Table of Contents

Laboratory - 0

MATLAB FUNDAMENTALS

Matlab stands for Matrix Laboratory.

Matlab is a high performance language for technical computing. Matlab is an interactive system whose ***data element is an array*** that does not require dimensioning.

It integrates computation, visualization and programming in an easy to use environment where problems and solutions are expressed in mathematical notations.

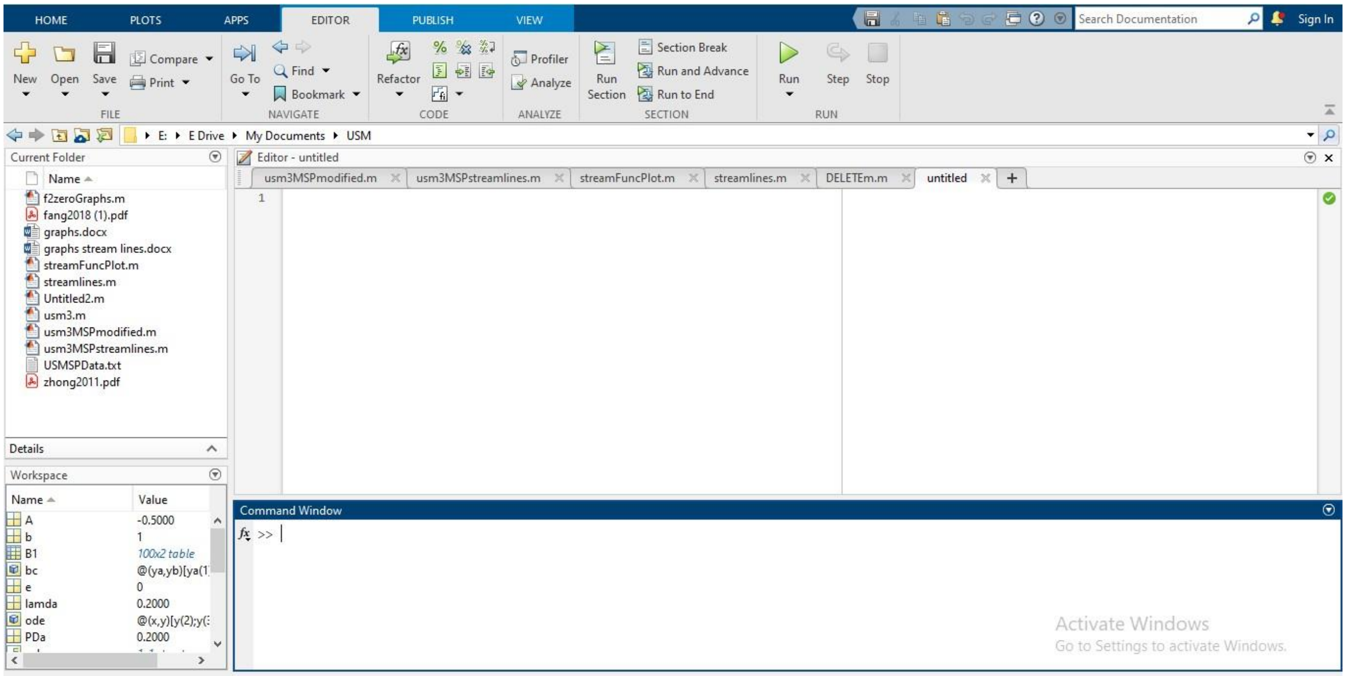
Applications of Matlab include
i) Math and Computation
ii) Algorithm development
iii) Modeling, Simulation and Prototyping
iv) Data analysis, exploration and visualization
v) Scientific and Engineering graphics
vi) Application development

Matlab features a family of application specific solution called toolboxes. Toolboxes allow us to learn and apply specialized technology. Toolboxes are comprehensive collection of Matlab functions (M files) that extend the Mablab environment to solve particular class of problems. Areas in which toll boxes are available include Fuzzy logic, Neural Networks, Signal Processing, Control Systems, Wavelets, Simulation and many more.

