**NIRMA UNIVERSITY**

**Institute of Technology**

**Department of Electronics & Communication Engineering**

**B. Tech. SEM. V**

**2ECOE76 MATLAB for Engineers**

**Laboratory Practical Manual**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Title** | **Date** | **Grade** | **Sign** |
|  | To work with MATLAB operators, expressions, variables and library functions. |  |  |  |
|  | To perform scalar and array operations on MATLAB variables, arrays, multi-dimensional arrays, sub-arrays. |  |  |  |
|  | To work with branching statements. |  |  |  |
|  | To work with MATLAB loops, logical arrays and vectorization. |  |  |  |
|  | To work with MATLAB user defined functions. |  |  |  |
|  | To work with additional data types in MATLAB. |  |  |  |
|  | To work with MATLAB strings and additional data plots. |  |  |  |
|  | To work with sparse arrays, cell arrays and structures in MATLAB. |  |  |  |
|  | To work with input/output functions in MATLAB. |  |  |  |
|  | To generate Graphical User Interfaces (GUI) in MATLAB. |  |  |  |

**Date:**

**Experiment 1**

**Aim:** To work with MATLAB operators, expressions, variables and library functions.

**Apparatus:** MATLAB Software

**Objective:**

1. To learn the basics of MATLAB software.
2. To be aware of MATLAB operators like addition, subtraction, multiplication, division etc.
3. To learn how to write expressions in MATLAB.
4. To learn different types of variables and how to work with.
5. To know different inbuilt library functions and their syntax.

**Theory:**

You are required to write the following things over here:

* Steps to open the MATLAB software
* About different windows in MATLAB like command window, command history, workspace, current folder, editor window.

1. **Operators:**
2. **Addition ‘+’ :**

**Function of this operator with example and screenshot attached**

1. **Subtraction ‘-‘ :**

**Function of this operator with example and screenshot attached**

1. **Multiplication ‘\*’ :**

**Function of this operator with example and screenshot attached**

1. **Division ‘/’ :**

**Function of this operator with example and screenshot attached**

1. **Exponentiation ‘^’ :**

**Function of this operator with example and screenshot attached**

1. **Basic Commands**

**Include all the basic commands we learnt, their descriptions, their examples and screen shots.**

**Quiz:**

* + 1. What is the purpose of the MATLAB Command Window? The Edit Window? The Figure Window?
    2. List the different ways that you get help in MATLAB.
    3. What is a workspace? How can you determine what is stored in a MATLAB workspace?
    4. How can you clear the contents of a workspace?
    5. The distance traveled by a ball falling in the air is given by the equation

Use MATLAB to calculate the position of the ball at time *t* =5 s if

* + 1. Suppose that *x=3* and *y=*4. Use MATLAB to evaluate the following expression:

**Conclusion:**

**Date:**

**Experiment 2**

**Aim:** To perform scalar and array operations on MATLAB variables, arrays, multi-dimensional arrays, sub-arrays.

**Apparatus:** MATLAB Software

**Objective:**

1. To learn how different library functions can be used with arrays and multi-dimensional arrays.
2. To learn how specific elements of an array can be accessed and utilized in coding.

**Problems:**

**Q-1.** Answer the following questions for the array shown here.



(*a*) Determine the size of an array1.

(*b*) Determine the value of an array1(1,4).

(*c*) Determine the size and value of an array1(:,1:2:5).

(*d*) Determine the size and value of an array1([1 3],end).

**MATLAB Program:**

**Here**

**Output:**

**Here (Screenshot or copy paste from command window)**

**The same is applicable to below questions**

**Q-2.** Are the following MATLAB variable names legal or illegal? Why?

(*a*) dog1

(*b*) 1dog

(*c*) Do\_you\_know\_the\_way\_to\_san\_jose

(*d*) \_help

(*e*) What's\_up?

**Q-3.** Determine the size and contents of the following arrays. Note that the later arrays may depend on the definitions of arrays defined earlier in this exercise.

(*a*) a = 2:3:8;

(*b*) b = [a' a' a'];

(*c*) c = b(1:2:3,1:2:3);

(*d*) d = a + b(2,:);

(*e*) w = [zeros(1,3) ones(3,1)' 3:5'];

(*f* ) b([1 3],2) = b([3 1],2);

(*g*) e = 1:-1:5;

**Q-4.** Assume that a, b, c, and d are defined as follows, and calculate the results of the following operations in MATLAB if they are legal. If an operation is illegal, explain why it is illegal.



