# MATLAB Review Lab

# Seth Ricks

# ECEN 390

# Brother Swenson

# ECEN 390: MATLAB Review Lab (50 points)

(jas, ECEN 390 MATLAB Review Lab.docx, 9/21/2023)

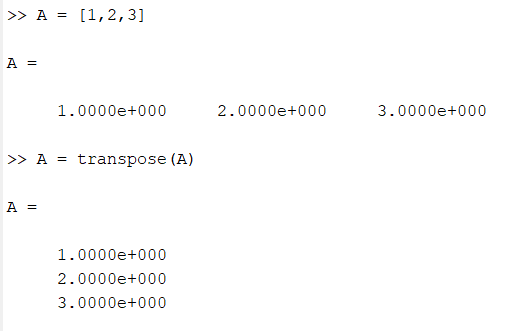
**Purpose:** The purpose of this lab is to familiarize yourself with using MATLAB, as it will be used throughout our study of Electromagnetics. The MATLAB help, along with the pdf document [MATLAB Basics and More](https://content.byui.edu/file/886a18ca-b656-47f6-a0ab-e340267fe0f5/1/Appendix_MATLAB.pdf) from the text “Introduction to Signals and Systems” by Mark A. Wickert, are both resources to be used to learn more about MATLAB.

**Note: While you are encouraged to work together and teach one another, each individual is to complete the following procedure. For your lab submission, add a cover page with your name, class and lab title along with your results included in the existing procedure, submitting the resulting document. A grading rubric is included at the end of this document.**

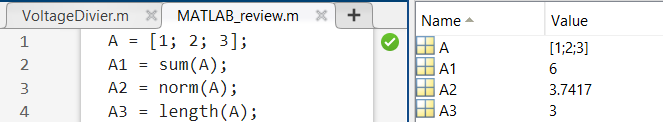
**Procedure:**

**Part 1 – Vector and Matrix Operations**

1. Enter the vector in the MATLAB Command Window. Then convert row vector A to a column vector by means of the MATLAB transpose operator. Include the MATLAB Command Window results below illustrating the original A matrix along with the transpose statement followed by the answer for the transpose of A. (3 points.)



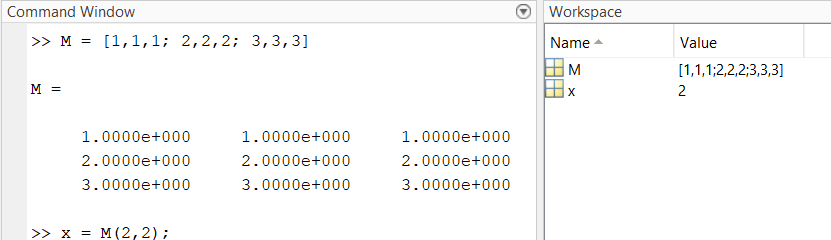
1. Apply the following MATLAB functions to the vector, and then briefly describe what each function returns for vector A: **sum**(A), **norm**(A), and **length**(A). Use the MATLAB help as needed. (3 points. 1 point each.)



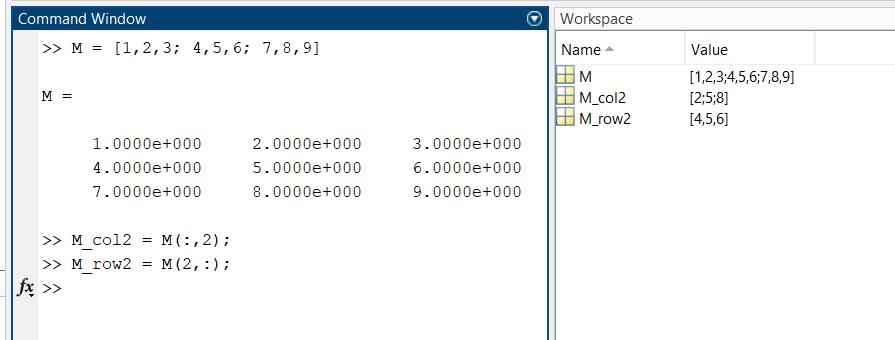
As expected, the sum of A is the sum of all the values added together, the norm is and the length is the length of the matrix.

