MAE 8 - Spring 2022
Homework 4

Instructions: Follow the homework solution template. Put all answers in a MATLAB script named **hw4.m**. For this homework, you will need to submit multiple files. Create a zip archive named **hw4.zip**. The zip archive should include the following files: **hw4.m**, **temperature.dat**, **lottery.m**, **winning_number.dat**, **piecewise2d.m** and **rgb_color.m**. Make sure that the two figures are plotted when your script is executed. Submit **hw4.zip** in CANVAS before 10 PM on 04/29/2022. Use double precision unless otherwise stated.

Problem 1: Create the following matrix **A**:

$$A = \begin{bmatrix} \ln(1/2) & \ln(1/3) & \ln(1/4) & \ln(1/5) \\ \ln(1/6) & \ln(1/7) & \ln(1/8) & \ln(1/9) \\ \ln(1/10) & \ln(1/11) & \ln(1/12) & \ln(1/13) \end{bmatrix}$$

a) Write the last column of matrix **A** into string **p1a** using %f with 9 characters including 6 decimal places for each element. The string must show a column vector.

b) Write the last row of matrix **A** into string **p1b** using %f with 7 characters including 4 decimal places for each element. The string must show a row vector.

c) Write matrix **A** into string **p1c** using %f with 10 characters including 7 decimal places.

d) Write matrix **A** into string **p1d** using %e with 10 characters including 7 decimal places.

Problem 2:

Download the file **temperature.dat** from CANVAS. The file includes the San Diego monthly-averaged temperature data (in Farenheit) from 1852 to 2020. The data is comma delimited and the columns represent the following: Year, Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, Annual mean. Load this data file into MATLAB and perform the following exercises. Provide the code needed to carry out the exercises. Do not obtain answers by inspection.

Consider the monthly average temperature (i.e., columns 2 to 13) in parts (a through g).

(a,b) Identify the month and the year when the monthly temperature is warmest on the record. Put the month in **p2a** and the year in **p2b**. For the month, use 1 for Jan, 2 for Feb, etc.

(c) Between 1900 and 2000, how many times had the monthly temperature been warmer than 75°? Put the answer in **p2c**.

(d) During the summer (e.g. Jun, Jul and Aug) between 1900 and 2000, how many times had the monthly temperature been warmer than 75°? Put the answer in **p2d**.

(e) Compute the annual cycle of the temperature by taking the average over the years for each of the months. Create **figure 1** to show the cycle with a bar graph. The x-axis shows the months (1 through 12) while the y-axis shows the temperature averaged over the years for the corresponding month. Set **p2e='See figure 1'**. Be sure to give the figure a title and label the axes.

(f) From figure 1, in what month of the cycle is the temperature warmest? Give the answer in a string **p2f = 'In general, the temperature peaks in ...'**.

(g) Write the annual cycle of the temperature into a file named **annual_cycle.dat** using ASCII format. The file should include 12 rows and 2 columns. The first column includes the months (1 through 12) and the second column includes the corresponding temperature. Set **p2g= evalc('type(''annual_cycle.dat'')')**. Note: Your homework script must produce the file during execution. If you include the file in your submission folder, the file will be removed before being graded.

Now compute the annual mean temperature (i.e., averaged over the twelve months). Create **figure 2** to show how the annual mean temperature varies over the years. Set **p2h='See figure 2'**. The figure should include the following:

- Use black solid line with a line width of 2 to show the annual mean temperature.
- Use red solid circle symbols to mark the years in which the temperature is warmer than 65°.
- Use blue solid diamond symbols to mark the years in which the temperature is cooler than 60°.
- Use red solid line with a line width of 4 to mark the two consecutive years over which the temperature increases the fastest.
- Use blue solid line with a line width of 4 to mark the two consecutive years over which the temperature decreases the fastest.
- Give the figure a title and a legend, and label the axes.

(i) In figure 2, does the temperature in San Diego get warmer or cooler over the years? Give the answer in a string **p2i = 'In general, the temperature ...'**.

