MACM 316 – Computing Assignment 6

- Read the *Guidelines for Assignments* first.
- Submit a one-page PDF report to Canvas and upload you Matlab scripts (as m-files). Do not use any other file formats.
- Keep in mind that Canvas discussions are open forums.
- You must acknowledge any collaborations/assistance from colleagues, TAs, instructors etc.

Polynomial interpolation and node distribution

The purpose of this assignment is to examine how the locations of the nodes $x_0, x_1, \ldots, x_n$ affect the accuracy and robustness of polynomial interpolation.

First, download the file *polyinterperr.m*. This is a Matlab function which takes as inputs a function $f(x)$ and a vector of nodes $[x_0, x_1, \ldots, x_n]$. Its output is the error

$$e_n = \max_{-1 \leq x \leq 1} |f(x) - P(x)|,$$

where $P(x)$ is the polynomial interpolant of $f(x)$ at the nodes. Using this function, write a code to compute the error for polynomial interpolation at the equally-spaced nodes

$$x_i = -1 + 2i/n, \quad i = 0, \ldots, n.$$

Apply your code to the function

$$f(x) = \exp(\sin(5x + 0.5)),$$

and plot $\log_{10}(e_n)$ versus n for $n = 1, 2, 3 \ldots, 100$. Discuss.

Hopefully, you will have seen that the error does not approach machine epsilon. As we discussed in class, equally-spaced nodes are usually poor choices for polynomial approximation. We are now going to search for better nodes using a so-called *greedy* algorithm.

Suppose that the first $n+1$ nodes $x_0, \ldots, x_n$ have been computed. We define the next point x_{n+1} to be the value of x that maximizes the function

$$V(x) = \prod_{k=0}^{n} |x - x_k|, \qquad -1 \leq x \leq 1.$$

Write a code that computes the nodes $x_1, \ldots, x_{100}$ according to this rule given the starting value $x_0 = -1$. You should replace the maximum over $-1 \leq x \leq 1$ by a maximum over a sufficiently fine grid of points, e.g. 10,000 equidistant points between -1 and $+1$. Let $X = [x_0, x_1, \ldots, x_{101}]$ be the vector of computed nodes. Sort this vector in increasing order (use `Xsort = sort(X)` in Matlab) and plot `Xsort`. How do your nodes compare with the equally-spaced nodes?

Next, run your code from $n = 1, 2, 3, \ldots, 100$ and use it to compute the interpolation error e_n based on these nodes for the function defined above. Plot your results and discuss.