**UDC 519.6**
**IRSTI 27.41.77**

# SPLINES INTERPOLATION ANALYSIS USING MAPLE PACKAGE

**Aqila Karimi, Yalda Qani**
Faryab University, Maymana, Afghanistan
e-mail: yalda.qani@gmail.com
ORCID ID: https://orcid.org/0000-0002-3030-5783

**Abstract** Expressing relationships in data with functions is both useful and convenient. It allows us to estimate dependent variables at values of independent variables not given in the data, take derivatives, integrate, and even solve differential equations. This is done by interpolation in a precise way. Interpolation is a set of function points that passes through each data point. Linear polynomial interpolation, cubic spline, Lagrange and Newton are common interpolation methods. In interpolation problems, spline interpolation is often preferred over polynomial interpolation because it gives similar results even when using low-order polynomials, while avoiding the phenomenon common at higher orders. will be We used simple linear functions to introduce some basic concepts and issues related to spline interpolation. Then we derive an algorithm for fitting quadratic lines to the data. Finally, we present material on the cubic spline, which is the most common and useful version in engineering practice. Cubic spline is easy to display and calculate by Maple package. Because Maple is closed, the same math symbols used in classrooms can be used to enter data. In addition, the Maple package has many features, including converting outputs to MATLAB codes and LaTeX commands, where computational problems related to the subject are presented with precise and explicit answers**.**

**Keywords:** cubic spline, interpolation, Maple package, natural spline**.**

## Introduction

Before examining spline interpolation, we first discuss about spline polynomials because in interpolation, the interpolation space is always involved and this space has foundations: for example, in Lagrange interpolation, which is a special type of polynomial interpolation, several Lagrange's terms form the basis of space, which is also the case in other types. Regarding spline interpolation, it can be said that this interpolation is a linear combination of spline polynomials such as $S_i(x)$, which is introduced below:

$$p(x) = \sum_{i=1}^{n} C_i S_i(x)$$

Once $S_i(x)$, is known, it suffices to obtain cis. In this case, the interpolation polynomials are discovered. For this purpose, we first discuss the space consisting of spline polynomials[1].

A spline is a function which, together with its several derivatives, is continuous on $[x_0, x_n]$ and is such that on each separate subinterval $[x_i,\ x_{i+1}]$ it is some algebraic polynomial. The terminology was introduced by I. J. Schoenberg in 1946, although the idea was used earlier informally by several authors. Technically a 'spline' is a draughtsman's flexible implement which is used to draw 'smooth' curves through a series of points [2]. Splines were first introduced in 1946 by a person named I. J. Schoenbery [1]. The term "spline" originated from a thin, lexible strip, known as a spline, used by draftsmen to draw smooth curves over a set of points marked by pegs or nails. The data points at which two splines meet are called knots. The most commonly used splines are cubic splines, which produce very smooth connections over adjacent intervals [6]. Polynomial interpolation is an ancient practice, but the heavy industrial use of interpolation began with cubic splines in the 20th century [8].The much broader application of splines to the areas of data fitting and computer-aided geometric design became evident with the widespread availability of computers in the 1960s [3]. Interpolation actually describes the problem of finding a curve (called an interpolation) that passes through a given set of real values $f_0, f_{1,\cdots,f_n}$ at real data points$x_0, x_{1,\cdots,x_n}$, which are sometimes called abscissae or nodes [10].Spline becomes very handy when considering functions that are used in applications requiring data interpolation and/or smoothing. Most graphic user interface software makes use of the spline interpolation for data such as ASCII and image dataset. The interesting part of spline interpolation is its flexibility to fits low-degree polynomials to small subsets of the values [5].

## Materials and methods

Because of these tasks, spline, cubic spline and B-spline interpolation were carefully calculated and analyzed using the following methods in MAPL.

### Spline

We precisely defined the specified data points using the Maple Spline package. The result is a piecewise polynomial. We also specified the independent variable and then defined the spline. As you can see, we enter a list of data points in the first step to analyze the issue accurately.

> [[−5 , 5] ,[0 , 2] ,[4 , 0]]

$$[[-5, 5], [0, 2], [4, 0]]$$

Now we specify the independent variable and then define the spline.

> $s := CurveFitting[Spline]((\mathbf{1}), x)$

$$s := \begin{cases} 2 - \frac{49}{90}x + \frac{1}{60}x^2 + \frac{1}{900}x^3 & x < 0 \\ 2 - \frac{49}{90}x + \frac{1}{60}x^2 - \frac{1}{720}x^3 & otherwise \end{cases}$$

Now if we want to calculate a spline with specified end conditions, in that The Spline routine computes a degree d piecewise polynomial in variable v that approximates the points $\{(x_0, y_0), (x_1, y_1), \cdots (x_n, y_n)\}$. It should be said that if v is a numerical value, the polynomial value is returned at this point. The default value of d is 3. This can be changed using the degree=d option.

