Experiment–1: Introduction to MATLAB

Signals and Systems Lab(EC2P002)
School of Electrical Sciences, IIT Bhubaneswar
Autumn Semester 2022

Agenda of the Experiment

In this lab session, we will do the following:

- Learn basic vector operations
- Learn basic matrix operations
- Save and load variables, matrices, audio files, image files.
- Learn loops, functions, decision making.

Vector Operations

1. Generate two vectors u and v by typing the code given below.
   $v_1 = [1\,2\,3\,4\,10]$
   $v_2 = [-1\ \ -2\ \ -3\ \ -4\ \ -10]$
2. Concatenate two vectors to form a new vector v_3. Use $v_3 = [u, v]$ or $v_3 = [u\ v]$
3. Type help ones and help zeros. Create vectors v_4 of 20 zeros and v_5 of 20 ones using the command. Use the size or length command to identify the size of the created vector.
4. Create a vector v_6 of length 20 with every entry equal to 5.
5. Add two vectors v_1 and v_2 by using $v_1 + v_2$. Assign it to v_7.
6. Add a constant to all entries of v_7 vector. Try $v_7 + 5$
7. Try 1:2:10. Observe the result. Use this to create vectors
   - $v_8 = [1\ 2\ 3 \ldots 10]$
   - $v_9 = [25\ 20\ 15 \ldots -20]$
8. Pick the third value in vector v_1. Use $v_1(3)$. Note that numbering of entries is from 1 to length(v_1). Try $v_1(6)$ and $v_1(0)$.

Vector Operations

9. Obtain v_{10} from v_1 by removing the third entry.
10. Obtain v_{11} from v_1 by picking only the odd indexed entries.
11. Obtain the transpose of v_1 and save it as v_{1t}.
12. Try the operations $v_1 * v_{1t}$ and $v_{1t} * v_1$
13. Try the operation $v_1 . * v_1$. Also try $v_1 .^\wedge 2$ and $v_1 ./ v_2$.
14. Add all the entries of v_1 and store it in s
15. Find the average of entries in v_1
16. Try $v_1 == 3$. Observe the result.
17. Find the number of entries in v_9 greater than or equal to 15. Do not use any loops.
18. Provide two different methods without using loops to find the sum of entries in v_9 which are greater than 15. Hint: Use '$*$' and '$.*$'

```
%vector_generation

clc;
clear all;
close all;

%1
v1=[1,2,3,4,10];
v2=-v1;

%2
v3=[v1,v2];

%3.1
v4=zeros(20,1);
%3.2
v5=ones(20,1);
%3.3
size(v4);
size(v5);

%4
v6=5*ones(20,1);

%5
v7=v1+v2;

%6
v7+5;

%7
%disp(1:2:10)
v8=[1:10];
v9=[25:-5:-20];

%8
%disp(v1(3))

%9
v10=[v1(1:2) v1(4:end)];

%10
v11=v1(1:2:5);

%11
v1t=transpose(v1);

%12
v1*v1t;
v1t*v1;

%13
v1.*v1;
v1.^2;
v1./v2;
```

```
%14
s=sum(v1);

%15
s/length(v1);

%16
v1==3;

%17
%disp(v9>=15);

%18.1
a=[v9(v9>15)];
%disp(a)
%disp(sum(a))

%18.2
%disp(sum(v9(v9>15)))
```

Published with MATLAB® R2023a

Matrix Operations

1. Create a 3×4 matrix A. Use $A = [1\ 2\ 3\ 4; 5\ 6\ 7\ 8; 9\ 10\ 11\ 12]$;
2. Print only second row of A using $A(2, :)$ and print only third column of A using $A(:, 3)$.
3. Print only first and third column of A.
4. Type help repmat. Use repmat command to create a 12×12 matrix B repeating the matrix A.
5. Use reshape command to change A into 2×6 matrix.
6. Use zeros, ones commands to create matrices C and D both of order 3×4.
7. Use horzcat to concatenate A and C and use vertcat to concatenate A and D.
8. Create an identity matrix using eye command.
9. Use diag command to obtain diagonal elements of B.
10. Use tril and triu commands on the matrix B. Apply these on matrix A also.
11. Using sum command find the sum of each row and the sum of each column of matrix A. Find the sum of all entries in the matrix. Use $sum(A(:))$ or 'all'.
12. Understand the difference between '$*$' and '$.*$' in matrix multiplication by choosing suitable matrices.

```
% matrix_operations

clc;
close all;
clear all;

%1
A=[1 2 3 4; 5 6 7 8; 9 10 11 12];
%2
%disp(A(2,:))
%disp(A(:,2))
%3
%disp(A(:,1:2:3))
%4
B=repmat(A,4,3);
%5
reshape(A,2,6);
%6
C=zeros(3,4);
D=ones(3,4);
%7
horzcat(A, C);
vertcat(A, D);
%8
eye(5);
%9
diag(B);
%10
tril(B);
triu(B);
%11
sum(A(:));
%12
A*D';
A.*D;
```

