MA 428: Homework 1: Interpolation

Due: Friday, 18 September 2015

Theoretical

[*40 points*]
Section 5.1 – 8,9; Section 5.3 – 3,8

Computational

Recall form the lectures the Lagrange form of the interpolating polynomial:
Suppose we wish to interpolate and arbitrary function $f(x)$ on a set of distinct nodes $x_0, x_1, \cdots, x_n$. First we define $n+1$ Lagrange polynomials $\mathcal{L}_{n,j}$ for $j = 0, \cdots n$ with the property:

$$\mathcal{L}_{n,j}(x_i) = \begin{cases} 0 & \text{if } j \neq i \\ 1 & \text{if } j = i \end{cases} \tag{1}$$

We can then interpolate an function f by the Lagrange form of the interpolation polynomial:

$$P_n(x) = \sum_{i=0}^{n} \mathcal{L}_{n,i}(x) f(x_i) \tag{2}$$

By construction when we evaluate $P_n(x)$ at x_j, we get $f(x_j)$:

$$P_n(x_j) = \sum_{i=0}^{n} \mathcal{L}_{n,i}(x_j) f(x_i) = \mathcal{L}_{n,j}(x_j) f(x_j) = 1 \cdot f(x_j). \tag{3}$$

Thus, P_n is the interpolating polynomial for the function f at the nodes $x_0, x_1, \cdots, x_n$. Recall that

$$\mathcal{L}_{n,j}(x) = \left(\frac{x - x_0}{x_j - x_0}\right)\left(\frac{x - x_1}{x_j - x_1}\right) \cdots \left(\frac{x - x_{j-1}}{x_j - x_{j-1}}\right)\left(\frac{x - x_{j+1}}{x_j - x_{j+1}}\right) \cdots \left(\frac{x - x_n}{x_j - x_n}\right), \quad (0 \leq j \leq n). \tag{4}$$

Exercise 1

[*10 points*]
In this exercise you will construct Lagrange polynomials on given nodes `xdata`. Your Matlab function should take in as input vectors containing the interpolating nodes and the x values to be evaluated.

1. Write a Matlab function m-file called lagrangep.m that computes the Lagrange polynomials for an $0 \leq k \leq n$. *You should of course adjust for MATLAB numbering which starts at 1)* A suggested signature for your function is:

```
function pval = lagrangep( j , xdata, xval )
% pval = lagrangepp( j , xdata, xval )
% returns the value of the Lagrange polynomial at each value in xval.
% j is the index for the Lagrange polynomial
% xdata is a set of specified  interpolation points
% xval= the set of points where the lagrange function is to be evalauted
% pval= an array of the values of the lagrange polynomial at each point in

% your name, data and comments
```

Your function should return the value of Lagrange polynomial at each value in xval.
Hint: You could implement a general formula using the following:

```
pval = 1;
for i = 1 : ???
  if i ~= j
      pval = pval .* ???   % elementwise multiplication
  end
end
```

Note: If xval is a vector of values, then pval will be a vector of the corresponding values, so that an elementwise multiplication (.*) is being performed.

2. Analyze your definition of the Lagrange polynomials. From the definition, test the output of your function for

```
lagrangep( 1, xdata, xval) for:
                                  xval=xdata(1),
                                  xval=xdata(2),
                                  xval=xdata(3).
```

Comment on your output.

Excercise 2

[*40 points*]

1. Write a Matlab function m-file that defines the lagrange interpolating polynomial for interpolating nodes `xdata` with associated `ydata` and evaluates this polynomial at `xval` Your function should have a signature:

```
function yval = eval_lag ( xdata, ydata, xval )
% yval = eval_lag ( xdata, ydata, xval )
% eval_lag fits and evaluates the lagrange interpolating polynomial
%xdata and ydata are specified points that the interpolating function has
%to pass through.
%yval is an array of values of the Lagrange interpolating polynomial at
%each point in xval.
```

2. Test your function eval_lag with the data

```
xdata = [0 1 2 ]
ydata = [0 1 8 ]
```

by evaluating at `xval = xdata`. You should obtain `ydata` back.

3. Test your function by interpolating the polynomial that passes through the following points, again use `xval = xdata`.

```
xdata= [   -3   -2 -1  0  1   2    3]
ydata= [1636 247 28  7  4  31  412]
```

4. Complete Problem 4 in Section 5.1.You may find the provided utility function `test_poly_interpolate` useful.

5. In problem 4 calculate the error from using 6 and 12 uniformly distributed interpolating points, respectively. Comment on your results.

Excercise 3

[*10 points*]

1. Consider the function $f(x) = \dfrac{1}{1+x^2}$. In Matlab you can define the function as follows:

```
function y= f(x)
%function y = f(x)
%y=1./(1+x^2);
y= 1./(1+x.^2);
```

2. Using the provided `test_poly_interpolate` function construct *xdata* containing *ndata* evenly spaced points on the interval $[-5, 5]$ and evaluate the error in interpolating the Runge function, fill in the following table

```
ndata =  5  Max Error = ________
ndata = 11  Max Error = ________
ndata = 21  Max Error = ________
ndata = 41  Max Error = ________
```

Comment on your results and give a possible explanation for the effects your observe. In particular pay close attention to the plots of the graph of f and the interpolant as n increases.

Submission

Email me your zipped m files, including your summary file with a discussion of your results for the computational part of the assignment. Your summary file must include all matlab output and answers to questions related to the output.