1. Generate the following M matrix in the Command Window.

Utilizing MATLABs array indexing, extract the element of the 2nd row and 2nd column of the M matrix, including the MATLAB Command Window results below illustrating the original matrix M, along with the required indexing statement followed by the answer for the extracted results. See the MATLAB help on Array Indexing if needed. (3 points.)



1. The following M matrix can be viewed as being composed of three row vectors or three column vectors. The MATLAB colon operator can be used to extract individual rows or columns of an array or matrix. Using array indexing utilizing the colon operator, extract just the middle row as M\_row2, followed by extracting the middle column as M\_col2 of the original M matrix. Include the MATLAB Command Window results below illustrating the original matrix M, along with the required array indexing statements for M\_row2 and M\_col2, followed by the answers for the extracted results. See the MATLAB help on the colon operator if needed. (5 points.)



**Part 2 – Array (Element-Wise) versus Matrix Operations**

Single-element variables in MATLAB are termed scalars. Mathematical operations involving scalars are straight forward with the symbol \* denoting multiplication, / denoting division and ^ denoting exponentiation or raising to a power. The same \*, /, and ^ operators also apply to multi-element variables multiplied, divided, or raised to a power by a scalar.

Multi-element variables in MATLAB can be considered matrices, vectors, or arrays.

For example, a variable consisting of m rows by n columns can either be considered a matrix or an array, as MATLAB makes no distinction between the two. However, there is a distinction between the arithmetic operations performed on matrices versus arrays. For example, matrix multiplication is fundamentally different than element-by-element multiplication between elements of an array. The following Array versus Matrix Operations description comes from the MATLAB help:

“MATLAB has two different types of arithmetic operations: array operations and matrix operations. You can use these arithmetic operations to perform numeric computations, for example, adding two numbers, raising the elements of an array to a given power, or multiplying two matrices.

Matrix operations follow the rules of linear algebra. By contrast, array operations execute element by element and support multidimensional arrays. The period character (.) distinguishes the array operations from the matrix operations. However, since the matrix and array operations are the same for addition and subtraction, the character pairs .+ and .- are unnecessary”.

So, in MATLAB the \* symbol between multi-element variables invokes matrix multiplication, whereas the two symbols .\* invoke element-wise multiplication. The same applies to exponentiation, i.e., raising to a power, with ^ denoting matrix exponentiation, and .^ denoting element-wise exponentiation. Matrix exponentiation involves matrix multiplication, with A^2 signifying the matrix A multiplied by itself. Matrix multiplication requires that the number of rows of the first matrix equals the number of columns of the second matrix, with MATLAB returning the error message: “Incorrect dimensions for matrix multiplication.”, when this requirement is violated.

In MATLAB, the / symbol when placed between scalars or a multi-dimensional variable to be divided by a scalar has been defined to invoke element-wise division. However, matrix division is not defined, rather matrix algebra utilizes the inverse matrix as described in **Part 1** of this lab. So, in MATLAB, the / symbol has been uniquely defined to solve a system of linear equations when placed between two multi-dimensional variables. Consequently, the / symbol should not be used between multi-dimensional variables, unless intending to solve a system of equations. Rather the ./ symbol pair should be used between multi-dimensional variables to denote element-by-element division.

In summary, a common error in MATLAB is to invoke matrix operations when element- wise operations are intended. So, in MATLAB, except when scalars are involved, use .\*, ./ and .^ to invoke element-wise multiplication, division, and exponentiation, respectively.

1. Complete **Table 1** below by adding the short description found in the MATLAB help for each of the functions listed. It is OK to use the MATLAB descriptions in this case, rather than using your own wording. (6 points total.)

