JHARGRAM RAJ COLLEGE

DEPARTMENT OF MATHEMATICS

MATLAB-1 Practical

Course Code: MATSEC01

Laboratory Manual:

Experiment No. 1

Name of the Experiment: - Find the sum, product, max, min of a list of number in an array, in a sub array without library function.

Instruction to the Students:

- **Create an Array:**
  - Prompt the user to enter the elements of an array.
  - Store these elements in a MATLAB array.
- **Calculate the Sum Without Built-In Functions:**
  - Initialize a variable `sum_val` to zero.
  - Use a loop to iterate through each element in the array.
  - Add each element to `sum_val` to accumulate the total sum.
- **Calculate the Product Without Built-In Functions:**
  - Initialize a variable `product_val` to one.
  - Use a loop to iterate through each element in the array.
  - Multiply each element with `product_val` to get the cumulative product.
- **Find the Maximum Value Without Built-In Functions:**
  - Assume the first element of the array is the maximum and store it in `max_val`.
  - Loop through the array starting from the second element.
  - If any element is greater than `max_val`, update `max_val` with that element.
- **Find the Minimum Value Without Built-In Functions:**
  - Assume the first element of the array is the minimum and store it in `min_val`.

- Loop through the array starting from the second element.
- If any element is smaller than `min_val`, update `min_val` with that element.

- **Working with a Subarray:**
  - Prompt the user to input the starting and ending indices of a subarray within the main array.
  - Apply the above calculations on this subarray by iterating only between the specified indices.

- **Output the Results:**
  - Display the calculated sum, product, maximum, and minimum values for both the main array and the selected subarray.

Experiment No. 2

Name of the Experiment: Find a sub matrix of the given matrix

Instruction to the Students:

- **Create a Matrix:**
    - Prompt the user to enter the size of the matrix (number of rows and columns).
    - Prompt the user to enter the elements of the matrix.
    - Store these elements in a MATLAB matrix.
- **Specify the Submatrix Boundaries:**
    - Prompt the user to enter the starting and ending row indices.
    - Prompt the user to enter the starting and ending column indices.
    - Ensure these indices are within the bounds of the matrix dimensions.
- **Extract the Submatrix:**
    - Use nested loops to iterate through the specified rows and columns.
    - Store the values within the specified boundaries into a new matrix.
- **Output the Submatrix:**
    - Display the extracted submatrix.

Experiment No. 3

Name of the Experiment: Find the coloumn sum, product, max, min of the given matrix without library function

Instruction to the Students:

- **Create a Matrix:**
  - Prompt the user to enter the size of the matrix (number of rows and columns).
  - Prompt the user to enter the elements of the matrix.
  - Store these elements in a MATLAB matrix.
- **Calculate Column-wise Sum Without Built-In Functions:**
  - Initialize a row vector `col_sum` with zeros, where each element represents the sum of a column.
  - Use nested loops: the outer loop iterates over each column, and the inner loop sums the elements of each column.
- **Calculate Column-wise Product Without Built-In Functions:**
  - Initialize a row vector `col_product` with ones, where each element represents the product of a column.
  - Use nested loops: the outer loop iterates over each column, and the inner loop multiplies the elements of each column.
- **Find the Maximum of Each Column Without Built-In Functions:**
  - Initialize a row vector `col_max` with the first row of the matrix.
  - Use nested loops to iterate over each column and compare each element, updating `col_max` if a larger element is found.
- **Find the Minimum of Each Column Without Built-In Functions:**
  - Initialize a row vector `col_min` with the first row of the matrix.
  - Use nested loops to iterate over each column and compare each element, updating `col_min` if a smaller element is found.
- **Output the Results:**
  - Display the calculated sum, product, maximum, and minimum for each column.

