MATLAB FILE

List of Programs in MATLAB

- Write a program in MATLAB for following problems:

1. To draw the tangent line at point on a given curve $y = 1 + x^2$ at the point (2,5) and also find the Radius of curvature at that point.
2. Graphically compare the function sin x and its Taylors series expansion $(up\ to\ degree\ 10)$ in the neighbourhood of 1.
3. Using inbuilt ode solvers ode23 and ode45 find $y(0.3)$, where y is the solution of the following initial value problem and hence compare this value with the actual answer:
$$\frac{dy}{dx} = \frac{y}{x}, \quad y(0) = 1.$$
4. Sketch the region enclosed by the curve $f(x) = x^2 - 3x^2 + 3x$ and $g(x) = x^2$ and find the area of the enclosed region.
5. Solve by the variation of parameters: $D^2y + 4y = \sec x$
6. Plot the surface defined by the function $f(x, y) = -xy\, e^{-2(x^2+y^2)}$ on the domain $-2 \le x \le 2\ and\ -2 \le y \le 2$. Find the values and locations of the maxima and minima of the function.
7. Write a program to show the consistency and inconsistency of the system of linear equations. If system is consistence then solve the given system of equation for unique/ infinite solution (with degree of freedom).
8. Write a program to determine the largest two eigen values of the following matrix:
$$\begin{pmatrix} 1 & 0 & 0 & 1 & -1 \\ 0 & 2 & 3 & 5 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 6 & 8 & 1 & 2 & -2 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$
9. Determine the characteristic polynomial of a matrix by evaluating the polynomial $P(\lambda)$ at $n + 1$ points. Then fit a polynomial to these points.

QUESTION 1:

To draw the tangent line at point on a given curve $\mathbf{y = 1 + x^2}$ at the point (2,5) and also find the Radius of curvature at that point.

THEORY:

Radius of curvature is given as

$$\rho(t) = \frac{|1 + f'^2(t)|^{3/2}}{|f''(t)|}.$$

FUNCTIONS USED:

ezplot(FUN) plots the function FUN(X) over the default domain -2*PI < X < 2*PI, where FUN(X) is an explicitly defined function of X.

ezplot(FUN2) plots the implicitly defined function FUN2(X,Y) = 0 over the default domain -2*PI < X < 2*PI and -2*PI < Y < 2*PI.

ezplot(FUN,[A,B]) plots FUN(X) over A < X < B.

ezplot(FUN2,[A,B]) plots FUN2(X,Y) = 0 over A < X < B and A < Y < B.

ezplot(FUN2,[XMIN,XMAX,YMIN,YMAX]) plots FUN2(X,Y) = 0 over XMIN < X < XMAX and YMIN < Y < YMAX.

sqrt(X) is the square root of the elements of X. Complex results are produced if X is not positive.

CODE:

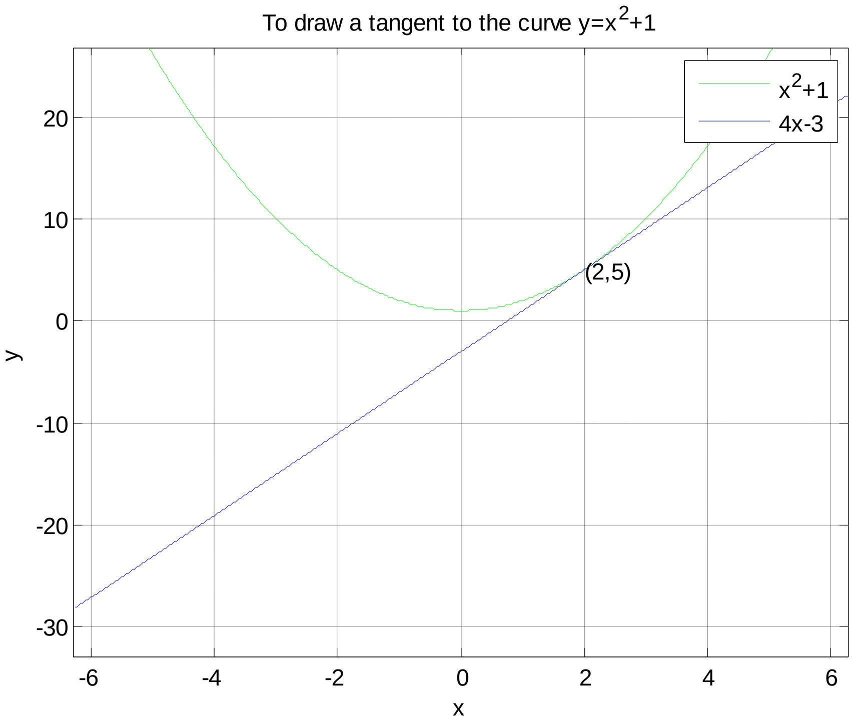
```
>> syms x; % defining symbol x

>> y1=1+x^2;

>> h=ezplot(y1); % plotting graph

>> set(h,'color','g'); % setting color of graph to green

>> grid % forming grid on graph

>> ydot1=diff(y1,x); % differentiating y1 with respect to x

>> s=subs(ydot1,x,2); % substituting values of x=2
```

```
>> y2=s*(x-2)+5;

>> hold on

>> k=ezplot(y2);

>> set(k,'color','b')

>> text(2,5,'(2,5)') % defining intersecting point on graph

>> hold off

>> title('To draw a tangent to the curve y=x^2+1')

>> xlabel('x')

>> ylabel('y')

>> legend('x^2+1','4x-3');

>> ydot2=diff(y2,x,2);

>> p=sqrt((1+(ydot1)^2)^3);

>> rc=p/ydot2;

>> rc=subs(p/ydot2,x,2) % to find radius of curvature
```