**Table 1**: Primary MATLAB Multiplication, Division and Exponentiation Functions.

|  |  |
| --- | --- |
| **Function** | **Description** |
| \* or **mtimes**() | [C](https://www.mathworks.com/help/releases/R2022b/matlab/ref/mtimes.html?searchHighlight=mtimes&s_tid=doc_srchtitle#btx9i74-C) = [A](https://www.mathworks.com/help/releases/R2022b/matlab/ref/mtimes.html?searchHighlight=mtimes&s_tid=doc_srchtitle#btx9i74-A)\*[B](https://www.mathworks.com/help/releases/R2022b/matlab/ref/mtimes.html?searchHighlight=mtimes&s_tid=doc_srchtitle#btx9i74-A) (or mtimes())is the matrix product of A and B. |
| ^ or **mpower**() | C = [A](https://www.mathworks.com/help/releases/R2022b/matlab/ref/mpower.html?searchHighlight=mpower&s_tid=doc_srchtitle#btx__27-1-A)^[B](https://www.mathworks.com/help/releases/R2022b/matlab/ref/mpower.html?searchHighlight=mpower&s_tid=doc_srchtitle#btx__27-1-A) computes A to the B power and returns the result in C. |
| / or **mrdivide**() | [x](https://www.mathworks.com/help/releases/R2022b/matlab/ref/mrdivide.html?searchHighlight=mrdivide&s_tid=doc_srchtitle#btg5p6j-x) = [B](https://www.mathworks.com/help/releases/R2022b/matlab/ref/mrdivide.html?searchHighlight=mrdivide&s_tid=doc_srchtitle#btg5p6j-A)/[A](https://www.mathworks.com/help/releases/R2022b/matlab/ref/mrdivide.html?searchHighlight=mrdivide&s_tid=doc_srchtitle#btg5p6j-A) solves the system of linear equations x\*A = B for x. |
| .\* or **times**() | C = [A](https://www.mathworks.com/help/releases/R2022b/matlab/ref/times.html?searchHighlight=times&s_tid=doc_srchtitle#btx_6a1-A).\*[B](https://www.mathworks.com/help/releases/R2022b/matlab/ref/times.html?searchHighlight=times&s_tid=doc_srchtitle#btx_6a1-A) multiplies arrays A and B by multiplying corresponding elements. |
| .^ or **power**() | C = [A](https://www.mathworks.com/help/releases/R2022b/matlab/ref/power.html?searchHighlight=power&s_tid=doc_srchtitle#btx_7d7-A).^[B](https://www.mathworks.com/help/releases/R2022b/matlab/ref/power.html?searchHighlight=power&s_tid=doc_srchtitle#btx_7d7-A) raises each element of A to the corresponding powers in B. |
| ./ or **rdivide**() | x = [A](https://www.mathworks.com/help/releases/R2022b/matlab/ref/rdivide.html?searchHighlight=rdivide&s_tid=doc_srchtitle#btg5tj1-1-A)./[B](https://www.mathworks.com/help/releases/R2022b/matlab/ref/rdivide.html?searchHighlight=rdivide&s_tid=doc_srchtitle#btg5tj1-1-A) divides each element of A by the corresponding element of B. |

1. Include below the necessary MATLAB statement to implement the following equation for F with theta being a 1 X 5 array. (3 points.)

theta = [1,2,3,4,5]; %example 1x5 matrix

F = cos(theta) ./ (sin(theta) + 2);

As previously mentioned, C = A\*B denotes the matrix multiplication of A and B. If A is an m by p (row by column) matrix and B is p by n matrix, then C is an m by n matrix (rows of A by columns of B) defined as follows:

For matrix multiplication to work, the number of columns of A must equal the number of rows of B, with p equal to the number of columns of A or the number of rows of B. For example, if Matrix A is a 2 X 3 matrix and Matrix B is a 3 X 2 matrix, then p equals 3, and the element C(1, 1) equals the three elements of the first row of A, multiplied by and summed together with the three elements of the 1st column of B, i.e., C(1,1) = a11b11 + a12b21 + a13b31.

1. In MATLAB, matrix multiplication is denoted by \* and the raising of a matrix by a power as ^. Using the above matrix multiplication formula manually, rather than with MATLAB, include below the resulting matrix expressions for C1 = A\*B, and C2 = A^2 = A\*A in terms of the associated aik and bkj elements of the matrices A and B. (Note: Each manually arrived at matrix should have four elements consisting of the appropriate combination of the terms a11, a12, a21, a22, b11, b12, b21 and b22.)

C1 = A\*B .

C2 = A^2

An element-wise operation simply involves an element-by-element operation of one array upon the corresponding elements of another array. For example, the element wise multiplication of a 2 by 2 array by another 2 by 2 array simply multiplies the elements of the first array by the equivalently indexed elements in the second array. Element wise operations are denoted by .\* for multiplication and .^ for raising to a power in MATLAB. If one of the arrays is multi-dimensional and the other is a scalar, then using either the \*, or .\* operators result in all elements of the multi-dimensional array being multiplied by the scalar. If both arrays have multiple elements, then the two arrays must be of the same size to perform element-wise operations.

