Math 442 Lab 1

1. Now, we will examine (numerically) the error analysis for interpolating polynomials that we have been discussing in class. If you check your handouts, Theorems 1 and 2 of your notes basically state that if $p_n(x)$ was the n-th degree polynomial interpolant of $f(x)$ on (a, b) and if $f(x)$ has $n+1$ continuous derivatives on $[a, b]$, then

$$f(x) - p_n(x) = e_n(x; c), \tag{1}$$

where the error function is explicitly given by

$$|e_n(x)| \le \frac{Mh^{n+1}}{4(n+1)},$$

where the nodes are equally spaced and

$$M = \max_{c \in [a,b]} \left| f^{(n+1)}(c) \right|.$$

In order to test this theorem, we will use the m-files **Pint.m**, **errdriver.m** and the ever useful **f.m**. Download these from the course webpage.

The m-file **f.m** should currently be returning function values of $f(x) = \sin(x)$, and the function **Pint.m** calculates the interpolating polynomial according to the linear system set up in Number 1a of Homework #2. It then evaluates the interpolating polynomial at a set of input values determined by the user. The m-file **errdriver.m** is the driver which initializes the appropriate variables, calls the aforementioned functions and plots the results. Open these files beginning with **errdriver.m**, read through them and run the driver to see what you see. The driver is currently set up to plot the function against a polynomial interpolant with 4 nodes. (ie. $n = 3$)

Assignment Modify the driver so that you can see a set of subplots which allows you to observe the uniform convergence of the interpolating polynomial as you increase the number of interpolation nodes. To do so, insert the following code:

```
figure(2)
x=[0:0.01:10];
for n = 1:9
subplot(3,3,n);
plot(x,f(x),'r-',x,Pint(0,10,x,n),'b-.');
end; % end-for-loop
```

When you (save first) run the driver, you should see the first 9 polynomial interpolants of $f(x) = \sin(x)$ using equally spaced nodes over the interval $[0, 10]$. Notice that the polynomials converge uniformly to the function $f(x) = \sin(x)$ as the number of nodes increase.

over $->$

(a) Label the figure with the subplots under the heading *Lab1, Figure 1.*

(b) On paper, explain how we can use the theorem discussed in class to mathematically justify the convergence you just observed numerically.

(c) Edit the m-file **f.m** to return Runge's function. (You should only need to delete the % sign on one line and insert a % sign in front of the *z=sin(x)* statement.) Modify the subplot code above to examine Runge's function over the interval $[-5, 5]$ and show the first $n = 16$ polynomial interpolants. Therefore, your subplot should be 4×4 instead of 3×3. Label this figure as *Lab 1, Figure 2* . Do the results in this figure contradict Theorem 2? Explain your answer. **Submit printed copies of your figures and your explanations.**

2. Next we examine the Newton polynomial form and Divided Differences. Download the m-files **newtdriver.m**, **newtpoly.m** and **nestmult.m**. The function **newtpoly.m** is basically the code from a book, and **nestmult.m** is a "rough" piece of code (a function) that implements nested multiplication to evaluate the polynomial at a given set of points. Of course, **newtdriver.m** is the driver which sets up the appropriate nodes and function values, etc. and then calls the other functions. Currently, **newtdriver.m** works through the construction of the Newton polynomial for the function $g(x) = \cos x$.

   (a) Modify the driver to plot the interpolating polynomial and the original function on the same axes. (so that you can compare the function to the polynomial) Label the figure as *Lab1 Figure 3.*

   (b) Now, modify the code to generate another figure which only plots the error between the interpolating polynomial and the original function. Once this is working, go back to Number 3 on Homework #2. Using the Newton polynomial, show that your prediction for the value of n does indeed result in an error of less than 10^{-10}. (The error estimate is the same for both polynomial approximations.) **Remember that you will have to adjust the function values, the interval over which you interpolate and the interval over which you plot**. Label the figure as *Lab1 Figure 4.*

   (c) Once you get the hang of plotting the error, can you make the value of n smaller to see where the true threshold lies for the error to be within 10^{-10}. In other words, is it necessary that n be quite as large as you computed using the theory? Often, the predicted value of n in the mathematical theory is not always the smallest value of n that will actually satisfy the error constraints. Investigate, and explain your results in a paragraph. **Submit printed copies of your figures and your explanations.**