# Cubic spline interpolation using Mathcad

**Conference Paper** · September 2009



Nicolae Dăneţ
Technical University of Civil Engineering of Bucharest

# CUBIC SPLINE INTERPOLATION USING MATHCAD

Nicolae Dăneţ

Abstract. In this paper we develop and implement in Mathcad an algorithm for the construction of the cubic spline functions with end slope boundary conditions. By effectuating simple modification of this algorithm we can obtain cubic spline functions that satisfy other boundary conditions. So we can show what are the boundary conditions used by Mathcad functions lspline, pspline and cspline.

2000 *Mathematics Subject Classification*: Primary 65D07. Secondary 68N30.

## 1. Introduction

Nowadays all computation are made using a computer and a modern software. But in some situations the software programs are not so well documented and the user does not know exactly in what conditions he can use a function, because the functions are created according to the principle of a black box. We know the name of the function, the inputs and the outputs, but we do not know the internal mechanism. A black box is defined by "what" it does and not by "how" it does it. We can successfully use functions like black boxes if we know the numerical method implemented in each function, its performance and its limitation, the cases for which it is recommended and the cases for which it is prohibited. But this information is not always given to the end user.

Let us consider the following example. It is well known that in order to obtain a cubic spline function we have to determine $4n$ unknowns and we have only $4n-2$ conditions (see Section 2 for details). To determine a unique cubic spline function it is necessary to add two additional boundary conditions. In literature, see for example [1, 2, 5, 6, 7], several conditions are known that lead to obtaining diverse spline functions which differ among themselves only near the endpoints (see Example 3.5).

Mathcad software has for interpolation three functions, `lspline`, `pspline` and `cspline`, that are used together with `interp` function. In Mathcad Help these functions have a short description, which contains no indications about the boundary conditions used by every of them.

In this paper we develop and implement in Mathcad an algorithm for the construction of the cubic spline functions with end slope boundary conditions. By effectuating simple modification of this algorithm we can obtain cubic spline functions that satisfy other boundary conditions. So we can show what are the boundary conditions used by Mathcad functions `lspline`, `pspline` and `cspline`. For this we construct spline functions in two ways: first, directly, by using Mathcad functions `lspline`, `pspline` and `cspline`, and, second, by using the proper algorithm implemented in Mathcad. Then we compare the values of these functions in some points.

## 2. Cubic spline functions

In this section we recall some basic facts about the cubic spline functions and give a unitary method of construction for these functions when they have different boundary conditions. In the first part we describe an algorithm for obtaining a cubic spline function with end slope boundary conditions, and then we show how this algorithm can be modified in order to obtain cubic spline functions that satisfy other boundary conditions.

**Definition 1** *Let $[a,b]$ be an interval of the real axis and $a = x_0 < x_1 < \cdots < x_n = b$ a partition of it. A function $S : [a,b] \longrightarrow \mathbf{R}$ is called a cubic spline function if it satisfies the following conditions:*

1) *$S$ and its derivatives $S'$ and $S''$ are continuous on the interval $[a,b]$.*

2) *On each subinterval $[x_i, x_{i+1}]$, $i = 0,1,\ldots,n-1$, $S(x)$ is a cubic polynomial: $S(x) = a_i + b_i x + c_i x^2 + d_i x^3$.*

Let $f : [a,b] \longrightarrow \mathbf{R}$ be a function such that we know its values only in the nodes $x_i$, $i = 0,1,\ldots,n$. We denote these values by $f_i = f(x_i)$, $i = 0,1,\ldots,n$. Our goal is to look for a cubic spline function $S$ satisfying the interpolation conditions

$$S(x_i) = f_i, \quad i = 0,1,\ldots,n. \tag{1}$$

To determine a cubic spline interpolant it is necessary to determine $4n$ unknowns $(a_i, b_i, c_i, d_i)$, $i = 0,1,\ldots,n-1$. For this operation we have $n+1$

interpolation conditions (1) and $3(n-1)$ continuousness conditions

$$\begin{cases} S(x_i-0)=S(x_i+0), & i=1,2,\ldots,n-1,\\ S'(x_i-0)=S'(x_i+0), & i=1,2,\ldots,n-1,\\ S''(x_i-0)=S''(x_i+0), & i=1,2,\ldots,n-1. \end{cases} \tag{2}$$

Therefore we have $3(n-1)+(n+1)=4n-2$ conditions for $4n$ unknowns. There are two degrees of freedom left.

To obtain a unique spline function two boundary conditions must be added to remove these two degrees of freedom.

**Case 1. End slope boundary conditions**

$$S'(x_0)=f'(x_0), \quad S'(x_n)=f'(x_n). \tag{3}$$

These endpoint derivative conditions assure the existence of an unique cubic spline function.

**Theorem 1** *There is a unique cubic spline function $S(x)$ that satisfies the interpolation conditions*

$$S(x_i)=f_i, \quad i=0,1,\ldots,n,$$

*and the end slope boundary conditions*

$$S'(x_0)=f'(x_0), \quad S'(x_n)=f'(x_n).$$

*Proof.* We give a short proof of this theorem because the algorithm we implemented in Mathcad used this method of construction for a cubic spline function and the notation established here (see also [3] and [4]).

