Topic : Direct Method
Simulation : Graphical Simulation of the Method
Language : Mathmatica 4.1
Authors : Nathan Collier, Autar Kaw
Date : 15 July 2002
Abstract : This simulation illustrates the direct method of interpolation. Given n data points of y versus x, you are then required to find the value of y at a particular value of x using first, second, and third order interpolation. So on has to first pick the needed data point, and then use those points to interpolate the data

In[890]:=
```
Clear[x, y, xData, yData]
```

■ INPUTS: Enter the Following

Array of x data

In[891]:=
```
x = {10, 0, 20, 15, 30, 22.5};
```

Array of y data

In[892]:=
```
y = {227.04, 0, 517.35, 362.78, 901.67, 602.97};
```

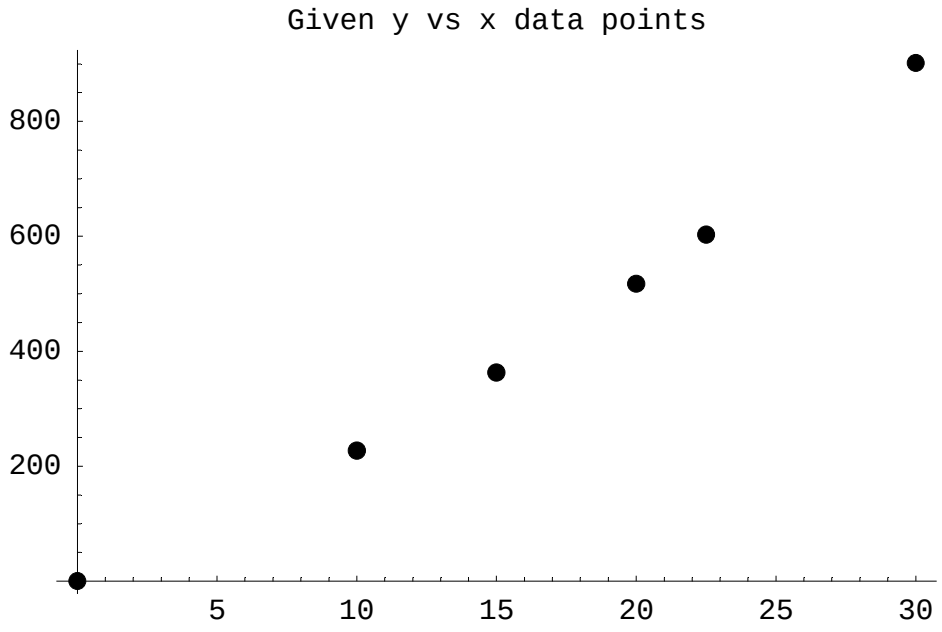
In[893]:=
```
xdesired := 16
```

In[894]:=
```
n := Abs[Dimensions[x]][[1]]
```

In[895]:=
```
xy = Table[0, {i, n}, {j, 2}];
```

In[896]:=
```
Do[xy[[i, 1]] = x[[i]]; xy[[i, 2]] = y[[i]], {i, 1, n}]
```

In[897]:=
```
data = ListPlot[xy, PlotStyle → PointSize[0.02],
   PlotLabel -> "Given y vs x data points", TextStyle → {FontSize → 11}];
```



■ SOLUTION

The following considers the x and y data and selects the two closest data points that bracket the desired value of x.

In[898]:=
```
comp := Abs[x - xdesired]
```

In[899]:=
```
c := Min[comp]
```

In[900]:=
```
Do[If[comp[[i]] == c, ci = i], {i, 1, n}]
```

In[901]:=
```
R = Table[0, {i, 1, n}];
```

In[902]:=
```
If[x[[ci]] < xdesired, q = 1;
 Do[If[x[[i]] > xdesired, R[[q]] = x[[i]]; q = q + 1], {i, 1, n}];
 RR = Table[RR[[i]] = R[[i]], {i, 1, q - 1}];
 b = Min[RR]; Do[If[x[[i]] == b, bi = i], {i, 1, n}]]
```

In[903]:=
```
If[x[[ci]] > xdesired, q = 1;
 Do[If[x[[i]] < xdesired, R[[q]] = x[[i]]; q = q + 1], {i, 1, n}];
 RR = Table[RR[[i]] = R[[i]], {i, 1, q - 1}];
 b = Max[RR]; Do[If[x[[i]] == b, bi = i], {i, 1, n}]]
```

In[904]:=
```
firsttwo := {ci, bi}
```

If more than two values are desired, the following selects the subsequent values and puts all the values into a matrix, maintaining the original data order.

