

Topic : Additional Interpolation Topics
Simulation : Extrapolation is a bad idea
Language : Mathematica 4.1
Authors : Nathan Collier, Autar Kaw
Date : 12 December 2002

Abstract : During the technology boom of the late 1990's, the NASDAQ Composite Index that tracks mainly technology stocks had a phenomenal increase in its value. For example, at the end of year 1998, the NASDAQ index was at an all time high of 1570.35. It increase to 2192.69 by the end of 1999 and further to 4069.31 by the end of 2000. Many people who had never ventured into stocks started investing in the technology stocks and mutual funds. Dreams of doubling their money every year were not considered unrealistic. So given below is the data of the NASDAQ index from 1994 to 1999. If you were extrapolating the data, what would you estimate the NASDAQ index to be at the end of 2000 and 2001? How different did it turn out to be?

```
In[129]:= ClearAll;
```

■ INPUTS: Enter the following

Range of years actually between 1994 (Year 1) and 1999 (Year 6). Numbers start from 1 to avoid round-off errors and near singularity in matrix calculations. End of Year NASDAQ Composite Data taken from www.bigcharts.com

```
In[130]:= EndofYear := {1, 2, 3, 4, 5, 6}
```

```
In[131]:= NASDAQ := {751.96, 1052.13, 1291.03, 1570.35, 2192.69, 4069}
```

Extrapolated Values

```
In[132]:= e1 := 7;  
          e2 := 8;
```

In the above number scheme, years 7 and 8 correspond to years 2000 and 2001 respectively

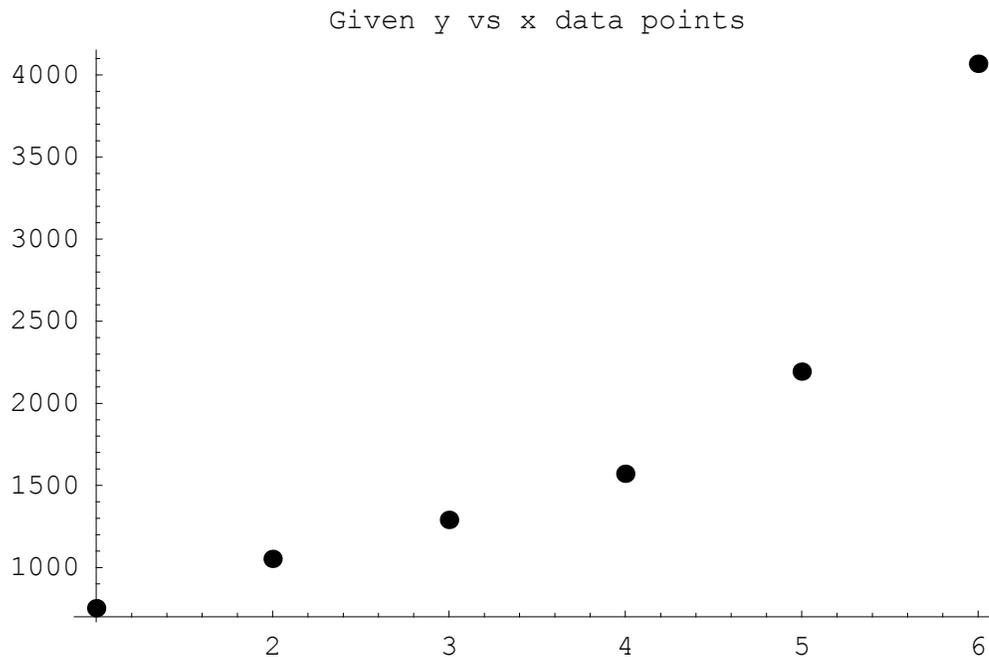
■ SOLUTION

```
In[134]:= n := Abs[Dimensions[EndofYear]][[1]]
```

```
In[135]:= xy = Table[0, {i, n}, {j, 2}];
```

```
In[136]:= Do[xy[[i, 1]] = EndofYear[[i]]; xy[[i, 2]] = NASDAQ[[i]], {i, 1, n}]
```

```
In[137]:= data = ListPlot[xy, PlotStyle -> PointSize[0.02],
  PlotLabel -> "Given y vs x data points",
  TextStyle -> {FontSize -> 11}, AxesOrigin -> {1, 700}];
```



```
In[138]:= exactual = Table[0, {2}, {2}];
```

```
In[139]:= exactual[[1, 1]] = e1;
exactual[[2, 1]] = e2;
exactual[[1, 2]] = 2470.52;
exactual[[2, 2]] = 1950.4;
```

```
In[143]:= extrapa =
  ListPlot[exactual, PlotStyle -> PointSize[0.02], DisplayFunction -> Identity];
```