Let $h_i=x_{i+1}-x_i$, $i=0,1,\ldots,n-1$. Denote by $m_i=S''(x_i)$, $i=0,1,\ldots,n$, the values of the second derivatives of the unknown spline function $S(x)$ at the nodes. $m_0,m_1\ldots,m_n$ are called the moments of $S(x)$.

In the first step of the proof we show that the spline function $S(x)$ is characterized by its moments $m_i$, and in the second step that the moments $m_i$ can be calculated as solution of a system of linear equations.

*Step* 1. Since the second derivative $S''(x)$ is a linear function on the interval $[x_i,x_{i+1}]$ and $S''(x_i)=m_i, S''(x_{i+1})=m_{i+1}$, then this linear function has the form

$$S''(x)=m_i\frac{x_{i+1}-x}{h_i}+m_{i+1}\frac{x-x_i}{h_i}. \tag{4}$$

By integration we have

$$S(x) = m_i \frac{(x_{i+1} - x)^3}{6h_i} + m_{i+1} \frac{(x - x_i)^3}{6h_i} + A_i \frac{x_{i+1} - x}{h_i} + B_i \frac{x - x_i}{h_i},$$

for $x \in [x_i, x_{i+1}]$. Using the interpolation conditions (1) we obtain the values for the constants of integration $A_i$ and $B_i$:

$$A_i = f_i - m_i \frac{h_i^2}{6}, \quad B_i = f_{i+1} - m_{i+1} \frac{h_i^2}{6}.$$

Therefore we have the following representation of the cubic spline function $S(x)$ in terms of its moments $m_i$:

$$\begin{aligned} S(x) = {} & m_i \frac{(x_{i+1} - x)^3}{6h_i} + m_{i+1} \frac{(x - x_i)^3}{6h_i} + \\ & + \left( f_i - m_i \frac{h_i^2}{6} \right) \frac{x_{i+1} - x}{h_i} + \left( f_{i+1} - m_{i+1} \frac{h_i^2}{6} \right) \frac{x - x_i}{h_i}, \end{aligned} \tag{5}$$

where $x \in [x_i, x_{i+1}]$ and $i = 0, 1, \ldots, n-1$. The resulting function $S(x)$ is continuous on the interval $[a, b]$ and satisfies the interpolation conditions (1).

*Step* 2. To determine the moments $m_0, \ldots, m_n$ we use the condition that the first derivative $S'(x)$ is continuous in the internal nodes $x_1, \ldots, x_{n-1}$, that is,

$$S'(x_i - 0) = S'(x_i + 0), \quad i = 1, 2, \ldots, n-1. \tag{6}$$

By using the expression of the derivative $S'(x)$ for $x \in [x_i, x_{i+1}]$ and for $x \in [x_{i-1}, x_i]$, the continuousness conditions (6) yield the equalities

$$m_i \frac{h_{i-1}}{2} + \frac{f_i - f_{i-1}}{h_{i-1}} - \frac{m_i - m_{i-1}}{6} h_{i-1} = -m_i \frac{h_i}{2} + \frac{f_{i+1} - f_i}{h_i} - \frac{m_{i+1} - m_i}{6} h_i.$$

After the rearrangement of terms we obtain

$$\frac{h_{i-1}}{6} m_{i-1} + \frac{h_{i-1} + h_i}{3} m_i + \frac{h_i}{6} m_{i+1} = \frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}, \tag{7}$$

for $i = 0, 1, \ldots, n-1$.

In order to simplify the form of these equations we introduce the following notation

$$b_i = \frac{h_{i-1}}{h_{i-1} + h_i}, \quad c_i = \frac{h_i}{h_{i-1} + h_i}, \tag{8}$$

$$d_i = \frac{6}{h_{i-1} + h_i} \left( \frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}} \right). \tag{9}$$

We observe that

$$b_i > 0, c_i > 0, \text{ and } b_i + c_i = 1. \tag{10}$$

Then the equations (7) are written in a uniform format

$$b_i m_{i-1} + 2\, m_i + c_i m_{i+1} = d_i, \quad i = 1, \dots, n-1. \tag{11}$$

These equations form a system of $n-1$ equations with $n+1$ unknowns $m_0, m_1, \dots, m_n$. The two necessary equations to obtain a system of $n+1$ equations with $n+1$ unknowns are obtained from the end slope boundary conditions (3).

The condition $S'(x_0) = f'(x_0)$ gives the equation

$$2\, m_0 + m_1 = d_0, \tag{12}$$

where

$$d_0 = \frac{6}{h_0}\left(\frac{f_1 - f_0}{h_0} - f'_0\right). \tag{13}$$

Analogously, the condition $S'(x_n) = f'(x_n)$ gives the equation

$$m_{n-1} + 2m_n = d_n, \tag{14}$$

with

$$d_n = \frac{6}{h_{n-1}}\left(f'_n - \frac{f_n - f_{n-1}}{h_{n-1}}\right). \tag{15}$$

The equations (11), (12) and (14) lead to the following system of $n+1$ linear equations with $n+1$ unknowns $m_0, m_1, \dots, m_n$, where we put

$$c_0 = 1,\ b_n = 1 \tag{16}$$

for the uniformity of the writing.

