M.Sc. in High-Performance Computing 5633A - Numerical Methods for High-performance Computing Programming Assignment 3

Kirk M. Soodhalter (ksoodha@maths.tcd.ie)
School of Mathematics, TCD

Rules

To submit, make a single tar-ball with all your code and a pdf of any written part you want to include. Submit this via Microsoft Teams by 16. November, 2020. Late submissions without prior arrangement or a valid explanation will result in reduced marks.

Question

For this assignment, we are going to explore root-finding and interpolation algorithms. Please implement functions

$$\begin{aligned} \texttt{root} &= \texttt{bisection}(\texttt{f},\texttt{a},\texttt{b},\texttt{tol}),\\ \texttt{root} &= \texttt{newton}(\texttt{f},\texttt{fderiv},\texttt{x0},\texttt{tol}),\\ \texttt{p} &= \texttt{ninterp}(\texttt{xvals},\texttt{fvals}),\ \textit{and}\\ \texttt{y} &= \texttt{polyeval}(\texttt{p},\texttt{x}) \end{aligned}$$

where `f` and `fderiv` are anonymous functions of a function and its derivative as discussed in the lecture, `xvals` and `fvals` are vectors containing x-values and function values, and `p` is a vector of Newton polynomial coefficients.

For the following, develop MATLAB scripts to carry out the requested tasks.

1. (a) Use your `bisection` and `newton` implementations to find all solutions to the following equations

$$\begin{aligned} e^{-x} &= x(2-x)\\ 10\sin\left(x-\frac{1}{10}\right) &= \frac{x^7}{1000}. \end{aligned}$$

For `bisection`, you may roughly bracket these solutions by any means, including plotting a graph to have a look. For `newton` you may do the same to find an initial value "close enough" to the root to ensure convergence. Please explain with code comments what you did.

(b) What might you do if you were not able to visualize the problem with `plot`-type functions?

2. Now use your `ninterp` function to interpolate the same two problems from the previous example on 2, 3, and 4 points, and use the roots of the interpolated polynomials as approximate solutions to each equation.

   (a) For each problem, what seems to be the best degree of interpolating polynomial to approximate these solutions? Why?

   (b) Where does it seem to be best to place the interpolation nodes such that the roots seem to be good approximations of the solutions to each problem?