Extrapolating using polynomial interpolation

```
In[144]:= M = Table[EndofYear[[i + 1]]^j, {i, 0, n - 1}, {j, 0, n - 1}];
M[[ (n + 1) / 2, 1]] = 1;
A = LinearSolve[M, NASDAQ];

Set::pspec :
  Part specification  $\frac{7}{2}$  is neither an integer nor a list of integers.

In[147]:= f1[z_] := Sum[A[[i]] * z^(i - 1), {i, 1, n}]

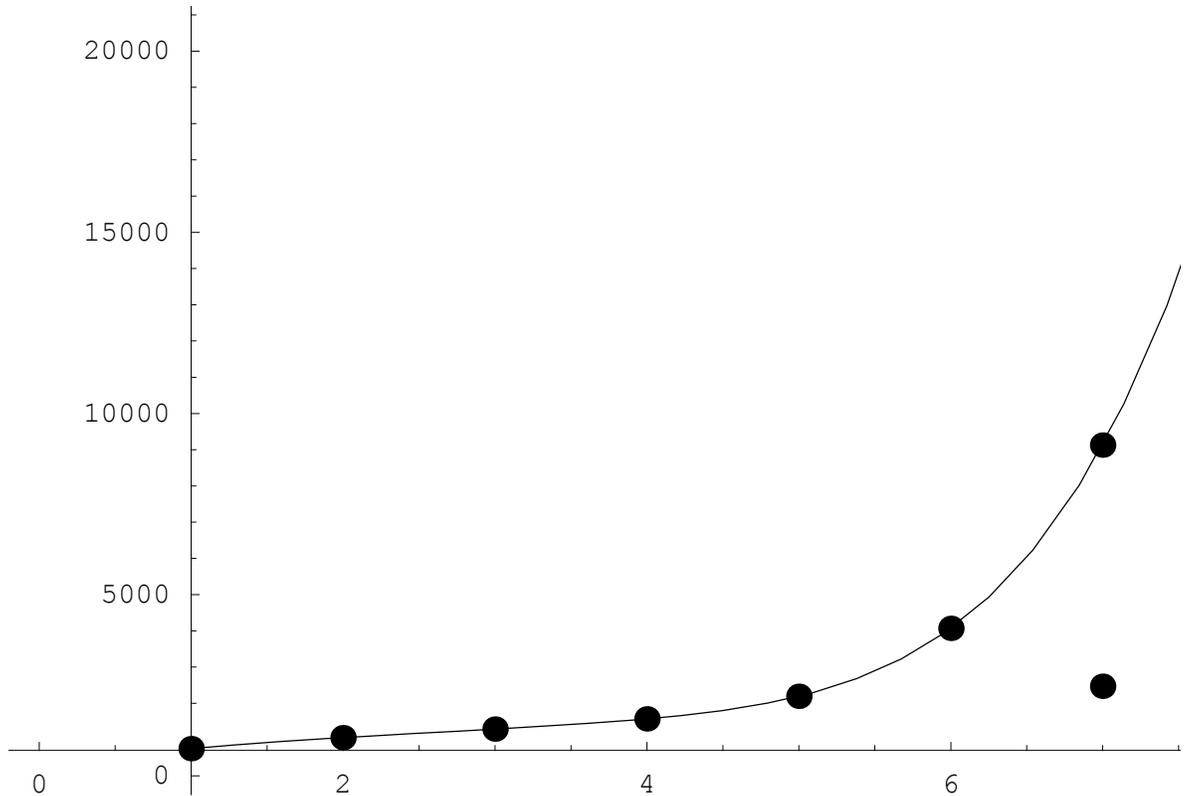
In[148]:= plot1 = Plot[f1[z], {z, 1, 8}, DisplayFunction -> Identity];

In[149]:= ex1 = Table[0, {2}, {2}];

In[150]:= ex1[[1, 1]] = e1;
ex1[[2, 1]] = e2;
ex1[[1, 2]] = f1[e1];
ex1[[2, 2]] = f1[e2];

In[154]:= extrap1 =
  ListPlot[ex1, PlotStyle -> PointSize[0.02], DisplayFunction -> Identity];
```

```
In[155]:= Show[data, extrap1, extrapa, plot1, PlotLabel ->
  "Data from 1994 to 1999 extrapolated to yield data for 2000 and 2001"];
  Data from 1994 to 1999 extrapolated to yield data for 2000 and 2001
```



