Topic : Additional Interpolation Topics
Simulation : The Effect of Choice of Points on Interpolation
Language : Mathematica 4.1
Authors : Nathan Collier,Autar Kaw
Date : 26 November 2002
Abstract : In 1901, Carl Runge published his work on dangers of higher order interpolation. He took a simple looking function f(x)=1/(1+25x2) on the interval [-1,1]. He took points equidistantly spaced in [-1,1] and interpolated the points with polynomials. He found that as he took more points, the polynomials and the original curve differed considerably. However, if he took data points close to the ends of the interval [-1,1], the problem of large differences between interpolated and actual values was less pronounced. This simulation shows you this phenomena.

In[451]:=
```
Clear[x, y, f, M, f1, A, xy]
```

■ INPUTS: Enter the following

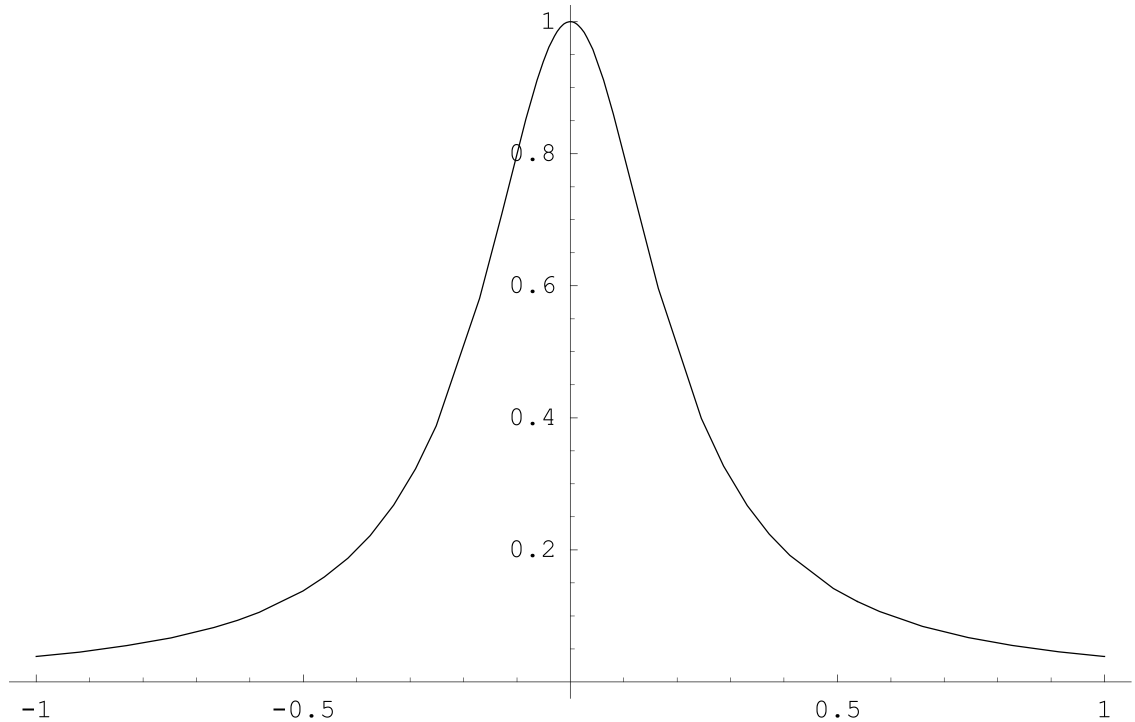
Enter the number of points chosen for interpolation in [-1,1]. For the number scheme to function properly, the number of points selected should be odd.

