Topic : Spline Interpolation
Simulation : Graphical Simulation of the Method
Language : Mathmatica 4.1
Authors : Nathan Collier, Autar Kaw
Date : 18 July 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 *Mathematica*'s internal function are also given.

```
In[1]:= ClearAll;
```

■ INPUTS: Enter the Following

Array of x data

```
In[2]:= x = {10, 0.000001, 20, 15, 30, 22.5};
```

> Note: If you have a zero in the x matrix, the simulation will not work. Try making the zero a near zero.

Array of y data

```
In[3]:= y = {227.04, 0, 517.35, 362.78, 901.67, 602.97};
```

Value of x at which y is desired

```
In[4]:= xdesired := 16
```

■ SOLUTION

The following sorts the x and y arrays in ascending order.

```
In[5]:= nn := Abs[Dimensions[x]]

In[6]:= n := nn[[1]]

In[7]:= xy = Table[0, {i, n}, {j, 2}];

In[8]:= Do[xy[[i, 1]] = x[[i]]; xy[[i, 2]] = y[[i]], {i, 1, n}]

In[9]:= xy = Sort[xy];
```

In[10]:=
```
x = xy[[All, 1]];
y = xy[[All, 2]];
```

The sorted x and y values

In[12]:=
```
x // MatrixForm
```

Out[12]//MatrixForm=

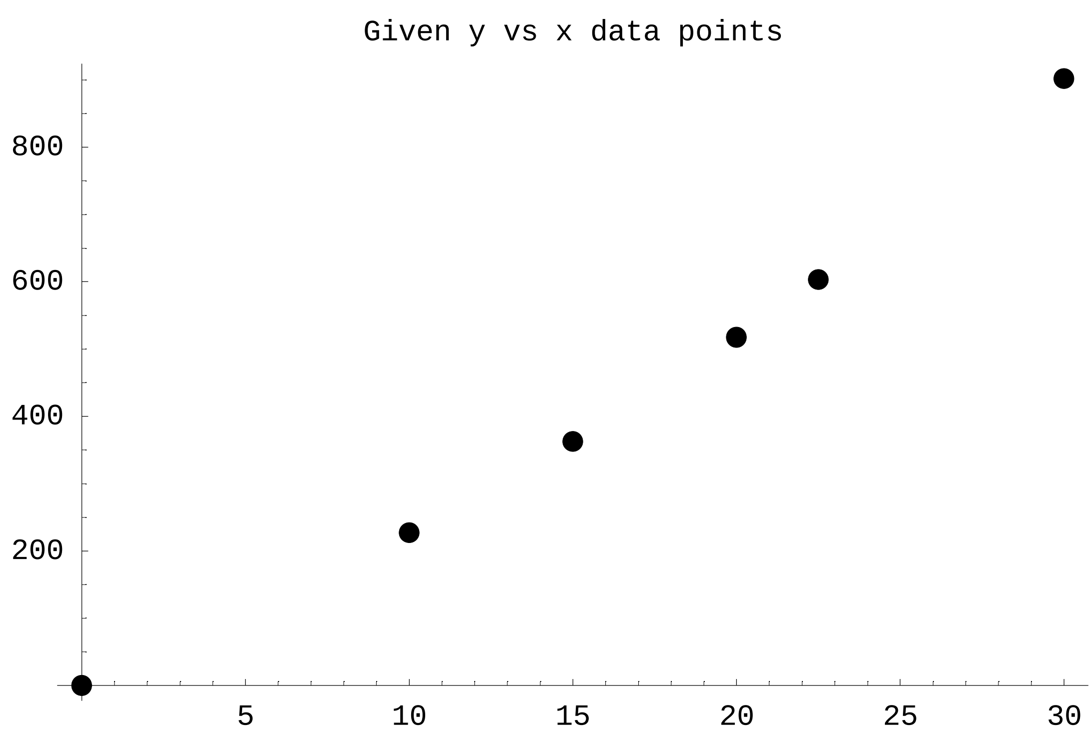
$$\begin{pmatrix} 1.\times 10^{-6} \\ 10 \\ 15 \\ 20 \\ 22.5 \\ 30 \end{pmatrix}$$

In[13]:=
```
y // MatrixForm
```

Out[13]//MatrixForm=

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

In[14]:=
```
data = ListPlot[xy, PlotStyle → PointSize[0.02], PlotLabel ->
    "Given y vs x data points", TextStyle → {FontSize → 11}, PlotRange → All];
```