In[905]:=
```
A = Table[0, {i, n}, {j, 3}];
```

```
In[906]:= Do[A[[i, 2]] = i; A[[i, 1]] = comp[[i]], {i, 1, n}]

In[907]:= A = Sort[A];

In[908]:= Do[A[[i, 3]] = i, {i, 1, n}]

In[909]:= T = A[[All, 1]];

In[910]:= Do[A[[i, 1]] = A[[i, 2]]; A[[i, 2]] = T[[i]], {i, 1, n}]

In[911]:= A = Sort[A];

In[912]:= d = A[[All, 3]];

In[913]:= If[d[[firsttwo[[2]]]] ≠ 2, temp = d[[firsttwo[[2]]]]; d[[firsttwo[[2]]]] = 1;
           Do[If[i ≠ firsttwo[[2]] && i ≠ firsttwo[[1]] && d[[i]] ≤ temp,
             d[[i]] = d[[i]] + 1]; d[[firsttwo[[1]]]] = 1, {i, 1, n}]]
```

Linear Interpolation

Pick two data points

```
In[914]:= datapoints = 2;

In[915]:= xData = Table[0, {i, 1, datapoints}];
          yData = Table[0, {i, 1, datapoints}];

In[917]:= p = 1; Do[If[d[[i]] ≤ datapoints,
            xData[[p]] = x[[i]]; yData[[p]] = y[[i]]; p = p + 1], {i, 1, n}]

In[918]:= xData // MatrixForm

Out[918]//MatrixForm=
(20
 15)

In[919]:= yData // MatrixForm

Out[919]//MatrixForm=
(517.35
 362.78)

In[920]:= M = {{1, xData[[1]]}, {1, xData[[2]]}} // MatrixForm

Out[920]//MatrixForm=
(1  20
 1  15)
```

In[921]:=
```
Co = LinearSolve[%, yData]
```
Out[921]=
```
{-100.93, 30.914}
```

In[922]:=
```
f[x_] := Co[[1]] + Co[[2]] * x
```

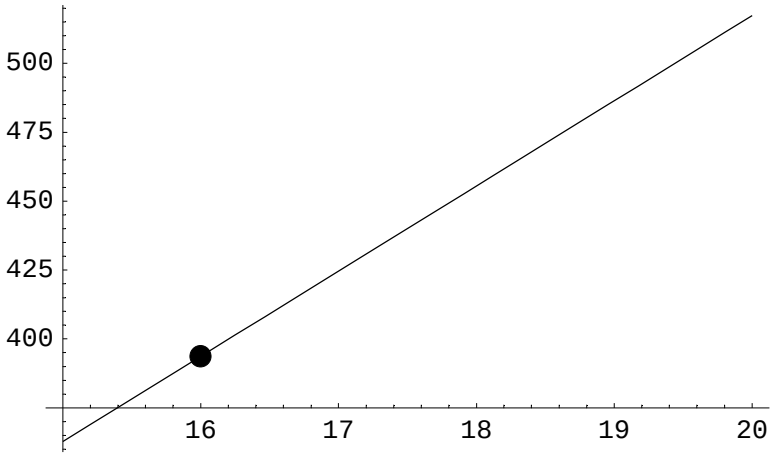
In[923]:=
```
f[xdesired]
```
Out[923]=
```
393.694
```

