HW 5
Due by CMS by 11:59 on Monday, April 4

Remember that you may (and should!) talk about the problems amongst yourselves, or discuss them with me or the TA, providing attribution for any good ideas you might get – but your final write-up should be your own.

1: Differentiating an interpolant. The code hw5newton (available on CMS) computes a vector of divided differences for use in the Newton form of interpolation; see the end of section 7.3.3 in Heath. Using these divided differences, write a routine hw5neval that evaluates the interpolating polynomial \( p \) and its derivative \( p' \):

\[
% [pxx, dpxx] = hw5neval(x, fdd, xx)
%
% Evaluate the Newton form of the interpolant at points xx.
% Inputs:
% x  -- coordinates of interpolation nodes
% fdd -- table of divided differences returned by hw5newton
% xx -- evaluation points
%
% Outputs:
% pxx -- interpolating polynomial evaluated at the points xx
% dpxx -- derivative of the interpolating polynomial evaluated at xx

function [pxx, dpxx] = hw5neval(x, fdd, xx)

Your code should only involve \( O(n) \) work per evaluation point, where \( n \) is the number of interpolation points. You may want to test your code using the hw5pitest script (also on CMS).
2: Fitting a Lorentzian. A Lorentzian is a function with the form

\[ L(x; A, c, \sigma) = \frac{A}{1 + 4(x - c)^2/\sigma^2}. \]

Lorentzian functions occur frequently in scattering theory, where they describe peaks in a measured response due to resonance phenomena. The parameters \( A, c, \) and \( \sigma \) respectively describe the amplitude, center, and width of the peak at half amplitude.

Given a set of points \( \{x_j\}_{j=1}^N \) and corresponding measurements \( \{y_j\}_{j=1}^N \), write a program to find \( A, x_0, \) and \( \sigma \) to minimize the sum-of-squares error

\[ \phi(A, c, \sigma) = \sum_{j=1}^{N} (L(x_j; A, c, \sigma) - y_j)^2. \]

Your code should have the form

```matlab
function [A,c,sigma] = hw5fit(x,y)
% Least-squares fit a Lorentzian of the form
% \( L(x) = \frac{A}{1+4*(x-c)^2/\sigma^2} \)
% to measured values y(i) at points x(i)
[I recommend the following strategy:

1. Form initial estimates of \( A \) and \( c \) based on the the maximum \( y \) value and the corresponding \( x \) value. Estimate \( \sigma \) based on the range of \( x \) values where the corresponding \( y \) is at least \( \hat{A}/2 \), where \( \hat{A} \) is the initial estimate of the peak amplitude.

2. Optimize \( A, c, \) and \( \sigma \) by a few steps of a basic Gauss-Newton iteration (I used ten). You do not need to implement a line search — the basic Gauss-Newton step will be fine for our purposes. The Gauss-Newton iteration is described in section 6.6.1 of Heath.

3. Test your code using \texttt{hw5p2test} (available on CMS). You should get answers that are correct to within rounding error in the case of no noise; you may be surprised to see how well the estimation procedure works even in the presence of noise.