Linear spline interpolation

```
In[15]:= m[i_, z_] := y[[i + 1]] + (y[[i + 1]] - y[[i]]) / (x[[i + 1]] - x[[i]]) * (z - x[[i + 1]])
```

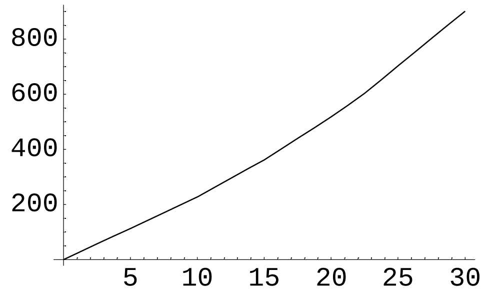
```
In[16]:= flinear[z_] := If[z < x[[2]], i = 1; m[i, z],
           Do[If[z ≤ x[[j + 1]] && z > x[[j]], i = j], {j, 1, n - 1}]; m[i, z]]
```

Value of function at desired value

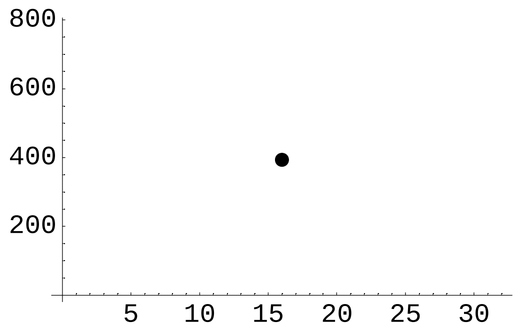
```
In[17]:= flinear[xdesired]
Out[17]= 393.694
```

```
In[18]:= fprev = %;
```

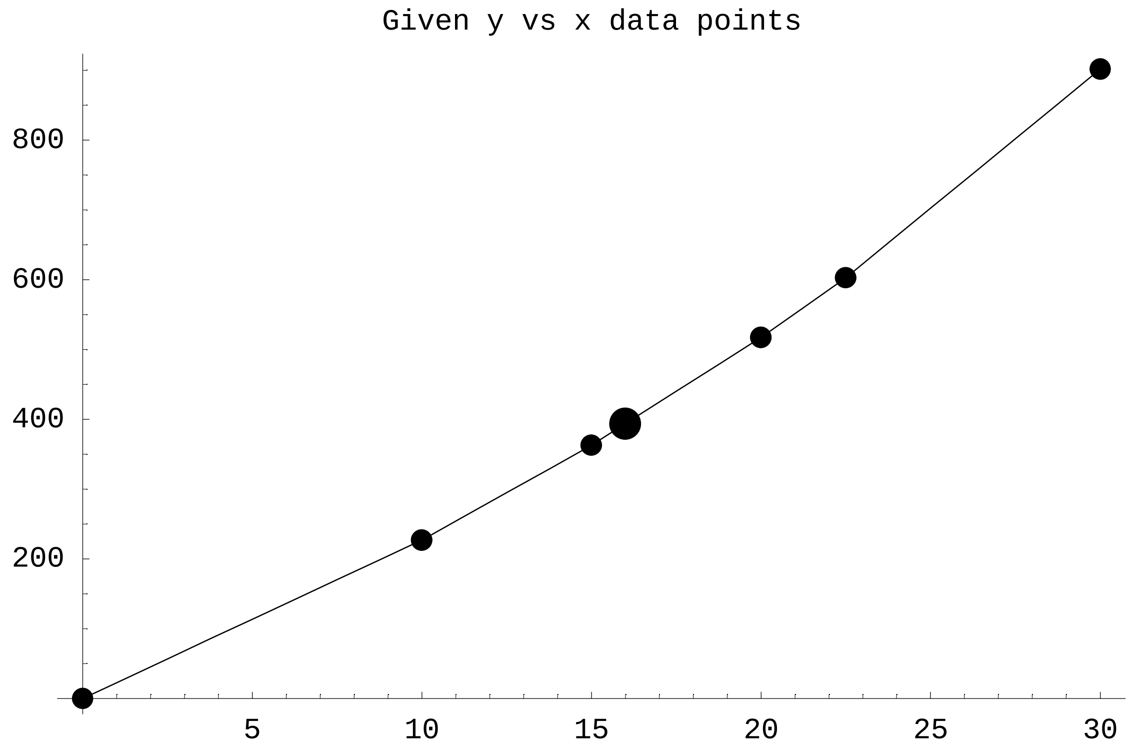
```
In[19]:= fline = Plot[flinear[z], {z, Min[x], Max[x]}];
```




```
In[20]:= desire =
           ListPlot[{{xdesired, flinear[xdesired]}}, PlotStyle → PointSize[0.03]];
```