OUTPUT:

```
rc =

  35.0464
```

GRAPH:



To draw a tangent to the curve y=x^2+1
x^2+1
4x-3
(2,5)
20
10
0
-10
-20
-30
y
-6
-4
-2
0
2
4
6
x

QUESTION 2:

Graphically compare the function sin x and its Taylors series expansion $(\mathbf{up\ to\ degree\ 10})$ in the neighbourhood of $\mathbf{1}$.

THEORY:

The Taylor series of a real or complex-valued function $f(x)$ that is infinitely differentiable in a neighborhood of a real or complex number a is the power series

$$f(a)+\frac{f'(a)}{1!}(x-a)+\frac{f''(a)}{2!}(x-a)^2+\frac{f^{(3)}(a)}{3!}(x-a)^3+\cdots.$$

which can be written in the more compact sigma notation as

$$\sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n$$

where $n!$ denotes the factorial of n and $f^{(n)}(a)$ denotes the nth derivative of f evaluated at the point a. The derivative of order zero f is defined to be f itself and $(x-a)^0$ and 0! are both defined to be 1. In the case that $a = 0$, the series is also called a Maclaurin series.

FUNCTIONS USED:

taylor(f,x,a)

is the fifth order Taylor polynomial approximation of f with respect to x about the point a. x and a can be vectors. If x is a vector and a is scalar, then a is expanded into a vector of the same size as x with all components equal to a. If x and a both are vectors, then they must have same length.

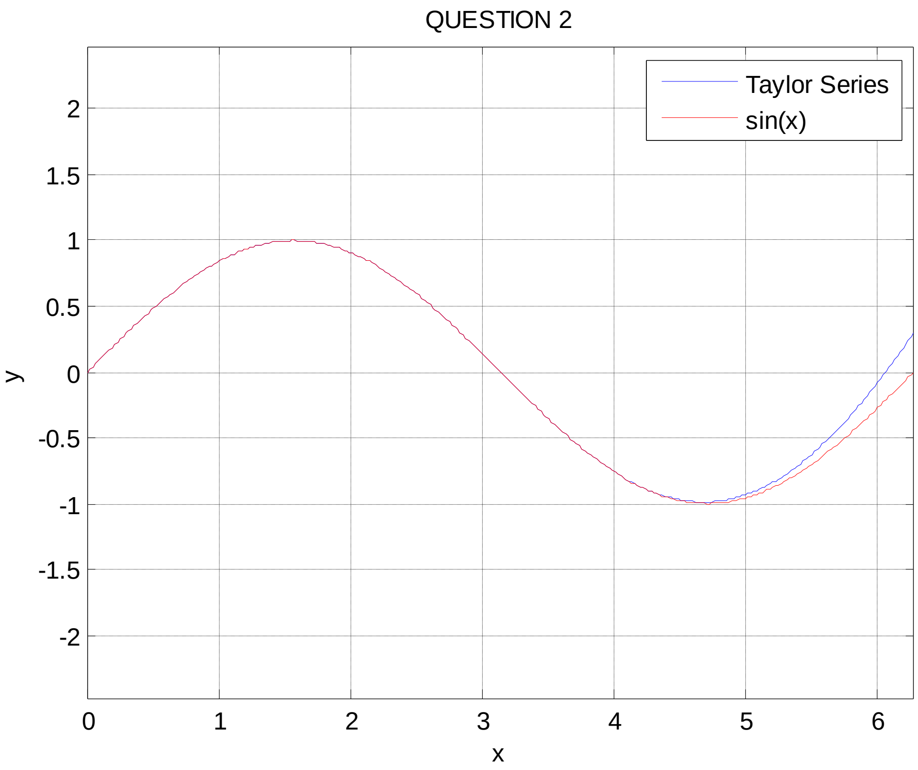
CODE:

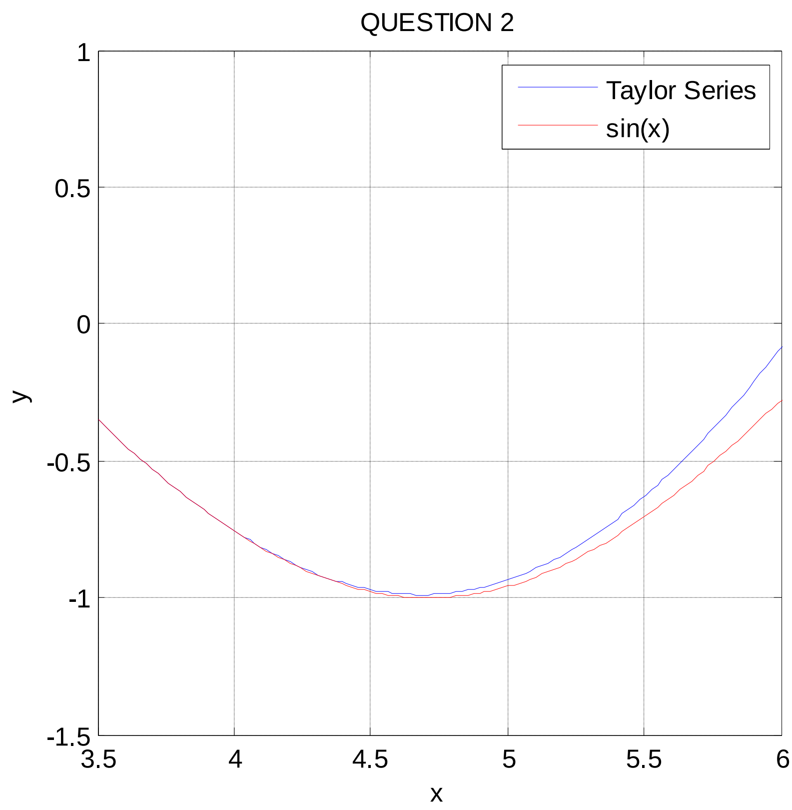
```
>> syms f x; % creates symbolic object

>> f=taylor(sin(x),11,1) % taylor is the standard function to define taylor series %



f =
```

sin(1) - (sin(1)*(x - 1)^2)/2 + (sin(1)*(x - 1)^4)/24 - (sin(1)*(x - 1)^6)/720 + (sin(1)*(x - 1)^8)/40320 - (sin(1)*(x - 1)^10)/3628800 + cos(1)*(x - 1) - (cos(1)*(x - 1)^3)/6 + (cos(1)*(x - 1)^5)/120 - (cos(1)*(x - 1)^7)/5040 + (cos(1)*(x - 1)^9)/362880

```
>> h=ezplot(x,f); % plots the graph

>> set(h,'color','b'); %sets the color of curve to blue

>> grid

>> hold on;

>> y=sin(x);

>> k=ezplot(x,y);

>> set(k,'color','r');

>> hold off;

>> title('QUESTION 2'); % assigns titile to the figure as QUESTION 2

>> legend('Taylor Series','sin(x)')

>> axis([3.5,6,-1.5,1])
```

GRAPH:


QUESTION 2
Taylor Series
sin(x)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
y
0
1
2
3
4
5
6
x


QUESTION 2
Taylor Series
sin(x)
1
0.5
0
-0.5
-1
-1.5
y
3.5
4
4.5
5
5.5
6
x

QUESTION 3:

Using inbuilt ode solvers ode23 and ode45 find $\mathbf{y(0.3)}$, where $\mathbf{y}$ is the solution of the following initial value problem is and hence compare this value with the actual answer

$$\frac{dy}{dx} = \frac{y}{x}, \quad y(0) = 1.$$

THEORY:

odefn are Runge-Kutta methods.