Experiment No. 4

Name of the Experiment: Find the row sum, product, max, min of the given matrix without library function

Instruction to the Students:

- **Create a Matrix:**
    - Prompt the user to enter the size of the matrix (number of rows and columns).
    - Prompt the user to enter the elements of the matrix.
    - Store these elements in a MATLAB matrix.
- **Calculate Row-wise Sum Without Built-In Functions:**
    - Initialize a column vector `row_sum` with zeros, where each element represents the sum of a row.
    - Use nested loops: the outer loop iterates over each row, and the inner loop sums the elements of each row.
- **Calculate Row-wise Product Without Built-In Functions:**
    - Initialize a column vector `row_product` with ones, where each element represents the product of a row.
    - Use nested loops: the outer loop iterates over each row, and the inner loop multiplies the elements of each row.
- **Find the Maximum of Each Row Without Built-In Functions:**
    - Initialize a column vector `row_max` with the first element of each row.
    - Use nested loops to iterate over each row and compare each element, updating `row_max` if a larger element is found.
- **Find the Minimum of Each Row Without Built-In Functions:**
    - Initialize a column vector `row_min` with the first element of each row.
    - Use nested loops to iterate over each row and compare each element, updating `row_min` if a smaller element is found.
- **Output the Results:**
    - Display the calculated sum, product, maximum, and minimum for each row.

Experiment No. 5

Name of the Experiment: Define any transcendental function and then find and show the table of its functional values

Instruction to the Students:

- **Define a Transcendental Function:**
    - Choose a transcendental function
    - Define this function in the program.
- **Specify the Range of Input Values:**
    - Prompt the user to enter the start, end, and step size for the range of x-values over which the function will be evaluated.
    - Store these x-values in a vector.
- **Calculate the Function Values:**
    - Use a loop to calculate the function value for each x-value without directly using MATLAB's built-in transcendental functions
    - Stop the series at a reasonable number of terms for accuracy (e.g., 10 terms).
- **Display the Results in a Table:**
    - Create a table showing each x-value alongside its corresponding function value

Experiment No. 6

Name of the Experiment: Plotting of graph of functions e^{ax+b}, $\log(ax+b)$, $\log\frac{1}{(ax+b)}$, $\sin(ax+b)$, $\cos(ax+b)$, $|ax+b|$ and to illustrate the effect of a & b on the graph

Instruction to the Students:

Step 1: Define Variables and Range

- Open MATLAB and start a new script.
- Define values for parameters a and b to explore their effects. Start with $a = 1$ and $b = 0$, then try different values of a and b to observe changes.
- Define a range for x, e.g., from -10 to 10, with enough points for smooth plot

Step 2: Define the Functions

- Define each function in terms of $(ax + b)$:
    - Exponential: e^{ax+b}
    - Logarithm: $\log(ax + b)$
    - Inverse Logarithm: $\log\frac{1}{(ax+b)}$
    - Sine: $\sin(ax + b)$
    - Cosine: $\cos(ax + b)$
    - Absolute Value: $|ax + b|$

Step 3: MATLAB Code Implementation

- Use the following MATLAB code to define, compute, and plot each function with initial values of $a = 1$ and $b = 0$. Experiment by changing a and b.

Step 4: Experiment with Different Values of a and b

- Change a: Observe how the graph's slope changes. For example, setting $a > 1$ increases steepness, while $0 < a < 1$ makes the function grow/decay slower.
- Change b: Observe horizontal shifts. A positive b shifts the graph left, while a negative b shifts it right.

Experiment No. 7

Name of the Experiment: Plotting the graphs of polynomial of degree 4 and 5, the derivative graph, the second derivative graph and comparing them

Instruction to the Students:

Step 1: Define the Polynomials

1. Open MATLAB and start a new script.
2. Define the 4th-degree polynomial
3. Define the 5th-degree polynomial

Step 2: Compute the Derivatives

1. First Derivative: Use the power rule to compute the first derivatives
2. Second Derivative: Similarly, compute the second derivatives.

Step 3: MATLAB Code Implementation

1. Use the relevant MATLAB code to define, compute, and plot each polynomial and its derivatives.

Step 4: Analyze the Graphs

1. Observe the polynomial graphs and describe the key features, such as the number of peaks, valleys, and general curvature.
2. Compare the first derivatives: Note where each derivative crosses zero, corresponding to peaks and valleys in the polynomial.
3. Examine the second derivatives: Identify regions where the graph is concave up or concave down.

