Topic : Newton Divided Difference
Simulation : Graphical Simulation of the Method
Language : Mathmatica 4.1
Authors : Nathan Collier, Autar Kaw
Date : 15 July 2002
Abstract : This simulation illustrates the Newton's Divided Difference 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

■ INPUTS: Enter the Following

Array of x data

```
In[1054]:= x = {10, 0, 20, 15, 30, 22.5};
```

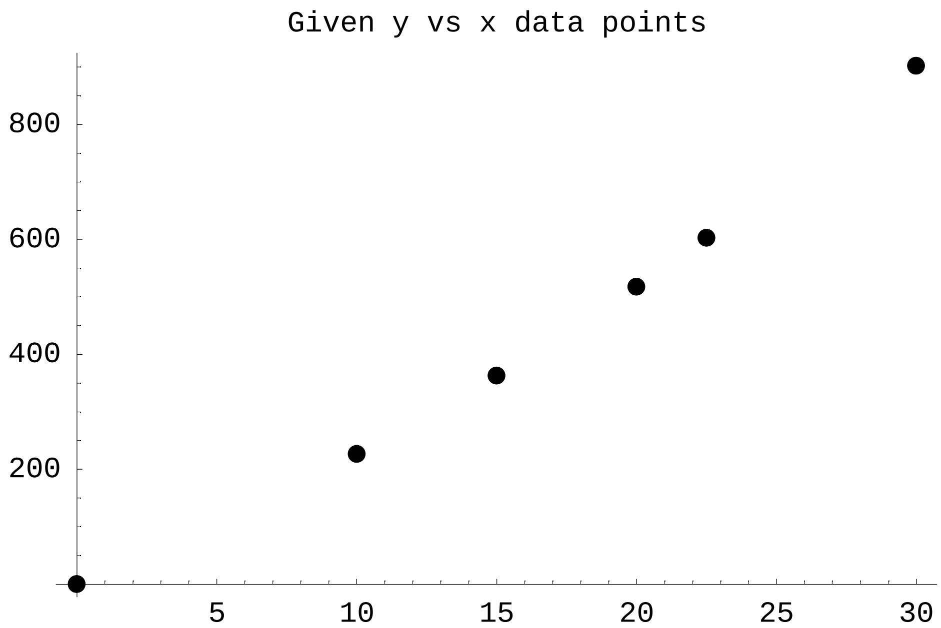
Array of y data

```
In[1055]:= y = {227.04, 0, 517.35, 362.78, 901.67, 602.97};
```

Value of x at which y is desired

```
In[1056]:= xdesired := 16

In[1057]:= nn := Abs[Dimensions[x]]

In[1058]:= n := nn[[1]]

In[1059]:= xy = Table[0, {i, n}, {j, 2}];

In[1060]:= Do[xy[[i, 1]] = x[[i]]; xy[[i, 2]] = y[[i]], {i, 1, n}]
```

In[1061]:=
```
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[1062]:=
```
comp := Abs[x - xdesired]
```

In[1063]:=
```
c := Min[comp]
```

In[1064]:=
```
Do[If[comp[[i]] == c, ci = i], {i, 1, n}]
```

In[1065]:=
```
R = Table[0, {i, 1, n}];
```

In[1066]:=
```
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[1067]:=
```
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[1068]:=
```
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[1069]:=
```
A = Table[0, {i, n}, {j, 3}];
```

