APPLIED MATHEMATICS LAB USING SCILAB

ETMA 252

EXPERIMENT NO. 1

AIM: Algebra of matrices

1. To find transpose of a matrix
2. To find addition of two matrices
3. To find multiplication of matrices

Code:

1.Transpose of a matrix:

```
clc
m=input('Enter number of rows in the Matrix: ');
n=input('Enter number of columns in the Matrix: ');
disp('Enter the Matrix')
fori=1:m
for j=1:n
A(i,j)=input('..');
end
end
B=zeros(n,m);
fori=1:n
for j=1:m
B(i,j)=A(j,i)
end
end
disp('Entered matrix is')
disp(A)
disp('Transposed matrix is')
disp(B)
```

OUTPUT:

```
Enter number of rows in the Matrix: 3
Enter number of columns in the Matrix: 3

 Enter the Matrix
..23
..34
..45
..6
..8
..97
..45
..21
..35

 Entered matrix is

    23.    34.    45.
    6.     8.     97.
    45.    21.    35.

 Transposed matrix is

    23.    6.     45.
    34.    8.     21.
    45.    97.    35.

-->
```

2. To find addition of two matrices :

```
clc
m=input('Enter number of rows in the Matrices: ');
n=input('Enter number of columns in the Matrices: ');
disp('Enter the first Matrix')
for i=1:m
for j=1:n
A(i,j)=input('..');
end
end
disp('Enter the second Matrix')
fori=1:m
for j=1:n
B(i,j)=input('..');
end
```

```
end
fori=1:m
for j=1:n
C(i,j)=A(i,j)+B(i,j);
end
end
disp('The first matrix is')
disp(A)
disp('The Second matrix is')
disp(B)
disp('The sum of the two matrices is')
disp(C)
```

OUTPUT:

```
Enter number of rows in the Matrices: 3
Enter number of columns in the Matrices: 2

 Enter the first Matrix
..12
..34
..5
..6
..78
..9

 Enter the second Matrix
..23
..4
..5
..67
..85
..91

 The first matrix is

    12.    34.    45.
    5.     6.     97.
    78.    9.     35.

 The Second matrix is

    23.    4.     45.
    5.     67.    21.
    85.    91.    35.

 The sum of the two matrices is
```

```
    35.     38.
    10.     73.
    163.    100.

-->
```

3) Multiplication of two matrices

```
clc
m=input('Enter number of rows in the first Matrix: ');
n=input('Enter number of columns in the first Matrix: ');
p=input('Enter number of rows in the second Matrix: ');
q=input('Enter number of columns in the second Matrix: ');
if n==p
disp('Matrices are conformable for multiplication')
else
disp('Matrices are not conformable for multiplication')
break;
end
disp('Enter the first Matrix')
fori=1:m
for j=1:n
A(i,j)=input('..');
end
end
disp('Enter the second Matrix')
fori=1:p
for j=1:q
B(i,j)=input('..');
end
end
C=zeros(m,q);
fori=1:m
for j=1:q
for k=1:n
C(i,j)=C(i,j)+A(i,k)*B(k,j);
end
end
```

```
end
disp('The first matrix is')
disp(A)
disp('The Second matrix is')
disp(B)
disp('The product of the two matrices is')
disp(C)
```

OUTPUT:

```
Enter number of columns in the first Matrix: 2
Enter number of rows in the second Matrix: 2
Enter number of columns in the second Matrix: 4

 Matrices are conformable for multiplication

 Enter the first Matrix
..2
..3
..45
..6
..78
..23

 Enter the second Matrix
..34
..54
..65
..86
..72
..2
..39
..45

 The first matrix is

    2.      3.      45.
    45.     6.      97.
    78.     23.     35.

 The Second matrix is

    34.     54.     65.     86.
    72.     2.      39.     45.
    85.     91.     35.     0.

 The product of the two matrices is

    284.      114.      247.      307.
    1962.     2442.     3159.     4140.
    4308.     4258.     5967.     7743.

-->
```