Experiment No. 8

Name of the Experiment: Sketching parametric curves (eg. Trochoid, cycloid, epicycloids, hypocycloid)

Instruction to the Students:

Step 1: Define the Parameter Range

- Open MATLAB and start a new script.
- Define the parameter ttt with an appropriate range, e.g., from 0 to 4π with sufficient points

Step 2: Define the Parametric Equations

- Cycloid: Define x and y as:
    - $x = r(t - sin\,(t))$

- $y = r(1 - cos\,(t))$

  Where r is the radius of the rolling circle.

- Epicycloid: Define x and y as:
  - $x = (R + r)cos\,(t) - rcos\,(\frac{R+r}{r} \cdot t)$
  - $y = (R + r)sin\,(t) - rsin\,(\frac{R+r}{r} \cdot t)$

  Where R is the radius of the fixed circle, and r is the radius of the rolling circle.

- Hypocycloid: Define x and y as:
  - $x = (R - r)cos\,(t) + rcos\,(\frac{R-r}{r} \cdot t)$
  - $y = (R - r)sin\,(t) - rsin\,(\frac{R-r}{r} \cdot t)$

- Trochoid: Define x and y as:
  - $x = r \cdot t - dsin\,(t)$
  - $y = r - dcos\,(t)$

  Where d is the distance from the point tracing the curve to the center of the rolling circle.

Step 3: MATLAB Code Implementation

- Use the relevant MATLAB code to define, compute, and plot each parametric curve. Adjust values of r, R, and d as needed to explore variations in shape.

Step 4: Analyze the Curves

- Cycloid: Observe the repetitive pattern created by the rolling circle.
- Epicycloid and Hypocycloid: Note how the path changes based on whether the circle rolls on the outside (epicycloid) or inside (hypocycloid).
- Trochoid: Examine how changing d affects the curve shape, making it closer to a cycloid when $d = r$

Experiment No. 9

Name of the Experiment: Tracing of conics in cartesian coordinates/polar coordinates

Instruction to the Students:

Step 1: Define Variables and Range

- Open MATLAB and start a new script.
- Define the range for x and y in Cartesian coordinates

Step 2: Define Conics in Cartesian Coordinates

- Write the equation of conics in general form

Step 3: MATLAB Code Implementation

- Use the relevant MATLAB code to trace the conic and plot

Step 4: Analyze the Graphs

- Display the nature of the conic

Experiment No. 10

Name of the Experiment: Sketching ellipsoid,hyperboloid of one and two sheets, elliptic cone, elliptic cone, elliptic paraboloid and hyperbolic paraboloid using cartesian coordinates

Instruction to the Students:

Step 1: Define Variables and Range

- Open MATLAB and start a new script.
- Define the range for x, y & z in Cartesian coordinates

Step 2: Define Conics in Cartesian Coordinates

- Write the equations of ellipsoid, hyperboloid of one and two sheets, elliptic cone, elliptic cone, elliptic paraboloid and hyperbolic paraboloid

Step 3: MATLAB Code Implementation

- Use the relevant MATLAB code to sketch the conicoids

Step 4: Analyze the Graphs

- Display the coloured shape of the conicoid

JHARGRAM RAJ COLLEGE

DEPARTMENT OF MATHEMATICS

MATLAB-2 Practical

Course Code: MATSEC02

Laboratory Manual:

Experiment No. 1

Name of the Experiment: Fitting a curve for given data

Instruction to the Students:

Step 1: Define the Data

- Load or manually define the data points (x, y).

Step 2: Construct the Vandermonde Matrix (A)

- Construct a Vandermonde matrix A

Step 3: Solve for Polynomial Coefficients

- Solve the equation $A \cdot c = y$ to find the polynomial coefficients c using matrix inversion.

  Use the equation: $c = (A^T A)^{-1} A^T y$

Step 4: Generate the Fitted Curve

- Use the obtained coefficients to evaluate the polynomial at new points (e.g., for smooth plotting).