(*a*) result = a + b;

(*b*) result = a \* d;

(*c*) result = a .\* d;

(*d*) result = a \* c;

(*e*) result = a .\* c;

(*f* ) result = a \ b;

(*g*) result = a .\ b;

(*h*) result = a .^ b;

**Q-5.** Evaluate each of the following expressions in MATLAB:

(*a*) 11 / 5 + 6

(*b*) (11 / 5) + 6

(*c*) 11 / (5 + 6)

(*d*) 3 ^ 2 ^ 3

(*e*) 3 ^ (2 ^ 3)

(*f* ) (3 ^ 2) ^ 3

(*g*) round(-11/5) + 6

(*h*) ceil(-11/5) + 6

(*i*) floor(-11/5) + 6

**Q-6.** Solve the following system of simultaneous equations for *x*:

-2.0 x1 + 5.0 x2 + 1.0 x3 + 3.0 x4 + 4.0 x5 - 1.0 x6 = 0.0

2.0 x1 - 1.0 x2 - 5.0 x3 - 2.0 x4 + 6.0 x5 + 4.0 x6 = 1.0

-1.0 x1 + 6.0 x2 - 4.0 x3 - 5.0 x4 + 3.0 x5 - 1.0 x6 = -6.0

4.0 x1 + 3.0 x2 - 6.0 x3 - 5.0 x4 - 2.0 x5 - 2.0 x6 = 10.0

-3.0 x1 + 6.0 x2 + 4.0 x3 + 2.0 x4 - 6.0 x5 + 4.0 x6 = -6.0

2.0 x1 + 4.0 x2 + 4.0 x3 + 4.0 x4 + 5.0 x5 - 4.0 x6 = -2.0

**Q-7.** The distance between two points (*x*1, *y*1) and (*x*2, *y*2) on a Cartesian coordinate plane is given by the equation,



Write a program to calculate the distance between any two points (*x*1, *y*1) and (*x*2, *y*2) specified by the user. Use good programming practices in your program. Use the program to calculate the distance between the points (-3, 2) and (3, -6).



**Conclusion:**

**Post-Lab Problems**

**Q-1.** Engineers often measure the ratio of two power measurements in *decibels*, or dB. The equation for the ratio of two power measurements in decibels is



where *P*2 is the power level being measured, and *P*1 is some reference power level.

(*a*) Assume that the reference power level *P*1 is 1 milliwatt, and write a program that accepts an input power *P*2 and converts it into dB with respect to the 1 mW reference level. (Engineers have a special unit for dB power levels with respect to a 1 mW reference: dBm.) Use good programming practices in your program.

(*b*) Write a program that creates a plot of power in watts versus power in dBm with respect to a 1 mW reference level. Create both a linear *xy* plot and a log-linear *xy* plot.

**Q-2.** The voltage across a resistor is related to the current flowing through it by Ohm’s law



and the power consumed in the resistor is given by the equation



Write a program that creates a plot of the power consumed by a 1000 Ω resistor as the voltage across it is varied from 1 V to 200 V. Create two plots, one showing power in watts, and one showing power in dBW (dB power levels with respect to a 1 W reference).



**Q-3.** The force required to compress a linear spring is given by the equation



where *F* is the force in newtons and *k* is the spring constant in newtons per meter. The potential energy stored in the compressed spring is given by the equation



where *E* is the energy in joules. The following information is available for four springs:



Determine the compression of each spring, and the potential energy stored in each spring. Which spring has the most energy stored in it?

**Q-4.** A simplified version of the front end of an AM radio receiver is shown in below figure. This receiver consists of an *RLC* tuned circuit containing a resistor, capacitor, and an inductor connected in series. The *RLC* circuit is connected to an external antenna and ground as shown in the picture. The tuned circuit allows the radio to select a specific station out of all the stations transmitting on the AM band. At the resonant frequency of the circuit, essentially all of the *V0* signal appearing at the antenna appears across the resistor, which represents the rest of the radio. In other words, the radio receives its strongest signal at the resonant frequency. The resonant frequency of the LC circuit is given by the equation



where *L* is inductance in henrys (H) and *C* is capacitance in farads (F). Write a program that calculates the resonant frequency of this radio set

given specific values of *L* and *C*. Test your program by calculating the frequency of the radio when *L* = 0.25 mH and *C =* 0.10 nF.