1. Include below the resulting array expressions for C3 = A.\*B and C4 = A.^2 in terms of the associated aik and bkj elements of the arrays A and B manually, rather than with MATLAB. The resulting C3 and C4 matrixes from element wise multiplication should be different than the C1 and C2 matrixes arrived at from matrix multiplication. (4 points. 2 points each.)

C3 = A.\*B .

C4 = A.^2

**Part 3 – Complex Numbers and Plotting**

1. Using MATLAB, generate an m-file to produce a plot of | ej2πft |, Re{ej2πft}, and Imaginary{ej2πft}, for f = 1 Hz, with units of volts, along with a unique color for each trace like the plot shown below. Use the command t = 0:0.01:2; to generate a vector of time values ranging from 0 to 2 seconds. Include a title, using the **title**(‘Your Name’) command, a **legend**() command, an x-axis label using the **xlabel**() command and a y-axis label using the **ylabel**() command, along with the **ylim**() command to plot the amplitude in volts ranging from -1.1 to 1.1. (See MATLAB help on “Exponential”, “Absolute Value and complex magnitude”, “Real”, and “Imaginary”, along with how to include “Greek Letter and Special Characters in Chart Text” for help on superscripts and Greek characters to include in your plot.)
2. Run your m-file file to generate a graph of the magnitude, real, and imaginary parts of , for f = 1 Hz, like the one below. Replace the figure below with your version. (8 points.)

A graph of a function

AI-generated content may be incorrect.

Figure : Overlay Plot of the Magnitude, Real, and Imaginary parts of e^(j2pi(t))

**Part 4 – 3-D Plotting**

Electromagnetic (EM) waves propagating in space are Transverse Electromagnetic Waves (TEM), meaning that the Electric Field Intensity, Magnetic Field Intensity and direction of propagation are all transverse, i.e., at right angles to each other. A 3-dimensional plot is necessary to visualize a traveling EM wave, with a lossless traveling EM wave illustrated below in **Figure 2**.

Using MATLAB, and the **plot3**() function, generate an m-file to produce a 3-dimensional plot of EM wave propagation associated with a sinusoidal electric and magnetic field intensity.

1. First generate a y vector corresponding to length ranging from 0 to 15.7 m, with a 0.1 m step size. Next generate the vector z1 = sin(y), corresponding to the electric field intensity. Then generate an x1 vector consisting of all zeros having the same length as y and Z1.
2. Next generate the vector x2 = (1/377)\*sin(y), corresponding to the magnetic field intensity, where a lossless EM wave propagating in free space has magnetic field intensity magnitude that is 377 times less than the magnitude of the electric field intensity. Finally generate a z2 vector equal to all zeros like the x1 vector previously generated.
3. Then produce a 3-D plot of the above generated vectors using the following commands:

plot3(x1, y, z1, 'ro', x2, y, z2, 'b+'), grid on;

hold on

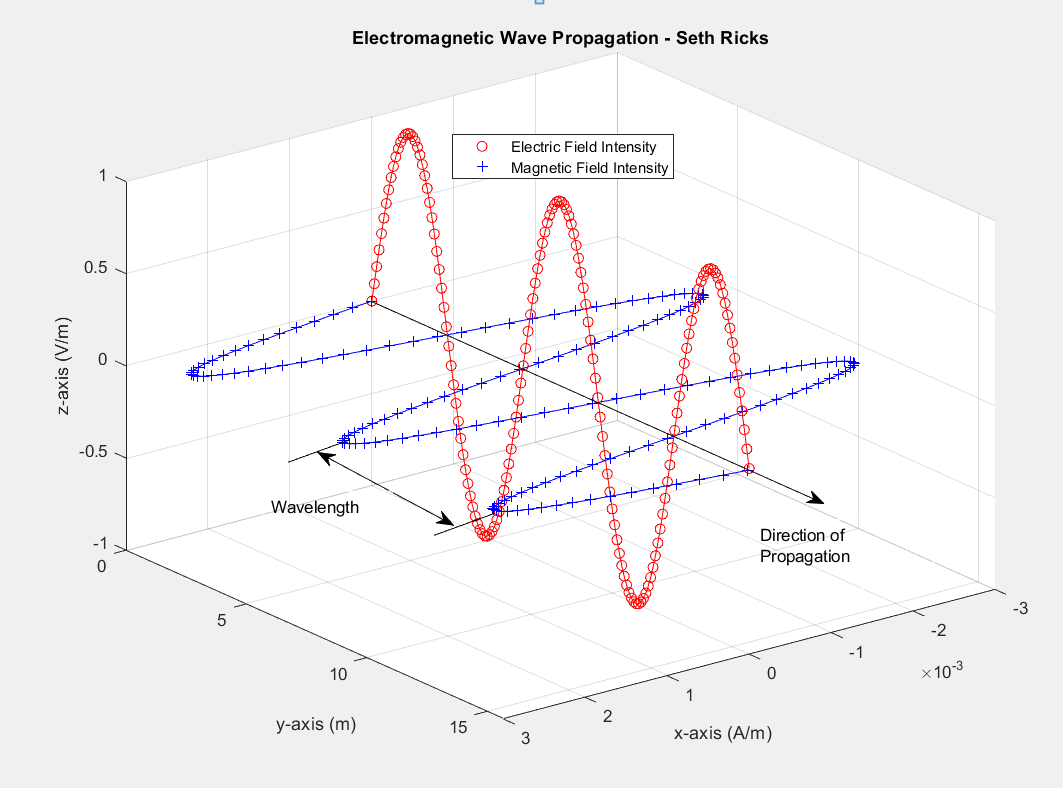
plot3(x1, y, z1, '-r', x2, y, z2, '-b'), grid on;