Published with MATLAB® R2023a

Random Matrices Generation

1. Use rand command to generate a random matrix R of order 3×4.
2. Note that all the random values generated lies between 0 and 1. Suitably modify the code to obtain random matrices with entries between 5 and 10.
3. Use randi command to generate a matrix of uniformly distributed integers. The integers generated should belong to the interval -5 to 5.
4. Use randn command to generate a matrix of random values following standard normal distribution.
5. Modify the code to obtain entries from a normal distribution with mean 10 and variance 4.
6. Do the same for Rayleigh and Exponential distribution with scale parameter = 2 and mean parameter = 10 respectively.

```
% Random Matrices Generation

clc;
clear all;
close all;

%1
R1=rand(3,4);
%2
R2=5+(10-5).*rand(3,4);
%3
R3=randi([-5 5],3,4);
%4
R4=randn(1,4);
%5
R5=2*randn(1,4)+10;
%6.1
r1=raylrnd(10,2) ;
%6.2
r2=exprnd(10,2);
```

Published with MATLAB® R2023a

Saving and Loading

1. Create and save two variables A and B to a file savefile.mat
2. Create a third variable C and append it to savefile.mat
3. Load all three variables from savefile.mat
4. Create a string array containing the names of your three friends and their ages. Use writematrix command to write the string array to an excel file.
5. Use readmatrix command to read from the saved excel file and compute the average age of your three friends.
6. Use audioread command to read an audio file.
7. Use audiowrite command to write a portion of the file to a new audio file.
8. Use imread to read a jpg image to a three dimensional matrix F.
9. Use image function to display the image.
10. Make the entries in $F(:,:,1)$ all zero and display the image. Restore the image and do the same for $F(:,:,2)$ and $F(:,:,3)$.

```
% Save & Load file

%1
A=[1, 2, 3];
B=[4, 5, 6];
save('ACE_LAB_EXP_1\savefile.mat','A','B');

%2
load('ACE_LAB_EXP_1\savefile.mat');
C=[11,10,101];
save('ACE_LAB_EXP_1\savefile.mat','C','-append');

%3
%disp(A);
%disp(B);
%disp(C);

%4-5
friends=["Archie","Harshil","Himanshu";
            "26","26","27"];
filename='ACE_LAB_EXP_1\friends_info.xlsx';
writematrix(friends,filename);

readmatrix("ACE_LAB_EXP_1\friends_info.xlsx");
sum=0.00;
for i=1:3
    n=str2double(friends(end,i));
    sum=sum+n;
    n=0;
end
avg=sum/3;

%6-7
[y,Fs]=audioread('ACE_LAB_EXP_1\Alok & Dynoro - On & On [hithotmusic] House my-free-mp3s.com .mp3');

startSample=1000;
endSample=2000;
portion=y(startSample:endSample,:);
audiowrite('ACE_LAB_EXP_1\portion.mp3',portion,Fs);

%8-9
[img,Fs]=imread('ACE_LAB_EXP_1\sk-CNBRg1K9QvQ-unsplash.jpg');
%disp(image(img));

%10
imgF1=img;
imgF2=img;
imgF3=img;

imgF1(:,:,1)=0;
%disp(image(imgF1));

imgF2(:,:,2)=0;
%disp(image(imgF2));
```

```
imgF3(:,:,3)=0;
%disp(image(imgF3));
```

Published with MATLAB® R2023a

Loops and functions

1. Create a 100 × 100 random matrix K having integer valued elements uniformly distributed in the interval 1 to 10. Use *for loop* to find the sum of square of the entries in the matrix K.
2. Do part 1 without using for loop.
3. Use tic toc command to compute the time elapsed in Part 1 and Part 2. Read more about vectorization.
4. Find the number of entries greater than 5 and less than 3 in the generated matrix K. Use *if else* statement
5. Do part 3 without using *if else* statement.
6. Demonstrate the use of *while loop* by a small program.
7. Demonstrate the use of *switch* statement by a small program.
8. Define a function to find the factorial of a number.

```
% Loops & Functions

%1.1
K=randi([1,10],100);
%1.2
tic;
arrsum=0;
for i=1:100
    for j=1:100
        arrsum=arrsum+K(i,j);
    end
end
(arrsum);
toc;
%2-3
tic;
array_sum_without_loop=sum(sum(K));
toc;
%sum by looping takes more time
%4
count_greater_than_5=0;
count_less_than_3=0;
%arrsum=0;
for i=1:100
    for j=1:100
        if K(i,j)<3
            count_less_than_3=count_less_than_3+1;
        end
        if K(i,j)>5
            count_greater_than_5=count_greater_than_5+1;
        end
    end
end
count_greater_than_5;
count_less_than_3;

%5
K1=K>5;
count_greater_than_five=sum(sum(K1(:,:)));
K2=K<3;
count_less_than_three=sum(sum(K2(:,:)));

%6
x=1;n=0;
while x<=10
    n=n+x;x=x+2;
end
x;

%7.1
factorial(5);
%7.2
n=5;
fact=1;
while n>0
    fact=fact*n;
    n=n-1;
end
fact;
```

```
Elapsed time is 0.001333 seconds.
Elapsed time is 0.000115 seconds.
```

Published with MATLAB® R2023a