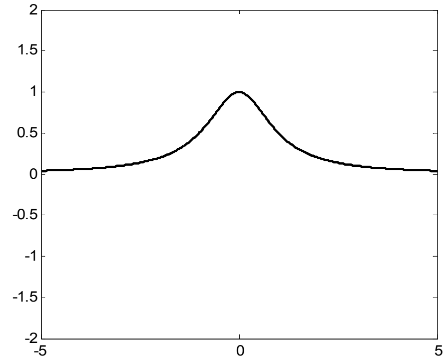
Section 5.5 Piecewise Interpolation

We have considered polynomial interpolation to sets of distinct data like $\mathbf{S} = \{(\mathbf{x_i}, \mathbf{y_i}) \,|\, \mathbf{i} = 0, 1, 2, \ldots, \mathbf{n}\}$.

The result was a **unique** polynomial $\mathbf{P_n(x)}$ of degree n or less.



Unfortunately for n large the resulting polynomial may contain undesired oscillations. A classic example is to generate polynomial interpolants of higher and higher degree to the data sets at equispaced points in the interval [-5, 5] determined by function

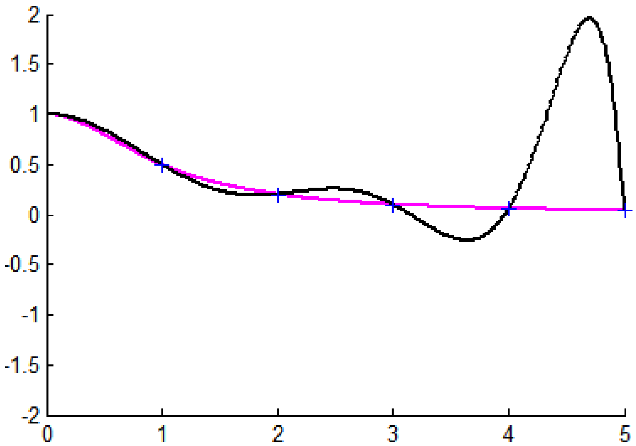
$$\mathbf{f(x)} = \frac{\mathbf{1}}{\mathbf{1 + x^2}}$$

To demonstrate this we have MATLAB routine **rungedem.** For successive experiments choose spacing h for data in [-5, 5] using

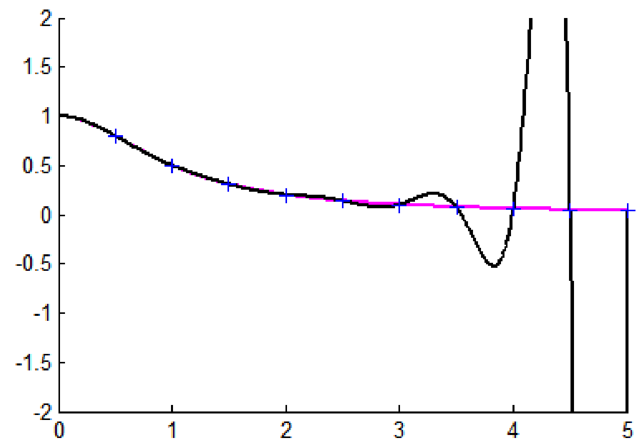
$$h = 1,\, h = 1/2,\, h = 1/4,\, h = 1/5.$$

We obtain the following graphs where we have used that f(x) is symmetric about the y-axis so we only show the graphs ob [0, 5].