EXPERIMENT NO. 2

AIM :To find the inverse of square matrix using Gauss- Jordan method

CODE:

```
clc
disp('Enter a 3 by 3 matrix row-wise, make sure that diagonal
elements are non zero')
fori=1:3
for j=1:3
A(i,j)=input('..');
end
end
disp('Entered Matrix is')
disp(A)
ifdet(A)==0
disp('Matrix is singular, Inverse does not exist')
break;
end
//Taking the augmented matrix
B=[A eye(3,3)]
disp('Augumented matrix is:')
disp(B)
// Making B(1,1)=1
B(1,:) = B(1,:)/B(1,1);
//Making B(2,1) and B(3,1)=0
B(2,:) = B(2,:) - B(2,1)*B(1,:);
B(3,:) = B(3,:) - B(3,1)*B(1,:);
//Making B(2,2)=1 and B(1,2), B(3,2)=0
B(2,:) = B(2,:)/B(2,2);
B(1,:) = B(1,:) - B(1,2)*B(2,:);
B(3,:) = B(3,:) - B(3,2)*B(2,:);
//Making B(3,3)=1 and B(1,3), B(2,3)=0
B(3,:) = B(3,:)/B(3,3);
```

```
B(1,:) = B(1,:) - B(1,3)*B(3,:);

B(2,:) = B(2,:) - B(2,3)*B(3,:);
disp('Augumented matrix after row operations is:')
disp(B)
B(:,1:3)=[ ]
disp('Inverse of the Matrix is')
disp(B)
```

OUTPUT :

```
 Enter a 3 by 3 matrix row-wise, make sure that diagonal elements are non zero
..1
..3
..5
..3
..2
..4
..5
..4
..2

 Entered Matrix is

    1.    3.    5.
    3.    2.    4.
    5.    4.    2.

 Augumented matrix is:

    1.    3.    5.    1.    0.    0.
    3.    2.    4.    0.    1.    0.
    5.    4.    2.    0.    0.    1.

 Augumented matrix after row operations is:

    1.    0.    0.  - 0.3     0.35     0.05
    0.    1.    0.    0.35  - 0.575    0.275
    0.    0.    1.    0.05    0.275  - 0.175

 Inverse of the Matrix is

  - 0.3     0.35     0.05
    0.35  - 0.575    0.275
    0.05    0.275  - 0.175

-->
```

EXPERIMENT NO.3

AIM :To find eigen values and eigen vectors of a square matrix

CODE:

```
clc
disp('Enter the 2 by 2 Matrix row-wise')
fori=1:2
for j=1:2
A(i,j)=input('..');
end
end
b=A(1,1)+A(2,2);
c=A(1,1)*A(2,2)-A(1,2)*A(2,1);

e1 = (b + sqrt(b^2 - 4*c))/2;
e2 = (b- sqrt(b^2 - 4*c))/2;

if A(1, 2)  ==0
X1 = [A(1,2); e1-A(1,1)];
X2 = [A(1,2); e2-A(1,1)];
elseif A(2, 1)  ==0
X1 = [e1-A(2,2); A(2,1)];
X2 = [e2-A(2,2); A(2,1)];
else
X1 = [1; 0];
X2 = [0; 1];
end
disp('First Eigen value is:');
disp(e1)
disp('First Eigen vector is:');
disp (X1)
disp('Second Eigen value is:');
disp(e2)
disp('Second Eigen vector is:');
disp (X2)
```

OUTPUT:

```
 Enter the 2 by 2 Matrix row-wise
..4
..-5
..1
..-2

 First Eigen value is:

    3.

 First Eigen vector is:

    1.
    0.

 Second Eigen value is:

  - 1.

 Second Eigen vector is:

    0.
    1.

-->
```

EXPERIMENT NO.4

AIM :Curve Fitting

1.To fit a straight line from a given data

2.To fit a parabola from a given data

CODE:

1. For fitting a straight line to a given set of data points (x,y)

```
clc;
n =input('Enter the no. of pairs of values (x,y):')
disp('Enter the values of x:')
fori=1:n
x(i)=input('..')
end
disp('Enter the corresponding values of y:')
fori=1:n
y(i)=input('..')
end
sumx=0; sumx2=0; sumy=0; sumxy=0
fori=1:n
sumx=sumx+x(i);
sumx2=sumx2+x(i)*x(i);
sumy=sumy+y(i);
sumxy=sumxy+x(i)*y(i);
end
A=[sumx n; sumx2 sumx];
B=[sumy;sumxy];
C=inv(A)*B
printf("The line of best fit is y =(%g)x+(%g)",C(1,1),C(2,1))
```