In[924]:=
```
fprev = %;
```

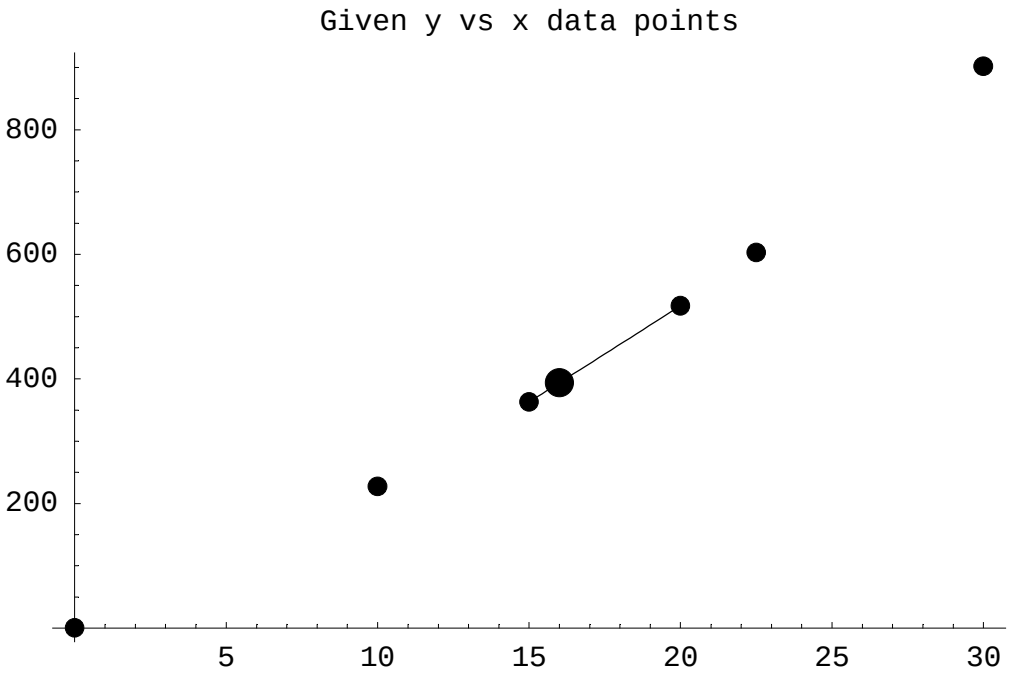
In[925]:=
```
lin = Plot[f[x], {x, Min[xData], Max[xData]}];
```

In[926]:=
```
desire = ListPlot[{{xdesired, f[xdesired]}}, PlotStyle → PointSize[0.03]];
```

In[927]:=
```
Show[desire, lin];
```



In[928]:=
```
Show[data, lin, desire, PlotRange → All];
```



Quadratic Interpolation

Pick two data points

In[929]:=
```
datapoints = 3;
```

In[930]:=
```
xData = Table[0, {i, 1, datapoints}];
yData = Table[0, {i, 1, datapoints}];
```

In[932]:=
```
p = 1; Do[If[d[[i]] ≤ datapoints,
  xData[[p]] = x[[i]]; yData[[p]] = y[[i]]; p = p + 1], {i, 1, n}]
```

In[933]:=
```
xData // MatrixForm
```

Out[933]//MatrixForm=

$$\begin{pmatrix} 10 \\ 20 \\ 15 \end{pmatrix}$$

In[934]:=
```
yData // MatrixForm
```

Out[934]//MatrixForm=

$$\begin{pmatrix} 227.04 \\ 517.35 \\ 362.78 \end{pmatrix}$$

In[935]:=
```
M = {{1, xData[[1]], xData[[1]]^2}, {1, xData[[2]], xData[[2]]^2},
   {1, xData[[3]], xData[[3]]^2}} // MatrixForm
```

