AM 147: Computational Methods and Applications: Winter 2021

Homework #4

Instructor: Abhishek Halder

Due: February 02, 2021

NOTE: Please submit your Homework as a single zip file named `YourlastnameYourfirstnameHW4.zip` via CANVAS. For example, `HalderAbhishekHW4.zip`. Please strictly follow the capital and small letters in the filename of the zip file you submit. You may not receive full credit if you do not follow the file-naming conventions. Your zip file should contain all .m files (MATLAB scripts) for the questions below.

Your zip file must be uploaded to CANVAS by 11:59 PM Pacific Time on the due date. The uploads in CANVAS are time-stamped, so please don't wait till last moment. Late homework will not be accepted.

Problem 1

Fixed point recursion **(25 points)**

Consider the Kepler's equation in celestial mechanics (see Lec. 7, p. 3) of the form $x = \underbrace{m + \varepsilon \sin x}_{g(x)}$, where we fix $m = \varepsilon = 0.5$. Then, $g : [0,1] \mapsto [0,1]$.

Using `format long`, write a MATLAB code with filename `YourlastnameYourfirstnameHW4p1.m` to compute the fixed point of the Kepler's equation via fixed point recursion with 20 iterations. Your code should generate a figure plotting the recursion index k (in the horizontal axis) versus x_k (in the vertical axis) for 50 randomly chosen initial guesses in $[0,1]$. For generating random initial guesses in $[0,1]$, you can use the MATLAB command `rand`.

Problem 2

Square linear system **(25 points)**

Using `format short`, write a MATLAB code with filename `YourlastnameYourfirstnameHW4p2.m` that finds a polynomial $p(t) = x_1 + x_2 t + x_3 t^2 + x_4 t^3$ satisfying the following conditions

$$\int_0^1 p(t)\mathrm{d}t = 0.1, \quad \int_0^1 t p(t)\mathrm{d}t = 0.2, \quad \int_0^1 t^2 p(t)\mathrm{d}t = 0.3, \quad \int_0^1 t^3 p(t)\mathrm{d}t = 0.4.$$

To do so, you need to formulate the problem as that of solving a square linear system of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ for appropriate $\boldsymbol{A}, \boldsymbol{b}$, and then solve it in MATLAB using `A\b`. Your code (same file) should then make a plot of t (in the horizontal axis) versus $p(t)$ (in the vertical axis) for $t \in [-1, 1]$, i.e., a plot of the computed polynomial curve.