  Create a fine grid of x-values (x_fit) to evaluate the polynomial.

Step 5: Plot the Data and the Fitted Curve

- Plot the original data points and the fitted curve using plot

Experiment No. 2

Name of the Experiment: Plotting of given data: Graph plotting, multiple plots, matrix plots, polar plots, 3D plotting (line, surface, mesh and contour) of three dimensional data

Instruction to the Students:

Step 1: Plotting a 2D Line Plot

- Create two vectors representing the data to plot

Step 2: Multiple Plots in One Figure

- Use the subplot function to create multiple plots in one figure

Step 3: Matrix Plot

- Use the imagesc function to plot a matrix.

Step 4: Polar Plot

- Create a polar plot to visualize data in polar coordinates

Step 5: 3D Line Plot

- Use the plot3 function to create a 3D line plot

Step 6: 3D Surface Plot

- Use the surf function to create a 3D surface plot.

Step 7: 3D Mesh Plot

- Use the mesh function to create a 3D mesh plot

Step 8: 3D Contour Plot

- Use the contour function to create a 3D contour plot

Step 9: Analyze the Results

- **2D Line Plot**: Shows the relationship between two variables.
- **Multiple Plots**: Displays different functions in a single figure for comparison.
- **Matrix Plot**: Visualizes matrix data as color-coded images, often used for heatmaps.
- **Polar Plot**: Visualizes data in polar coordinates, useful for cyclic phenomena.
- **3D Line Plot**: Displays data in 3D space, where each point has X, Y, and Z coordinates.
- **3D Surface Plot**: Visualizes a 3D surface, showing how the surface behaves based on input values.
- **3D Mesh Plot**: Similar to the surface plot, but displays the surface as a mesh grid.
- **3D Contour Plot**: Displays contours of a 3D surface in 2D, representing levels of constant values

Experiment No. 3

Name of the Experiment: Obtaining surface of revolution of curves

Instruction to the Students:

Step 1: Define the Curve to Be Revolved

- Choose a function or curve that you want to revolve

Step 2: Create the Parameter Grid

- Define the range of the curve and the angle of revolution.
    - Let the curve $y = f(x)$ be revolved around the x-axis.
    - Create a grid for x values and θ values for the revolution

Step 3: Parametric Equations for the Surface of Revolution

- For revolution around the x-axis: $Y = f(X) \cdot \cos(\theta), Z = f(X) \cdot \sin(\theta)$Using the function f and the parameter grid, calculate the corresponding Y and Z values

Step 4: Plot the Surface

- Use the surf function to plot the surface of revolution

Experiment No. 4

Name of the Experiment: Find the sum, product, max, min, sort of a list of number in an array, in a sub array using library function

Instruction to the Students:

Step 1: Define the Array

- Create an array with a list of numbers

Step 2: Perform Sum Operation

- Use the sum function to calculate the sum of the array elements

Step 3: Perform Product Operation

- Use the prod function to calculate the product of the array elements.

Step 4: Find Maximum and Minimum Values

- Use the max and min functions to find the maximum and minimum values in the array

Step 5: Sort the Array

- Use the sort function to sort the array in ascending order.

Step 6: Perform Operations on a Sub-array

- To work with a sub-array, first select a range of elements from the original array.

Step 7: Analyze the Results

- **Sum and Product**: The sum and product should be straightforward calculations of all the elements in the array or sub-array.
- **Max and Min**: The max and min functions should return the largest and smallest values in the array or sub-array.
- **Sorting**: The sorted array should be in ascending order.
- **Sub-array Operations**: The operations performed on the sub-array should be similar to those performed on the full array, but only for the selected portion of the array.

