Homework #3: Fun with Interpolation

For this problem set you will need some auxiliary files and programs. Download and unpack ProblemSet03.tar.gz from the website.

1. Consider data at the three points $(x_0, y_0) = (0, 0)$, $(x_1, y_1) = (1, 2)$, $(x_2, y_2) = (2, 2)$

   (a) Find the interpolating polynomial $P(x)$ that passes through these three points using the following bases

      i. monomial: $P(x) = a_0 + a_1 x + a_2 x^2$

      ii. Lagrange: $P(x) = \sum_{k=0}^{2} y_k L_k(x)$

      iii. Newton: $P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$ (Note: we didn't talk about the newton basis, but see my notes Interpolation.pdf)

   (b) Show that all three bases return the same polynomial (i.e. $P(x)$ is unique).

   (c) Use the uniqueness of the interpolating polynomial to show that for general $N + 1$ points

   $$\sum_{k=0}^{N} L_k(x) = 1$$

   at any value of x. (I.e. that the interpolant of a constant is a constant). (hint: consider the Newton basis and uniqueness).

2. Unfortunately, for evenly spaced points, the Lagrange interpolants can vary considerably from 1 near the ends. Figure 1 shows Lagrange Polynomials for 4 and 8 evenly spaced points on the unit interval $x = [0, 1]$.

   (a) For $N = 4$ find the position and value of the maximum value of $|L_k(x)|$ (hint: the position can be found analytically).

   (b) Now estimate $|L_k|_{\max}$ for general $N > 2$ by writing a matlab function

   `LMax = lagrangeError(N)`

   that returns the maximum absolute value of all Lagrange Polynomials for N evenly spaced points. (hints: (1) use my routine `L=lagrange(x,xNodes)` to calculate the lagrange polynomials at points x. (2) note that the maximum excursions

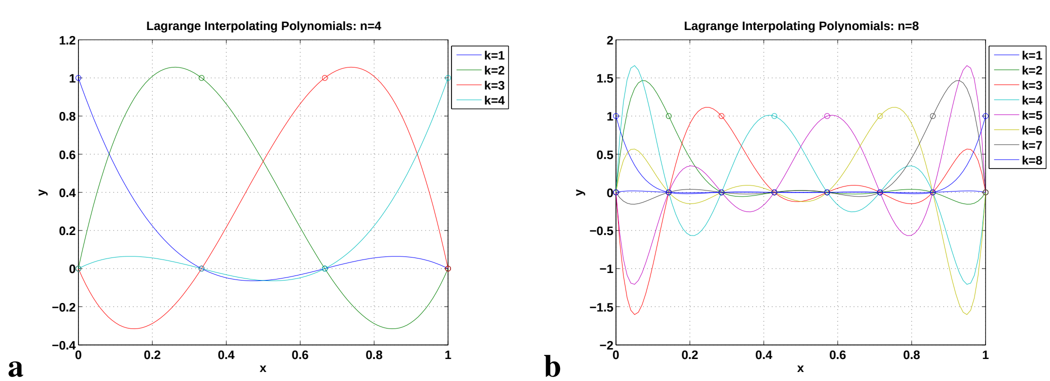

Figure 1: Lagrange Polynomials for N evenly spaced points on the unit interval $x = [0, 1]$. (*a*) $N = 4$, (*b*) $N = 8$. Both figures were produced using `xNodes=linspace(0,1,N); seeLagrange(xNodes)`

are always between the first two points (3) you might want to look at the matlab functions `max` and `abs`). You can check your answers against mine using the test function `plotLagrangeError(@lagrangeError)`.

(c) From these results, what maximum value of N do you believe is "safe" for full Polynomial interpolation with evenly spaced points?

3. Fun with Pchips.

(a) Show that the Piecewise Cubic Hermite polynomial between two points x_k and x_{k+1} with values y_k, y_{k+1} and first derivates d_k, d_{k+1} can be written as

$$P(s) = (1 - 3s^2 + 2s^3)y_k + (3s^2 - 2s^3)y_{k+1} + s(1-s)^2 h_k d_k + s^2(s-1)h_k d_{k+1}$$

where $h_k = (x_{k+1} - x_k)$ is the interval length and $s = (x - x_k)/h_k$ is the "fractional distance" ($s \in [0, 1]$) between grid points.

(b) Modify Moler's Function `pchiptx` to create a new function

```
function y = pchipMean(xNodes,yNodes,x)
```

that calculates the slopes d_k as the *arithmetic mean* of the adjacent δ_k's i.e.

$$d_k = \frac{1}{2}(\delta_k + \delta_{k-1})$$

and assumes $d_1 = \delta_1$, $d_{n+1} = \delta_n$.

(c) Compare the behavior of `pchiptx`, `pchipMean` and `interp1` using spline interpolation for the following data

```
xNodes=linspace(0,1,7);
yNodes=sin(2*pi*xNodes);
```

and produce a plot that looks like Figure 2 (and don't forget the labels)

(d) What is the relative errors of the different interpolants at $x = 0.25$?

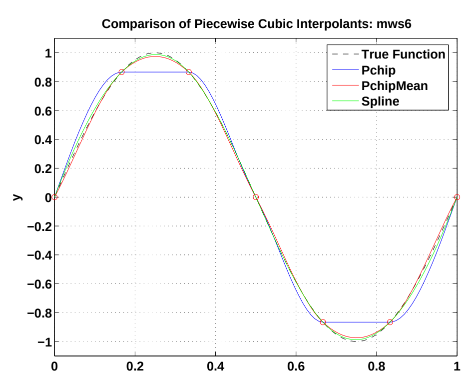

Figure 2: Comparison of Pchip, PchipMean and Spline interpolants to true function $y = \sin(2\pi x)$ through 7 points.