Problem 3: Write a function **lottery.m** that checks for winning tickets and returns winning dollar amount. The function should have the following declaration: **function amount = lottery(ticket)** where **ticket** is an input vector with 6 lottery numbers and the output **amount** denotes the winning amount. Download the file **winning_number.dat** from CANVAS. The file includes 6 winning numbers. Load the file inside function **lottery**, i.e. **load('winning_number.dat')**, and use it to determine the winning dollar amount which depends on the matching between the numbers on the ticket and the winning numbers. Use the following formula:

- No match: amount = 0
- 1 matching numbers: amount = 10
- 2 matching numbers: amount = 100

- 3 matching numbers: amount = 1,000
- 4 matching numbers: amount = 10,000
- 5 matching numbers: amount = 1,000,000
- 6 matching numbers: amount = 100,000,000

Use **if** statement and function **intersect** to check for matching numbers. Remember to include a description for the function. Note that a different **winning_number.dat** file will be used during grading.

(a) Set **p3a = evalc('help lottery')**.
(b) Set **p3b = lottery([2, 3, 4, 5, 6, 7])**.
(c) Set **p3c = lottery([12, 23, 24, 34, 50, 61])**.
(d) Set **p3d = lottery([22, 33, 44, 50, 51, 61])**.
(e) Set **p3e = lottery([32, 43, 54, 44, 51, 61])**.
(f) Set **p3f = lottery([42, 53, 34, 44, 51, 61])**.
(g) Set **p3g = lottery([42, 23, 34, 44, 51, 61])**.
(h) Set **p3h = lottery([12, 23, 34, 44, 51, 61])**.

Problem 4: Write a function **piecewise2d.m** to compute the following 2-dimensional function f(x,y):

$$f(x,y) = \begin{cases} 10x + 10y & for\, x \geq 0 \quad y > 0 \\ -10x + 10y & for\, x < 0 \quad y \geq 0 \\ -10x - 10y & for\, x \leq 0 \quad y < 0 \\ 10x - 10y & for\, x > 0 \quad y \leq 0 \end{cases}$$

Set **f** to be zero for other conditions of x and y. Function **piecewise2d.m** should have the function declaration: **function f = piecewise2d(x,y)** where **x** and **y** are number inputs and **f** is a number output. Remember to include a description for the function. Use **if elseif** statements.

(a) Set **p4a=evalc('help piecewise2d')**.
(b) Set **p4b=piecewise2d(1, 1)**.
(c) Set **p4c=piecewise2d(1, -1)**.
(d) Set **p4d=piecewise2d(-1, 1)**.
(e) Set **p4e=piecewise2d(-1, -1)**.
(f) Set **p4f=piecewise2d(0, 0)**.
(g) Set **p4g=piecewise2d(0, 1)**.
(h) Set **p4h=piecewise2d(0, -1)**.
(i) Set **p4i=piecewise2d(1, 0)**.
(j) Set **p4j=piecewise2d(-1, 0)**.

Problem 5: Write a function **rgb_color.m** to display color (red, green, blue, yellow, magenta, cyan, white) as a result of mixing primary colors (red, green, blue). The function

should have the following declaration: **function color = rgb_color(rgb)** where the input **rgb** is a 3-element vector having value of either 0 or 1 to denote the three primary colors (red, green, blue) respectively. The output **color** is a string denoting the color of the mixture.

See `http://en.wikipedia.org/wiki/Additive_color` for information on how colors are mixed. For example, when **rgb = [1 0 0]**, the resulting color is red. When **rgb = [1 1 1]**, the resulting color is white. If the input **rgb** is not a valid input, the output **color** should be the string **'Invalid input'**. Remember to give the function a description. Use **nested if** statements.

(a) Set **p5a=evalc('help rgb_color')**.
(b) Set **p5b=rgb_color([1 1 1])**.
(c) Set **p5c=rgb_color([1 0 0])**.
(d) Set **p5d=rgb_color([0 1 0)**.
(e) Set **p5e=rgb_color([0 0 1])**.
(f) Set **p5f=rgb_color([1 1 0])**.
(g) Set **p5g=rgb_color([0 1 1])**.
(h) Set **p5h=rgb_color([1 0 1])**.
(i) Set **p5i=rgb_color([0 0 0])**.