**Q-5.** The average (rms) voltage across the resistive load in above same figure varies as a function of frequency according to Equation



where *w=2пf* and *ƒ* is the frequency in hertz. Assume that *L* = 0.25 mH, *C* = 0.10 nF, R=50Ω, and *V0=*mV.

(*a*) Plot the rms voltage on the resistive load as a function of frequency. At what frequency does the voltage on the resistive load peak? What is the voltage on the load at this frequency? This frequency is called the resonant frequency ƒ*0* of the circuit.

(*b*) If the frequency is changed to 10% greater than the resonant frequency, what is the voltage on the load? How selective is this radio receiver?

(*c*) At what frequencies will the voltage on the load drop to half of the voltage at the resonant frequency?

**Q-6.** An object moving in a circular path at a constant tangential velocity *v* is shown in below figure. The radial acceleration required for the object to move in the circular path is given by the Equation



where *a* is the centripetal acceleration of the object in m/s2, *v* is the tangential velocity of the object in m/s, and *r* is the turning radius in meters. Suppose that the object is an aircraft, and answer the following questions about it:

(*a*) Suppose that the aircraft is moving at Mach 0.85, or 85% of the speed of sound. If the centripetal acceleration is 2 g, what is the turning radius of the aircraft? (*Note*: For this problem, you may assume that Mach 1 is equal to 340 m/s, and that 1 g=9.81 m/s2.)

(*b*) Suppose that the speed of the aircraft increases to Mach 1.5. What is the turning radius of the aircraft now?

(*c*) Plot the turning radius as a function of aircraft speed for speeds between Mach 0.5 and Mach 2.0, assuming that the acceleration remains 2 g.

(*d*) Suppose that the maximum acceleration that the pilot can stand is 7 g. What is the minimum possible turning radius of the aircraft at Mach 1.5?

(*e*) Plot the turning radius as a function of centripetal acceleration for accelerations between 2 g and 8 g, assuming a constant speed of Mach 0.85.



**Conclusion:**

**Date:**

**Experiment 3**

**Aim:** To work with branching statements.

**Apparatus:** MATLAB Software

**Objective:** To learn how to control the order in which statements are executed using if-else and switch-case.

**Problems:**

**Q-1.** The following statements are intended to alert a user to dangerously high

oral thermometer readings (values are in degrees Fahrenheit). Are they correct

or incorrect? If they are incorrect, explain why and correct them.

if temp < 97.5

disp('Temperature below normal');

elseif temp > 97.5

disp('Temperature normal');

elseif temp > 99.5

disp('Temperature slightly high');

elseif temp > 103.0

disp('Temperature dangerously high');

end

**Q-2.** Write a MATLAB program to evaluate the function:

for any user-specified value of *x*, where *x* is a number <1.0 (note that ln is the natural logarithm, the logarithm to the base *e*). Use an if structure to verify that the value passed to the program is legal. If the value of *x* is legal, calculate *y*(*x*). If not, write a suitable error message and quit.

**Q-3.** Examine the following MATLAB statements. Are they correct or incorrect?

If they are correct, what do they output? If they are incorrect, what

is wrong with them?

1. if volts > 125

disp('WARNING: High voltage on line.');

if volts < 105

disp('WARNING: Low voltage on line.');

else

disp('Line voltage is within tolerances.');

end

1. color = 'yellow';

switch ( color )

case 'red',

disp('Stop now!');

case 'yellow',

disp('Prepare to stop.');

case 'green',

disp('Proceed through intersection.');

otherwise,

disp('Illegal color encountered.');

end

1. if temperature > 37

disp('Human body temperature exceeded.');

elseif temperature > 100

disp('Boiling point of water exceeded.');

end

**Q-4** Write a program that allows a user to enter a string containing a day of the week (“Sunday,” “Monday,” “Tuesday,” etc.) and uses a switch construct to convert the day to its corresponding number, where Sunday is considered the first day of the week and Saturday is considered the last day of the week. Print out the resulting day number. Also, be sure to handle the case of an illegal day name! (*Note:* Be sure to use the 's' option on function input so that the input is treated as a string.)

**Q-5.** The gain *G* of a certain microwave dish antenna can be expressed as a function of angle by the equation



where θis measured in radians from the boresite of the dish, and sinc *x*=sin *x*/*x*. Plot this gain function on a polar plot, with the title “**Antenna** **Gain vs**” in boldface.