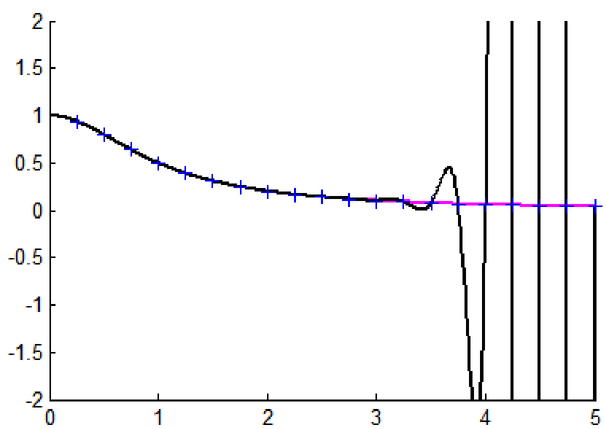
h = 1



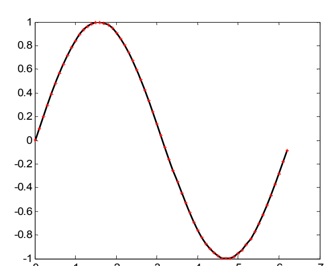
h = ½



h = ¼



For this function f(x) it is possible to show that the interpolation error goes to infinity as the spacing h of the points goes to zero (or equivalently the degree of the interpolant becomes larger and larger). Note that f(x) is infinitely differentiable. But this doesn't always happen. Let g(x) = sin(x) over [0, 2π]. The interpolant to the equispaced points with h = 0.1is shown as red dots on the sine function.