Out[935]//MatrixForm=

$$\begin{pmatrix} 1 & 10 & 100 \\ 1 & 20 & 400 \\ 1 & 15 & 225 \end{pmatrix}$$

In[936]:=
```
Co = LinearSolve[%, yData]
```

Out[936]=
```
{12.05, 17.733, 0.3766}
```

In[937]:=
```
f[x_] := Co[[1]] + Co[[2]] * x + Co[[3]] * x^2
```

In[938]:=
```
f[xdesired]
```

Out[938]=
```
392.188
```

In[939]:=
```
fnew = %;
```

In[940]:=
```
∈a = Abs[(fnew - fprev) / fnew * 100]
```

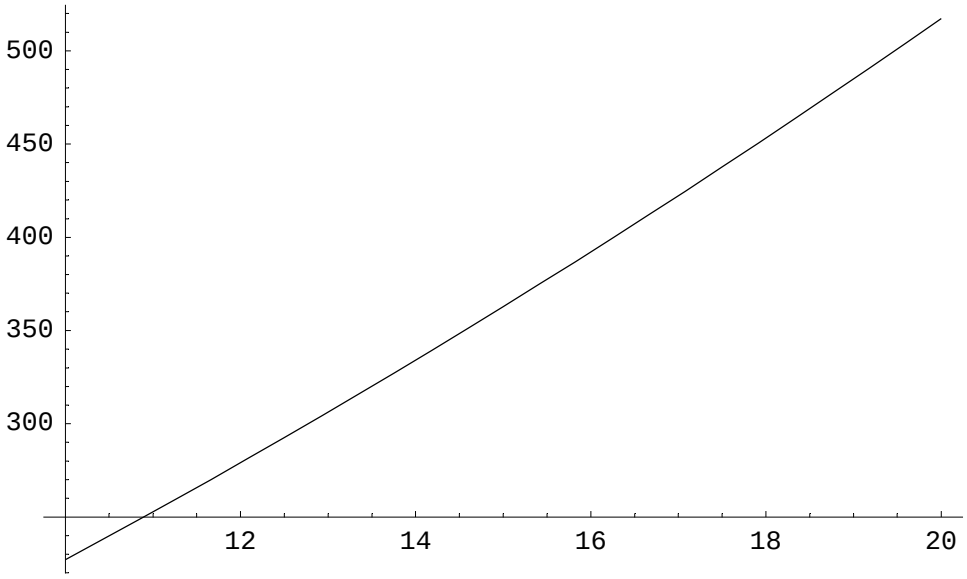
Out[940]=
```
0.384102
```

In[941]:=
```
sigdig = Floor[2 - Log[10, (∈a / 0.5)]]
```

Out[941]=
```
2
```

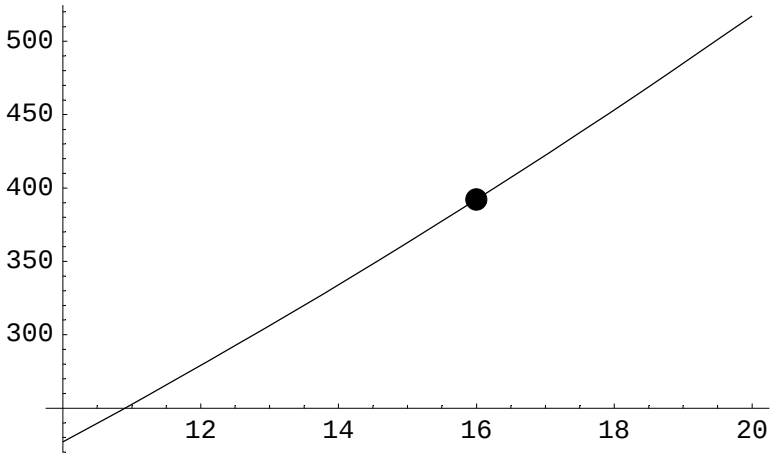
In[942]:=
```
fprev = %%%;
```

In[943]:=
```
lin = Plot[f[x], {x, Min[xData], Max[xData]}];
```

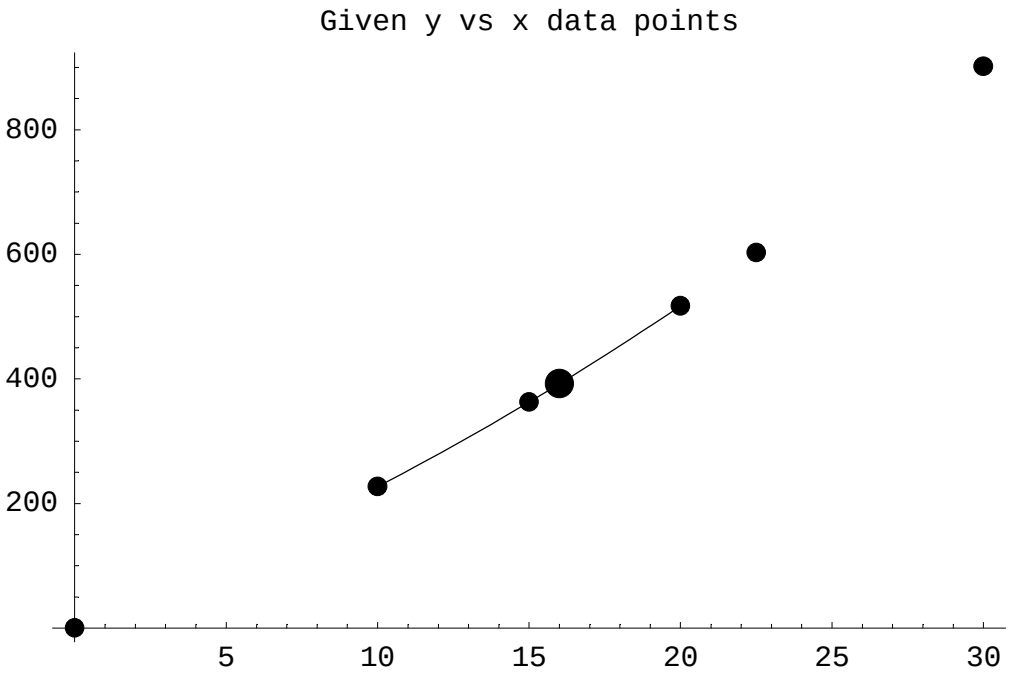


In[944]:=
```
desire = ListPlot[{{xdesired, f[xdesired]}}, PlotStyle → PointSize[0.03]];
```

