MATH 353: Engineering Mathematics III – Section 012

Spring 2014 (F.–J. Sayas) | Lab # 4 | March 10

Open Matlab and *move to the Desktop or to a folder where you can find your work* at the end of thesession. Type these two lines

```
>> diary myworkMarch10
>> format long
>> format compact
```

Download the functions `evaluatelagrange.m` from my website.

1. Here's an easy one. Open the editor and copy the following lines in a script. Save it as `scriptMarch10.m`

```
f = @(x) 1./(2+sin(x.^2));
x=[0,1,2];
plot(x,f(x),'o')      % plot interpolation points
hold on               % keep the points for the next figure
xx=0:0.1:2;          % many points in [0,2]
yy=evaluatelagrange(x,f(x),xx);
plot(xx,yy)
```

   Now run the script. What did we do? What function are we interpolating? Where? What is the degree of the polynomial that we are plotting?

2. Add a line to the previous code so that you see the graph of the function `f` on top of everything. Make the plot in color red. (To do this, look for help `plot` or `fplot`. If you cannot figure it out, ask.)

3. Repeat everything, interpolating in the points $0, 1, 2$, and $3/2$.

4. **An infamous example.** We want to interpolate the function

$$f(x) = \frac{1}{1 + 12\,x^2}$$

   on several equally spaced points in the interval $[-1, 1]$. Exactly in these points

$$x_1 = -1, \quad x_2 = -1 + \frac{2}{n}, \quad x_3 = -1 + \frac{4}{n}, \quad \ldots \quad x_n = -1 + \frac{2n-2}{n}, \quad x_n = 1 = -1 + \frac{2n}{n}.$$

   This can also be seen as

$$x_i = a + \frac{b-a}{n}\,(i-1) \qquad i = 1, \ldots, n+1, \qquad \text{with } a = -1, b = 1.$$

   - We are going to try this with several different choices of n, so n should be defined to have a value right at the beginning of the script and we will change that later.

- Other than that, follow the same structure as the script we already had, but plot everything in the interval $[-1, 1]$. You will need quite some points to plot (this is `xx` in the code).
- Once you are done, run the code for

$$n = 5, \qquad n = 10, \qquad n = 15, \qquad n = 20.$$

  At each precise value of n, what is thedegree of the interpolating polynomial? *Can you see the train wreck?*

5. **The Chebyshev points.** What happened in the previous example is a phenomenon that is very well understood and can be phrased as: *interpolating a function in an increasing number of equidistant points may lead to a diverging approximation.* One way to avoid this is to use the following collection of points. In order to approximate in the interval $[a, b]$ we use the points

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{(2i-1)\pi}{2n+2}\right) \qquad i = 1, \ldots, n+1.$$

   Repeat the previous example, substituting the atuomatic generation of the equispaced opoints by this new choice of points. Run again the code for increasing values of n.

6. **The Lagrange polynomials.** Given points $x_1, \ldots, x_{n+1}$ (which we are going to suppose to be given in increasing order, although this is not important), we can define the polynomials

$$L_k(x) = \frac{x - x_1}{x_k - x_1} \cdots \frac{x - x_{k-1}}{x_k - x_{k-1}} \frac{x - x_{k+1}}{x_k - x_{k+1}} \cdots \frac{x - x_{n+1}}{x_k - x_{n+1}} \qquad k = 1, \ldots, n+1.$$

   The computation of these polynomials is at the heart (in the inner loop) of the function `evaluatelagrange`. Create a script that: uses the points

$$x_1 = 1, \quad x_2 = 3, \quad x_3 = 4, \quad x_4 = 6, \quad x_5 = 7$$

   and plots the polynomials $L_1(x), \ldots, L_5(x)$ (what is the degree of these polynomials, by the way?) in the interval $[1, 7]$. **Hint.** A particular choice of values of `y` will give you each of the polynomials, and there's basically nothing to program. You just need to run `evaluatelagrange` five times.