Matlab System consists of five parts, namely

1) Development Environment: It includes Matlab desktop, command window, a command history, a help browser, the workspace, files and search path.
2) Matlab Mathematical function library: It consists of all mathematical functions like sin, cos, log, ... till fast Fourier Transform
3) The Matlab Language: This is a high level matrix / array language with control flow statements, functions, data structures, input/output and so on. It allows us to write programs to perform the defined task.
4) Handle Graphics: Tool for visualization, image processing, animations, presentations of graphics.
5) Matlab Application:

The typical Matlab environment is as shown below.



Command window - opens with a command prompt in the beginning.
Calculator mode operates in a sequential way
Default output variable name is “ans”.

Assignment:

Scalars -

>> a = 4		(takes the value of a as 4 and display a = 4)
>> a = 4;		(takes the value of a as 4 and supresses the output)
>> b = 4 + i*2 or	b = 4 +j*2	assigning a complex number (in the output only i displays)

Exercise:

Perform all the arithmetic operations with the complex numbers.

Library functions

clc - clear the screen

clear - clear all the variables used so far in the command window

help NAME - Display help text in Command Window

(If NAME is not specified, help displays content relevant to your previous actions.)

who - lists the variables in the current workspace

sqrt(),
sin(), cos(), tan(), : Trigonometric functions in radians
sind(), cosd(), tand(), : Trigonometric functions in degrees
exp(), sinh(), cosh(), : Exponential or hypergeometric functions
log(), log10(), log2(), : Logarithmic functions
asin(), acos(), atan() : Inverse Trigonometric functions

Arrays: Collection of values represented by single value name.
Examples:
A=[1 2 3 4 5 6] or A=[1,2,3,4,5,6] a row matrix

B=[1; 2; 3; 4; 5; 6]
or
B=[1
2
3 a column matrix
4
5
6]

C=[1; 2; 3; 4; 5; 6]’ transposed matrix

D=[1 2 3;4 5 6] an array of size 2 x 3

Z=[0 0 0;0 0 0;0 0 0] a zero matrix of size 3 x 3

Z can also be generated by a command *zeros(m,n)*

U=[1 1 1; 1 1 1; 1 1 1] a unity matrix of size 3 x 3
U can also be generated by a command *ones(m,n)*

I=[1 0;0 1] or I=[1 0 0;0 1 0;0 0 1] Identity matrix of size 2 or 3.
I can also be generated by a command *eye*(n)

Colon (:) operator

It is used to generate a vector with the uniform spacing between the points

Syntax

<variable name> = a : n : b
where **a** is the initial value, **n** is the increment/decrement (default is 1), **b** is the final value

Example:

1) X=1:10

X =

1 2 3 4 5 6 7 8 9 10

2) Y=10:-1:1

Y =

10 9 8 7 6 5 4 3 2 1

3) Z=3.5:0.25:5.75

Z =

3.5000 3.7500 4.0000 4.2500 4.5000 4.7500 5.0000 5.2500 5.5000 5.7500

4) Z=3.5:0.25:5.7

Z =

3.5000 3.7500 4.0000 4.2500 4.5000 4.7500 5.0000 5.2500 5.5000

linspace

linspace Linearly spaced vector.

Syntax

linspace(X_1, X_2)
generates a row vector of 100 linearly equally spaced points between X_1 & X_2.
linspace(X_1, X_2, N)
generates N points between X_1 and X_2.

Note: for N = 1, linspace returns X_2.

Examples:

Generate vectors X,Y,Z using linspace command

X=1:10 → linspace(1,10,10)

Y=10:-1:1 → linspace(10,1,10)

Z=3.5:0.25:5.75 → linspace(3.5,5.75,10)

Matrix Operations:

Operator	Function	Command
+	Addition	A+B
-	Subtraction	A-B
*	Multiplication	A*B
/	Division	A/B (computes $A*B^{-1}$)
.*	Element by element multiplication	A.*B
./	Element by element division	A./B
inv	Inverse of a non-singular matrix	inv(A)

Logical operators

Symbol	Meaning
==	Equal
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
~=	Not equal to

Laboratory Exercises

1) Create an identity matrix of size 4.

2) Create zero matrix of order 3x4. Also, transpose the same.

3) Create two matrices A and B of same size and perform the above operations.

4) Multiply an identity matrix with A and verify A*I=I*A=A.

5) Compute the inverse of matrix A. Verify that $A*A^{-1}=A^{-1}*A=I$.