$$\left\{\begin{array}{ccccccccccc}
2m_0 & + & c_0 m_1 & & & & & & & = & d_0, \\
b_1 m_0 & + & 2m_1 & + & c_1 m_2 & & & & & = & d_1, \\
 & & b_2 m_1 & + & 2m_2 & + & c_2 m_3 & & & = & d_2, \\
 & & & \ddots & & \ddots & & \ddots & & & \vdots \\
 & & & & b_{n-1} m_{n-2} & + & 2m_{n-1} & + & c_{n-1} m_n & = & d_{n-1} \\
 & & & & & & b_n m_{n-1} & + & 2m_n & = & d_n.
\end{array}\right.$$

In matrix notation, this system has the form

$$\begin{bmatrix} 2 & c_0 & 0 & 0 & \cdots & 0 \\ b_1 & 2 & c_1 & 0 & \cdots & 0 \\ 0 & b_2 & 2 & c_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & b_{n-1} & 2 & c_{n-1} \\ 0 & 0 & \cdots & 0 & b_n & 2 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}. \tag{17}$$

If we denote by $\mathbf{A}$ the $(n+1) \times (n+1)$ matrix of the system, with $\mathbf{m} = (m_0, m_1, \ldots, m_n)^T$ the vector of the moments and with $\mathbf{d} = (d_0, d_1, \ldots, d_n)^T$ the vector of the right hand side, then the system (17) can be written

$$\mathbf{Am} = \mathbf{d}. \tag{18}$$

Since $c_0 = 1$, $b_n = 1$, and $b_i + c_i = 1$, $i = 1, \ldots, n-1$, the matrix $\mathbf{A}$ is strictly diagonally dominant, therefore is nonsingular ([2], p.404), and the system (18) has a unique solution. (For a short proof that $\mathbf{A}$ is a nonsingular matrix see [7], p.101.) Because the matrix $\mathbf{A}$ is tridiagonal this system can be solved easily and rapidly using a $LU$ decomposition method (Crout factorization for tridiagonal linear systems, see [2], p.414). $\square$

End slope boundary conditions lead to an accurate approximation since they include more information about the function ([2], p.154). In order to use these conditions, it is necessary to have either the values of the derivative at the end points $x_0$ and $x_n$, or an accurate approximation of those values.

Mathcad does not have a specific function dedicated to the construction of a cubic spline function that satisfies the end slope boundary conditions. But we have the possibility to implement in Mathcad the algorithm developed in the proof of Theorem 1 (see Example 3.1).

**Case 2. Natural boundary conditions**

$$S''(x_0) = S''(x_n) = 0. \tag{19}$$

These conditions will generally give less accurate results that the end slope boundary conditions near the ends of the interval $[x_0, x_n]$ (unless the function $f$ happens to nearly satisfy $f''(x_0) = f''(x_n) = 0$). The cubic spline function which satisfies the natural boundary conditions is called a *natural spline*.

**Theorem 2** *There is a unique cubic spline function $S(x)$ that satisfies the interpolation conditions*

$$S(x_i) = f_i, \quad i = 0, 1, \ldots, n,$$

*and the natural boundary conditions*

$$S''(x_0) = S''(x_n) = 0.$$

*Proof.* In this case we have $m_0 = S''(x_0) = 0$ and $m_n = S''(x_n) = 0$. The $n-1$ equations (11) are sufficient for the determination of the $n-1$ moments $m_1, m_2, \ldots, m_{n-1}$. If we change the relations (16) with $c_0 = 0$ and $b_n = 0$, and define $d_0 = 0$ and $d_n = 0$ in (13) and (15), respectively, then we can use the system (17) for the determination of the moments $m_0, m_1, \ldots, m_n$. The matrix of this system is also strictly diagonally dominant. □

Natural splines are obtained in Mathcad using the functions `lspline` and `interp` (see Example 3.2). With `lspline` we obtain the vector **m**, here denoted by **ml** in order to know that it was obtained with `lspline` function. More precisely, we have

```
ml:=lspline(x,f),
```

where $\mathbf{x} = (x_0, \ldots, x_n)^T$ is the vector of nodes and $\mathbf{f} = (f_0, \ldots, f_n)^T$ is the vector of the values of function $f$ at these nodes. The corresponding cubic spline function, denoted here by $SL(z)$, is obtained by using Mathcad function `interp`.

```
SL(z):=interp(ml,x,f,z).
```

This function could be written in one line as follows

```
SL(z):=interp(lspline(x,f),x,f,z).
```

**Case 3. Cubic spline functions with the boundary conditions**

$$S''(x_0) = S''(x_1), \quad S''(x_{n-1}) = S''(x_n). \tag{20}$$

This assumption states that $S''(x)$ is constant in the first and last intervals. This implies that $S(x)$ is quadratic in these intervals, $[x_0, x_1]$ and $[x_{n-1}, x_n]$.