In[1070]:=
```
Do[A[[i, 2]] = i; A[[i, 1]] = comp[[i]], {i, 1, n}]
```

In[1071]:=
```
A = Sort[A];
```

In[1072]:=
```
Do[A[[i, 3]] = i, {i, 1, n}]
```

In[1073]:=
```
T = A[[All, 1]];
```

In[1074]:=
```
Do[A[[i, 1]] = A[[i, 2]]; A[[i, 2]] = T[[i]], {i, 1, n}]
```

In[1075]:=
```
A = Sort[A];
```

In[1076]:=
```
d = A[[All, 3]];
```

In[1077]:=
```
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[1078]:=
```
datapoints = 2;
```

In[1079]:=
```
xData = Table[0, {i, 1, datapoints}];
yData = Table[0, {i, 1, datapoints}];
```

In[1081]:=
```
p = 1; Do[If[d[[i]] ≤ datapoints,
   xData[[p]] = x[[i]]; yData[[p]] = y[[i]]; p = p + 1], {i, 1, n}]
```

In[1082]:=
```
xData // MatrixForm
```

Out[1082]//MatrixForm=

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

In[1083]:=
```
yData // MatrixForm
```

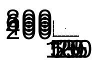
Out[1083]//MatrixForm=

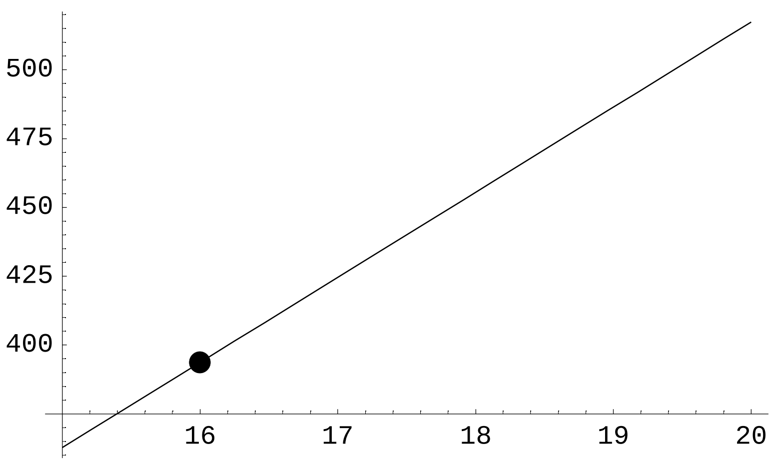
$$\begin{pmatrix} 517.35 \\ 362.78 \end{pmatrix}$$

In[1084]:=
```
b0 = yData[[1]]
```

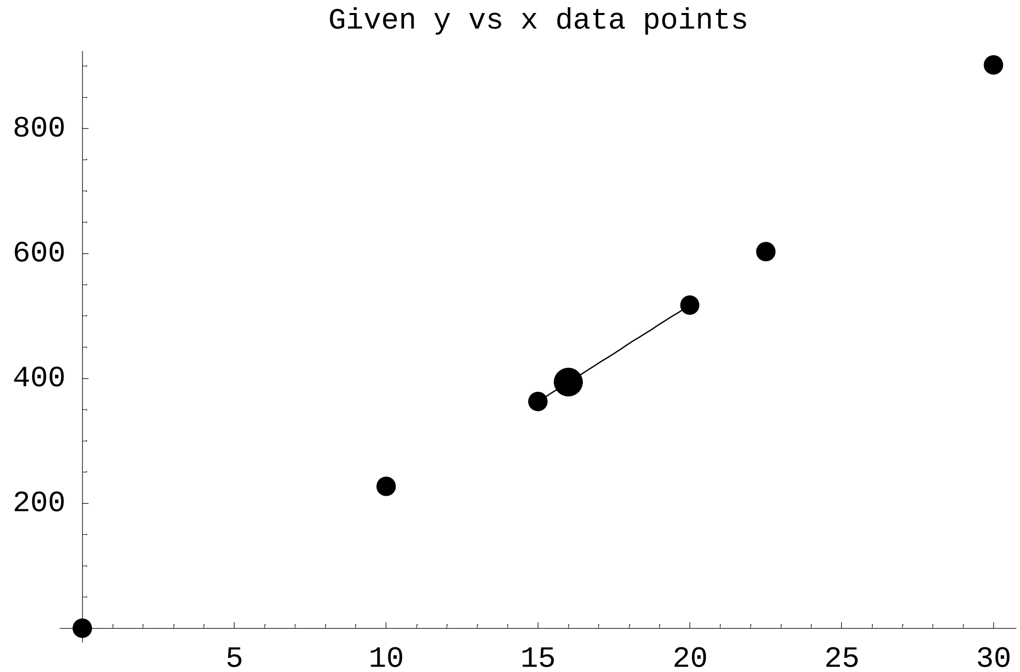
Out[1084]= 517.35

In[1085]:=
```
b1 = ((yData[[2]] - yData[[1]]) / (xData[[2]] - xData[[1]]))
```
Out[1085]=
```
30.914
```
In[1086]:=
```
f1[x_] := b0 + b1 * (x - xData[[1]])
```
In[1087]:=
```
f1[xdesired]
```
Out[1087]=
```
393.694
```
In[1088]:=
```
fprev = %;
```
In[1089]:=
```
lin = Plot[f1[x], {x, Min[xData], Max[xData]}];
```


In[1090]:=
```
desire = ListPlot[{{xdesired, f1[xdesired]}}, PlotStyle → PointSize[0.03]];
```


In[1091]:=
```
Show[desire, lin];
```


In[1092]:=
```
Show[data, lin, desire, PlotRange → All];
```



Quadratic Interpolation

Pick two data points

In[1093]:=
```
datapoints = 3;
```

In[1094]:=
```
xData = Table[0, {i, 1, datapoints}];
yData = Table[0, {i, 1, datapoints}];