The main purpose of the Spline function is to calculate and return the piecewise formula of a spline interpolant. On numerical data, to perform fast spline interpolation, the Curve Fitting [Array Interpolation] command should be used instead. We can call the Spline routine in two ways. First, Spline ($xy$data,$v$,$d$gr,endpts) accepts a list, array, or matrix , $[[x_0, y_0], [x_1, y_1], \cdots [x_n, y_n]]$ of data points.

**Examples**

> $with(CurveFitting)$ :

> $Spline([[0, 0], [1, 1], [2, 4], [3, 3]], v)$

$$\begin{cases} \frac{4}{5}v^3 + \frac{1}{5}v & v < 1 \\ -2v^3 + \frac{42}{5}v^2 - \frac{41}{5}v + \frac{14}{5} & v < 2 \\ \frac{6}{5}v^3 - \frac{54}{5}v^2 + \frac{151}{5}v - \frac{114}{5} & otherwise \end{cases}$$

> $Spline([[0, 0], [1, 5], [2, -1], [3, 0]], v, degree = 2, endpoints = 'periodic')$

$$\begin{cases} \frac{16}{5}v^2 + \frac{24}{5}v & v < \frac{1}{2} \\ -\frac{44}{5}v^2 + \frac{84}{5}v - 3 & v < \frac{3}{2} \\ \frac{28}{5}v^2 - \frac{132}{5}v + \frac{147}{5} & v < \frac{5}{2} \\ \frac{16}{5}v^2 - \frac{72}{5}v + \frac{72}{5} & otherwise \end{cases}$$

Returns the linear system of equations used to solve the cubic spline interpolants.

**Description**

- The LinearSystem command retrieves the matrix and vector in the linear system of equations that were solved when computing the cubic spline interpolants.

- To calculate the cubic spline interpolants, the linear system of equations was solved and the matrix and vector were recovered using the LinearSystem command.
- A POLYINTERP structure is created using the CubicSpline command.
- The LinearSystem command only accepts interpolation structures created using theCubicSpline command, because the cubic spline interpolation method is the only method that has an associated linear system [https://www.maplesoft.com].

## Examples

> $with(Student_{NumericalAnalysis}):$

> $xy := \left[[0,1], \left[\frac{1}{2}, 1\right], \left[1, \frac{11}{10}\right], \left[\frac{3}{2}, \frac{3}{4}\right], \left[2, \frac{7}{8}\right], \left[\frac{5}{2}, \frac{9}{10}\right], \left[3, \frac{11}{10}\right], \left[\frac{7}{2}, 1\right]\right]$

$$xy := \left[[0,1], \left[\frac{1}{2}, 1\right], \left[1, \frac{11}{10}\right], \left[\frac{3}{2}, \frac{3}{4}\right], \left[2, \frac{7}{8}\right], \left[\frac{5}{2}, \frac{9}{10}\right], \left[3, \frac{11}{10}\right], \left[\frac{7}{2}, 1\right]\right]$$

> $p1 := CubicSpline(xy, independentvar = 'x'):$

> $LinearSystem(p1)$

$$\left[\begin{array}{cccccccc} 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0.5000000000 & 2. & 0.5000000000 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0.5000000000 & 2. & 0.5000000000 & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.5000000000 & 2. & 0.5000000000 & 0. & 0. & 0. \\ 0. & 0. & 0. & 0.5000000000 & 2. & 0.5000000000 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0.5000000000 & 2. & 0.5000000000 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0.5000000000 & 2. & 0.5000000000 \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. \end{array}\right], \left[\begin{array}{c} 0. \\ 0.8291995878 \\ -2.116798351 \\ 2.237993817 \\ -1.135176915 \\ 1.102713844 \\ -1.175678461 \\ 0. \end{array}\right]$$

**Linear Splines**

The simplest connection between two points is a straight line. First-order contours for a group of ordered data points can be defined as a set of linear functions [4]. With linear splines, straight lines (linear functions) are used for interpolation between the data points [6]. An arbitrary collection of n data points can be successfully interpolated by the linear spline. Linear splines, however, are not smooth. This flaw in linear splines is intended to be fixed by cubic splines. Degree 3 (cubic) polynomials are used in place of linear functions between the data points by a cubic spline [8].