Experiment No. 5

Name of the Experiment: Find the column sum, product, max, min, sort of the given matrix using library function

Instruction to the Students:

Step 1: Define the Matrix

- Create a matrix for which the operations will be performed

Step 2: Column-wise Sum

- Use the sum function to calculate the sum of elements in each column

Step 3: Column-wise Product

- Use the prod function to calculate the product of elements in each column

Step 4: Find Maximum and Minimum Values of Each Column

- Use the max function to find the maximum value in each column.
- Use the min function to find the minimum value in each column

Step 5: Sort Each Column

- Use the sort function to sort the elements of each column in ascending order

Step 6: Analyze the Results

- **Column Sum**: The sum of each column is computed, which adds up the elements along each column.
- **Column Product**: The product of the elements in each column is calculated by multiplying the elements of each column.
- **Column Maximum and Minimum**: The max function returns the largest element in each column, while the min function returns the smallest element in each column.
- **Column Sorting**: The sort function sorts the values in each column independently in ascending order.

Experiment No. 6

Name of the Experiment: Find the row sum, product, max, min of the given matrix using library function
Instruction to the Students:

Step 1: Define the Matrix

- Create a matrix for which the operations will be performed

Step 2: Row-wise Sum

- Use the sum function to calculate the sum of elements in each Row

Step 3: Row-wise Product

- Use the prod function to calculate the product of elements in each Row

Step 4: Find Maximum and Minimum Values of Each Row

- Use the max function to find the maximum value in each Row.
- Use the min function to find the minimum value in each Row

Step 5: Sort Each Row

- Use the sort function to sort the elements of each Row in ascending order

Step 6: Analyze the Results

- **Row Sum**: The sum of each Row is computed, which adds up the elements along each Row.
- **Row Product**: The product of the elements in each Row is calculated by multiplying the elements of each Row.
- **Row Maximum and Minimum**: The max function returns the largest element in each Row, while the min function returns the smallest element in each Row.
- **Row Sorting**: The sort function sorts the values in each Row independently in ascending order.

Experiment No. 7

Name of the Experiment: Conversion of one number system to another number system among decimal, binary, octal, hexadecimal

Instruction to the Students:

Step 1: Decimal to Binary Conversion

- To convert a **decimal number** to **binary**, use the dec2bin function

Step 2: Binary to Decimal Conversion

- To convert a **binary number** to **decimal**, use the bin2dec function

Step 3: Decimal to Octal Conversion

- To convert a **decimal number** to **octal**, use the dec2oct function

Step 4: Octal to Decimal Conversion

- To convert an **octal number** to **decimal**, use the oct2dec function

Step 5: Decimal to Hexadecimal Conversion

- To convert a **decimal number** to **hexadecimal**, use the dec2hex function

Step 6: Hexadecimal to Decimal Conversion

- To convert a **hexadecimal number** to **decimal**, use the hex2dec function

Experiment No. 8

Name of the Experiement: Solution of a square, under determined and over determined system of linear equation

Instruction to the Students:

Step 1: Square System (Unique Solution)

- Consider the system of equations

Step 2: Under-Determined System (Infinite Solutions)

- Consider the system with fewer equations than unknowns

Step 3: Over-Determined System (Least-Squares Solution)

- Consider the system with more equations than unknowns

Step 4: Analyze the Results

- **Square system**: If the coefficient matrix A is invertible, you will get a unique solution.
- **Under-determined system**: The system will have infinitely many solutions, and the solution found is the one that minimizes the least-squares error.
- **Over-determined system**: The system may not have an exact solution, but MATLAB will return the best approximation using least squares

Experiment No. 9

Name of the Experiement: Different problems for root, eigen values and eigen vectors of the matrix

Instruction to the Students:

Step 1: Matrix Input

- Consider a square matrix

Step 2: Find the eigen values and eigen vectors of the matrix

The function eig returns a matrix of eigenvectors and a diagonal matrix of eigen values

Experiment No. 10

Name of the Experiment: Plotting of recursive sequences

Instruction to the Students:

Step 1: Define the Recursive Sequence

- Choose a recursive sequence to study.