**Q-6.** The author of this book now lives in Australia. Australia is a great place to live, but it is also a land of high taxes. In 2002, individual citizens and residents of Australia paid the following income taxes:



In addition, a flat 1.5% Medicare levy is charged on all income. Write a program to calculate how much income tax a person will owe based on this information. The program should accept a total income figure from the user and calculate the income tax, Medicare levy, and total tax payable by the individual.

**Q-7.** Assume that the complex function *f* (*t*) is defined by the equation



**Q-8.** Below figure shows a simple high-pass filter consisting of a resistor and a capacitor. The ratio of the output voltage *Vo* to the input voltage *Vi* is given by the equation



Assume that *R* =16 kΩ and *C* =1µF. Calculate and plot the amplitude and phase response of this filter as a function of frequency.

**Q-9.** When a satellite orbits the Earth, the satellite’s orbit will form an ellipse with the Earth located at one of the focal points of the ellipse. The satellite’s orbit can be expressed in polar coordinates as



where *r* and θ are the distance and angle of the satellite from the center of the Earth, *p* is a parameter specifying the size of the orbit, and ε is a parameter representing the eccentricity of the orbit. A circular orbit has an eccentricity ε of 0. An elliptical orbit has an eccentricity of 0<= ε<=1. If ε >1, the satellite follows a hyperbolic path and escapes from the Earth’s gravitational field.

Consider a satellite with a size parameter *p* =1000 km. Plot the orbit

of this satellite if (*a*) ε =0; (*b*) ε =0.25; (*c*) ε =0.5. How close does each orbit come to the Earth? How far away does each orbit get from the Earth? Compare the three plots you created. Can you determine what the parameter *p* means from looking at the plots?

**Conclusion:**

**Date:**

**Experiment 4**

**Aim:** To work with MATLAB loops, logical arrays and vectorization.

**Apparatus:** MATLAB Software

**Objective:**

1. To learn how to execute a sequence of statements more than once using different loops like for loop, while loop.
2. To learn the different applications of logical arrays.
3. To learn the benefits of vectorization by comparing the same logical code using loop and vectorization.

**Problems:**

**Q-1.** Write an M-file to evaluate the equation y(x)=x2-3x+2 for all values of *x* between -1 and 3, in steps of 0.1. Do this twice, once with a for loop and once with vectors. Plot the resulting function using a 3-pointthick dashed red line.

**Q-2.** Examine the following for statements and determine how many times each loop will be executed in MATLAB.

(*a*) for ii = -32768:32767

(*b*) for ii = 32768:32767

(*c*) for kk = 2:4:3

(*d*) for jj = ones(5,5)

**Q-3.** Examine the following for loops and determine the value of ires at the end of each of the loops, and also the number of times each loop executes.

(*a*) ires = 0;

for index = -10:10

ires = ires + 1;

end

(*b*) ires = 0;

for index = 10:-2:4

if index == 6

continue;

end

ires = ires + index;

end

(*c*) ires = 0;

for index = 10:-2:4

if index == 6

break;

end

ires = ires + index;

end

(*d*) ires = 0;

for index1 = 10:-2:4

for index2 = 2:2:index1

if index2 == 6

break

end

ires = ires + index2;

end

end

**Q-4.** Examine the following while loops and determine the value of ires at the end of each of the loops and the number of times each loop executes.

(*a*) ires = 1;

while mod(ires,10) ~= 0

ires = ires + 1;

end

(*b*) ires = 2;

while ires <= 200

ires = ires^2;

end

(*c*) ires = 2;

while ires > 200

ires = ires^2;

end

**Q-5.** What is contained in array arr1 after each of the following sets of statements have been executed in MATLAB?

(*a*) arr1 = [1 2 3 4; 5 6 7 8; 9 10 11 12];

mask = mod(arr1,2) == 0;

arr1(mask) = -arr1(mask);

(*b*) arr1 = [1 2 3 4; 5 6 7 8; 9 10 11 12];

arr2 = arr1 <= 5;

arr1(arr2) = 0;

arr1(~arr2) = arr1(~arr2).^2;

**Q-6.** Write a MATLAB program to evaluate the function

for any user-specified value of *x*, where ln is the natural logarithm (logarithm to the base *e*). Write the program with a while loop, so that the program repeats the calculation for each legal value of *x* entered into the program. When an illegal value of *x* is entered, terminate the program. (Any x>=1 is considered an illegal value.)