In[452]:=
```
n := 13;
```

■ SOLUTION

In[453]:=
```
f[x_] := 1 / (1 + 25 * x^2)
```

In[454]:=
```
Runge = Plot[f[x], {x, -1, 1}];
```



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

In[455]:=
```
x := Table[2 / (n - 1) * i - 1, {i, 0, n - 1}]
```

In[456]:=
```
y := f[x]
```

In[457]:=
```
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[459]:=
```
M = Table[x[[i + 1]] ^j, {i, 0, n - 1}, {j, 0, n - 1}];
```

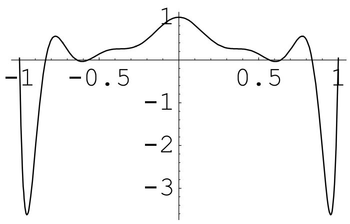
Power::indet : Indeterminate expression 0^0 encountered.

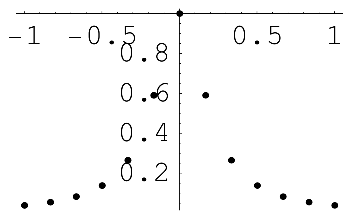
In[460]:=
```
M[[(n + 1) / 2, 1]] = 1;
```

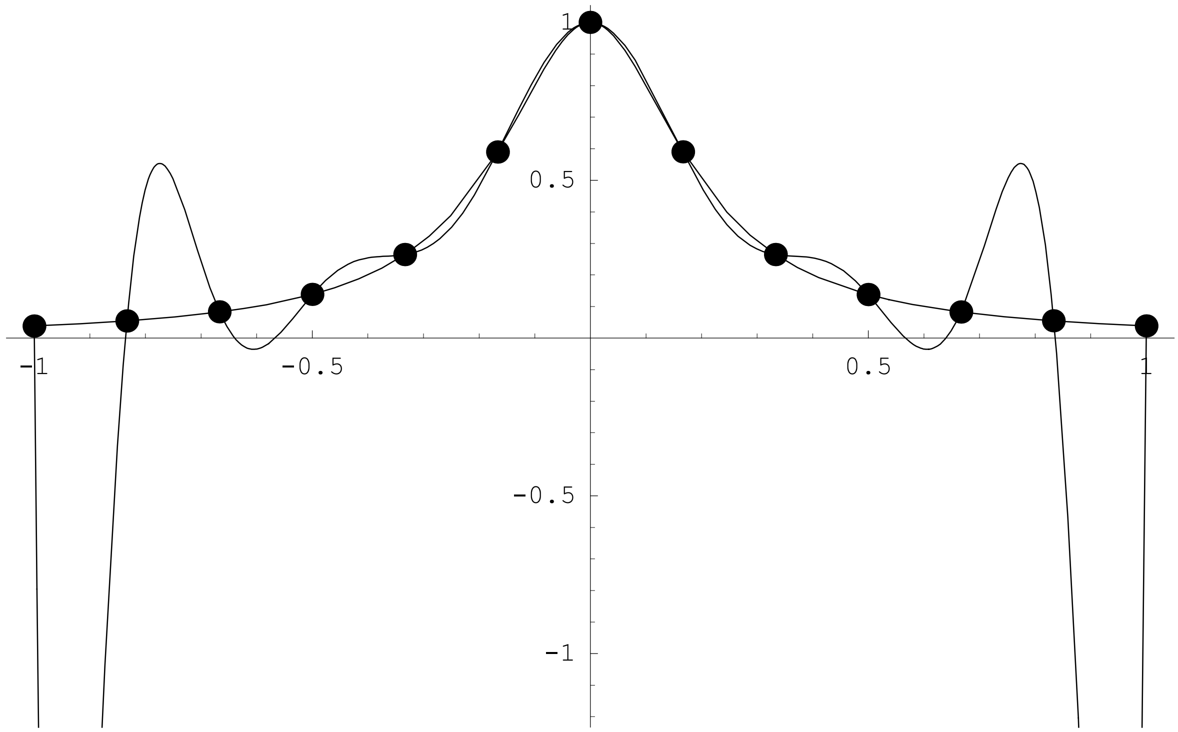
In[461]:=
```
A = LinearSolve[M, y];
```

In[462]:=
```
f1[z_] := Sum[A[[i]] * z^(i - 1), {i, 1, n}]
```

In[463]:=
```
even = Plot[f1[z], {z, -1, 1}];
data = ListPlot[xy, PlotStyle → PointSize[0.02],
   TextStyle → {FontSize → 11}, PlotRange → All];
Show[Runge, even, data];
```







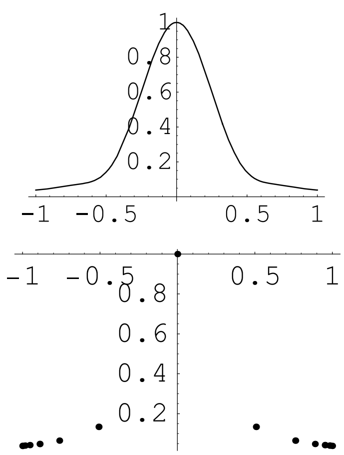
Fitted Polynomial with more points on the ends.

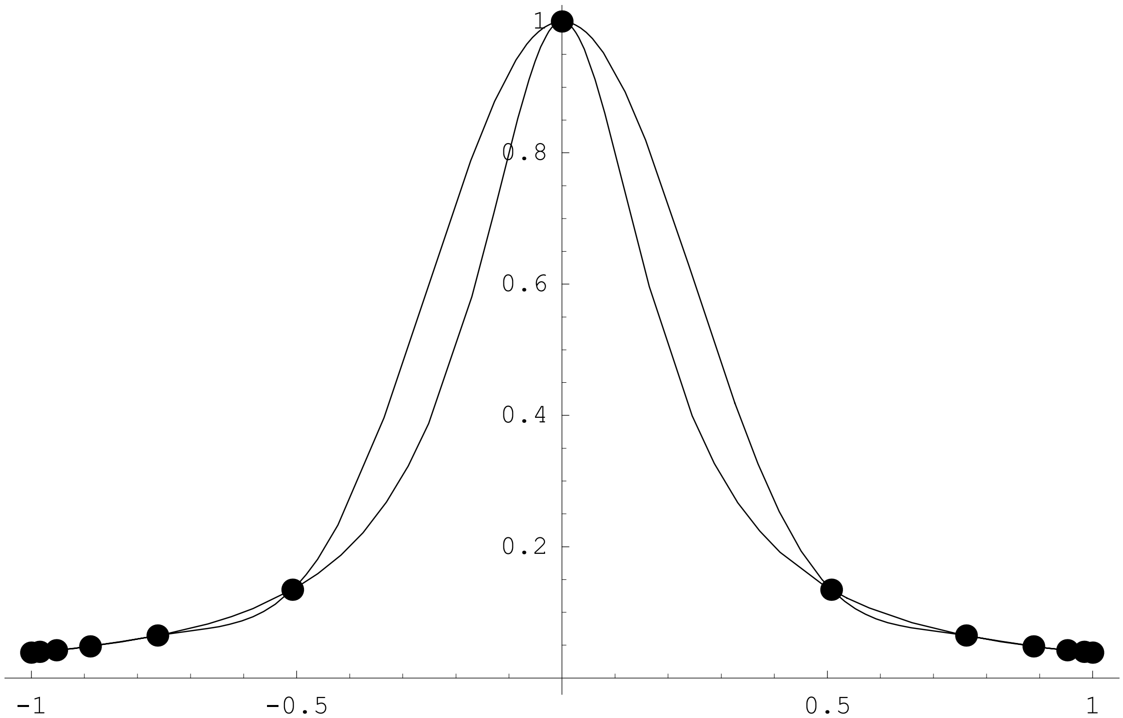
The idea of this is to place more points at the ends of the interval than in the middle. If the distance between the first and second points is x, then the distance between the second and third is 2*x. Each subsequent point is twice as far from the next point as the previous. After passing the middle, the function reverses.