**Theorem 3** *There is a unique cubic spline function $S(x)$ that satisfies the interpolation conditions*

$$S(x_i) = f_i, \quad i = 0, 1, \ldots, n,$$

*and the boundary conditions*

$$S''(x_0) = S''(x_1), \quad S''(x_{n-1}) = S''(x_n).$$

*Proof.* In this case we have $m_0 = m_1$ and $m_{n-1} = m_n$. By using these conditions in the first and last equations (11) we obtain the folloowing linear system

$$\begin{bmatrix} b_1+2 & c_1 & 0 & 0 & \cdots & 0 \\ 0 & b_2 & 2 & c_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & b_{n-2} & 2 & c_{n-2} \\ 0 & 0 & \cdots & 0 & b_{n-1} & 2+c_{n-1} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-2} \\ m_{n-1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} \end{bmatrix}$$

which has a strongly diagonally dominant matrix. Hence the system has a unique solution. $\square$

If we change the relations (16) with $c_0 = -2$ and $b_n = -2$, and put $d_0 = 0$ and $d_n = 0$ in (13) and (15), respectively, then we can have the same system (17) for the determination of the moments $m_0, m_1, \ldots, m_n$ (see Example 3.3).

This type of cubic spline function is obtained in Mathcad using the functions `pspline` and `interp` (see Example 3.3). More precisely, we have

```
SP(z):=interp(pspline(x,f),x,f,z).
```

**Case 4. Cubic spline with $S''(x)$ values at endpoints as linear extrapolations**

In this case, the values of $S''(x_0)$ and $S''(x_n)$ are taken as linear extrapolation of the $S''(x)$ values of the two nearest nodes [6]. For $i = 1$ the relation (4), that is, the expression of $S''(x)$ on the interval $[x_1, x_2]$, becomes

$$S''(x) = m_1 \frac{x_2 - x}{h_1} + m_2 \frac{x - x_1}{h_1}, \quad x \in [x_1, x_2].$$

The value of this function at the point $x_0 \notin [x_1, x_2]$ is

$$S''(x_0) = m_1 \frac{x_2 - x_0}{h_1} + m_2 \frac{x_0 - x_1}{h_1}. \tag{21}$$

By using the notation $m_0 = S''(x_0)$, $h_0 = x_1 - x_0$, and $h_1 = x_2 - x_1$, the relation (21) can be written in the form

$$m_0 - m_1 \left(1 + \frac{h_0}{h_1}\right) + m_2 \frac{h_0}{h_1} = 0. \tag{22}$$

Similarly, by using the expression of $S''(x)$ on the interval $[x_{n-2}, x_{n-1}]$ and taken the value of this function at $x_n \notin [x_{n-2}, x_{n-1}]$, we obtain the following relation between $m_{n-2}, m_{n-1}, m_n$ :

$$m_{n-2}\frac{h_{n-1}}{h_{n-2}} - m_{n-1}\left(1 + \frac{h_{n-1}}{h_{n-2}}\right) + m_n = 0. \tag{23}$$

**Theorem 4** *There is a unique cubic spline function* $S(x)$ *that satisfies the interpolation conditions*

$$S(x_i) = f_i, \quad i = 0, 1, \ldots, n,$$

*and the boundary conditions* (22) *and* (23).

*Proof.* The equations (11) together with (22) and (23) forms a system of $n+1$ equations with $n+1$ unknowns, $m_0, m_1, \ldots, m_n$. In order to see that this system has a unique solution we eliminate the unknown $m_0$ between the equation (22) and the first equation (11) and the unknown $m_m$ between the equation (23) and the last equation (11). The obtained linear system with the unknown $m_1, m_2, \ldots, m_{n-1}$ has the matrix

$$\begin{bmatrix} 2 + \dfrac{h_0}{h_1} & 1 - \dfrac{h_0}{h_1} & 0 & 0 & \cdots & 0 \\ b_2 & 2 & c_2 & 0 & \cdots & 0 \\ 0 & b_3 & 2 & c_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & 0 \\ 0 & \cdots & 0 & b_{n-2} & 2 & c_{n-2} \\ 0 & \cdots & 0 & 0 & 1 - \dfrac{h_{n-1}}{h_{n-1}} & 2 + \dfrac{h_{n-1}}{h_{n-1}} \end{bmatrix}$$

which is strongly diagonally dominant. Therefore the system has a unique solution $(m_1, m_2, \ldots, m_{n-1})$. The value of $m_0$ is obtained from the equation (22) and the value of $m_n$ from equation (23). $\square$

We can use the same system (17) to compute the solution $(m_0, m_1, \ldots, m_n)$ if we delete the relations (16), put in the first row the coefficients of equation (22) and in the last row the coefficirnts of equation (23), and put $d_0 = 0$ and $d_n = 0$ in (13) and (15), respectively. Then the matrix of the linear system 17 with these changes made is (see Example 3.4):

$$\begin{bmatrix} 1 & -1-\dfrac{h_0}{h_1} & \dfrac{h_0}{h_1} & 0 & \cdots & 0 \\ b_1 & 2 & c_1 & 0 & \cdots & 0 \\ 0 & b_2 & 2 & c_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ 0 & 0 & \cdots & b_{n-1} & 2 & c_{n-1} \\ 0 & 0 & \cdots & \dfrac{h_{n-1}}{h_{n-2}} & -1-\dfrac{h_{n-1}}{h_{n-2}} & 1 \end{bmatrix}.$$