**Q-7.** The *n*th Fibonacci number is defined by the following recursive equations:

f(1)=1;

f(2)=2;

f(n)=f(n-1)+f(n-2)

Therefore, f(3)=f(2)+f(1)=2+1=3, and so forth for higher numbers. Write an M-file to calculate and write out the *n*th Fibonacci number for n>2, where *n* is input by the user. Use a while loop to perform the calculation.

**Q-8.** The current flowing through the semiconductor diode shown in Figure 4.7 is given by the equation

Where, voltage across the diode, in volts

current flow through the diode, in amps

leakage current of the diode, in amps

q = charge on an electron, 1.602 x 10-19 coulombs

k = Boltzmann’s constant, 1.38 x 10-23 joule/K

T = temperature, in kelvins (K)

The leakage current of the diode is 2.0 µA. Write a program to calculate the current flowing through this diode for all voltages from -1.0 V to +0.6 V, in 0.1 V steps. Repeat this process for the following temperatures: 75°F and 100°F, and 125°F. Create a plot of the current as a function of applied voltage, with the curves for the three different temperatures appearing as different colors.

**Q-9.** Engineers often measure the ratio of two power measurements in

*decibels*, or dB. The equation for the ratio of two power measurements

in decibels is

where is the power level being measured and is some reference power level. Assume that the reference power level is 1 watt, and write a program that calculates the decibel level corresponding to power levels between 1 and 20 watts, in 0.5 W steps. Plot the dB-versus-power curve on a log-linear scale.

**Conclusion:**

**Date:**

**Experiment 5**

**Aim:** To work with MATLAB user defined functions.

**Apparatus:** MATLAB Software

**Objective:** To learn how to develop and debug sub-tasks independently before building the final program.

**Problems:**

**Q-1.** Modify function random0(Solved example 5.4) so that it can accept 0, 1, or 2 calling arguments. If it has no calling arguments, it should return a single random value. If it has 1 or 2 calling arguments, it should behave as it currently does.

**Q-2.** Write a function that uses function random0 to generate a random value in the range [-1.0,1.0). Make random0 a subfunction of your new function.

**Q-3.** It is often useful to be able to simulate the throw of a fair die. Write a MATLAB function dice that simulates the throw of a fair die by returning some random integer between 1 and 6 every time that it is called. (*Hint:* Call random0 to generate a random number. Divide the possible values out of random0 into six equal intervals and return the number of the interval that a given random value falls into.)

**Q-4.** A function is said to be recursiveif the function calls itself. MATLAB functions are designed to allowrecursive operation. To test this feature, write a MATLAB function toevaluate the factorial function, which is defined as follows:

**

where *N* is a positive integer. The function should check to make sure that there is a single argument *N*, and that *N* is a nonnegative integer. If it is not, generate an error using the error function. If the input argument is a nonnegative integer, the function should evaluate *N*! using Equation.

**Q-5.** Antiship missiles (ASMs) are sometimes used to attack naval ships, and the ships being attacked use surface-to-air missiles (SAMs) to try to kill the attacking missiles before they hit the ship. Answer the following questions about this situation using the function developed in previous exercise.

(*a*) If the ship uses three SAMs to attack an incoming ASM and the probability of success of each attack is 0.3, what is the probability that the ship will destroy the ASM before it is hit?

(*b*) If the ship uses three SAMs to attack an incoming ASM and the probability of success of each attack is 0.5, what is the probability that the ship will destroy the ASM before it is hit?

(*c*) If the ship uses three SAMs to attack an incoming ASM and the probability of success of each attack is 0.7, what is the probability that the ship will destroy the ASM before it is hit?

(*d*) If the ship uses three SAMs to attack an incoming ASM and the probability of success of each attack is 0.9, what is the probability that the ship will destroy the ASM before it is hit?

**Q-6.** Suppose a designer wanted to ensure that there is a 90% probability that the ship will survive an ASM attack without being hit. How many SAMs should the ship fire to destroy the incoming ASM if the individual probability of success of for a SAM is (*a*) 30%? (*b*) 50%? (*c*) 70%?

**Q-7.** Write three MATLAB functions to calculate the hyperbolic sine, cosine, and tangent functions:



Use your functions to plot the shapes of the hyperbolic sine, cosine, and

tangent functions.

