000

Table 12: Error of the Cubic B-spline and Finite Difference Method for Ex 6

Figure 20 displays the error graph for this example. Here, we also find the same conclusion as before regarding the errors.

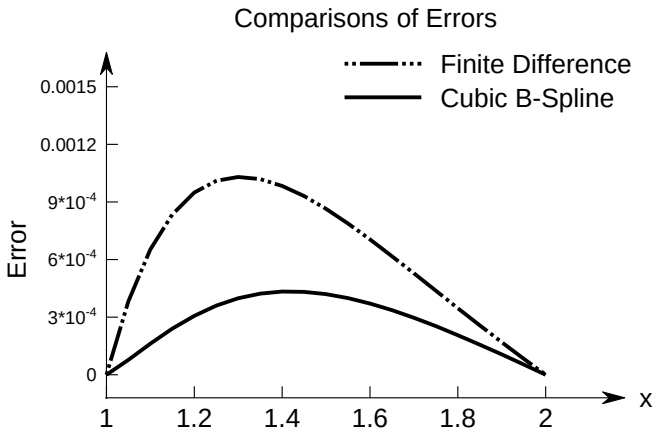


Figure 20: The Cubic B-spline and Finite Difference Errors for Ex 6

In all our examples, the error for the cubic B-spline method shows significant difference compared to the central finite difference method. It is observed that the results obtained using the cubic B-spline method is superior to the results obtained by the finite difference method.

Conclusion

We derived the cubic B-spline and made use of the functions for two kind of applications: (i) to interpolate and (ii) to obtain solutions of second-order linear boundary value problems. In order to interpolate given data, we used two types of boundary conditions, the natural and complete boundary conditions. We compared our interpolation results with results obtained using the Lagrange interpolation. It is observed that our method gives superior results than the results obtained from Lagrange interpolation. We also found that the complete cubic B-spline gives better approximations than the natural cubic B-spline near the end points. For our second application, we used cubic B-splines to obtain numerical solutions of boundary value problems. Results for these examples were presented and compared with results using the central finite difference method. In all cases, our investigation shows that cubic B-splines gave significantly better approximations to the exact solution than the central finite difference method. Also, the cubic B-spline method creates unique cubic polynomials at each subinterval, regardless of the value of N. This leads to smooth results.

REFERENCES

Bhatta D., (2008). Use of modified Bernstein polynomials to solve KdV-Burgers equation numerically. *Applied Mathematics and Computation*, 206, 457-464.

Bhatta D. & Bhatti M., (2006). Numerical solution of KdV equation using modified Bernstein polynomials. *Applied Mathematics and Computation*, 174, 1255-1268.

Bhatti M. I. & Bracken P., (2006). Solutions of Differential Equations in a Bernstein Polynomial Basis. *Journal of Computational and Applied Mathematics*, 205, 272-280.

Birkhoff G. & De Boor C., (1964). Error Bounds for Spline Interpolation. *Journal of Mathematics and Mechanics*, 13(5), 827-836.

Burden R. L. & Faires J. D., (2011). Numerical Analysis. *Brooks/Cole*, 105-144 & 672-696.

Cheng F. Y. & Zizhi F., (1990). Computational Mechanics in Structural Engineering: Recent developments and future trends. *Taylor & Francis Group*, 77-83.

Fang Q., Tsuchiya T., & Yamamoto T., (2001). Finite Difference, Finite Element and Finite Volume Methods Applied to Two-point Boundary Value Problems. *Journal of Computational and Applied Mathematics*, 139, 9-19.

Hall C. A., (1968). On Error Bounds for Spline Interpolation. *Journal of Approximation Theory*, 1(2), 209-218.

Kincaid D. & Cheney W., (2002). Numerical Analysis: Mathematics of Scientific Computing. *AMS*, 308-377 & 572-589.

Phillips G. M., (2003). Interpolation and Approximation by Polynomials. *Springer*, 215-246.

Raghavarao C. V., Sanyasiraju Y. V. S. S., & Suresh S., (1993). A Note on Application of Cubic Splines to Two Point Boundary Value Problems. *Computers & Mathematics with Application*, 27(11), 45-48.

Zarowski C. J., (2004). An Introduction to Numerical Analysis for Electrical and Computer Engineers. *WILEY*, 269-284.

Zill D. G. & Wright W. S., (2012). Differential Equations with Boundary-Value Problems. *Brooks/Cole*, 380-384.

BIOGRAPHICAL SKETCH

Maria Munguia, born August 6, 1989, grew up in Mission, TX. She graduated from Sharyland High school in 2008. She attended South Texas College earning an Associates of Science in Mathematics in 2010. Later she transferred to the University of Texas-Pan American where she graduated summa cum laude in 2012 with a Bachelors of Science in Applied Mathematics. She went on to earn her Masters of Science in Mathematical Science with Concentration in Applied Mathematics from the University of Texas-Pan American in 2014. While pursuing her degree, she served as a graduate teaching assistant. She was a member in the Society for Industrial and Applied Mathematics and Phi Theta Kappa. Her permanent mailing address is 123 Quebec, Mission, TX, 78572.