Topic : Spline Interpolation
Simulation : Graphical Simulation of the Method
Language : Mathcad 2001
Authors : Nathan Collier, Autar Kaw, Ginger Fisher
Date : 26 June 2002
Abstract : This simulation illustrates the spline interpolation method. Given 'n' data points of y versus x, and then required to find the value of 'y' at a particular 'x', you are asked to used quadrature spline interpolation using MATHCAD's internal function are also given.

INPUTS: Enter the following

Array of x data:

$$x := \begin{pmatrix} 10 \\ 0 \\ 20 \\ 15 \\ 30 \\ 22.5 \end{pmatrix}$$

Array of y data

$$y := \begin{pmatrix} 227.04 \\ 0 \\ 517.35 \\ 362.78 \\ 901.67 \\ 602.97 \end{pmatrix}$$

Value of x at which y is desired: $x_{desired} := 16$

SOLUTION

The following functions effectively sort the matrix x and y arrays in ascending order. The augment command, puts the two matrices x and y together.

$tosort := augment(x, y)$

The csort command sorts the matrix by the specified column. In this case the 0 column.

$sorted := csort(tosort, 0)$

$n := rows(x) - 1$

$x := submatrix(sorted, 0, n, 0, 0)$

$y := submatrix(sorted, 0, n, 1, 1)$

$$x = \begin{pmatrix} 0 \\ 10 \\ 15 \\ 20 \\ 22.5 \\ 30 \end{pmatrix} \qquad y = \begin{pmatrix} 0 \\ 227.04 \\ 362.78 \\ 517.35 \\ 602.97 \\ 901.67 \end{pmatrix}$$

Linear spline interpolation:

$$f_{linear}(z) := \left| \begin{array}{l} n \leftarrow \text{rows}(x) - 1 \\ d \leftarrow y_1 + \dfrac{y_1 - y_0}{x_1 - x_0} \cdot (z - x_1) \\ \text{for } i \in 1 .. n-1 \\ \quad \left| \begin{array}{l} m \leftarrow \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i} \\ d \leftarrow y_i + m \cdot (z - x_i) \text{ if } z > x_i \wedge z \le x_{i+1} \end{array} \right. \\ d \end{array} \right.$$

Value of function at desired value

$f_{linear}(x_{desired}) = 393.694$

$f_{prev} := f_{linear}(x_{desired})$

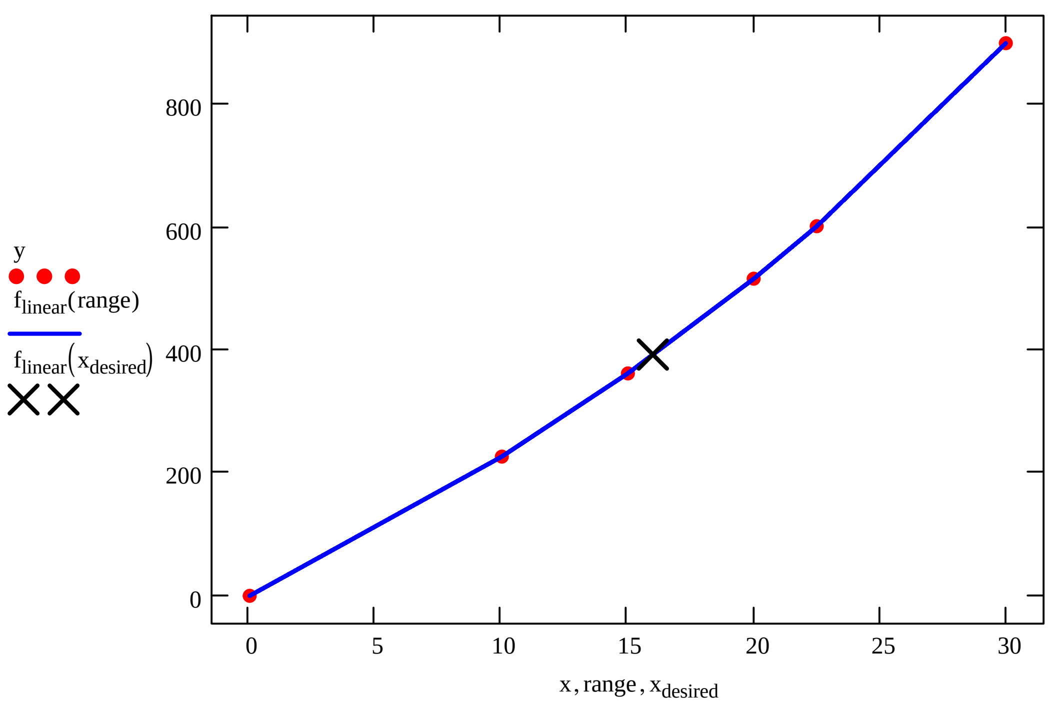
$w := \max(x) - \min(x)$

$h := \max(y) - \min(y)$

$l := \text{rows}(x) - 1$

$\text{range} := x_0, x_0 + \dfrac{w}{10000} .. x_l$

Linear spline interpolation



Quadratic spline interpolation:

The following function assembles the matrix whose inverse is needed to solve for the coefficients of the polynomial splines that fits the data.

$$
A := \left|\begin{array}{l}
n \leftarrow \text{rows}(x) \\
\text{for } i \in 0..3(n-1)-1 \\
\quad \text{for } j \in 0..3(n-1)-1 \\
\quad\quad M_{i,j} \leftarrow 0 \\
\text{for } i \in 1..n-1 \\
\quad \text{for } j \in 0..1 \\
\quad\quad \text{for } k \in 0..2 \\
\quad\quad\quad M_{2\cdot i-1+j,\,3\cdot i-3+k} \leftarrow \left(x_{i-1+j}\right)^{k} \\
\text{for } i \in 1..n-2 \\
\quad \text{for } j \in 0..1 \\
\quad\quad \text{for } k \in 0..1 \\
\quad\quad\quad M_{2(n-1)+i,\,3\cdot i-2+k+j\cdot 3} \leftarrow (-1)^{j}\left(2\cdot x_{i}\right)^{k} \\
M_{0,2} \leftarrow 1 \\
M
\end{array}\right.
$$

The matrix contains several patterns so a loop structure is used to assemble the matrix more efficiently. This loop puts the top part of the matrix which is developed from equating function values at data points.

This loop assembles the lower portion of the matrix which corresponds to equating derivatives of the functions at data points.

A =

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	10	100	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	10	100	0	0	0	0	0	0	0	0	0
4	0	0	0	1	15	225	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	1	15	225	0	0	0	0	0	0
6	0	0	0	0	0	0	1	20	400	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	1	20	400	0	0	0
8	0	0	0	0	0	0	0	0	0	1	22.5	06.25	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	22.5	06.25
10	0	0	0	0	0	0	0	0	0	0	0	0	1	30	900
11	0	1	20	0	-1	-20	0	0	0	0	0	0	0	0	0
12	0	0	0	0	1	30	0	-1	-30	0	0	0	0	0	0
13	0	0	0	0	0	0	0	1	40	0	-1	-40	0	0	0
14	0	0	0	0	0	0	0	0	0	0	1	45	0	-1	-45

This assembles the Y matrix also needed to determine the coefficients of the polynomial splines.

$$
Y := \begin{array}{|l}
n \leftarrow \mathrm{rows}(x) \\
\text{for } i \in 0..3(n-1)-1 \\
\quad M_i \leftarrow 0 \\
\text{for } i \in 0..n-2 \\
\quad \text{for } j \in 0..1 \\
\quad\quad M_{2\cdot(i+1)-1+j} \leftarrow y_{i+j} \\
M
\end{array}
$$

$Y =$

	0
0	0
1	0
2	227.04
3	227.04
4	362.78
5	362.78
6	517.35
7	517.35
8	602.97
9	602.97
10	901.67
11	0
12	0
13	0
14	0

$$C := A^{-1}\cdot Y$$

$C =$

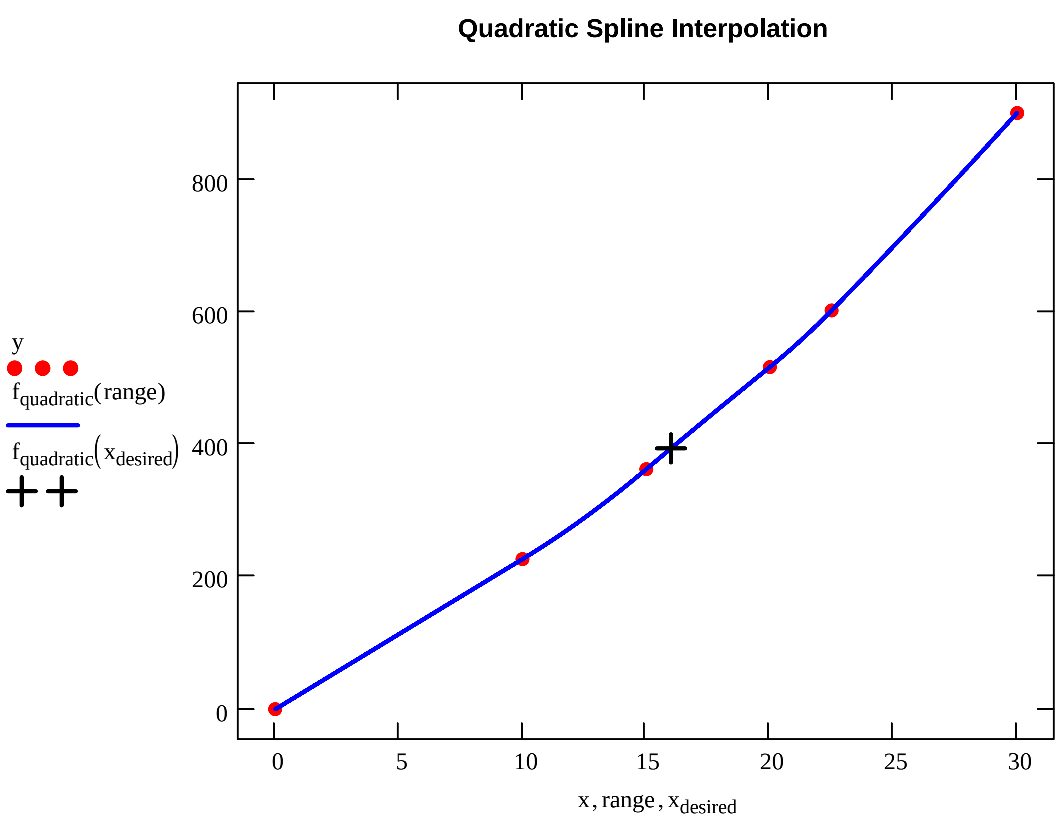
	0
0	0
1	22.704
2	0
3	88.88
4	4.928
5	0.889
6	-141.61
7	35.66
8	-0.136
9	554.55
10	-33.956
11	1.605
12	-152.13
13	28.86
14	0.209