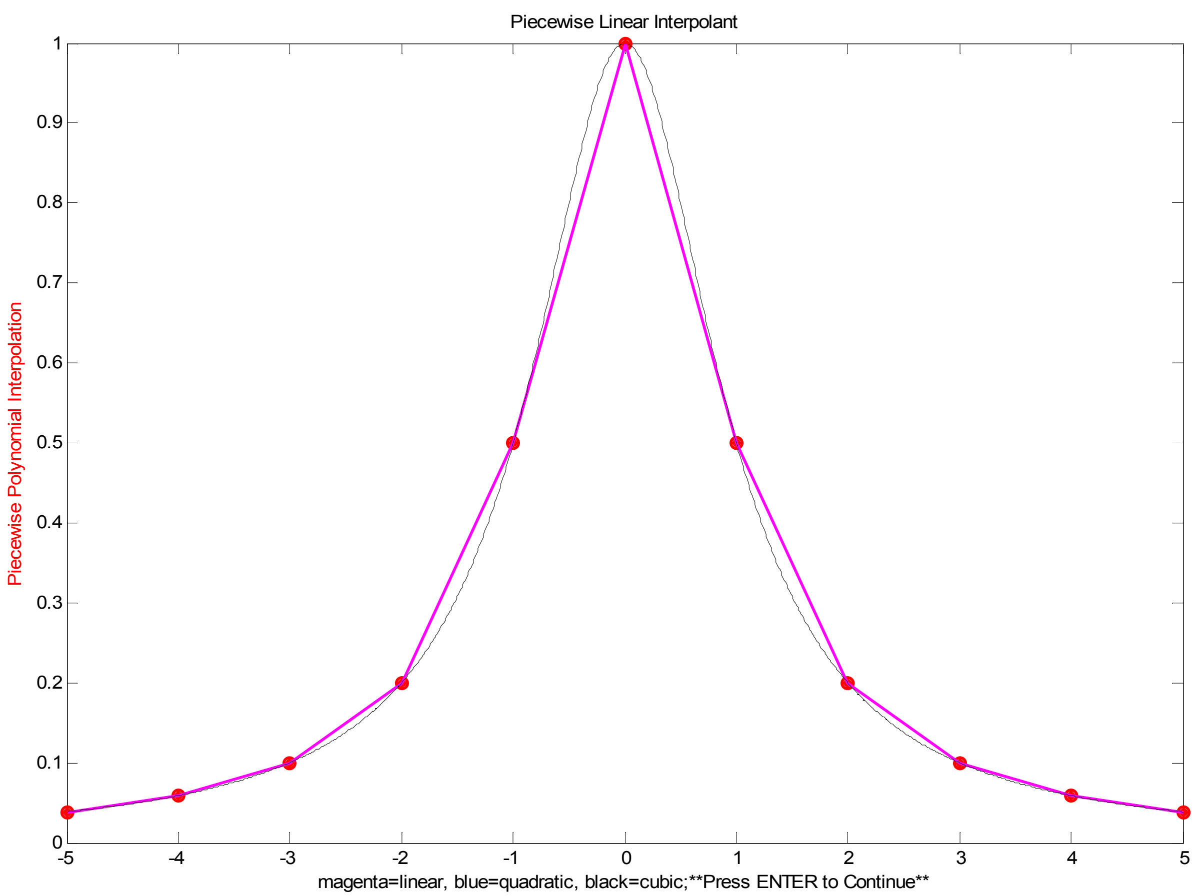


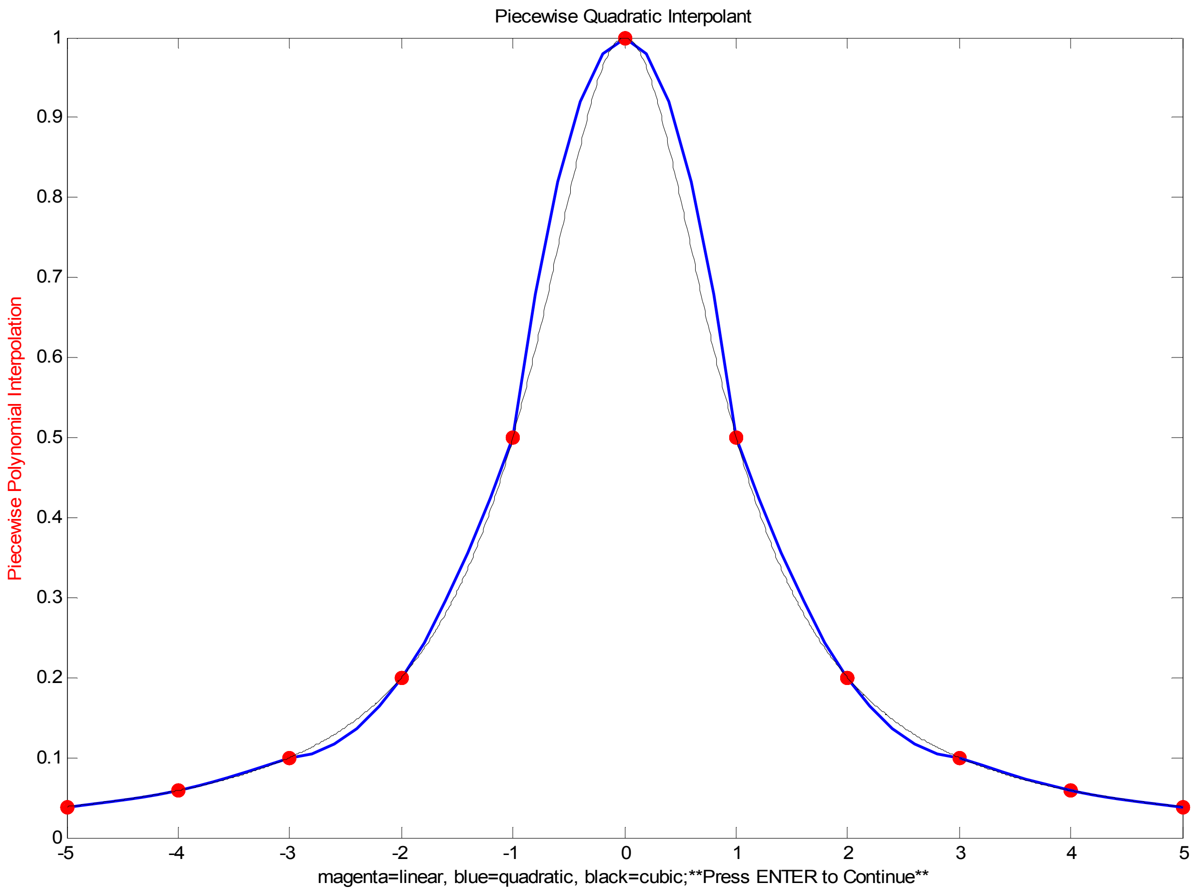
The behavior described for $f(x) = \frac{1}{1+x^2}$ is called the **“Runge” phenomena** or the **“polynomial wiggle problem”**.

Runge, Carl (1901), "Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten". *Zeitschrift für Mathematik und Phvsik* **46**: 224–243.