This type of cubic spline function is obtained in Mathcad using the functions `cspline` and `interp` (see Example 3.4). More precisely, we have

```
SC(z):=interp(cspline(x,f),x,f,z).
```

## 3. Cubic spline interpolation using Mathcad

In this section we give some examples of cubic spline interpolation using Mathcad. In Example 3.1 we construct a cubic spline function with end slope boundary conditions using the algorithm described in Theorem 1. Examples 3.2, 3.3 and 3.4 show how cubic spline functions which satisfy the boundary conditions described in Cases 2, 3 and 4 of Section 2 can be obtained either with Mathcad functions or by a short modification of the algorithm of Example 3.1. In Example 3.5 we show the importance of a good knowledge of the boundary conditions for spline interpolation near the end nodes.

In all these examples we use the following conventions: 1) The math regions of Mathcad are written with italic font. 2) The text regions of Mathcad are written we the usual font of all the text of the paper. 3) The graph regions are introduced like images. 4) The vector results are shown in transposed position. 5) All computation are made with fifteen digits, but shown in a shorter form.

**Example 3.1**

In this example we construct a cubic spline function with end slope boundary conditions using the algorithm described in the proof of Theorem 1. We recall that Mathcad does not have a special function for this type of cubic spline interpolation.

The input data:

x - the vector of the nodes;
f - the vector of the values of the function at nodes;
v - a vector which contains some points from the interval of interpolation.

$$x := \begin{bmatrix} 1.00 \\ 1.75 \\ 3.00 \\ 4.10 \\ 5.00 \\ 5.60 \\ 7.00 \end{bmatrix} \qquad f := \begin{bmatrix} 5.25 \\ 2.95 \\ 3.40 \\ 5.60 \\ 4.25 \\ 6.10 \\ 4.75 \end{bmatrix} \qquad v := \begin{bmatrix} 1.50 \\ 3.25 \\ 4.70 \\ 6.55 \end{bmatrix}$$

The length of vector x: $n := last(x)$ $n = 6$

The end slope boundary conditions: $df_0 := -3$ $df_n := -1$

The differences between nodes: $i := 0..n-1$ $h_i := x_{i+1} - x_i$

$$h^T = (0.75 \quad 1.25 \quad 1.10 \quad 0.90 \quad 0.60 \quad 1.40)$$

The coefficients given by the formulas (8):

$$j := 1..n-1 \qquad b_j := \frac{h_{j-1}}{h_{j-1} + h_j} \qquad c_j := \frac{h_j}{h_{j-1} + h_j}$$

The coefficients below are introduced for the uniformity of the writing of the system (17). This line must be modified if we use other boundary conditions.

$$c_0 := 1 \qquad b_n := 1$$

The construction of matrix A:

$a_{0,0} := 2$ $a_{0,1} := c_0$

$j := 1..n-1$ $a_{j,j-1} := b_j$ $a_{j,j} := 2$ $a_{j,j+1} := c_j$

$a_{n,n-1} := b_n$ $a_{n,n} := 2$

$A := a$

$$A = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0.375 & 2 & 0.625 & 0 & 0 & 0 & 0 \\ 0 & 0.532 & 2 & 0.468 & 0 & 0 & 0 \\ 0 & 0 & 0.55 & 2 & 0.45 & 0 & 0 \\ 0 & 0 & 0 & 0.6 & 2 & 0.4 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 2 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

The construction of vector d:

$$D1f_i := \frac{f_{i+1} - f_i}{h_i}$$

$$d_0 := \frac{6}{h_0}(D1f_0 - df_0)$$

$$j := 1..n-1 \qquad d_j := \frac{6}{h_{j-1} + h_j}(D1f_j - D1f_{j-1})$$

$$d_n := \frac{6}{h_0}(df_n - D1f_{n-1})$$

$$d^T = (-0.53 \quad 10.28 \quad 4.19 \quad -10.5 \quad 18.33 \quad -12.14 \quad -0.15)$$

Remark. If we use this algorithm to construct a cubic spline function which satisfies other boundary conditions we must change the definition of the components $d_0$ and $d_n$.

The linear system Am = d is solved in Mathcad using the function lsolve.

$$m := lsolve(A, d)$$

The solution is:

$$m^T = (-2.61 \quad 4.69 \quad 3.00 \quad -9.20 \quad 13.90 \quad -9.85 \quad 4.85)$$

The definition of the cubic spline function on a partial interval (see formula (5)):

$$s(z, x, i) := m_i \frac{(x_{i+1} - z)^3}{6h_i} + m_{i+1} \frac{(z - x_i)^3}{6h_i} \ldots$$
$$+ \left[ f_i - m_i \frac{(h_i)^2}{6} \right] \frac{x_{i+1} - z}{h_i} + \left[ f_{i+1} - m_{i+1} \frac{(h_i)^2}{6} \right] \frac{z - x_i}{h_i}$$

The definition of the cubic spline function on the whole interval:

$$S(z) := \left| \begin{array}{l} vs \leftarrow 0 \\ for\ i \in 0..n-1 \\ vs \leftarrow s(z,x,i) \quad if \quad x_i \leq z \leq x_{i+1} \end{array} \right.$$

Now we can use this function for interpolation. We can compute the value of the cubic spline function at a point, for example,

$$S(2.15) = 2.423$$

or in many point simultaneously if they are declared in a vector (v, in our case).

$$p := 0..last(v) \qquad \begin{array}{ll} v_p = & S(v_p) = \\ 1.50 & 3.626 \\ 3.25 & 4.112 \\ 4.75 & 4.373 \\ 6.55 & 5.532 \end{array}$$

To plot the nodes, we define a vector g that has all components equal with zero:

$k := 0..last(x) \qquad g_k := 0$

Finally, we realize the graphical representation shown in Figure 1.

