University of Louisiana at Lafayette
EECE 260, Fall 2019

Homework Assignment #1

Work each of the following problems in its own **individual script file**. Use comments to label which parts of your code correspond to each problem section (e.g., `%1a` or `%2c` placed above corresponding code). Each of these 10 problems is worth 10% of your grade for this assignment. Your individual script files should be named "YourName_HW1_Problem#.m" and your .zip file submission on Moodle should be named "YourName_HW1.zip".

Problem 1

Use MATLAB to make the following calculations using the values $x = 5$ and $y = 2$:

a) $u = x + y$
b) $v = 3xy$
c) $w = (4x) / y$
d) $z = \cos(x)$
e) $s = 0.3\sin(2y)$
f) $m = 2y^3 + x^4$

Problem 2

Use MATLAB to evaluate the following expressions:

a) $\frac{82}{7} + \frac{\pi}{120} + (5)(4.2^2)(358.9^{-1})$

b) $\sin\left(\sqrt{\frac{(45.2)(2)}{38} + \frac{1}{7.77}}\right)$

c) $(\frac{1.1}{4.23})(4 + 2.58)^5(\frac{4.9}{8} - \frac{0.3}{2.2*2.2})^{-0.5}$

Problem 3

Use MATLAB to calculate the following values **without** converting radians to degrees or vice versa:

a) $\sin(\pi / 4)$
b) $\cos(40°)$
c) $\sin^{-1}(0.32)$ (in radians)
d) $\cos^{-1}(0.71)$ (in degrees)

Problem 4

Use MATLAB to make the following calculations using $x = -5 + 6i$ and $y = 7 + 8i$:

a) $u = 2x + 3y$
b) $v = x^2$
c) $w = y / (x + y)$

University of Louisiana at Lafayette
EECE 260, Fall 2019

Problem 5
Use the `roots()` function to find the roots of the following polynomial:

$x^5 - 18x^4 + 95x^3 - 70x^2 - 456x + 448$

After displaying the roots, put them into the `poly()` function to find and display the coefficients of the original polynomial. The MathWorks website and the Function Browser contain information on the use of these functions.

Problem 6
Do the following:
a) Use two methods to create a vector *x* that has exactly 70 linearly spaced elements with values ranging between 0 and 15.
   Note: Typing the elements in by hand is not considered an option except in the case of small vectors (~5 or less elements)!
b) Use two methods to create a vector *y* with a linear spacing of exactly 0.25 between adjacent elements and that ranges from a minimum value of -2 to a maximum value of 20
c) Create a vector *z* with 43 logarithmically spaced values ranging from 1 to 1,000,000.
   Note: Pay attention to how the `logspace()` function works! Don't just type 100 and 100,000 as arguments; your results will be wrong!

Problem 7
Use MATLAB to plot the following functions over the interval $0 \leq x \leq 10$ on the same plot:

$u = 8\sin(x) + 1.5$
$v = 5\cos(2x) - 3$

Title the plot "u & v", label the x-axis as "x", and label the y-axis as "y". Make sure to also use the grid command. Generate *x* using `linspace()` such that there are **exactly** 101 elements in the vector, starting with 0 and ending at 10.

Problem 8

$$\begin{bmatrix} 2 & -2 & 4 & 3 \\ -5 & -9 & 10 & 8 \\ 12 & -13 & 5 & 1 \\ 5 & 5 & 7 & 10 \end{bmatrix}$$

Generate the array above (call it "*A*") in the following four ways:
a) Create a vector for each row of four elements and then concatenate these vectors vertically. Name these vectors "*a1*", "*b1*", "*c1*", and "*d1*".
b) Create a vector for each column of four elements and then concatenate these vectors horizontally. Name these vectors "*a2*", "*b2*", "*c2*", and "*d2*".
c) Create the full array using one line of code: the assignment statement for *A*.

University of Louisiana at Lafayette
EECE 260, Fall 2019

Problem 9

Regenerate the array *A* from Problem 8 using your preferred method. Do the following with array indexing:

a) Create a 4x2 array called "*B*" that consists of all elements in the third and fourth columns of *A*
b) Create a 3x4 array called "*C*" consisting of all elements in the first through third rows of *A*
c) Create a 1x1 called "*D*" consisting of the element in the top-left corner of *A* (meaning *D* = [2])

Problem 10

Given the following arrays:

$$A = \begin{bmatrix} 2 & 7 & 1 \\ 3 & 6 & 12 \\ 11 & -5 & 40 \\ 73 & -8 & 31 \end{bmatrix}$$

$$B = log_{10}(A) + 2$$

Write MATLAB expressions to do the following:

a) Select just the first column of *B*, calling this vector *b*
b) Evaluate the sum of elements in the fourth row of *A*
c) Create a matrix *C* by multiplying the first column of *B* and the last column of *A* **element-by-element**
d) Use the `min()` function to find the lowest value element in matrix *C*
e) Divide the third column of *A* by the last column of *B* **element-by-element**