Math 128A, Fall 2018, Wilkening
Programming Assignment 2
due Tues, Oct 9

In this assignment we will explore the accuracy of polynomial interpolation for large n. It turns out that the interpolating polynomial usually does not converge to the function as $n \to \infty$ when the grid points are uniformly spaced, but there is another set of points, the Chebyshev grid points, in which it is guaranteed to converge under very mild assumptoins on f. (f only needs to be Lipshitz continuous.)

1. Write a program that begins `function yout = interpolate1(f,xin,xout)`. Here `f` is a function handle (like we have been using for Newton's method, etc.), `xin` contains the interpolation points, and `xout` contains the evaluation points. The program should return `yout`, the values of the interpolating polynomial $P(x)$ evaluated at each point in `xout`. Use the formula

$$P(x) = \sum_{j=0}^{n} f(x_j) L_j(x), \qquad L_j(x) = \prod_{k \neq j} \frac{x - x_k}{x_j - x_k},$$

where the x_j are the entries of the vector `xin` and x ranges over the entries of `xout`. **Do this with $\mathbf{n = 10, 19, 50, 99}$ on two grids:**

```
xinU = linspace(-1,1,n+1)          (uniform),
xinC = cos(linspace(-pi,0,n+1))    (Chebyshev).
```

Note that the Chebyshev grid points are clustered more closely at the endpoints of the interval. For output, use `xout=linspace(-1,1,500)`, and for the function, use `f = @(x) 1.0 ./ (1+9*x.^2)`.

2. Write a second program `function yout = interpolate2(f,xin,xout)` that does exactly the same thing as above, but using the following (barycentric) formula for the interpolating polynomial:

$$P(x) = \left\{ \begin{array}{ll} f(x_j) & x = x_j \\ \sum_{j=0}^{n} \frac{\lambda_j f(x_j)}{x - x_j} \Big/ \sum_{j=0}^{n} \frac{\lambda_j}{x - x_j}, & x \notin \{x_0, \ldots, x_n\} \end{array} \right\}, \qquad \lambda_j = \frac{1}{\prod_{k \neq j} (x_j - x_k)}.$$

In case you are curious, the derivation of barycentric interpolation is given in chapter 5 of Trefethen's book, *Approximation Theory and Approximation Practice*, which is posted on bCourses.

Submit your codes `interpolate1.m`, `interpolate2.m` and a write-up briefly explaining how your codes work and showing 8 plots: First four: for $n \in \{10, 19\}$ and `xin` $\in$ {`xinU`,`xinC`}, plot $f(x)$ and $P(x)$ on top of each other, using either code for $P(x)$. Last four: for $n \in \{50, 99\}$ and `xin` $\in$ {`xinU`,`xinC`}, plot `semilogy(xout,1.0e-18+abs([youtA1-f(xout),youtA2-f(xout)]),'linewidth',1)`, where `A=U` or `A=C` and `youtA1` and `youtA2` are the results of the two codes you wrote. Examples with $n = 14$ and $n = 99$ are given below. (Modify the instructions above as needed if you are not using Matlab).