Extrapolated Value at 2000 and 2001 respectively

```
In[156]:= f1[e1]
Out[156]= 9126.02
```

```
In[157]:= f1[e2]
Out[157]= 20713.7
```

Actual Values at 2000 and 2001 respectively

```
In[158]:= exactual[[1, 2]]
Out[158]= 2470.52
```

```
In[159]:= exactual[[2, 2]]
Out[159]= 1950.4
```

Relative True Error for 2000 and 2001 respectively

```
In[160]:= err1 = Abs[(f1[e1] - exactual[[1, 2]]) / exactual[[1, 2]] * 100]
Out[160]= 269.397
```

```
In[161]:= err2 = Abs[(f1[e2] - exactual[[2, 2]]) / exactual[[2, 2]] * 100]
```

```
Out[161]= 962.024
```

Extrapolating using cubic spline interpolation

Mathematica's Cubic Spline function works in a parametric fashion. The user specifies the relative position in the data set, and the function interpolates based on that. For more information, see the help file. For this reason, we must write our own cubic spline process. The following was adapted from an algorithm found in Burden's *Numerical Analysis*, 4th ed. page 131.

```

In[162]:= CubicSplineCoeff[X0_, Y0_] := Module[{X = X0, Y = Y0},
  GettingCoeff := Module[{},
    n := Length[X] - 1;
    a = b = d = Alpha = h = Table[0, {n}];
    l = u = z = c = Table[0, {n + 1}];
    Do[a[[i]] = Y[[i]], {i, 1, n}];
    Do[h[[i]] = X[[i + 1]] - X[[i]], {i, 1, n}];
    Do[Alpha[[i]] = 3 * (Y[[i + 1]] * h[[i - 1]] - Y[[i]] * (X[[i + 1]] - X[[i - 1]])) +
      Y[[i - 1]] * h[[i]]) / (h[[i - 1]] * h[[i]]), {i, 2, n}];
    l[[1]] := 1;
    u[[1]] := 0;
    z[[1]] := 0;
    Do[
      l[[i]] = 2 * (X[[i + 1]] - X[[i - 1]]) - h[[i - 1]] * u[[i - 1]];
      u[[i]] = h[[i]] / l[[i]];
      z[[i]] = (Alpha[[i]] - h[[i - 1]] * z[[i - 1]]) / l[[i]],
      {i, 2, n}];
    l[[n + 1]] := 1;
    z[[n + 1]] := 0;
    c[[n + 1]] := 0;
    j = n + 1; While[(j = j - 1) >= 1,
      c[[j]] = z[[j]] - u[[j]] * c[[j + 1]];
      b[[j]] = (Y[[j + 1]] - Y[[j]]) / h[[j]] - h[[j]] * (c[[j + 1]] + 2 * c[[j]]) / 3;
      d[[j]] = (c[[j + 1]] - c[[j]]) / (3 * h[[j]]);];];
  SplineFunction[t_] := Module[{i},
    For[i = 1, i <= n, i++,
      If[X[[i]] <= t && t < X[[i + 1]], k = i];];
    If[t < X[[1]], k = 1];
    If[t > X[[n + 1]], k = n];
    q = t - X[[k]];
    Return[((d[[k]] * q + c[[k]]) * q + b[[k]]) * q + a[[k]]];];
  GettingCoeff;];

```

```

In[163]:= CubicSplineCoeff[EndofYear, NASDAQ]

```

```

In[164]:= plot2 = Plot[SplineFunction[z], {z, 1, 8}, DisplayFunction -> Identity];

```

```

In[165]:= ex2 = Table[0, {2}, {2}];