$f_{quadratic}(z) :=$

```
n ← rows(x)
for  i ∈ 0 .. n − 2
  if  z ≤ x_{i+1} ∧ z > x_i
      d ← 0
      for  j ∈ 0 .. 2
        d ← d + C_{3·i+j}·z^j
d
```

Value of function at desired value of x

$f_{quadratic}(x_{desired}) = 394.236$



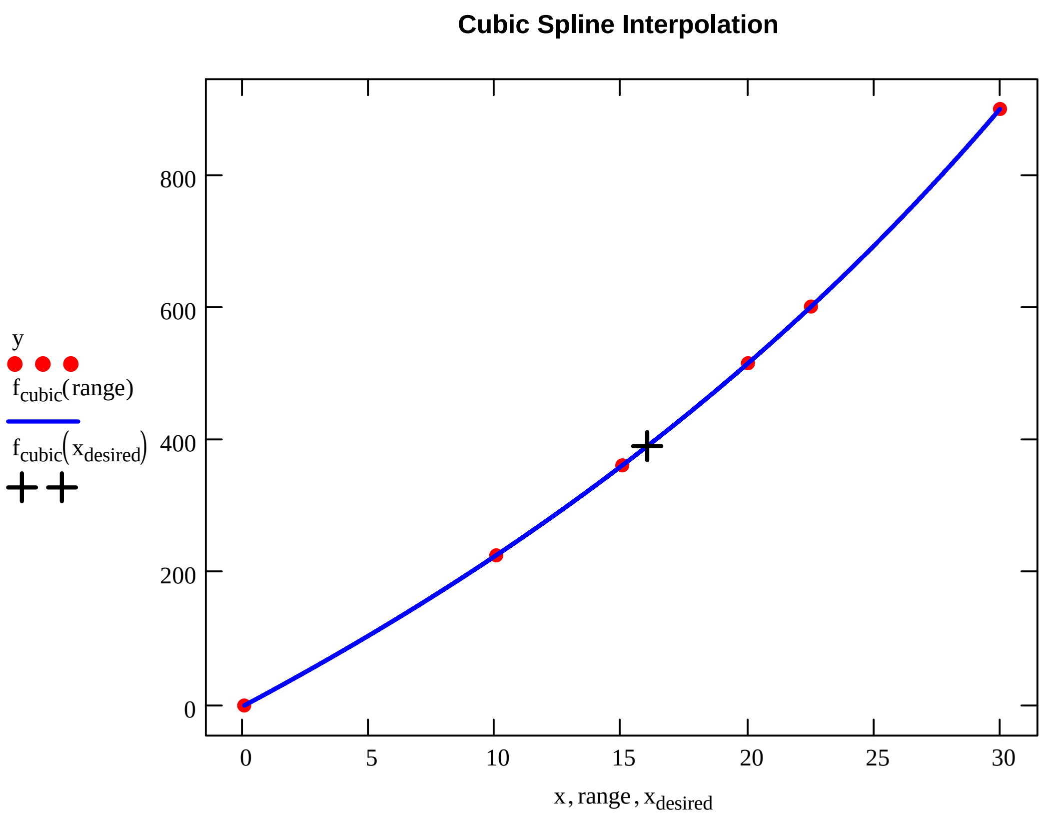
Cubic Spline Interpolation:

$S := \mathrm{cspline}(x, y)$

$f_{cubic}(z) := \mathrm{interp}(S, x, y, z)$

Value of function at desired value of x

$f_{cubic}\left(x_{desired}\right) = 392.071$



$$results := \left| \begin{array}{l} d_0 \leftarrow x_{desired} \\ d_1 \leftarrow f_{linear}(x_{desired}) \\ d_2 \leftarrow f_{quadratic}(x_{desired}) \\ d_3 \leftarrow f_{cubic}(x_{desired}) \\ d^T \end{array} \right.$$

		Spline Type		
	$X_{desired}$	**Linear**	**Quadratic**	**Cubic**
results =	16	393.69	394.24	392.07