Step 2: Plot the Recursive Sequence

- Plot the sequence to visualize how the terms evolve over time

Step 3: Analyze the Plot

- The plot will show the terms of the Fibonacci sequence

Experiment No. 11

Name of the Experiment: Study the convergence of sequences through plotting

Instruction to the Students:

Step 1: Define the Sequence

- Choose a sequence $\{a_n\}$ to study

Step 2: Plot the Sequence Terms

- Plot the terms of the sequence to visualize the behaviour as n increases

Step 3: Analyze the Plot

- **Convergent Sequence**: If the sequence $\{a_n\}$ tends towards a finite value as $n \to \infty$ the plot will show that the terms approach this value.
- **Divergent Sequence**: If the sequence $\{a_n\}$ grows without bound or oscillates, the plot will show that the terms keep increasing or oscillating without settling on a finite value.

Experiment No. 12

Name of the Experiment: Verify Bolzano -Weirstrass theorem through plotting of sequences and hence identify convergent sub sequences from the plot

Instruction to the Students:

Step 1: Define the Sequence

- Define a bounded sequence that oscillates

Step 2: Plot the Sequence

- Plot the terms of the sequence to visualize how it behaves. The sequence will oscillate, but the values will be bounded.

Step 3: Identify and Plot Convergent Sub sequences

- A bounded sequence can have sub sequences that converge to different limits

Step 4: Analyze the Plot

- Verify Bolzano –Weirstrass Theorem

Experiment No. 13

Name of the Experiment: Study the convergence/divergence of infinite series by plotting their sequences of partial sum

Instruction to the Students:

Step 1: Define the Sequence

- Choose an infinite series $\sum a_n$ for which you want to study the convergence/divergence

Step 2: Compute the Partial Sums

- Compute the partial sums using the **cumsum** function.

Step 3: Plot the Partial Sums

- Plot the partial sums of the series to visualize their behaviour as n increases

Step 4: Analyze the Plot

- For a **convergent series**, the partial sums should approach a finite value as n increases. The graph should flatten out after a certain point.
- For a **divergent series**, the partial sums should increase without bound or oscillate as n increases

Step 5: Interpret the Results

- **Convergent Series**: The partial sums should approach a finite value, indicating that the series converges.
- **Divergent Series**: The partial sums should grow without bound, indicating that the series diverges

Experiment No. 14

Name of the Experiment: Cauchy's root test by plotting nth roots

Instruction to the Students:

Step 1: Define the Sequence

- Choose a sequence a_n for which you want to apply the Root Test

Step 2: Compute the nth Root of the Terms

- Compute the n-th root of the absolute value of the terms of the sequence.

Step 3: Plot the nth Roots of the Terms

- Plot the n-th roots of the terms to visualize the behaviour as n increases

Step 4: Analyze the Plot

- The plot shows how the n-th roots of the terms behave as n increases.
- If the n-th root tends towards a value less than 1, the series is likely to converge.
- If the n-th root approaches a value greater than 1, the series will diverge.

Step 5: Interpret the Results

- If the n-th root of $|\ a_n\ |$ tends towards a value less than 1 as n increases, the series is convergent.
- If the n-th root is greater than 1, the series diverges.
- If the n-th root fluctuates around 1, the test is inconclusive.

Experiment No. 15

Name of the Experiment: Ratio test by plotting the ratio of nth and (n+1) th term

Instruction to the Students:

Step 1: Define the Sequence

- Choose a sequence a_n for which you want to apply the Ratio Test

Step 2: Compute the Ratio of nth and (n+1)th Terms

- Compute the ratio of the (n+1)th term to the nth term

Step 3: Plot the Ratio of Terms

- Plot the ratio of the terms to visualize the behavior of the ratio

Step 4: Analyze the Plot

- The plot shows how the ratio $\frac{a_{n+1}}{a_n}$ behaves as n increases.
- If the ratio tends towards a value less than 1, the series is likely to converge.
- If the ratio approaches a value greater than 1, the series will diverge

Step 5: Interpret the Results

- If the ratio $\frac{a_{n+1}}{a_n}$ is consistently below 1, the series is converging.
- If the ratio exceeds 1 for large values of n, the series is diverging.
- If the ratio fluctuates around 1, the Ratio Test is inconclusive, and further analysis (such as using other convergence tests) may be needed