OUTPUT :

```
Enter the no. of pairs of values (x,y):4

 Enter the values of x:
(.)50
(.)70
(.)100
(.)120

 Enter the corresponding values of y:
(.)12
(.)15
(.)21
(.)25
The line of best fit is y =(0.187931)x+(2.27586)
-->
```

2. For fitting a parabola to given set of data points(x,y)

```
clc;
n =input('Enter the no. of pairs of values (x,y):')
disp('Enter the values of x:')
fori=1:n
x(i)=input('(.)')
end
disp('Enter the corresponding values of y:')
fori=1:n
y(i)=input('(.)')
end
sumx=0; sumx2=0; sumx3=0; sumx4=0; sumy=0; sumxy=0;
sumx2y=0;
fori=1:n
sumx=sumx+x(i);
sumx2=sumx2+x(i)*x(i);
sumx3=sumx3+x(i)*x(i)*x(i);
```

```
sumx4=sumx4+x(i)*x(i)*x(i)*x(i);
sumy=sumy+y(i);
sumxy=sumxy+x(i)*y(i);
sumx2y=sumx2y+x(i)*x(i)*y(i);
end
A=[sumx2 sumx n; sumx3 sumx2 sumx; sumx4 sumx3 sumx2];
B=[sumy;sumxy;sumx2y];
C=inv(A)*B
printf("The fitted parabola is
y=(%g)x^2+(%g)x+(%g)",C(1,1),C(2,1),C(3,1))
```

OUTPUT

```
Enter the no. of pairs of values (x,y):5

 Enter the values of x:
(.)-3
(.)-2
(.)0
(.)3
(.)4

 Enter the corresponding values of y:
(.)18
(.)10
(.)2
(.)2
(.)5
The fitted parabola is y=(0.872832)x^2+(-2.64547)x+(1.82466)
-->
```

EXPERIMENT NO. 5

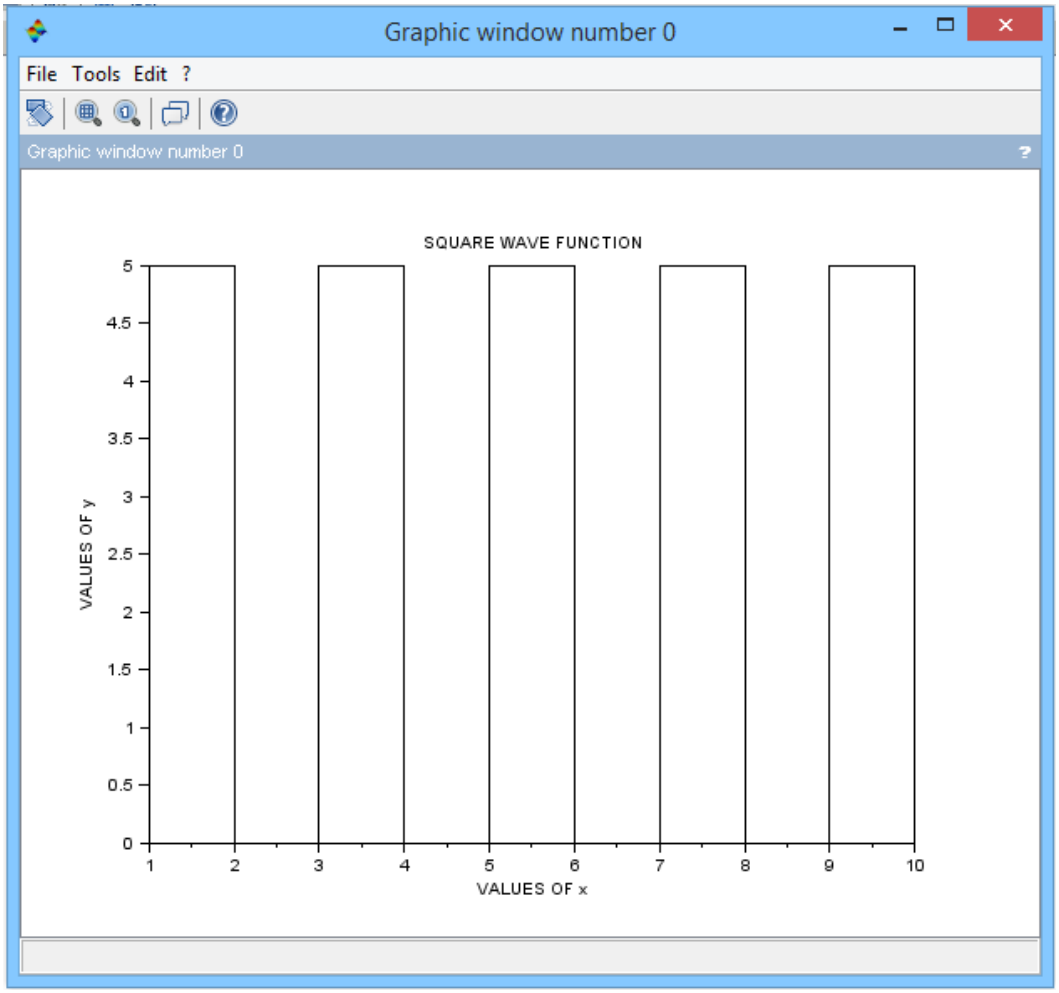
AIM: To plot unit step function and square wave function