FUNCTIONS USED:

ode23

Solve non-stiff differential equations, low order method. [TOUT,YOUT] = ode23(ODEFUN,TSPAN,Y0) with TSPAN = [T0 TFINAL] integrates the system of differential equations y' = f(t,y) from time T0 to TFINAL with initial conditions Y0. ODEFUN is a function handle. For a scalar T and a vector Y, ODEFUN(T,Y) must return a column vector corresponding to f(t,y). Each row in the solution array YOUT corresponds to a time returned in the column vector TOUT. To obtain solutions at specific times T0,T1,...,TFINAL (all increasing or all decreasing), use TSPAN = [T0 T1 ... TFINAL].

INPUT:

1. First order differential equation is ydot=y/x
2. Creating a function file diffy.m
   function ydot=diffy(x,y);
   ydot=y/x;

3. **On command window**

(i) Using ode23

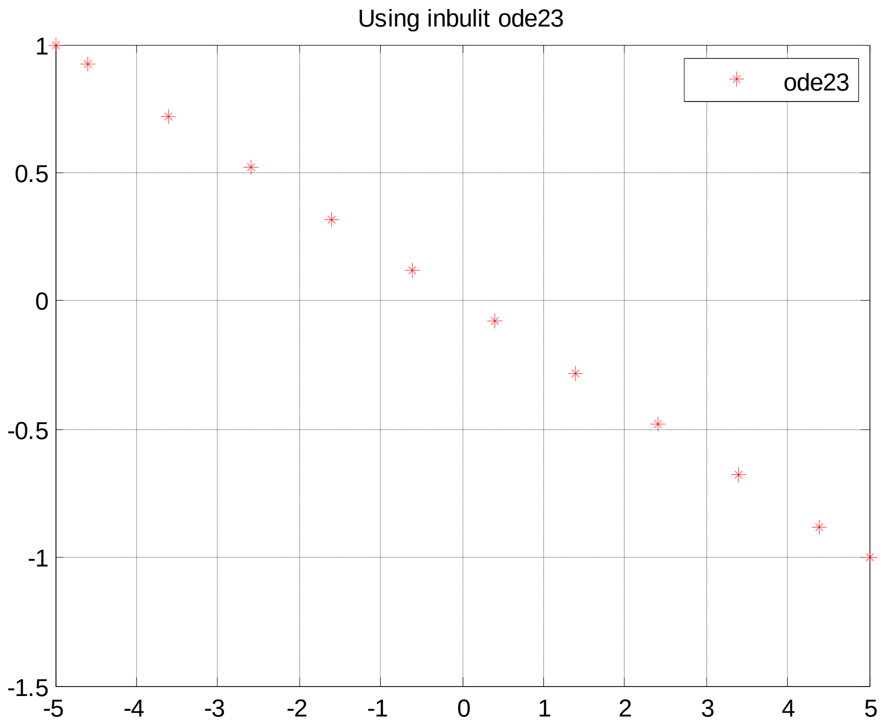
```
>> yspan=[-5 5];

>> y0=1;

>> [x,y]=ode23('diffy',yspan,y0);

>> plot(x,y,'r*');

>> y1=y(3)
```

```
>> grid

>> title('Using inbulit ode23');

>> legend('ode23');
```

OUTPUT:

```
y1 =

 0.7200
```

GRAPH:



(ii) Using ode45

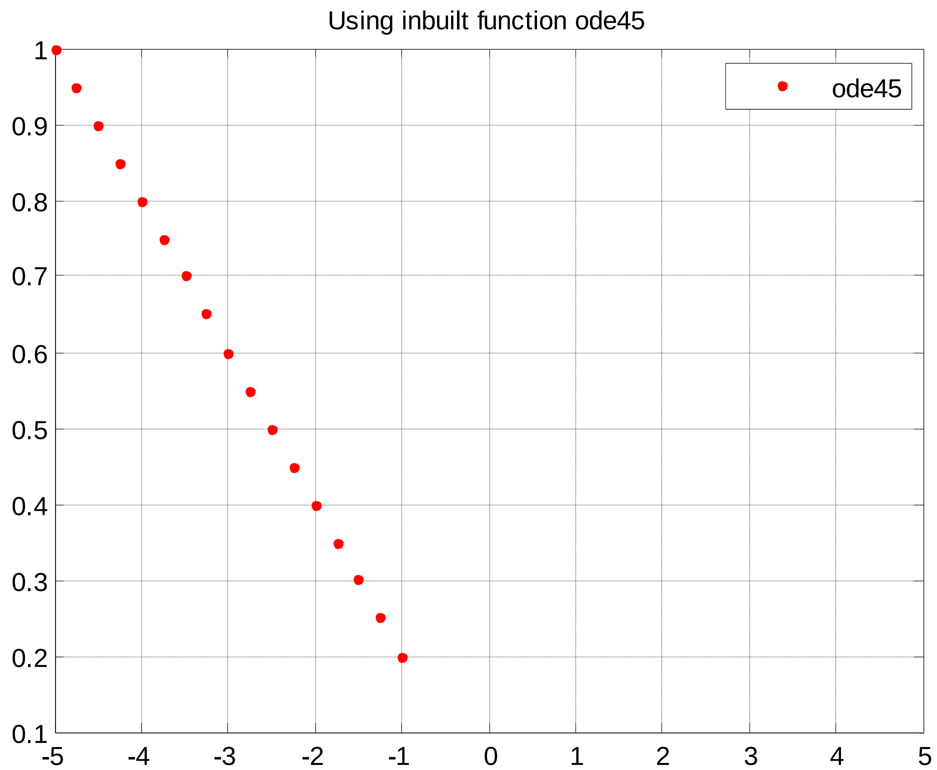
```
>> [x,y]=ode45('diffy',yspan,y0);

>> plot(x,y,'r.')

>> y2=y(3)

>> title('Using inbuilt function ode45');
```

```
>> legend('ode45')

>> grid
```

OUTPUT:

```
y2 =

   0.9000
```

GRAPH:



QUESTION 4:

Sketch the region enclosed by the curve $\mathbf{f(x) = x^2 - 3x^2 + 3x}$ and $\mathbf{g(x) = x^2}$ and find the area of the enclosed region.

CODE:

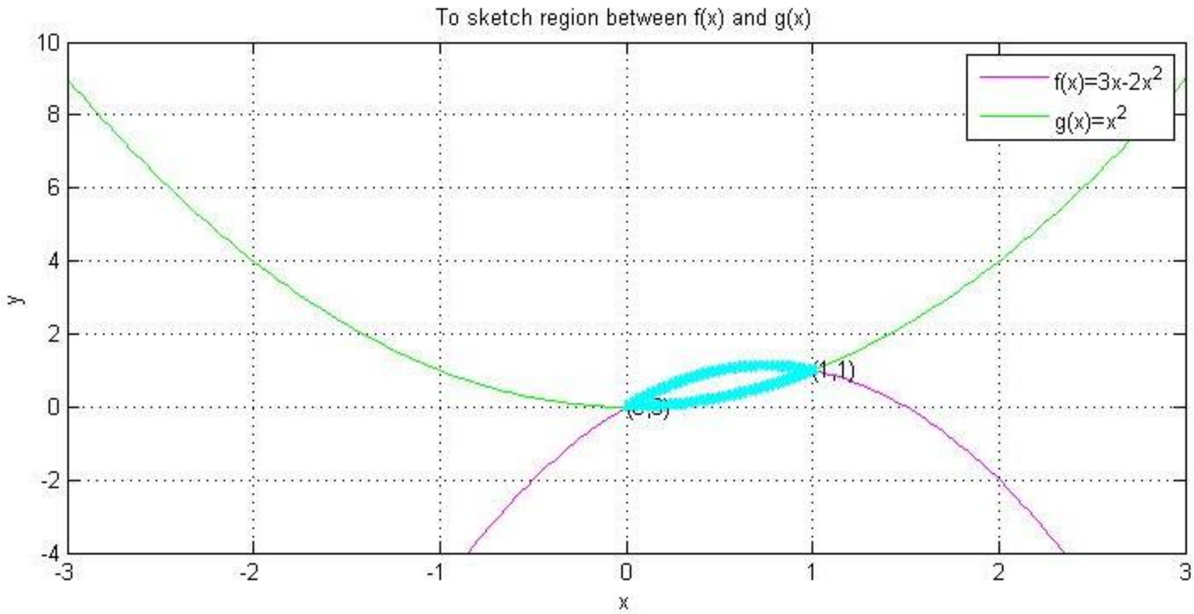
```
>> syms x;

>> f= x^2-3*x^2+3*x;

>> h=ezplot(f);

>> set(h,'color','m')

>> g=x^2;

>> hold on

>> k=ezplot(g);

>> set(k,'color','g')

>> hold off

>> grid

>> syms r;

>> r=int(f,0,1)-int(g,0,1) % int is used to integrate
```