6) Let P be a matrix of size 5x6. Extract the elements of
   i) First row
   ii) First column
   iii) Second row 3rd to 6th elements
   iv) Second column 3rd to 5th elements.

Laboratory - 1

CARTESIAN AND POLAR PLOTS

Aim: To plot standard Cartesian curves in xy-plane (2D plots) and polar curves in rθ-plane.

Matlab commands: syms, plot, fplot, fimplicit, polarplot

Syntax: **syms Short-cut for constructing symbolic variables.**

syms arg1 arg2 ...

plot – creates 2D plot

plot(X,Y) plots vector Y versus vector X

fplot **Plot 2-D function**

fplot(FUN) plots the function FUN between the limits of the current axes, with a default of [-5 5].

fplot(FUN,LIMS)

fimplicit Plot implicit function

fimplicit(FUN) plots the curves where FUN(X,Y)==0 between the axes limits,
with a default range of [-5 5].

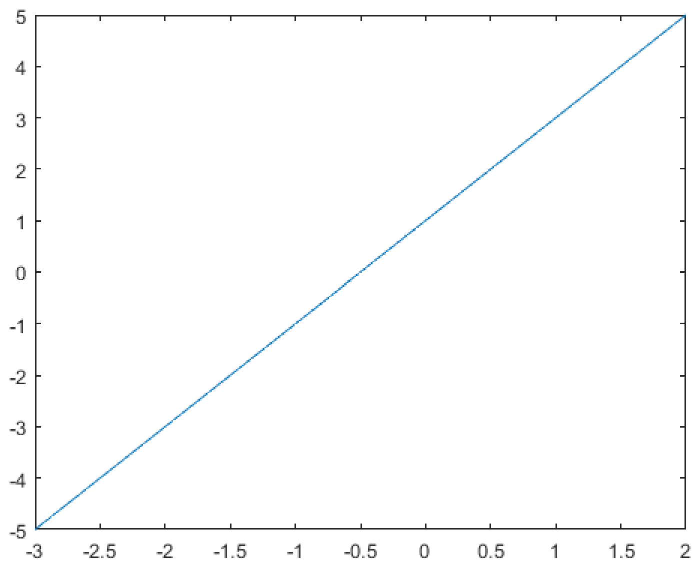
fimplicit(FUN,LIMS) uses the given limits. LIMS can be [XMIN XMAX YMIN YMAX]

polarplot Polar plot.

polarplot(TH,R) plots vector TH vs R. The values in TH are in radians

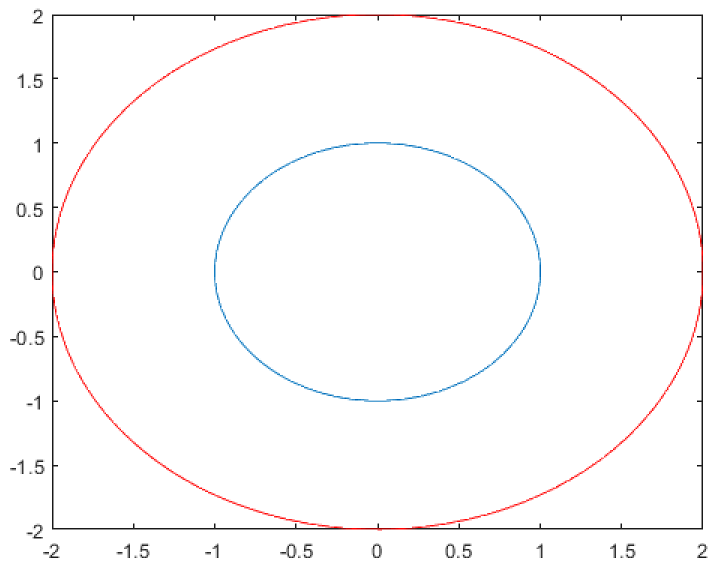
Task is to plot a straight line

```
syms x y
m=2;
c=1;
fimplicit(y == m*x + c)
```



