AM 147: Computational Methods and Applications: Winter 2022

Homework #2

Instructor: Abhishek Halder

Due: January 19, 2022

NOTE: Please submit your Homework as a single zip file named `YourlastnameYourfirstnameHW2.zip` via CANVAS. For example, `HalderAbhishekHW2.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

Numerical errors **(20 points)**

For real x, consider the function

$$\mathrm{sinc}(x) := \begin{cases} \dfrac{\sin(x)}{x} & \text{for} \quad x \neq 0, \\ 1 & \text{for} \quad x = 0. \end{cases}$$

This function (see Fig. 1) arises frequently in signal processing and physical sciences. It has a Taylor series sum expansion

$$\mathrm{sinc}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n+1)!}.$$

In 1735, Euler proved an infinite product expansion

$$\mathrm{sinc}(x) = \prod_{m=1}^{\infty} \cos\left(\frac{x}{2^m}\right).$$

Consider computing the 7-term series approximation of sinc(5) in two different ways:

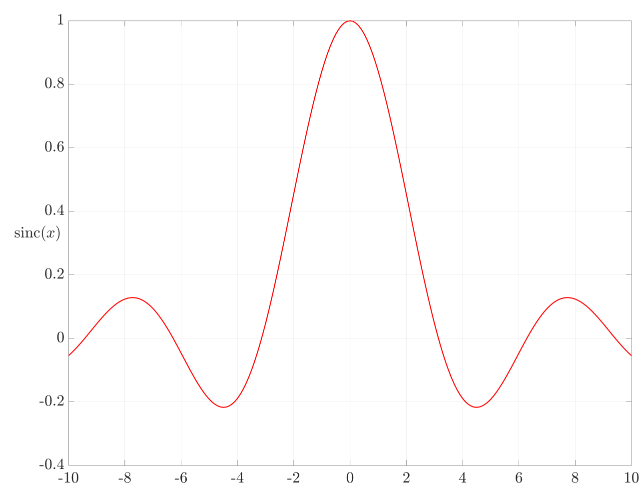

Figure 1: The function sinc(x) plotted for $x \in [-10, 10]$.

(i) $\mathrm{sinc}(5) \approx \sum_{n=0}^{6} \frac{(-1)^n 5^{2n}}{(2n+1)!}$, (ii) $\mathrm{sinc}(5) \approx \prod_{m=1}^{7} \cos\left(\frac{5}{2^m}\right)$.

Typing `sin(5)/5` in MATLAB command prompt (with default double precision short format) returns -0.1918. Let us call this as "true value". Submit a MATLAB script (.m file) named `YourlastnameYourfirstnameHW2p1.m` that computes the relative errors in using formula (i) and formula (ii). For your script, you may find the MATLAB in-built commands `factorial`, `sum`, and `prod` useful.

Problem 2

Weird integers **(**$20 + 10 = 30$ **points)**

We say that a positive integer is *weird* if the sum of its divisors, including 1 but excluding itself, is larger than that integer, and no subsets of those divisors sums to that integer.

Example: 70 is a weird integer because its divisors, including 1 but excluding itself, are $1, 2, 5, 7, 10, 14, 35$. We have $1 + 2 + 5 + 7 + 10 + 14 + 35 = 74 > 70$, and no subsets of $\{1, 2, 5, 7, 10, 14, 35\}$ sums to 70.

Non-example: 12 is NOT a weird integer because its divisors, including 1 but excluding itself, are $1, 2, 3, 4, 6$. Even though $1+2+3+4+6 = 16 > 12$, the subset $\{2, 4, 6\} \subset \{1, 2, 3, 4, 6\}$ sums to $2 + 4 + 6 = 12$.

You can easily verify by hand that there does not exist any single digit weird integer, i.e., all positive integers smaller than 10 are NOT weird. It turns out that 70 is the only possible two digit weird integer but this gets tedious to verify by hand. Computers can help.

Write a MATLAB script `YourlastnameYourfirstnameHW2p2.m` that takes two positive integers $n_{\min}, n_{\max}$ as inputs where $n_{\min} < n_{\max}$, and outputs all weird integers in the interval $[n_{\min}, n_{\max}]$.

For example, if the inputs to your code are $n_{\min} = 1, n_{\max} = 9$, then the output of your code should be the empty array $\{\}$. If the inputs to your code are $n_{\min} = 10, n_{\max} = 99$, then the output of your code should be the singleton array $\{70\}$.

Use your code to output all weird integers having no more than four digits, i.e., all weird integers in the interval $[10, 9999]$.

Hint: look up the commands `divisors`, `nchoosek`, `length`, `sum`, `prod` in MATLAB documentation. You may use some of these commands if you wish, but other than that your code should only use basic loops and logical operators.