OUTPUT:

r =

½

GRAPH:



QUESTION 5:

Solve by the variation of parameters: $\mathbf{D^2y + 4y = \sec x}$

THEORY:

We solve non homogeneous linear differential equation by variation of parameters using formula

$$\sum_{i=1}^{n} \int \frac{W_i(x)}{W(x)} dx \, y_i(x).$$

Where W_i is wroskian of i^{th} solution and W is the wroskian of the complementary solutions

CODE:

```
>> syms x t;

>> yc=dsolve('D2y=-4*y') % finding complementary function. dsolve is used to find complementary solutions of a differential equation%

 yc =

C2*cos(2*t) + C3*sin(2*t) % complementary function

>> ydot1=diff('cos(2*t)',t) % differentiable function for y1=cos(2t)

 ydot1 =

-2*sin(2*t)

>> ydot2=diff('sin(2*t)',t) % differentiable function for y2=sin2t

ydot2 =

2*cos(2*t)

>> w=[cos(2*t) sin(2*t);ydot1 ydot2] % wronskian of y1 and y2

w =

[    cos(2*t),   sin(2*t)]

[ -2*sin(2*t), 2*cos(2*t)]

>> r=det(w) % determinant of w
```

```
r =

2*cos(2*t)^2 + 2*sin(2*t)^2

>> w1=[0,sin(2*t);sec(t),ydot2]

w1 =

[       0,   sin(2*t)]

[ 1/cos(t), 2*cos(2*t)]

>> r1=det(w1)

r1 =

-sin(2*t)/cos(t)

>> w2=[cos(2*t) 0;ydot1 sec(t)]

w2 =

[    cos(2*t),        0]

[ -2*sin(2*t), 1/cos(t)]

>> r2=det(w2)

r2 =

cos(2*t)/cos(t)

>> u=int(r1/r)

u =

cos(t)

>> v=int(r2/r)

v =

log(sin(t) - 1)/4 - log(sin(t) + 1)/4 + sin(t)

>> yp=u*cos(2*t)+v*sin(2*t) % particular solution for D.E.

yp =

sin(2*t)*(log(sin(t) - 1)/4 - log(sin(t) + 1)/4 + sin(t)) + cos(2*t)*cos(t)

>> y=yc+yp % general solution
```

OUTPUT:

y =

sin(2*t)*(log(sin(t) - 1)/4 - log(sin(t) + 1)/4 + sin(t)) + cos(2*t)*cos(t) + C2*cos(2*t) + C3*sin(2*t)

QUESTION 6:

Plot the surface defined by the function f(x, y) = $-xye^{-2(x^2+y^2)}$ on the domain −2 ≤ x ≤ 2 and − 2 ≤ y ≤ 2. Find the values and locations of the maxima and minima of the function.

CODE:

```
>> syms x y;

>> [x,y]=meshgrid(-2:0.1:2,-2:0.1:2); % meshgrid  replicates the grid vectors x and y to produce a full grid.%

>> f=-x.*y.*exp(-2*(x.^2+y.^2));

>> figure(1)

>> mesh(x,y,f),xlabel('X'),ylabel('y'),grid % mesh is used to plot the surfaces%

>> figure(2)

>> contour(x,y,f)

>> xlabel('X'),ylabel('y'),grid,hold on

% finding and locating minimum and maximum values of f%

>> fmax=max(max(f)) % finding maximum value of f

 fmax =

   0.0920

>> kmax=find(f==fmax) % locating maximum value of f

kmax =

        641

       1041

>> pos=[x(kmax) y(kmax)]

pos =

  -0.5000    0.5000

   0.5000   -0.5000
```

```
>> plot(x(kmax),y(kmax),'*') % plotting maximum values on graph

>> text(x(kmax),y(kmax),'MAXIMUM')

>> fmin=min(min(f)) % finding minimum value of f

fmin =

  -0.0920

>> kmin=find(f==fmin) % locating min value of

kmin =

      631

     1051

>> pos1=[x(kmin) y(kmin)]

pos1 =

  -0.5000   -0.5000

   0.5000    0.5000

>>plot(x(kmin),y(min),'*') % plotting minimum value of f in graph

>> text(x(kmin),y(kmin),'MINIMUM')

>> hold off
```

