Homework set 4: Due April 5.

- Paper and pencil problems:
  Problems 7.1, 7.2, 7.4, and 7.5 on p. 236 of the text book.

- The following MATLAB program illustrates, through an example due to Runge, that polynomial interpolation using equidistant points can provide a very poor approximation even of a very smooth function, cf. section 7.2.6 of the text. I ran this program in class on March 24 and then revised it, using the Chebyshev points given by the formula on p. 227.

```matlab
% Script File: RungeEu
% For n = 7:2:15, the equal-spacing interpolants of f(x) = 1/(1+25x^2) on [-1,1]
% are of plotted.
close all
x = linspace(-1,1,100);
y = ones(100,1)./(1 + 25*x.^2);
for n=7:2:15
    figure
    xEqual = linspace(-1,1,n);
yEqual = ones(n,1)./(1+25*xEqual.^2);
cEqual=InterpN(xEqual,yEqual);
pValsEqual = HornerN(cEqual,xEqual,x);
plot(x,y,x,pValsEqual,xEqual,yEqual,'*')
title(sprintf('Equal Spacing (n = %2.0f)',n))
end
```
function c = InterpN(x,y)
% c = InterpN(x,y)
% The Newton polynomial interpolant.
% x is a column n-vector with distinct components and y is
% a column n-vector. c is a column n-vector with the property that if
% p(x) = c(1) + c(2)(x-x(1)) + ... + c(n)(x-x(1))... (x-x(n-1))
% then
% p(x(i)) = y(i), i=1:n.

n = length(x);
for k = 1:n-1
    y(k+1:n) = (y(k+1:n) - y(k))./(x(k+1:n) - x(k));
end
c = y;

function pVal = HornerN(c,x,z)
% pVal = HornerN(c,x,z)
% Evaluates the Newton interpolant on z where
% c and x are n-vectors and z is an m-vector.
% pVal is a vector the same size as z with the property that if
% p(x) = c(1) + c(2)(x-x(1)) + ... + c(n)(x-x(1)) ... (x-x(n-1))
% then
% pVal(i) = p(z(i)), i=1:m.

n = length(c);
pVal = c(n)*ones(size(z));
for k=n-1:-1:1
    pVal = (z-x(k)).*pVal + c(k);
end
Modify the program replacing the polynomial interpolant by the cubic spline interpolant provided by MATLAB; run help spline to get started. Plot the spline interpolant and the polynomial interpolant, if possible in the same plot and discuss the properties of the approximations obtained by the two procedures. Use both the equidistant and Chebyshev sets of points.

- Modify the function InterpN.m so that it can be used to compute the cubic Hermite interpolant of a function given by the data \( f(a), f(b), \frac{df(a)}{dx} \) and \( \frac{df(b)}{dx} \). If possible, also provide a program which computes a piecewise Hermite cubic interpolant. The data for a piecewise Hermite cubic interpolant is given by the function values and the first derivative of a function \( f(x) \) at a number of distinct points \( x(1), x(2), \ldots, x(N) \) and provides the Hermite cubic interpolant between \( x(1), x(2), x(2), x(3) \), etc. Try your program for the function \( \frac{1}{1 + 2x^2} \), i.e., the function of the Runge example.

- Do computer problem 7.4 on p. 237. If you wish, you can of course use the program developed for the previous problems.