

Topic : Additional Interpolation Topics  
Simulation : The Need for Spline Interpolation  
Language : Mathematica 4.1  
Authors : Nathan Collier,Autar Kaw  
Date : 12 December 2002

Abstract : This simulation shows the need for spline interpolation as opposed to using polynomial interpolation. The function chosen was first used by Runge in 1901 when he wanted to show that higher order interpolation is a bad idea. He took a simple looking function  $f[x]=1/(1+25*x^2)$  and chose equidistantly spaced points to interpolate the function

```
In[207]:= Clear[x, y, f, M, f1, A, xy]
```

#### ■ INPUTS: Enter the following

Enter the number of points chosen for interpolation in [-1,1]. For the number scheme to function properly, the number of points selected should be odd.

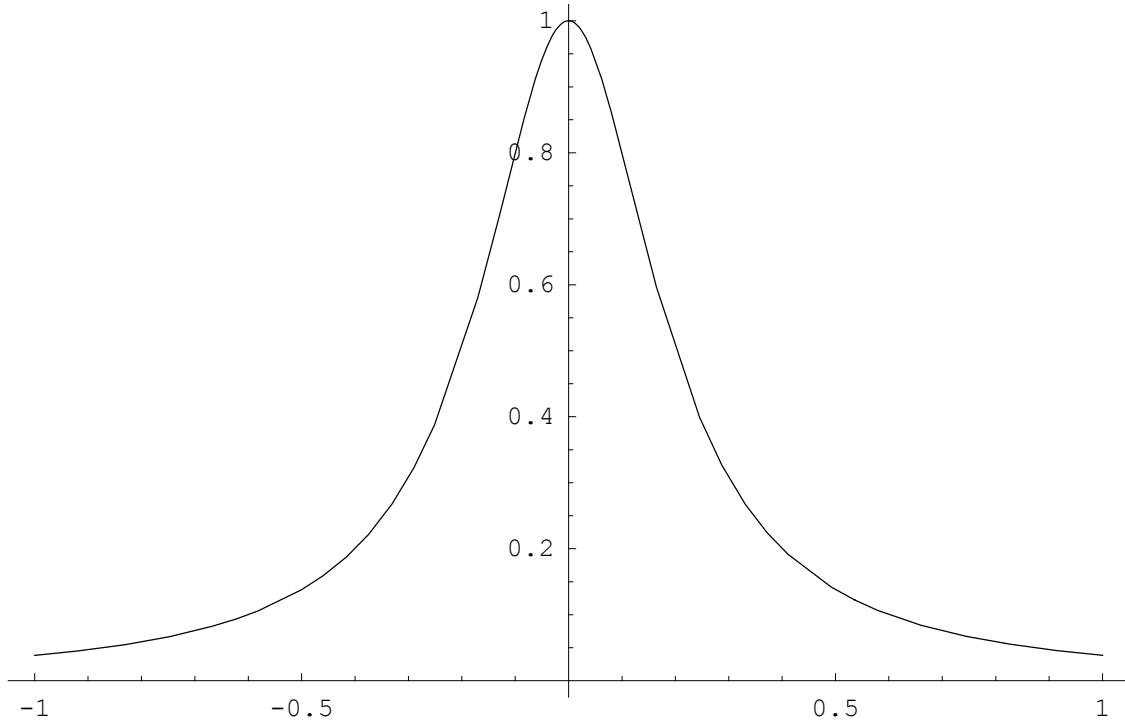
```
In[208]:= n := 9;
```

#### ■ SOLUTION

```
In[209]:= f[x_] := 1 / (1 + 25 * x^2)
```

```
In[210]:= Runge = Plot[f[x], {x, -1, 1}];
```

```
General::spell1 : Possible spelling error: new
symbol name "Runge" is similar to existing symbol "Range".
```



When  $n$  is given, this returns an array containing sequential values of  $x$ .

```
In[211]:= x := Table[2 / (n - 1) * i - 1, {i, 0, n - 1}]
```

```
In[212]:= y := f[x]
```

```
In[213]:= xy = Table[0, {i, 1, n}, {j, 1, 2}];
```

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

When  $x$  and  $y$  data and order is given, this constructs the matrix whose inverse is needed to find the coefficients of the polynomial which approximates the data.

```
In[215]:= M = Table[x[[i + 1]] ^ j, {i, 0, n - 1}, {j, 0, n - 1}];
```

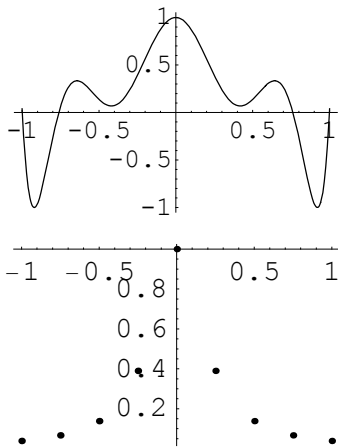
```
Power::indet : Indeterminate expression 00 encountered.
```

```
In[216]:= M[[ (n + 1) / 2, 1]] = 1;
```

```
In[217]:= A = LinearSolve[M, y];
```

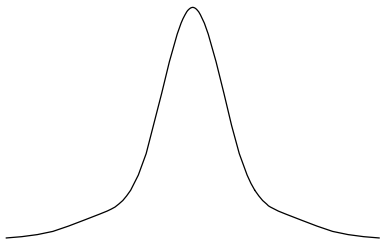
```
In[218]:= f1[z_] := Sum[A[[i]] * z ^ (i - 1), {i, 1, n}]
```

```
In[219]:= poly = Plot[f1[z], {z, -1, 1}];  
data = ListPlot[xy, PlotStyle -> PointSize[0.02],  
TextStyl e -> {FontSize -> 11}, PlotRange -> All];
```



```
In[221]:= << Graphics`Spline`
```

```
In[228]:= splin = Show[Graphics[{Spline[xy, Cubic]}], PlotRange -> All];
```



```
In[232]:= Show[splin, poly, data, Runge, Axes → True];
```