**Example 3.2**

In this example we construct a natural cubic spline function using the same input data like in Example 3.1. This is done firstly with Mathcad functions `interp` and `lspline,` and secondly by a simple modification of the algorithm of Example 3.1.

Using `lspline` function we compute the vector of moments:

$ml := lspline(x, f)$

$ml^T = (0 \quad 3 \quad 0 \quad 0 \quad 4.16 \quad 3.13 \quad -9.16 \quad 13.53 \quad -8.1 \quad 0)$

The first three components (0, 3, 0) are internal code of Mathcad used by `interp` function to know what type of interpolation to do. The fourth zero and the last zero correspond to the natural boundary conditions.

By using the vector ml and Mathcad function `interp` we define the cubic natural spline function denoted SL(x):

$SL(z) := interp(ml, x, f, z)$

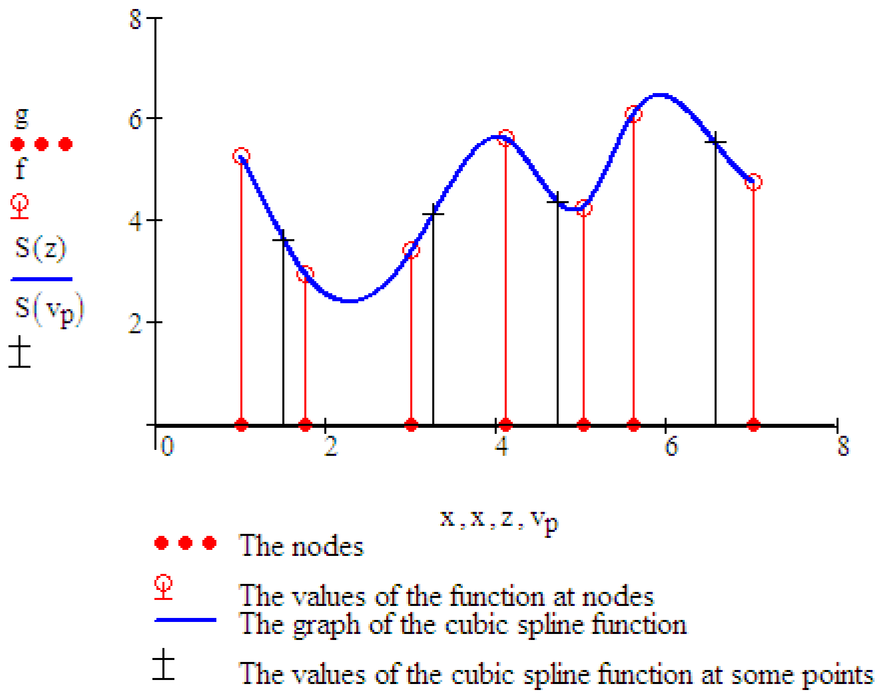


FIGURE 1. A cubic spline function with end slope boundary conditions.

The values of this function at the points contained in vector v are:

$$SL(v) = \begin{bmatrix} 3.572 \\ 4.101 \\ 4.390 \\ 5.947 \end{bmatrix}$$

For plotting the graph of this function we define the range variable t,

$n := last(x) \qquad t := x_0, x_0 + 0.01..x_n$

and a vector g with null components,

$k := 0..n \qquad g_k := 0$

The graph is shown in Figure 2.

To obtain the natural cubic spline using the algorithm of Example 3.1 we must make the following changes:

$c_0 := 0 \qquad b_n := 0 \qquad d_0 = 0 \qquad d_n = 0$

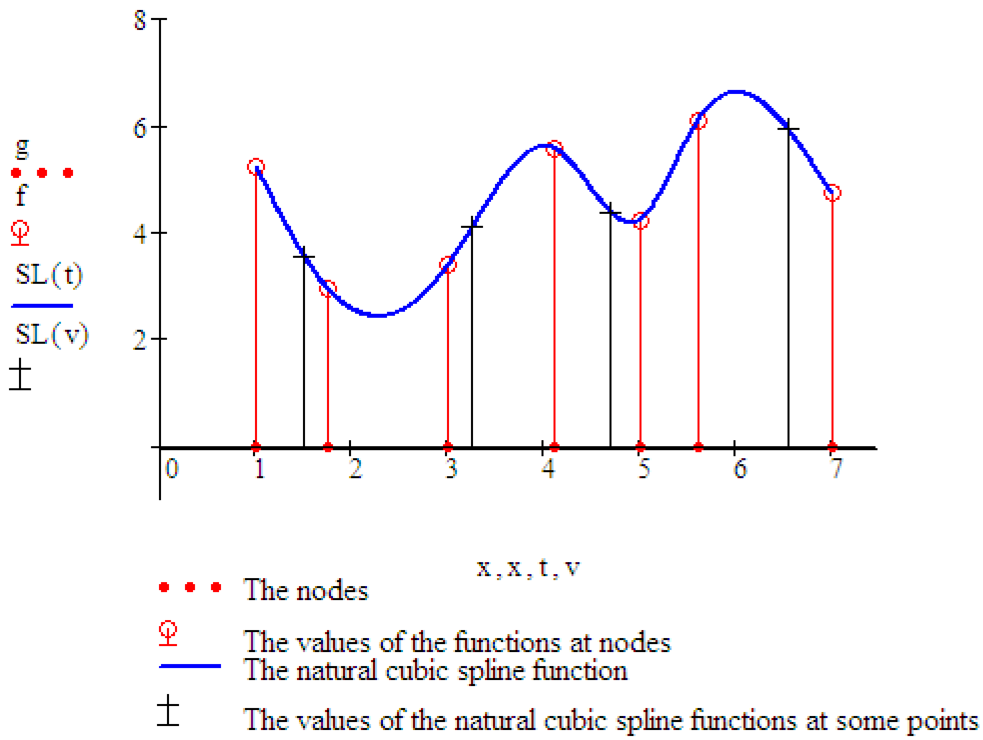


FIGURE 2. A natural cubic spline function.

The obtained spline function is denoted by Sl(z). The following table shows the values of this function at the components of vector v and the differences between these values and the values of the functions SL(x) at the same points.
$p := 0..last(v)$

| $v_p =$ | $Sl(v_p) =$ | $Sl(v_p) - SL(v_p) =$ |
|---|---|---|
| 1.50 | 3.5721518772 | 0.0000000000000 |
| 3.25 | 4.1014766405 | 0.0000000000000 |
| 4.75 | 4.3895442007 | 0.0000000000000 |
| 6.55 | 5.9466972985 | 0.0000000000000 |

**Example 3.3**
In this example we construct a cubic spline functions which satisfies the boundary conditions $S''(x_0) = S''(x_1)$, $S''(x_{n-1}) = S''(x_n)$. The input data are the same like in the above two examples. For this purpose we use first the

Mathcad functions `pspline` and `interp`, and second a modified version of the algorithm of Example 3.1.

$mp := pspline(x, f)$

$mp^T = (0 \quad 3 \quad 1 \quad 3.46 \quad 3.46 \quad 3.3 \quad -9.1 \quad 13.09 \quad -5.95 \quad -5.95)$

The first three components (0, 3, 1) represent the code given by this function to the `interp` function.

$SP(z) := interp(mp, x, f, z)$

$$SP(v) = \begin{bmatrix} 3.500 \\ 4.088 \\ 4.410 \\ 6.456 \end{bmatrix}$$

The function SP(z) could be obtained with the algorithm of Example 3.1 if we do the following changes in it:

$c_0 := -2 \qquad b_n := -2 \qquad d_0 = 0 \qquad d_n = 0$

The obtained function is denoted by Sp(z). We show the values of this function at all points existing in vector v and compare the values of Sp(z) with their of SP(z).

$p := 0..last(v)$

| $v_p =$ | $Sp(v_p) =$ | $Sp(v_p) - SP(v_p) =$ |
|---|---|---|
| 1.50 | 3.5004875631 | 0.0000000000000 |
| 3.25 | 4.0882334341 | 0.0000000000000 |
| 4.75 | 4.4097381712 | 0.0000000000000 |
| 6.55 | 6.4560788161 | 0.0000000000000 |

**Example 3.4**

In this example we construct a cubic spline function which satisfies the boundary conditions obtained by extrapolations (see Case 4 in Section 2). Obviously, we use the same input data like in the above examples.

$mc := cspline(x, f)$

$mc^T = (0 \quad 3 \quad 2 \quad 3.6 \quad 3.46 \quad 3.22 \quad -8.75 \quad 11.64 \quad 0.78 \quad -24.56)$

The first three components (0, 3, 2) represent internal code used by `interp` function.

$SC(z) := interp(mc, x, f, z)$

The values of these function at points contained in vector v are:

$$SC(v) = \begin{bmatrix} 3.497 \\ 4.078 \\ 4.468 \\ 8.048 \end{bmatrix}$$

This function can be obtained by using the algorithm of Example 3.1 if we make the following changes:

1) The coefficient $c_0$ and $b_n$ must be deleted.

2) The non null elements of the first and last line of the matrix A are defined as follows:

$$a_{0,0} := 1 \qquad a_{0,1} := -\left(1 + \frac{h_0}{h_1}\right) \qquad a_{0,2} := \frac{h_0}{h_1}$$

$$a_{n,n-2} := \frac{h_{n-1}}{h_{n-2}} \qquad a_{n,n-1} := -\left(1 + \frac{h_{n-1}}{h_{n-2}}\right) \qquad a_{n,n} := 1$$

3) The lines 2,3,...,n-1 rest unchanged.