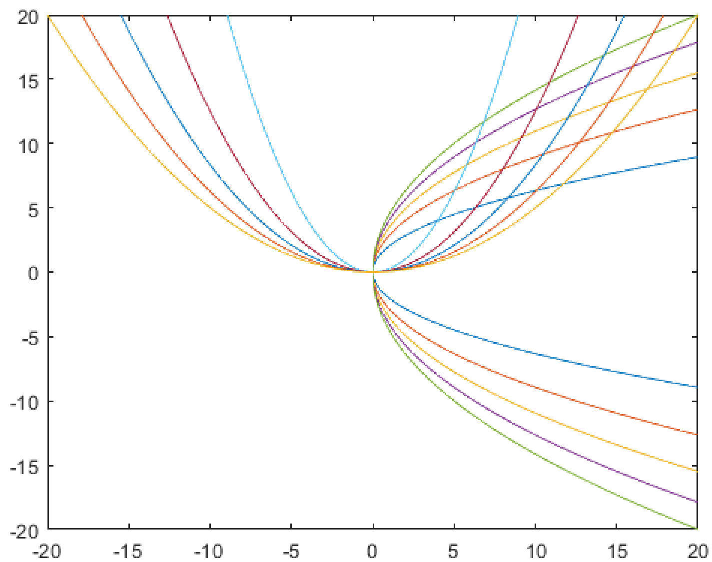
Task is to plot a circle of radius 1 and 2

```
syms x y
fimplicit(x^2 + y^2 == 1)
hold on
fimplicit(x^2 + y^2 == 4,'r')
hold off
```



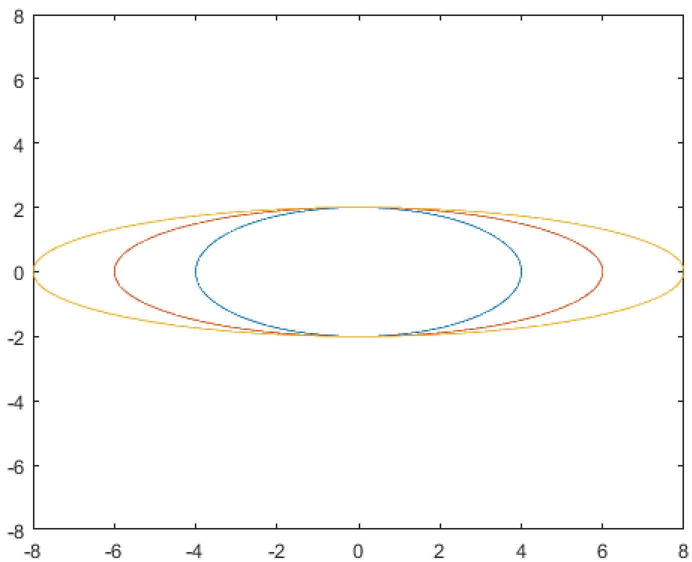
Task is to plot family of parabolas and mark the points of intersection.

```
syms x y
a=[1 2 3 4 5];
fimplicit(y^2 - 4*a*x == 0,[-20,20])
hold on
fimplicit(x^2 - 4*a*y == 0,[-20,20])
hold off
```



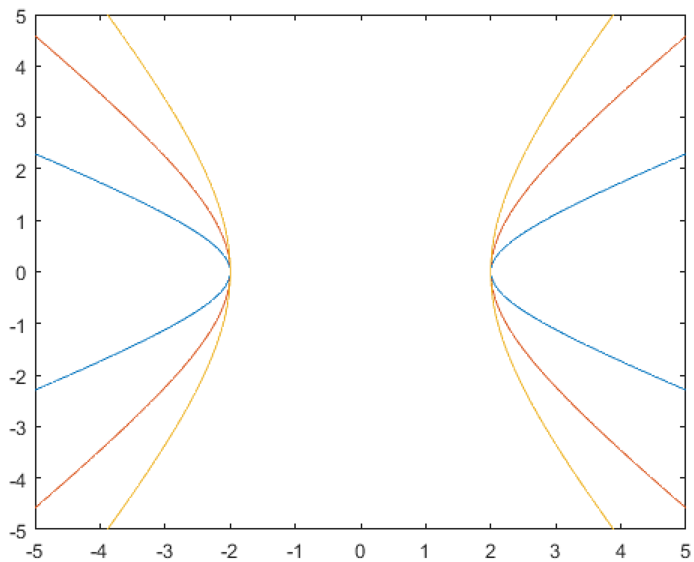
Task is to plot an ellipse

```
syms x y
for a=[4 6 8];
b=2;
fimplicit((x/a)^2 + (y/b)^2 == 1,[-a,a])
hold on
end
hold off
```