```

```

In[166]:= ex2[[1, 1]] = e1;
ex2[[2, 1]] = e2;
ex2[[1, 2]] = SplineFunction[e1];
ex2[[2, 2]] = SplineFunction[e2];

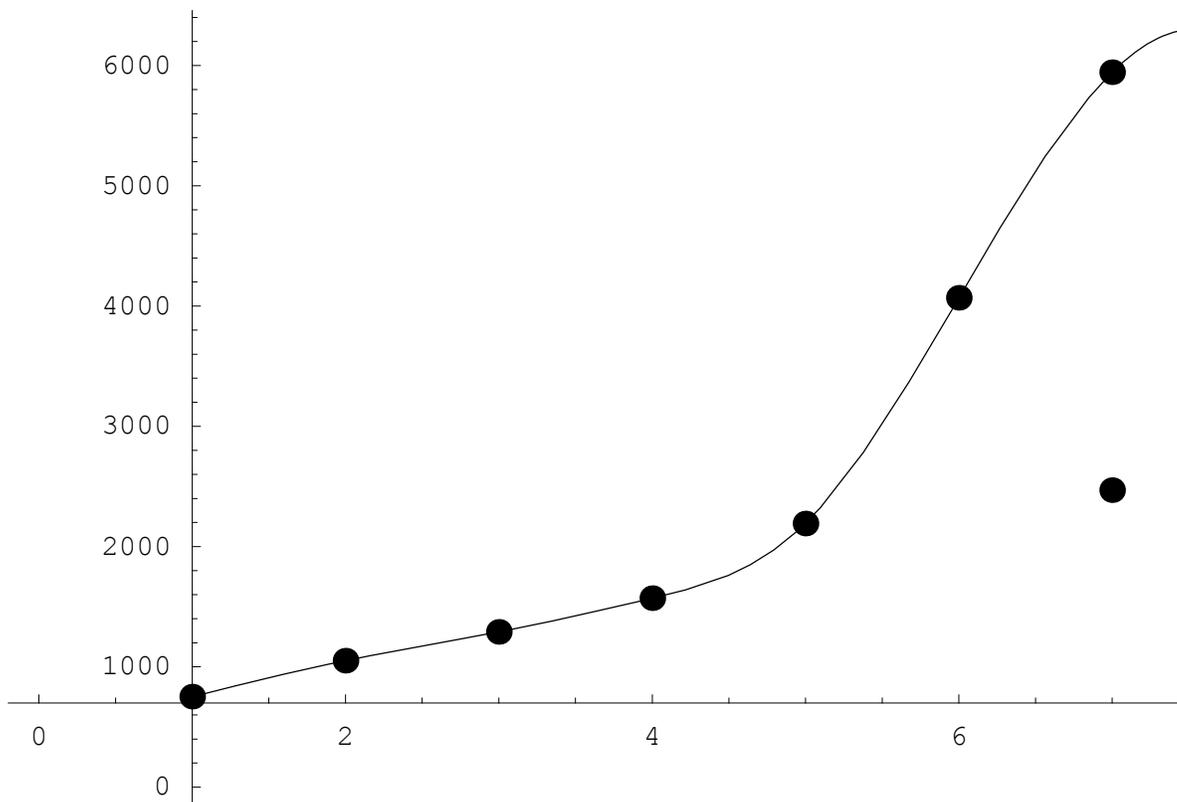
```

```

In[170]:= extrap2 =
  ListPlot[ex2, PlotStyle -> PointSize[0.02], DisplayFunction -> Identity];

```

```
In[171]:= Show[data, extrap2, extrapa, plot2, PlotLabel ->
  "Data from 1994 to 1999 extrapolated to yield data for 2000 and 2001"];
  Data from 1994 to 1999 extrapolated to yield data for 2000 and 2001
```



Extrapolated Value at 2000 and 2001 respectively

```
In[172]:= SplineFunction[e1]
```

```
Out[172]= 5945.31
```

```
In[173]:= SplineFunction[e2]
```

```
Out[173]= 5946.98
```

Actual Values at 2000 and 2001 respectively

```
In[174]:= exactual[[1, 2]]
```

```
Out[174]= 2470.52
```

```
In[175]:= exactual[[2, 2]]
```

```
Out[175]= 1950.4
```

Relative True Error for 2000 and 2001 respectively

```
In[176]:= err1 = Abs[(SplineFunction[e1] - exactual[[1, 2]]) / exactual[[1, 2]] * 100]
```

```
Out[176]= 140.65
```

```
In[177]:= err2 = Abs[(SplineFunction[e2] - exactual[[2, 2]]) / exactual[[2, 2]] * 100]
```

```
Out[177]= 204.911
```

Compare the two curves

```
In[179]:= Show[data, extrap1, extrap2, extrapa, plot1, plot2];
```

Given y vs x data points