OUTPUT:

```
fmax =

   0.0920

fmin =

  -0.0920
```

GRAPHS:

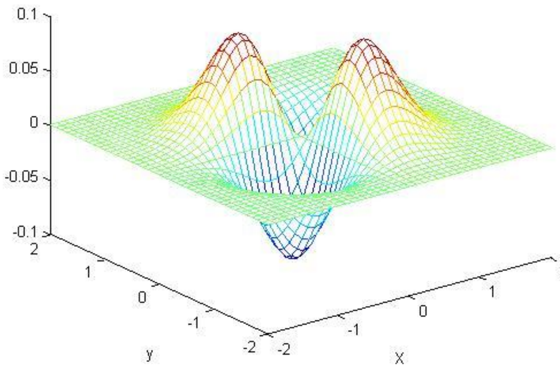


FIGURE 1

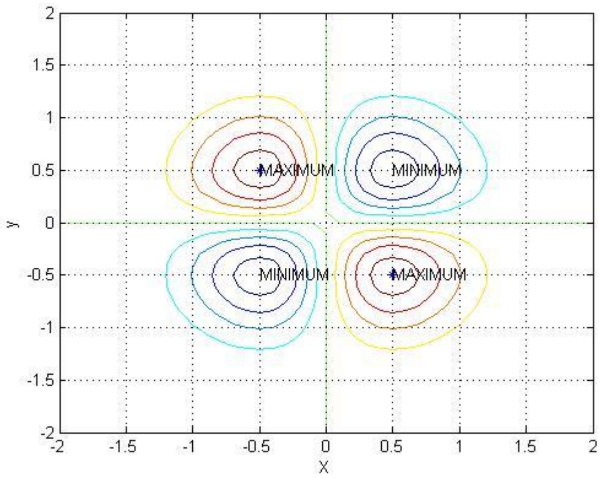


FIGURE 2

QUESTION 7:

Write a program to show the consistency and inconsistency of the system of linear equations. If system is consistence then write a program to solve the given system of equation for unique/ infinite solution with degree of freedom.

INPUT AND OUTPUT:

```
%First we need to create function file solns.m%

function [val] =sols( A,B )

rankA= rank(A); % to find rank of a matrix

rankAB= rank([A B]);

[m,n]=size(A); % to find size of matrix

disp(['There are ' int2str(m) ' equations']) % int2str() converts integer to string

disp(['with' int2str(n) ' variables'])

if rankA~=rankAB

   disp('There are no solutions')

elseif rankA==n

      disp('There is a unique solution')

   elseif rankA<n

dof =(n-rank(A));

disp(['There are infinite number of solutions with ' int2str(dof) ' degree of freedom'])

end
```

On command window:

```
>> a1=[1];b1=1;

>> a2=[1 0;0 1]; b2=[1;2];

>> a3=[1 1; 1 -1]; b3=[1;2];

>> a4=[1 1; 1 1];b4=[1;2];
```

```
>> a5=[1 1; 1 1];b5=[1;1];

>> sols(a1,b1)

There are 1 equations

with1 variables

There is a unique solution

>> sols(a2,b2)

There are 2 equations

with2 variables

There is a unique solution

>> sols(a3,b3)

There are 2 equations

with2 variables

There is a unique solution

>> sols(a4,b4)

There are 2 equations

with2 variables

There are no solutions

>> sols(a5,b5)

There are 2 equations

with2 variables

There are infinite number of solutions with 1 degree of freedom

>> a1\b1

ans =

    1

>> a2\b2

ans =
```

```
     1
     2
>> a3\b3
ans =
    1.5000
   -0.5000
>> a4\b4
Warning: Matrix is singular to working precision.
ans =
  -Inf
   Inf
>> a5\b5
Warning: Matrix is singular to working precision.
ans =
   NaN
   NaN
```

QUESTION 8:

Write a program to determine the largest two Eigen values of the following matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & -1 \\ 0 & 2 & 3 & 5 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 6 & 8 & 1 & 2 & -2 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