**Q-8.** Write a single MATLAB function hyperbolic to calculate the hyperbolic sine, cosine, and tangent functions as defined in the previous problem. The function should have two arguments. The first argument will be a string containing the function names 'sinh', 'cosh', or 'tanh', and the second argument will be the value of *x* at which to evaluate the function. The file should also contain three subfunctions sinh1, cosh1, and tanh1 to perform the actual calculations, and the primary function should call the proper subfunction depending on the value in the string. [*Note:* Be sure to handle the case of an incorrect number of arguments, and also the case of an invalid string. In either case, the function should generate an error.]

**Q-9.** Write a function that attempts to locate the maximum and minimum values of an arbitrary function over a certain range. The function being evaluated should be passed to the function as a calling argument. The function should have the following input arguments:

first\_value—The first value of *x* to search

last\_value—The last value of *x* to search

num\_steps—The number of steps to include in the search

func—The name of the function to search

The function should have the following output arguments:

xmin—The value of *x* at which the minimum was found

min\_value—The minimum value of found

xmax—The value of *x* at which the maximum was found

max\_value—The maximum value *f* s*x*d found

Be sure to check that there are a valid number of input arguments, and that the MATLAB help and lookfor commands are properly supported.

**Q-10.** Use function random0 to generate a set of three arrays of random numbers. The three arrays should be 100, 1000, and 2000 elements long. Then, use functions tic and toc to determine the time that it takes function ssort to sort each array. How does the elapsed time to sort increase as a function of the number of elements being sorted? (*Hint*: On a fast computer, you will need to sort each array many times and calculate the average sorting time in order to overcome the quantization error of the system clock.)

**Q-11.** The gravitational force *F* between two bodies of masses and is given by the equation



where G is the gravitation constant (6.672 x 10-11 N m2 / kg2), m1 and m2 are the masses of the bodies in kilograms, and *r* is the distance between the two bodies. Write a function to calculate the gravitational force between two bodies given their masses and the distance between them. Test you function by determining the force on an 800 kg satellite in orbit 38,000 km above the Earth. (The mass of the Earth is 5.98 x 1024 kg.)

**Conclusion:**

**Date:**

**Experiment 6**

**Aim:** To work with additional data types in MATLAB.

**Apparatus:** MATLAB Software

**Objective:**

**Problems:**

**Q-1.** Write a function to\_polar that accepts a complex number c, and returns two output arguments containing the magnitude mag and angle theta of the complex number. The output angle should be in degrees.

**Q-2.** Write a function to\_complex that accepts two input arguments containing

the magnitude mag and angle theta of the complex number in degrees, and returns the complex number c.

**Q-3.** In a sinusoidal steady-state AC circuit, the voltage across a passive element is given by Ohm’s Law:

V=IZ

Where, **V** is the voltage across the element, **I** is the current though the element, and ***Z*** is the impedance of the element. Note that all three of these values are complex and that these complex numbers are usually specified in the form of a magnitude at a specific phase angle expressed in degrees. For example, the voltage might be **V=**12030V.



Write a program that reads the voltage across an element and the impedance of the element, and calculates the resulting current flow. The input values should be given as magnitudes and angles expressed in degrees, and the resulting answer should be in the same form. Use the function to\_complex to convert the numbers to rectangular for the actual computation of the current, and the function to\_polar to convert the answer into polar form for display.

**Q-4.** Below figure shows a series *RLC* circuit driven by a sinusoidal ac voltage source whose value is 1200volts. The impedance of the inductor in this circuit is ZL=j2пfL, where *j* is , ƒ is the frequency of the voltage source in hertz and *L* is the inductance in henrys. The impedance of the capacitor in this circuit is ZC=, where *C* is the capacitance in farads. Assume that R=100mH, and C=0.25nF.

The current **I** flowing in this circuit is given by Kirchhoff ’s Voltage Law to be



(*a*) Calculate and plot the magnitude of this current as a function of frequency as the frequency changes from 100 kHz to 10 MHz. Plot this information on both a linear and a log-linear scale. Be sure to include a title and axis labels.

(*b*) Calculate and plot the phase angle in degrees of this current as a function of frequency as the frequency changes from 100 kHz to 10 MHz. Plot this information on both a linear and a log-linear scale. Be sure to include a title and axis labels.

(*c*) Plot both the magnitude and phase angle of the current as a function of frequency on two sub-plots of a single figure. Use log-linear scales.