CODE:

1.SQUARE WAVE FUNCTION

```
clc
x = [1 2 3 4 5 6 7 8 9 10 ];
y = [5 0 5 0 5 0 5 0 5 0 ];
plot2d2(x,y)
xlabel('VALUES OF x');
ylabel('VALUES OF y');
title('SQUARE WAVE FUNCTION');
```

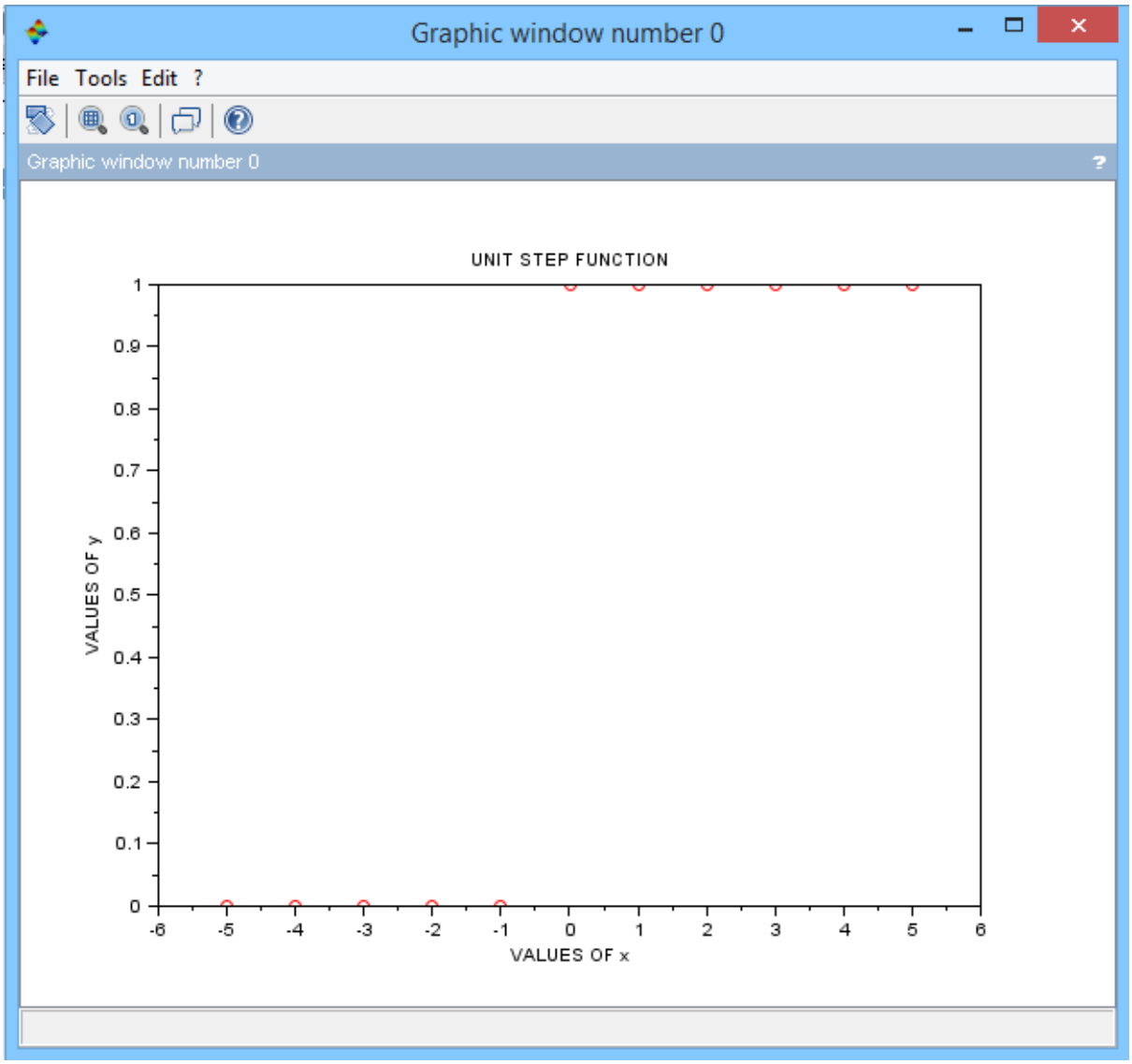
OUTPUT :