In[21]:=
```
Show[data, fline, desire];
```



Quadratic spline interpolation

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

In[22]:=
```
A = Table[0, {i, 3*n-3}, {j, 3*n-3}];
```

In[23]:=
```
Do[Do[Do[A[[2*i-1+j+1, 3*i-3+1+k]] = (x[[i-1+j+1]])^(k), {k, 0, 2}],
  {j, 0, 1}], {i, 1, n-1}]
```

In[24]:=
```
Do[Do[Do[A[[2*(n-1)+i+1, 3*i-2+k+j*3+1]] = (-1)^j*(2*x[[i+1]])^k,
   {k, 0, 1}], {j, 0, 1}], {i, 1, n-2}]
```

In[25]:=
```
<< LinearAlgebra`MatrixManipulation`
```

In[26]:=
```
A[[1, 3]] = 1;
```

In[27]:= **A // MatrixForm**

Out[27]//MatrixForm=

$$\begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1.\times10^{-6} & 1.\times10^{-12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 10 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 10 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 15 & 225 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 15 & 225 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 20 & 400 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 20 & 400 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 22.5 & 506.25 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 22.5 & 506.25 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 30 & 900 \\
0 & 1 & 20 & 0 & -1 & -20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 30 & 0 & -1 & -30 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 40 & 0 & -1 & -40 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 45. & 0 & -1 & -45.
\end{pmatrix}$$

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

In[28]:= **Y = Table[0, {i, 3 * n - 3}];**

In[29]:= **Do[Do[Y[[2 * (i + 1) + j]] = y[[i + j + 1]], {j, 0, 1}], {i, 0, n - 2}]**

In[30]:= **Y // MatrixForm**

Out[30]//MatrixForm=

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

In[31]:= **Co = LinearSolve[A, Y];**

In[32]:=
```
Co // MatrixForm
```

Out[32]//MatrixForm=

$$\begin{pmatrix} -0.000022704 \\ 22.704 \\ 2.66454\times 10^{-16} \\ 88.8799 \\ 4.92801 \\ 0.8888 \\ -141.61 \\ 35.66 \\ -0.1356 \\ 554.55 \\ -33.956 \\ 1.6048 \\ -152.13 \\ 28.86 \\ 0.208889 \end{pmatrix}$$

In[33]:=
```
fquadratic[z_] := If[z ≤ x[[2]],
  Co[[1]] + Co[[2]] * z + Co[[3]] * z^2, Do[If[z ≤ x[[i + 2]] && z > x[[i + 1]],
    d = 0; Do[d = d + Co[[3 * i + j + 1]] * z^j, {j, 0, 2}]], {i, 1, n - 2}]; d]
```

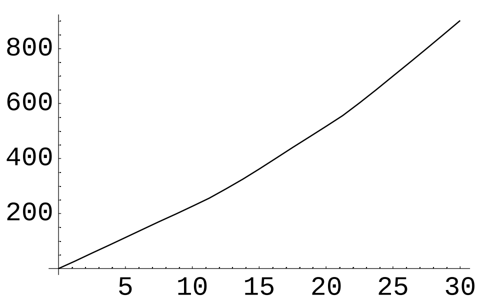
Value of function at desired value

In[34]:=
```
fquadratic[xdesired]
```

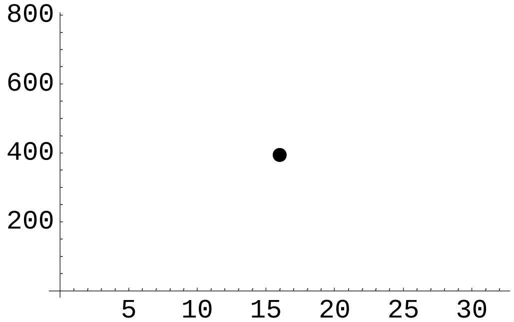
Out[34]= 394.236

In[35]:=
```
fprev = %;
```

In[36]:=
```
fline = Plot[fquadratic[z], {z, Min[x], Max[x]}];
```



In[37]:=
```
desire =
  ListPlot[{{xdesired, fquadratic[xdesired]}}, PlotStyle → PointSize[0.03]];
```



In[38]:=
```
Show[data, fline, desire];
```