**Q-5.** Plot the function for 0 <=*t* <= 10 using the function plot(t,v). What is displayed on the plot?

**Q-6.** Euler’s equation defines *e* raised to an imaginary power in terms of sinusoidal functions as follows:

Create a two-dimensional plot of this function as varies from 0 to 2п. Create a three-dimensional line plot using function plot3 as varies from 0 to 2п (the three dimensions are the real part of the expression, the imaginary part of the expression, and ).

**Conclusion:**

**Date:**

**Experiment 7**

**Aim:** To work with MATLAB strings and additional data plots.

**Apparatus:** MATLAB Software

**Objective:**

1. To learn about different string operations.
2. To be aware of additional features of the simple two-dimensional plots.
3. To learn how to plot in three dimensions.

**Problems:**

**Q-1.** Write a program that accepts an input string from the user and determines the how many times a user-specified character appears within the string. (*Hint:* Look up the 's' option of the input function using the MATLAB Help Browser.)

**Q-2.** Modify the previous program so that it determines how many times a user-specified character appears within the string without regard to the case of the character.

**Q-3.** Write a program that accepts a string from a user with the input function, chops that string into a series of tokens, sorts the tokens into ascending order, and prints them out.

**Q-4.** MATLAB includes functions upper and lower, which shift a string to upper case and lower case respectively. Create a new function called caps, which capitalizes the first letter in each word and forces all other letters to be lower case. (*Hint:* Take advantage of functions upper, lower, and strtok.)

**Q-5.** Write a function that accepts a character string and returns a logical array with true values corresponding to each printable character that is *not* alphanumeric or whitespace (for example, $, %, #, etc.) and false values everywhere else.

**Q-6.** Write a function that accepts a character string and returns a logical array with true values corresponding to each vowel and false values everywhere else. Be sure that the function works properly for both lowercase and uppercase characters.

**Q-7.** Plot the function for *x* between 0 and 2 in steps of 0.1. Create the following plot types: *(a)* stem plot; *(b)* stair plot; *(c)* bar plot; *(d)* compass plot. Be sure to include titles and axis labels on all plots.

**Q-8.** Suppose that George, Sam, Betty, Charlie, and Suzie contributed $15, $5,$10, $5, and $15, respectively, to a colleague’s going-away present. Create a pie chart of their contributions. What percentage of the cost was paid by Sam?

**Q-9.** Plot the function f(x)=1/ over the range 0.1<=x<=10 using the function fplot. Be sure to label your plot properly.

**Q-10.** Create a mesh, surface plot, and contour plot of the function for the interval -1<=x<=1 and -2п<=y<=2п. In each case, plot the real part of *z* versus *x* and *y*.

**Conclusion:**

**Date:**

**Experiment 8**

**Aim:** To work with sparse arrays, cell arrays and structures in MATLAB.

**Apparatus:** MATLAB Software

**Objective:** To deal with special types of array like sparse arrays, cell arrays and structures.

**Problems:**

**Q-1.** Write a MATLAB function that will accept a cell array of strings and sortthem into ascending order according to the lexicographic order of theASCII character set. (You may use function c\_strcmp from Chapter 6for the comparisons if you wish.)

**Q-2.** Write a MATLAB function that will accept a cell array of strings and sort them into ascending order according to *alphabetical order*. (This implies that you must treat ‘A’ and ‘a’ as the same letter.)

**Q-3.** Create a sparse 100 x 100 array a in which about 5% of the elements contain normally distributed random values, and all of the other elements are zero (use function sprandn to generate these values). Next, set all of the diagonal elements in the array to 1. Next, define a 100-element sparse column array b, and initialize that array with 100 uniformly distributed values produced by function rand. Answer the following questions about these arrays:

(*a*) Create a full array a\_full from the sparse array a. Compare the memory required to store the full array and the sparse array. Which is more efficient?

(*b*) Plot the distribution of values in array a using function spy.

(*c*) Create a full array b\_full from the sparse array b. Compare the memory required to store the full array and the sparse array. Which is more efficient?

(*d*) Solve the system of equations a \* x = b using both the full arrays and the sparse arrays. How do the two sets of answers compare? Time the two solutions. Which one is faster?

**Q-4.** Create a function that accepts any number of numeric input arguments and

sums up all of individual elements in the arguments. Test your function by passing it the four arguments a=10, b= and d=[1 5 -2].