```
In[466]:= xb = Table[0, {i, 1, n}];
          d = 0; Do[d = d + 2 ^ (i - 2), {i, 2, (n - 1) / 2 + 1}]
          l = 1 / d;
          xb[[1]] = -1;
          Do[xb[[i]] = xb[[i - 1]] + 2 ^ (i - 2) * l, {i, 2, (n - 1) / 2 + 1}]
          Do[xb[[i]] = xb[[i - 1]] + 2 ^ (n - i) * l, {i, (n - 1) / 2 + 2, n}]
```

```
In[472]:= yb = f[xb];
```

```
In[473]:= xyb = Table[0, {i, 1, n}, {j, 1, 2}];
          Do[xyb[[i, 1]] = xb[[i]]; xyb[[i, 2]] = yb[[i]], {i, 1, n}];

In[475]:= M = Table[xb[[i + 1]]^j, {i, 0, n - 1}, {j, 0, n - 1}];

          Power::indet : Indeterminate expression 0^0 encountered.

In[476]:= M[[(n + 1) / 2, 1]] = 1;

In[477]:= A = LinearSolve[M, yb];

In[478]:= f2[z_] := Sum[A[[i]] * z^(i - 1), {i, 1, n}]

In[479]:= even = Plot[f2[z], {z, -1, 1}];
          data = ListPlot[xyb, PlotStyle → PointSize[0.02],
             TextStyle → {FontSize → 11}, PlotRange → All];
          Show[Runge, even, data];
```




1
0.8
0.6
0.4
0.2
-1
-0.5
0.5
1