Midterm Project

Math 300

Summer 2023

Submit a .pdf written in LaTeX solving the following problem. Your submission should include any code you used.

Consider the following function:

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ 3n+1 & \text{if } n \text{ is odd} \end{cases}$$

The famous Collatz conjecture says that repeatedly applying this function to any positive integer will eventually always yield 1. For example,

$$\begin{aligned} f(10) &= 5 \\ f(5) &= 16 \\ f(16) &= 8 \\ f(8) &= 4 \\ f(4) &= 2 \\ f(2) &= 1 \end{aligned}$$

Here it took 6 applications of the function to reach 1. If n is a positive integer, let the smallest number of iterations of f to reach 1 be denoted by $C(n)$. By our work above, we know that $C(10) = 6$. Furthermore, we can deduce that $C(1) = 0$ since the number

1 takes no iterations to get to 1.

Your goal is to use the programming you've learned in this course to find a vector $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ such that:

$$\underbrace{\begin{bmatrix} C(6) & C(2) & C(3) \\ C(4) & C(11) & C(6) \\ C(7) & C(8) & C(16) \end{bmatrix}}_{A} x = \underbrace{\begin{bmatrix} 14 \\ -4 \\ 17 \end{bmatrix}}_{b}$$

(There is a pattern to constructing A but you can do it manually)

Do this by:

1. Programming the function f using Python or any other language of your choice.
2. Programming the function C using Python or any other language of your choice.
3. Using your function C to calculate the entries of A.
4. Solving the equation $Ax = b$ using Python, Matlab or any other language of your choice to get x.

Be sure to include all of your code in your submission. Any necessary console commands should also be included.

Bonus:

(This is for fun if you want to try it and is worth no extra points)

Do the same thing, but for the system:

$$\begin{bmatrix} C(7) & C(2) & C(3) & C(4) & C(5) \\ C(6) & C(27) & C(8) & C(9) & C(10) \\ C(11) & C(12) & C(55) & C(14) & C(15) \\ C(16) & C(17) & C(18) & C(62) & C(20) \\ C(21) & C(22) & C(23) & C(24) & C(102) \end{bmatrix} x = \begin{bmatrix} 72 \\ 345 \\ 521 \\ 551 \\ 247 \end{bmatrix}$$

Hint: `np.fill_diagonal()` may be helpful here.