CODE:

```
>> A=[1 0 0 1 -1;...
0 2 3 5 0;...
-1 0 0 0 1;...
6 8 1 2 -2;...
1 1 1 1 1];
>> [v,d]=eigs(A,2,'LM') % eigs is used to select only certain eigen values.LM is used to find largest magnitude eigen values . v stores eigen vectors of A in columns and d stores eigenvalues in diagonal%
```

OUTPUT:

```
v =

   -0.0769    0.1400

   -0.6050    0.5780

   -0.0143    0.0270

   -0.7674   -0.8034

   -0.1974    0.0100
```

d =

```
    8.4127         0
         0   -4.8097
```

QUESTION 9:

Determine the characteristic polynomial of a matrix by evaluating the polynomial P(λ) at n + 1 points. Then fit a polynomial to these points.

CODE:

```
%First we need to create charpoly.m function file%

 function [ co ] = charpoly( A )

[m n]=size(A)

if m~=n% checking that matrix is square or not

   disp('matrix is not square')

   co=[]

   return

end

for i=1:(n+1)

   x(i)=(i-1)*pi/n;

   y(i)=det(A-x(i)*eye(n)); % det is used to find determinant of a matrix. eye(n) creates a
singular square matrix of order n %

end

co=polyfit(x,y,n); % polyfit used to fit a polynimial
```

On command window:

```
>> A=[1 2 3; 4 5 6 ; 7 8 9];

>> charpoly(A) % called function file

m =

    3

n =

    3
```

```
ans =

   -1.0000   15.0000   18.0000   -0.0000

>> z=length(ans);

>> syms x;

>> f=0;

>> i=4;

>> for y=1:1:z

f=f+ans(y).*x.^(i-1);

i=i-1;

end

>> f
```

OUTPUT:

```
 f =

- x^3 + 15*x^2 + 18*x
```

QUESTION 10:

Given two polynomials f = 15x^3 – 7x^2 + 2x + 4 and g = 9x^2 – 17x +3, do the following problems:

A. **Find the product of f and g.**
B. **Find the quotient and remainder of f divided by g.**
C. **Find the roots of g and f.**
D. **Find the value of f at x=3 and for g at x=2i**

FUNCTIONS USED:

1. **conv** Convolution and polynomial multiplication.

C = conv(A, B) convolves vectors A and B. The resulting vector is length MAX([LENGTH(A)+LENGTH(B)-1,LENGTH(A),LENGTH(B)]). If A and B are vectors of polynomial coefficients, convolving them is equivalent to multiplying the two polynomials.

2. **roots** Find polynomial roots.

roots(C) computes the roots of the polynomial whose coefficients are the elements of the vector C. If C has N+1 components, the polynomial is C(1)*X^N + ... + C(N)*X + C(N+1).

3. **polyval** Evaluate polynomial.

Y = polyval(P,X) returns the value of a polynomial P evaluated at X. P is a vector of length N+1 whose elements are the coefficients of the polynomial in descending powers.

Y = P(1)*X^N + P(2)*X^(N-1) + ... + P(N)*X + P(N+1)

If X is a matrix or vector, the polynomial is evaluated at all points in X.

4. **deconv** Deconvolution and polynomial division.

[Q,R] = deconv(B,A) deconvolves vector A out of vector B. The result is returned in vector Q and the remainder in vector R such that B = conv(A,Q) + R.

If A and B are vectors of polynomial coefficients, deconvolution is equivalent to polynomial division. The result of dividing B by A is quotient Q and remainder R.

CODE:

```
>> f=[15 -7 2 4]; % Defining the polynomial f

>> g=[9 -17 3]; % Defining the polynomial g

>> fg=conv(f,g) % Finding the product of f and g
```

OUTPUT A:

```
fg =

  135  -318   182   -19   -62    12

>> [Quotient, Remainder]=deconv(f,g) % Finding the quotient and remainder of f divided by g
```

OUTPUT B:

```
Quotient =

   1.6667    2.3704


Remainder =

        0         0   37.2963   -3.1111

>> roots(g) % Finding the roots of g
```

OUTPUT C:

```
ans =

   1.6919

   0.1970


>> roots(f) % Finding the roots of f

ans =

  0.4672 + 0.5933i

  0.4672 - 0.5933i

 -0.4676
```

```
>> polyval(f,3) % Finding the value of f at x=3
```

OUTPUT D:

```
ans =

   352


>> polyval(g,-2i) % Finding the value of g at x=2i

ans =

 -33.0000 +34.0000i
```