In[945]:=
```
Show[desire, lin];
```



In[946]:=
```
Show[data, lin, desire, PlotRange → All];
```



Cubic Interpolation

Pick four data points

In[947]:=
```
datapoints = 4;
```

In[948]:=
```
xData = Table[0, {i, 1, datapoints}];
yData = Table[0, {i, 1, datapoints}];
```

In[950]:=
```
d
```

Out[950]=
```
{3, 6, 2, 1, 5, 4}
```

In[951]:=
```
p = 1; Do[If[d[[i]] ≤ datapoints,
  xData[[p]] = x[[i]]; yData[[p]] = y[[i]]; p = p + 1], {i, 1, n}]
```

In[952]:=
```
xData // MatrixForm
```

Out[952]//MatrixForm=

$$\begin{pmatrix} 10 \\ 20 \\ 15 \\ 22.5 \end{pmatrix}$$

In[953]:=
```
yData // MatrixForm
```

Out[953]//MatrixForm=

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

In[954]:=
```
M = {{1, xData[[1]], xData[[1]]^2, xData[[1]]^3},
   {1, xData[[2]], xData[[2]]^2, xData[[2]]^3},
   {1, xData[[3]], xData[[3]]^2, xData[[3]]^3},
   {1, xData[[4]], xData[[4]]^2, xData[[4]]^3}} // MatrixForm
```

Out[954]//MatrixForm=

$$\begin{pmatrix} 1 & 10 & 100 & 1000 \\ 1 & 20 & 400 & 8000 \\ 1 & 15 & 225 & 3375 \\ 1 & 22.5 & 506.25 & 11390.6 \end{pmatrix}$$

In[955]:=
```
Co = LinearSolve[%, yData]
```

Out[955]=
```
{-4.254, 21.2655, 0.13204, 0.00543467}
```

In[956]:=
```
f[x_] := Co[[1]] + Co[[2]] * x + Co[[3]] * x^2 + Co[[4]] * x^3
```

In[957]:=
```
f[xdesired]
```

Out[957]=
```
392.057
```

In[958]:=
```
fnew = %;
```

In[959]:=
```
ϵa = Abs[(fnew - fprev) / fnew * 100]
```

Out[959]=
```
0.0332686
```

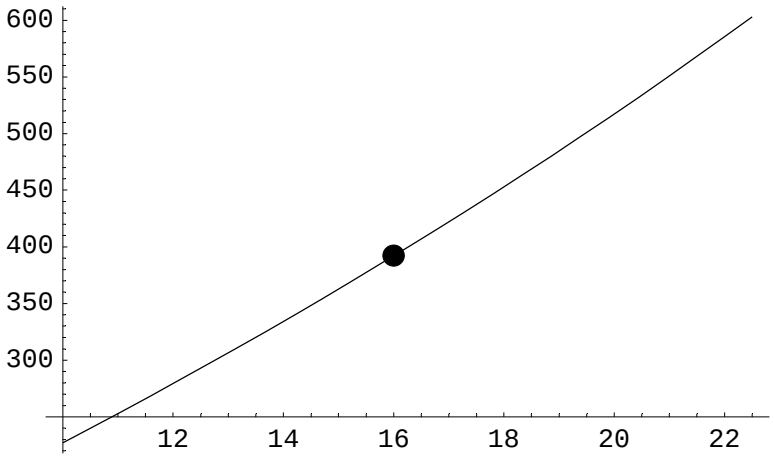
In[960]:=
```
sigdig = Floor[2 - Log[10, (ϵa / 0.5)]]
```

Out[960]=
```
3
```

In[961]:=
```
lin = Plot[f[x], {x, Min[xData], Max[xData]}];
```

In[962]:=
```
desire = ListPlot[{{xdesired, f[xdesired]}}, PlotStyle → PointSize[0.03]];
```

In[963]:= **Show[desire, lin];**



In[964]:= **Show[data, lin, desire, PlotRange → All];**