**Q-5.** Modify the function of the previous exercise so that it can accept either

ordinary numeric arrays or cell arrays containing numeric values. Test your

function by passing it the two arguments a and b, where , b{1} = [1 5 2] , and b{2} =.

**Q-6.** Create a structure array containing all of the information needed to plot a data set. At a minimum, the structure array should have the following fields:

* x\_data- *x*-data (one or more data sets in separate cells)
* y\_data -*y*-data (one or more data sets in separate cells)
* type -linear, semilogx, and so forth
* plot\_title -plot title
* x\_label -*x*-axis label
* y\_label -*y*-axis label
* x\_range -*x*-axis range to plot
* y\_range -*y*-axis range to plot

You may add additional fields that would enhance your control of the final plot.

After this structure array has been created, create a MATLAB function that accepts an array of this structure and produces one plot for each structure in the array. The function should apply intelligent defaults if some data fields are missing. For example, if the plot\_title field is an empty matrix, then the function should not place a title on the graph. Think carefully about the proper defaults before starting to write your function!

To test your function, create a structure array containing the data for three plots of three different types and pass that structure array to your function. The function should correctly plot all three data sets in three different figure windows.

**Q-7.** Define a structure **point** containing two fields x and y. The x field will contain the *x*-position of the point, and the y field will contain the *y*-position of the point. Then write a function **dist3** that accepts two points and returns the distance between the two points on the Cartesian plane. Be sure to check the number of input arguments in your function.

**Q-8.** Write a function that will accept a structure as an argument and return two cell arrays containing the names of the fields of that structure and the data types of each field. Be sure to check that the input argument is a structure and generate an error message if it is not.

**Q-9.** Write a function that will accept a structure array of student as defined in this chapter, and calculate the final average of each one, assuming that all exams have equal weighting. Add a new field to each array to contain the final average for that student, and return the updated structure to the calling program. Also, calculate and return the final class average.

**Q-10.** Write a function that will accept two arguments—the first a structure array, and the second a field name stored in a string. Check to make sure that these input arguments are valid. If they are not valid, print out an error message. If they are valid and the designated field is a string, concatenate all of the strings in the specified field of each element in the array, and return the resulting string to the calling program. Be sure that the function works properly for both lowercase and uppercase characters.

**Conclusion:**

**Date:**

**Experiment 9**

**Aim:** To work with input/output functions in MATLAB.

**Apparatus:** MATLAB Software

**Objective:** To learn about MATLAB’s input/output capabilities.

**Problems:**

**Q-1** The acceleration due to the Earth’s gravity at any height *h* above the surface of the Earth is given by the equation



where *G* is the gravitational constant (6.672 × 10-11 N m2 / kg2), *M* is the mass of the earth (5.98 × 1024 kg), *R* is the mean radius of the Earth (6371 km), and *h* is the height above the Earth’s surface. If *M* is measured in kg and *R* and *h* in meters, then the resulting acceleration will be in units of meters per second squared. Write a program to calculate the acceleration due to the Earth’s gravity in 500 km increments at heights from 0 km to 40,000 km above the surface of the Earth. Print out the results in a table of height versus acceleration with appropriate labels, including the units of the output values. Plot the data as well.

**Q-2.** Write a program that reads an arbitrary number of real values from a user specified input data file, rounds the values to the nearest integer, and writes the integers out to a user-specified output file. Make sure that the input file exists, and if not, tell the user and ask for another input file. If the output file exists, ask the user whether or not to delete it. If not, prompt for a different output file name.

**Q-3.** Write a program to generate a table containing the sine and cosine of for between 0° and 90°, in 1° increments. The program should properly label each of the column in the table.

**Q-4.** Write a program to read a set of integers from an input data file, and locate the largest and smallest values within the data file. Print out the largest and smallest values, together with the lines on which they were found. Assume that you do not know the number of values in the file before the file is read.

**Q-5.** Angles are often measured in degrees (º), minutes ('), and seconds ("), with 360 degrees in a circle, 60 minutes in a degree, and 60 seconds in a minute. Write a program that reads angles in radians from an input disk file and converts them into degrees, minutes, and seconds. Test your program by placing the following four angles expressed in radians into an input file and reading that file into the program: 0.0, 1.0, 3.141593, 6.0.

**Conclusion:**

**Date:**

**Experiment 10**

**Aim:** To generate Graphical User Interfaces (GUI) in MATLAB.

**Apparatus:** MATLAB Software

**Objective:** To learn the basic elements of the MATLAB GUIs.