```

In[1096]:=
```
p = 1; Do[If[d[[i]] ≤ datapoints,
  xData[[p]] = x[[i]]; yData[[p]] = y[[i]]; p = p + 1], {i, 1, n}]
```

In[1097]:=
```
xData // MatrixForm
```

Out[1097]//MatrixForm=

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

In[1098]:=
```
yData // MatrixForm
```

Out[1098]//MatrixForm=

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

In[1099]:=
```
b0 = yData[[1]]
```

Out[1099]= 227.04

In[1100]:=
```
b1 = ((yData[[2]] - yData[[1]]) / (xData[[2]] - xData[[1]]))
```

Out[1100]= 29.031

In[1101]:=
```
b2 = (((yData[[3]] - yData[[2]]) / (xData[[3]] - xData[[2]])) -
      ((yData[[2]] - yData[[1]]) / (xData[[2]] - xData[[1]]))) /
   (xData[[3]] - xData[[1]])
```

Out[1101]= 0.3766

In[1102]:=
```
f2[x_] := b0 + b1 * (x - xData[[1]]) + b2 * (x - xData[[1]]) * (x - xData[[2]])
```

In[1103]:=
```
f2[xdesired]
```

Out[1103]= 392.188

In[1104]:=
```
fnew = %;
```

In[1105]:=
```
∈a = Abs[(fnew - fprev) / fnew * 100]
```

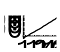
Out[1105]= 0.384102

In[1106]:=
```
sigdig = Floor[2 - Log[10, (∈a / 0.5)]]
```

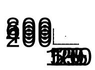
Out[1106]= 2

In[1107]:=
```
fprev = %%%;
```

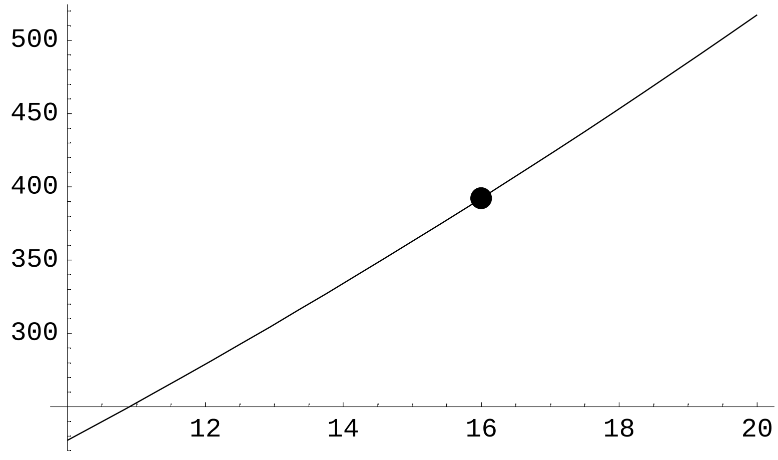
In[1108]:=
```
lin = Plot[f2[x], {x, Min[xData], Max[xData]}];
```



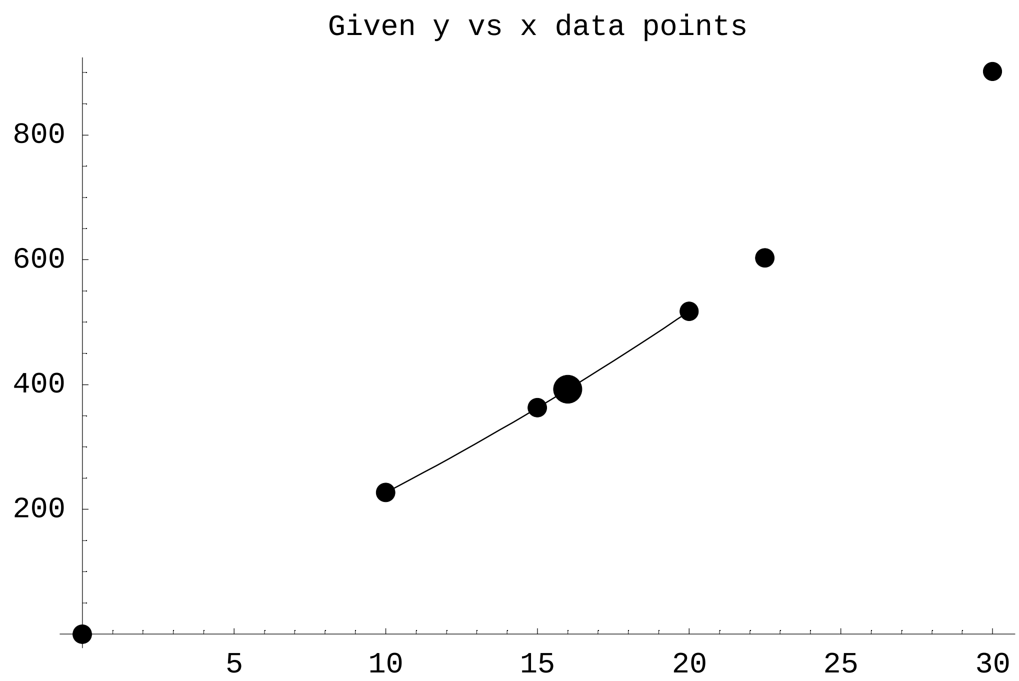
In[1109]:=
```
desire = ListPlot[{{xdesired, f2[xdesired]}}, PlotStyle → PointSize[0.03]];
```



In[1110]:= **Show[desire, lin];**



In[1111]:= **Show[data, lin, desire, PlotRange → All];**



Cubic Interpolation

Pick four data points

In[1112]:= **datapoints = 4;**

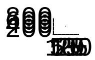
```
In[1113]:= xData = Table[0, {i, 1, datapoints}];
           yData = Table[0, {i, 1, datapoints}];