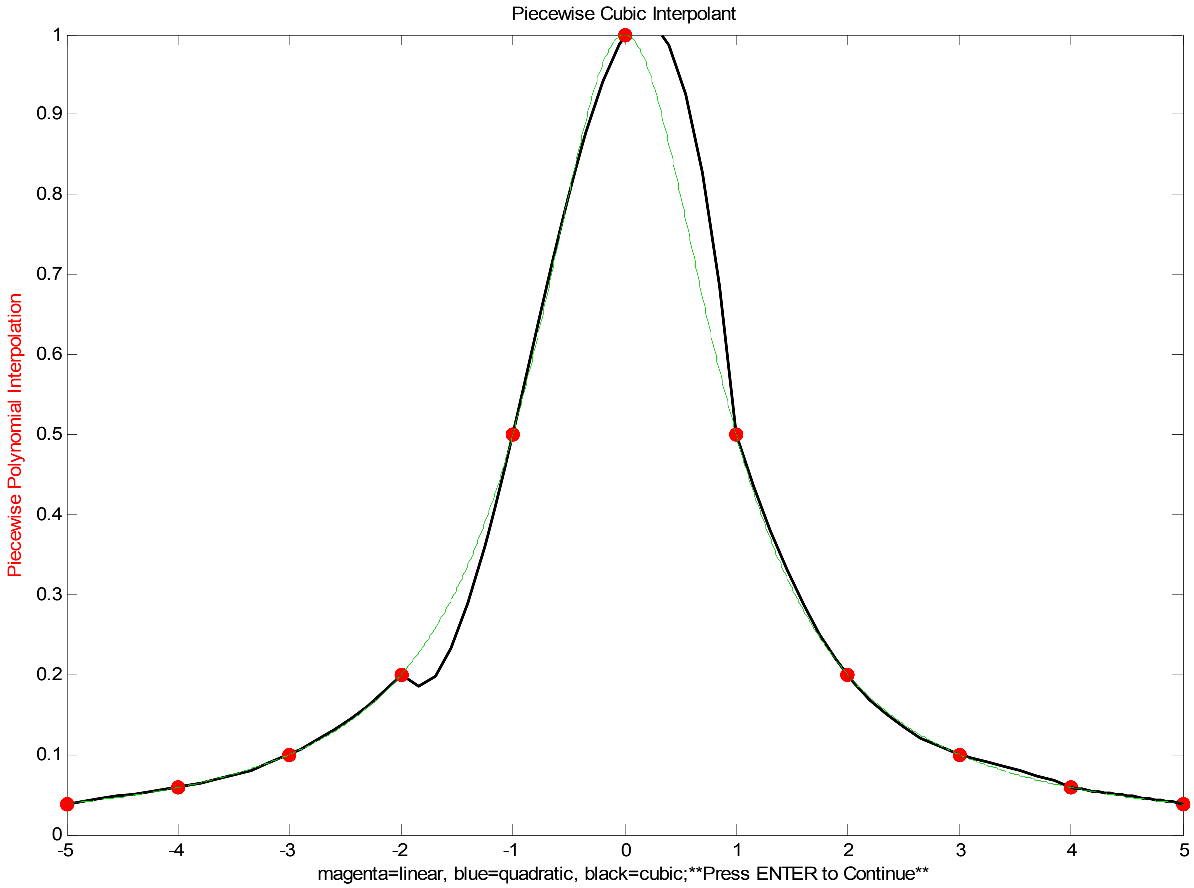
The “wiggle problem” suggests that we use low order polynomial interpolation over subsets of the data set. This is referred to as piecewise polynomial interpolation. We briefly consider linear, quadratic, and cubic piecewise interpolation.

The simplest such procedure is piecewise linear interpolation which amounts to connecting successive data points with straight lines (‘connect-the-dots’). While the interpolating function constructed in this way is continuous, it is usually not differentiable because ‘sharp points’ occur at the data points. Such procedures generalize to piecewise quadratic and piecewise cubic by appropriately subdividing the data set into successive subsets of three and four points respectively with one point overlap to ensure continuity. To experiment with piecewise polynomial interpolation of this type use routine MATLAB **pwinterp**.





Piecewise Quadratic Interpolant
Piecewise Polynomial Interpolation
magenta=linear, blue=quadratic, black=cubic;**Press ENTER to Continue**
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-5
-4
-3
-2
-1
0
1
2
3
4
5


Piecewise Cubic Interpolant
Piecewise Polynomial Interpolation
magenta=linear, blue=quadratic, black=cubic;**Press ENTER to Continue**
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-5
-4
-3
-2
-1
0
1
2
3
4
5