hold off

set(gca, 'XDir', 'reverse', 'YDir', 'reverse');

The first **plot3**() function plots individual o and + markers at each data point, whereas the second **plot3**() function generates a line plot. The hold on command retains both plots, resulting in a line plot with data point markers. The **set**(gca, …) function reverses the default direction of the x and y-axis plots, which is helpful in this case. This command must be placed after the occurrence of a **plot**() or **plot3**() function, so that there are axis definitions to reverse.

1. For your 3-D plot, include an x-axis label of “x-axis (A/m)”, a y-axis label of “y-axis (m)”, a z-axis label of “z-axis (V/m)”, a title of “Electromagnetic Wave Propagation – Your Name”, along with a legend denoting “Electric Field Intensity” and Magnetic Field Intensity”, as shown below in **Figure 2**.
2. After running the above **plot3**() functions, a 3-D plot window opens in which some additional adjustments and documentation to be added. First move the legend onto the 3-D plot surface. Next, from the pull-down menus in the upper right-hand corner of the plot window, place an arrow at the start at the point (0, 0, 0), parallel to the y-axis with the arrow pointing in the direction of increasing distance along the y-axis, by means of **Insert** 🡪 **Arrow**. Also include a text box, by means of **Insert** 🡪 **TextBox**, including the label “Direction of Propagation” near the end of the placed arrow. For the text box, choose an edge color of clear to remove the box, as illustrated in the **Figure** below. Then from the **Insert** pull-down menu, add a **Line**, **Double Arrow**, and **TextBox** including the word “Wavelength” indicating the wavelength of the traveling EM wave, as shown below. Replace the figure below with your version. (10 points.)



**Figure 2**: Transverse Electromagnetic Wave Propagation.

**MATLAB Review Lab Grading Rubric:** For your lab submission, add a cover page with your name, class and lab title along your results included in the existing procedure, submitting the resulting document. Points for each part of the lab are summarized in the following table.

|  |  |
| --- | --- |
| **Lab Items** | **Points** |
| Cover Page | 1 |
| Part 1 – Vector and Matrix Operations | 14 |
| Part 2 – Array (Element-Wise) versus Matrix Operations | 17 |
| Part 3 – Complex Numbers and Plotting  **Figure 1.** (8 points total. 1 point per trace (3 points total), (-0.5 points per trace not having a unique color), 1 point for title, 1 point for x-axis label (0.5 points for units), 1 point for y-axis label (0.5 points for units), 2 points for legend. -1 point for not using superscripts in legend.) | 8 |
| Part 4 – 3-D Plotting  **Figure 2.** (10 points total. 1 point per correct looking trace (2 points total), 1 point for title, 1 point for x-axis label (0.5 points for units), 1 point for y-axis label (0.5 points for units), 2 points for the added lines, double arrow and text box indicating wavelength, 1 point for legend, 1 point for added direction arrow, 1 point for added “Direction of Propagation” text box.) | 10 |
|  |  |
| **Total** | 50 |

Please give feedback on errors you find in this document.