In[1115]:= p = 1; Do[If[d[[i]] ≤ datapoints,
             xData[[p]] = x[[i]]; yData[[p]] = y[[i]]; p = p + 1], {i, 1, n}]

In[1116]:= xData // MatrixForm

Out[1116]//MatrixForm=
  10
  20
  15
  22.5

In[1117]:= yData // MatrixForm

Out[1117]//MatrixForm=
  227.04
  517.35
  362.78
  602.97

In[1118]:= b0 = yData[[1]]

Out[1118]= 227.04

In[1119]:= b1 = ((yData[[2]] - yData[[1]]) / (xData[[2]] - xData[[1]]))

Out[1119]= 29.031

In[1120]:= b2 = (((yData[[3]] - yData[[2]]) / (xData[[3]] - xData[[2]])) -
               ((yData[[2]] - yData[[1]]) / (xData[[2]] - xData[[1]]))) /
             (xData[[3]] - xData[[1]])

Out[1120]= 0.3766

In[1121]:= b3 = ((((yData[[4]] - yData[[3]]) / (xData[[4]] - xData[[3]])) -
                  ((yData[[3]] - yData[[2]]) / (xData[[3]] - xData[[2]]))) /
                (xData[[4]] - xData[[2]]) - b2) / (xData[[4]] - xData[[1]])

Out[1121]= 0.00543467

In[1122]:= f3[x_] := b0 + b1 * (x - xData[[1]]) + b2 * (x - xData[[1]]) * (x - xData[[2]]) +
             b3 * (x - xData[[1]]) * (x - xData[[2]]) * (x - xData[[3]])

In[1123]:= f3[xdesired]

Out[1123]= 392.057

In[1124]:= fnew = %;

In[1125]:= ϵa = Abs[(fnew - fprev) / fnew * 100]

Out[1125]= 0.0332686
```

In[1126]:=
```
sigdig = Floor[2 - Log[10, (∈a / 0.5)]]
```

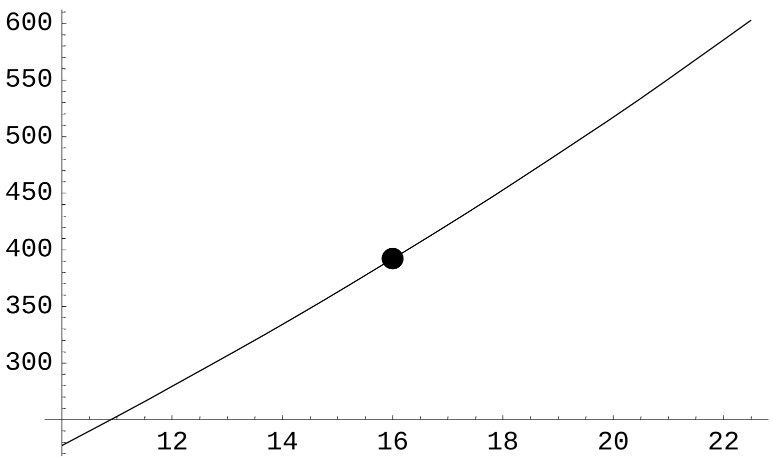
Out[1126]= 3

In[1127]:=
```
lin = Plot[f3[x], {x, Min[xData], Max[xData]}];
```

In[1128]:=
```
desire = ListPlot[{{xdesired, f3[xdesired]}}, PlotStyle → PointSize[0.03]];
```



In[1129]:=
```
Show[desire, lin];
```



In[1130]:=
```
Show[data, lin, desire, PlotRange → All];
```