**The B-Spline curve**

The B-Spline curve routine calculates a B-Spline curve based on control points. The B-Spline routine computes a piecewise function representing the kth order B-spline in the symbol v. The non-zero parts of this function are polynomials of degree$k-1$. If the knots option is not provided, the uniform knot list $[0,1,\cdots,k]$ is used. The node list must contain exactly $k+1$ elements. These elements must be in non-decreasing order. Otherwise, unexpected results may be produced. Nodes can have a number of up to $k-1$. If the number of a node is m, the continuity in that node is $C(k-m-1)$. This method returns a B-spline basis function.

Use the Curve Fitting[BSplineCurve] procedure to create a B-spline curve. This function is part of the Curve Fitting package and therefore it can be used in the form of BSpline(..) only after executing the command with (Curve Fitting). However, it can always be accessed via the long form of the command using Curve Fitting[BSpline] [https://www.maplesoft.com/].

**Examples**

> $with(CurveFitting):$

> $BSpline(2, u)$

$$\begin{cases} 0 & u < 0 \\ u & u < 1 \\ 2-u & u < 2 \\ 0 & 2 \le u \end{cases}$$

> $BSpline(2, u, knots = [0, a, 2])$

$$\begin{cases} 0 & u < 0 \\ \dfrac{u}{a} & u < a \\ \dfrac{-u+a}{2-a}+1 & u < 2 \\ 0 & 2 \leq u \end{cases}$$

**Cubic spline**

The concept of a spline originates from the drafting technique of using a thin, flexible strip (called a spline) to draw smooth curves through a set of points. In this technique, the draftsman places the paper on a wooden board and nails or pins into the paper (and the board) at the data points. A smooth cubic curve results from interweaving the tape between the pins. Hence, the name "cubic spline" has been adapted for polynomials of this type[3].Cubic splines give useful alternatives to plain polynomials. Cubic splines are piecewise cubic functions s that are continuous and have continuous first and second derivatives [9]. The objective in cubic splines is to derive a third-order polynomial for each interval between knots [4]. Cubic splines may guarantee the continuity of the first and second derivatives at the nodes and are therefore more commonly used in practice [7]. in cubic splines, third-degree polynomials are used to interpolate over each interval between data points. Suppose there are $n+1$ data points $(x_1, y_1), \ldots, (x_{n+1}, y_{n+1})$ so that there are n intervals and thus n cubic polynomials. Each cubic polynomial is conveniently expressed in the form

$$S_i(x) = a_i(x-x_1)^3 + b_i(x-x_1)^2 + C_i(x-x_i) + d_i, \quad i = 1,2,\cdots n$$

where $a_i$, $b_i$, $c_i$, $d_i$ $(i = 1, 2, \ldots, n)$ are unknown constants to be determined [6]. The Cubic Spline command interpolates the given $xy$ data points using the cubic spline method and stores all the computed information in a POLYINTERP structure. The POLYINTERP structure is then passed to various interpolation commands in the Student [Numerical Analysis] sub package, where information can be extracted from it and manipulated depending on the command. A residual term is not calculated by the Cubic Spline command, so it cannot be used in conjunction with the Remainder Term command or the Interpolant Remainder Term command. This method works numerically. That is, inputs that are not numeric are first evaluated to floating point numbers before continuing calculations [https://www.maplesoft.com/].

Examples

```
> with(Student[NumericalAnalysis]) :
> xy := [[0, 4.0], [0.5, 0], [1.0, -2.0], [1.5, 0], [2.0, 1.0], [2.5, 0], [3.0, -0.5]]
xy := [[0, 4.0], [0.5, 0], [1.0, -2.0], [1.5, 0], [2.0, 1.0], [2.5, 0], [3.0, -0.5]]
> p1 := CubicSpline(xy, independentvar = x) :
> expand(Interpolant(p1))
```

$$\begin{cases} 4. - 8.48076923076923\,x + 1.92307692307692\,x^3 \\ -5.13461538461539\,x + 3.44230769230769 - 6.69230769230769\,x^2 + 6.38461538461538 \\ 21.2884615384615 - 58.6730769230769\,x + 46.8461538461538\,x^2 - 11.4615384615385\,x \\ 15.0576923076923\,x - 15.5769230769231 - 2.30769230769231\,x^2 - 0.538461538461538 \\ -64.8076923076923 + 88.9038461538461\,x - 39.2307692307692\,x^2 + 5.61538461538461 \\ -52.4423076923077\,x + 52.9807692307692 + 17.3076923076923\,x^2 - 1.92307692307692 \end{cases}$$

```
> Draw(p1)
```