4) The first and the last components of the vector d are defined as equal to zero:

$$d_0 := 0 \qquad d_n := 0$$

The cubic spline function such constructed is denoted by Sc(z). Similar to Example 3.2 and 3.3, we show the values of this function at the points contained in v and compare these values with those obtained with SC(z) function.

$p := 0..last(v)$

| $v_p =$ | $Sc(v_p) =$ | $Sc(v_p) - SC(v_p) =$ |
|---|---|---|
| 1.50 | 3.4966223058 | 0.0000000000000 |
| 3.25 | 4.0781840882 | 0.0000000000000 |
| 4.75 | 4.4683196933 | 0.0000000000000 |
| 6.55 | 8.0478124572 | $-1.77635683940025 \cdot 10^{-15}$ |

**Example 3.5**

This example shows the importance of the boundary conditions for the cubic spline interpolation near the end points $x_0$ and $x_n$.

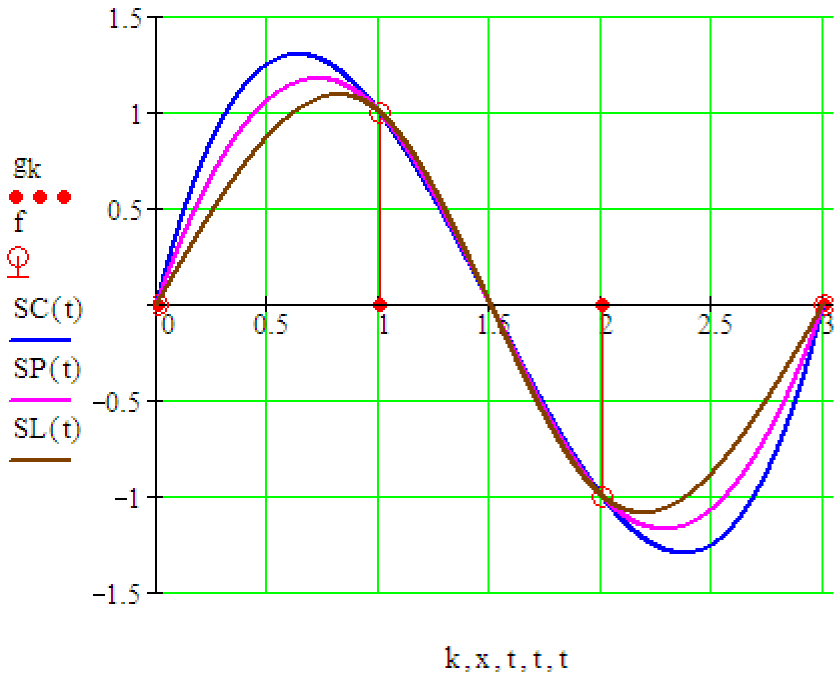


FIGURE 3. The differences between the functions SL, SP and SC near the end points.

The input data:

$$x := \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} \qquad f := \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$$

The cubic spline functions:

$SL(z) := interp(lspline(x, f), x, f, z)$

$SP(z) := interp(pspline(x, f), x, f, z)$

$SC(z) := interp(cspline(x, f), x, f, z)$

Elements necessary for plotting the nodes and these functions:

$n := last(x) \qquad k := 0..n \qquad g_k := 0$

$t := x_0, x_0 + 0.001..x_n$

The graph is shown in Figure 3.

At z = 0.5 these three functions has the following values:

$$SL(0.5) = 0.87500$$
$$SP(0.5) = 1.06250$$
$$SC(0.5) = 1.25000$$

## References

[1] Atkinson, K. E.: *An Introduction to Numerical Analysis*, Second Edition, John Wiley&Sons, New York, Toronto, Singapore, 1988.

[2] Burden, R. L. and Faires, J. D.: *Numerical Analysis*, Sixth edition, Brooks/Cole Publishing Company, ITP An International Thomson Publishing Company, Pacific Grove, London, New York, Paris, Tokyo, 1997.

[3] Dăneţ, N.: *Analiză numerică. Cu aplicaţii rezolvate în Mathcad*, (*Numerical Analysis. With Mathcad*), MatrixRom Technical and Scientific Publishing House, Bucharest, 2002.

[4] Dăneţ, N.: *Analiză numerică. Probleme rezolvate cu Mathcad*, (*Numerical Analysis. Solved Problems with Mathcad*), Mathcad electronic book, MatrixRom Technical and Scientific Publishing House, Bucharest, 2005.

[5] Prenter, P. M.: *Splines and variational methods*, A Wiley-Intersience Publication, John Wiley&Sons, New York, London, Sydney, Toronto, 1975.

[6] Rao, S. S. : *Applied Numerical Methods for Engineers and Scientiests*, Prentice Hall, Upper SaddleRiver, NJ, 2002.

[7] Stoer, J. and Burlish, R.: *Introduction to Numerical Analysis*, Springer-Verlag, New York Heidelberg Berlin, 1980.

Nicolae Dăneţ
Department of Mathematics and Computer Science
Technical University of Civil Engineering of Bucharest
124, Lacul Tei Blvd., 020396 Bucharest, Romania
email:*ndanet@cfdp.utcb.ro*