2.UNIT STEP FUNCTION

```
clc
x = [-1 -2 -3 -4 -5 0 1 2 3 4 5 ];
y = [0 0 0 0 0 1 1 1 1 1 1 ];
plot(x,y, 'ro')
xlabel(' VALUES OF x');
ylabel('VALUES OF y');
title('UNIT STEP FUNCTION');
```

OUTPUT :



EXPERIMENT NO.6

AIM:To find the solution of algebraic and transcendental Equations using

(a)Bisection Method

(b)Newton Raphson Method

CODE:

(A) BISECTION METHOD

```
clc
deff('y=f(x)','y=x^3+x^2-3*x-3')
a =input('Enter initial interval value: ');
b =input('Enter final interval value: ');

fa = f(a);
fb = f(b);
if sign(fa) == sign(fb)

disp('f must have different signs at the endpoints a and b')
error
end
e=input(' Answer correct upto : ');
iter=0;
printf('Iteration \t a \t\t b \t \t root \t \t f(root)\n')
while abs(a-b)>2*e
root = (a + b)/2
printf(' %i\t\t %f \t %f \t %f \t %f \n' ,iter,a,b,root,f(root))
if (root)*f(a) > 0
        a = root
else
b = root
end
iter=iter+1
end
```

printf("\n\n The solution of given equation is %f after %i iterations", root,iter-1)

OUTPUT :

```
Enter initial interval value: 1
Enter final interval value: 2
 Answer correct upto : 0.00001
Iteration        a               b               root            f(root)
 0               1.000000        2.000000        1.500000        -1.875000
 1               1.000000        1.500000        1.250000        -3.234375
 2               1.000000        1.250000        1.125000        -3.685547
 3               1.000000        1.125000        1.062500        -3.859131
 4               1.000000        1.062500        1.031250        -3.933563
 5               1.000000        1.031250        1.015625        -3.967770
 6               1.000000        1.015625        1.007813        -3.984130
 7               1.000000        1.007813        1.003906        -3.992126
 8               1.000000        1.003906        1.001953        -3.996078
 9               1.000000        1.001953        1.000977        -3.998043
 10              1.000000        1.000977        1.000488        -3.999022
 11              1.000000        1.000488        1.000244        -3.999511
 12              1.000000        1.000244        1.000122        -3.999756
 13              1.000000        1.000122        1.000061        -3.999878
 14              1.000000        1.000061        1.000031        -3.999939
 15              1.000000        1.000031        1.000015        -3.999969


 The solution of given equation is 1.000015 after 15 iterations
-->
```

(B) NEWTON RAPHSON METHOD :

```
clc
deff('y=f(x)','y=x^3+x^2-3*x-3')
deff('y=df(x)','y=3*x^2+2*x-3')
x(1)=input('Enter Initial Guess:');
e= input("Answer correct upto : ");
fori = 1 : 100
x(i+1)=x(i)-f(x(i))/df(x(i));
err(i)=abs((x(i+1)-x(i))/x(i));
if err(i) < e
break;
```

```
end
end
printf('The solution is %f',x(i))
```

OUTPUT :

```
Enter Initial Guess:1
Answer correct upto : 0.000001
The solution is 1.732051
-->
```