

Exercise 1.

- Calculate the following terms with MATLAB:

➤ $\frac{4 \cdot 7 - 8}{3^4 - 1}$

➤ $e^{\sin 3.2}$

➤ $\frac{5 + \log \frac{37}{2.9 \cdot 1.7}}{9}$

- Do the following calculations with MATLAB:

➤ $A = \begin{pmatrix} 3 & 5 & 1 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix}$

➤ $B = (-2, 1, -4)^T$

➤ $C = A^{-1}B$

- Create a sequence from -3 to 5 with step width 0.1
- Create a sequence from 10^{-3} to 10^5 with increments on a logarithmic scale, i.e. the resulting sequence should contain the numbers $10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4, 10^5$. Thereto, combine sequence generation with the (element-wise) power function. Write down the MATLAB commands.
- calculate the following term: $\sum_{k=1}^{100} k \cdot (101 - k)$. Thereto, create an increasing sequence \vec{u} and a decreasing sequence \vec{v} and calculate $\vec{u} \cdot \vec{v}^T$. Write down the MATLAB commands and the result.
- Create a 5×6 random two-dimensional array A, and use the logical 1 identifier to find all elements in A that are greater than 0.3 and less than 0.7 , and assign them to B.
- Create a 4×4 two-dimensional array A with all values 0 and a 4×3 two-dimensional array B with all values 1. Splice A and B into a 4×7 two-dimensional array.

- $A = \begin{pmatrix} 3 & 2 & 0 \\ -5 & 6 & 1 \\ 1 & 0 & 2 \end{pmatrix}$

- Use the “find” function to find the linear index of the non-zero elements of the A matrix.
- Find the subscripts of the rows and columns in matrix A with element 0.
- Find the official documentation and learn more about the use of the “find” function.

- $A = \begin{pmatrix} 1 & 25 & 7 & 20 \\ 8 & 90 & 30 & 18 \\ 12 & 23 & 15 & 176 \\ 50 & 0 & 31 & 9 \end{pmatrix}$

- Add a fifth column and assign the values [1, 0, 2, 8] in matrix A.
- Sum of the number of rows and columns of matrix A.
- Assign the elements of columns 1, 3 and 5 of rows 2-4 of matrix A to matrix B.
- Assign A to matrix C, delete the first and third rows of C and display them.
- Rearrange matrix A into a matrix of 5 rows and 4 columns.
- Find the mean value for each column of matrix A.
- Find the mean value for each row of matrix A.
- Calculate the mean value of all elements in matrix A.
- Assign the first and last values of matrix A to nan.
- Calculate the mean of all element in matrix A, excluding nan.

Exercise 2.

- plot $\sin(20x) \cdot \exp(-x)$ in the interval $[0,5]$ using solid line, then plot $\pm \exp(-x)$ on the same graph using dashed line; add descriptions and change default font to Times New Roman and font size to 15.

Exercise 3.

- Write a function in MATLAB that calculates the Euclidean distance between two vectors u and v : $\sqrt{\sum_i (u_i - v_i)^2}$

Exercise 4.

- plot contour for $\sqrt{((x - 0.5)^2 + (y - 0.5)^2) \cdot ((x + 0.5)^2 + (y + 0.5)^2)}$ in the interval $[-0.8,0.8] \times [-0.8,0.8]$ using blue line, then denote both minima using red cross; add axis description and a title.

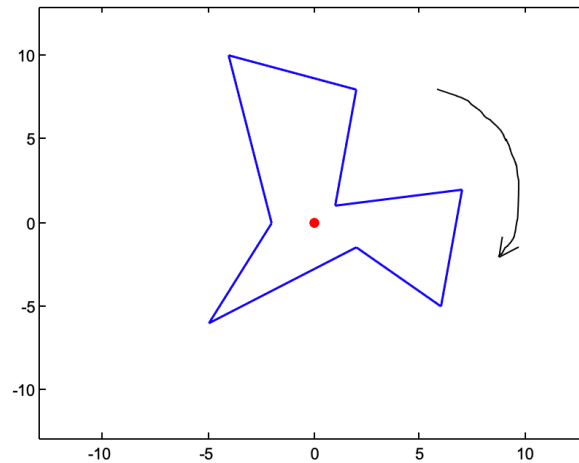
Exercise 5.

- Implement simple database storing people names and ages. Database should be stored in a .mat file and accessed through a function 'people.m' that can work in four modes as illustrated below (here, db_file is a string storing the database file name)
 - people(db_file,'reset') – erases the database
 - people(db_file,'list') – displays the database content (list of people)
 - people(db_file,'insert','John',25,'Anna',17) – inserts new items
 - people(db_file,'remove','Jack',15) – removes items from database
- Test your database using a sequence of commands below. The preferred format for storing data is a structure. Please use a given list to start.
 - Add [your name, your age] to the list; Display the list
 - Remove ["Tom"] from the list; Display the list
 - Add ["Joe",24; "Ella",28] to the list; Display the list
 - Reset the list; Display the list

Exercise 6.

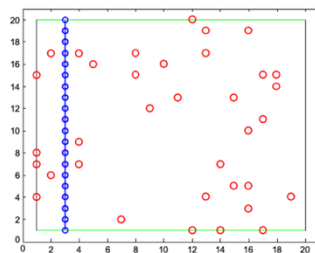
- Implement Matlab function that draws a polygon defined by its vertices and rotates it with respect to the center of the coordinate system as shown below. The angle of rotation should be determined by the user.

Test your code using **a given example**. Rotating clockwise and rotation angle is calculated using the following formula: $\text{mod}([\text{Your ID number}], 360)$.

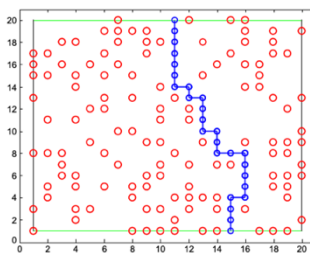


Exercise 7.

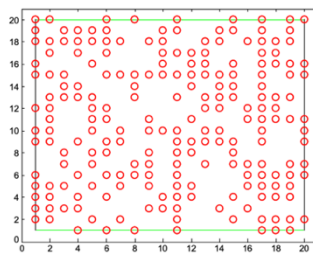
- Given $N \times N$ rectangular grid, mark certain number of randomly selected nodes as obstacles. Develop and implement an algorithm that searches for the (possibly shortest) path from the bottom edge of the grid to its top edge assuming that possible moves are only vertical or horizontal. The path must not pass through an obstacle. Visualize the grid, obstacles and the path similarly as shown below. Test your code for various cases (different value of N , different density of obstacles). **Please use the given obstacle maps.**



Small number of obstacles
=> Simple path



Larger number of obstacles
=> Path still can be found



Large number of obstacles
=> No path exists