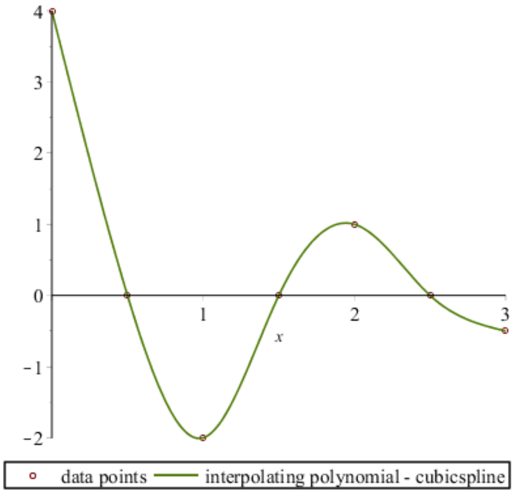


**Figure 1 -** Cubic spline interpolation natural boundary conditions.

> $p2 := CubicSpline(xy, independentvar = x, boundaryconditions = clamped(0, 6))$ :

> $Draw(p2)$

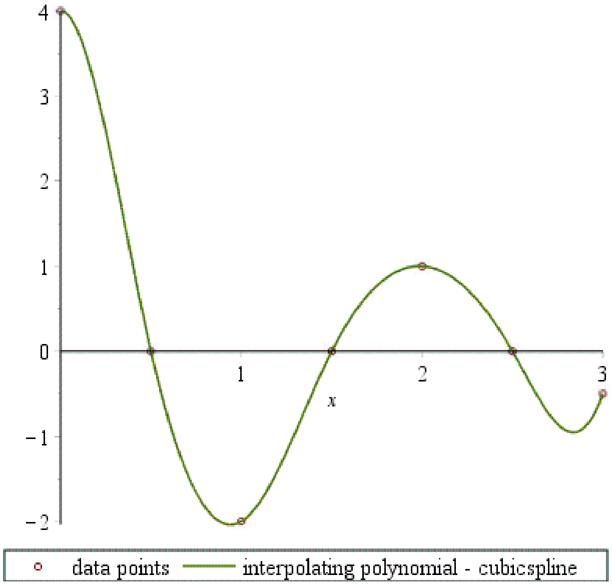


**Figure 2-** Cubic spline interpolation clamped boundary conditions.

## Conclusion

Interpolation is a set of function points that passes through each data point, and expressing relationships in data with functions is both useful and convenient. This is done by interpolation in a precise way. In interpolation problems, spline interpolation is often preferred over polynomial interpolation because it gives similar results even when using low-order polynomials, while avoiding the phenomenon common at higher orders. to be The Maple package has good computing capabilities. We chose Mapel package for accurate data analysis. From Mapel package, you can use mathematical symbols used in classrooms to enter data. In addition, the Maple package has many features, including converting outputs to MATLAB codes and LaTeX commands, where computational problems related to the subject are presented with precise and explicit answers. Using the capabilities of the LinearSystem command, we recovered the matrix and vector in the linear system of equations that were solved when calculating the cubic spline interpolants. Also, to calculate the cubic spline interpolants, the linear system of equations was solved and the matrix and vector were recovered using the LinearSystem command. We also showed that a POLYINTERP structure is created using the CubicSpline command. The LinearSystem command only accepts interpolation structures created using the CubicSpline command, because the cubic spline interpolation method is the only method that has an associated linear system.

**References**

[1] Allahviranloo T., Pedrycz W., Esfandiari A. Advances in Numerical Analysis Emphasizing Interval Data. CRC Press. 2022. 86.

[2] Bose S. K. Numerical methods of mathematics implemented in Fortran. Springer Singapore. 2019. 197

[3] Burden R. L., Faires J. D., Burden, A. M. Numerical analysis. Cengage learning. 2015. 143.

[4] Chapra S. C., Canale R. P. Numerical methods for engineers, Seventh., 2015. 33.

[5] Emetere, M. E. Numerical Methods in Environmental Data Analysis. Elsevier. 2022. 179.

[6] Esfandiari R. S. Numerical methods for engineers and scientists using MATLAB®. Crc Press. 2017. 194

[7] Li H. Introduction to Numerical Methods in Java. Numerical Methods Using Java: For Data Science, Analysis, and Engineering, 2022. 1-69. 249.

[8] Sauer T. Numerical analysis. Addison-Wesley Publishing Company. 2011. 138.

[9] Stewart D. Numerical Analysis: A Graduate Course. Springer. 2022. 258.

[10] Faul A. C. A Concise introduction to numerical analysis. Chapman and Hall/CRC. 2018. 70.

[11] https://www.maplesoft.com/