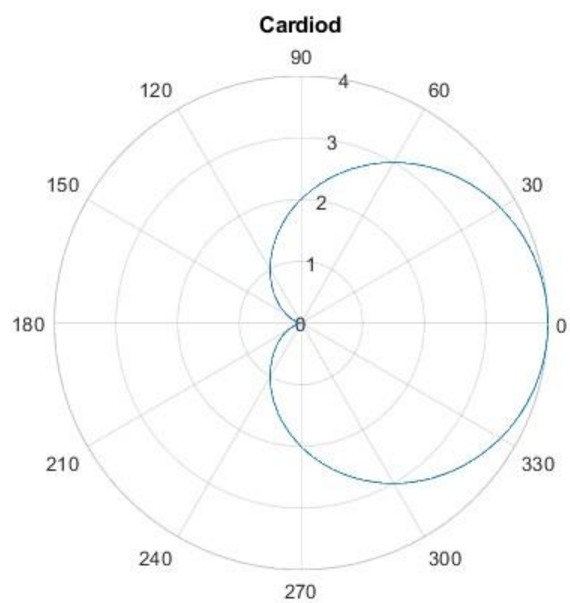
Task is to plot an hyperbola

```
syms x y
for b=[1 2 3];
a=2;
fimplicit((x/a)^2 - (y/b)^2 == 1)
hold on
end
hold off
```



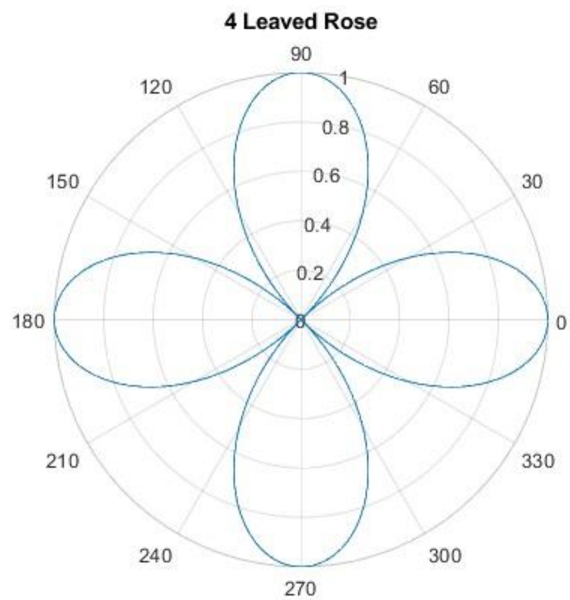
polar plot - Cardior

```
th=0:0.01:2*pi;
a=2;
r=a*(1+cos(th));
polarplot(th,r)
title('Cardiod')
```



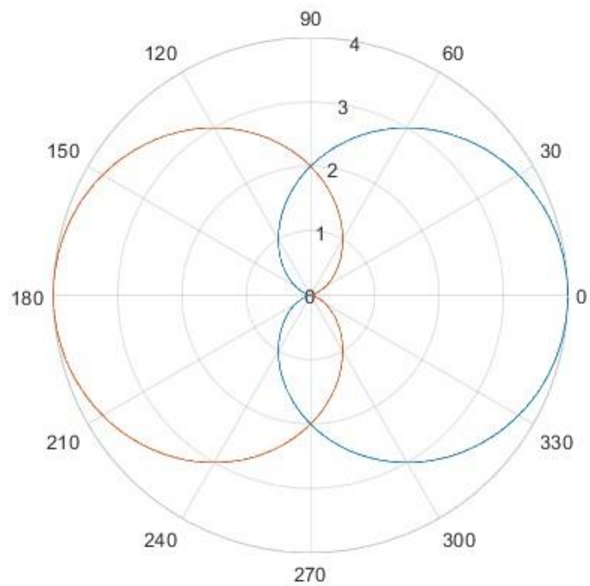
Plot the 4 petal rose r = cos2θ

```
th=0:0.01:2*pi;
a=2;
r=cos(2*th);
polarplot(th,r)
title('4 Leaved Rose')
plotedit off
```



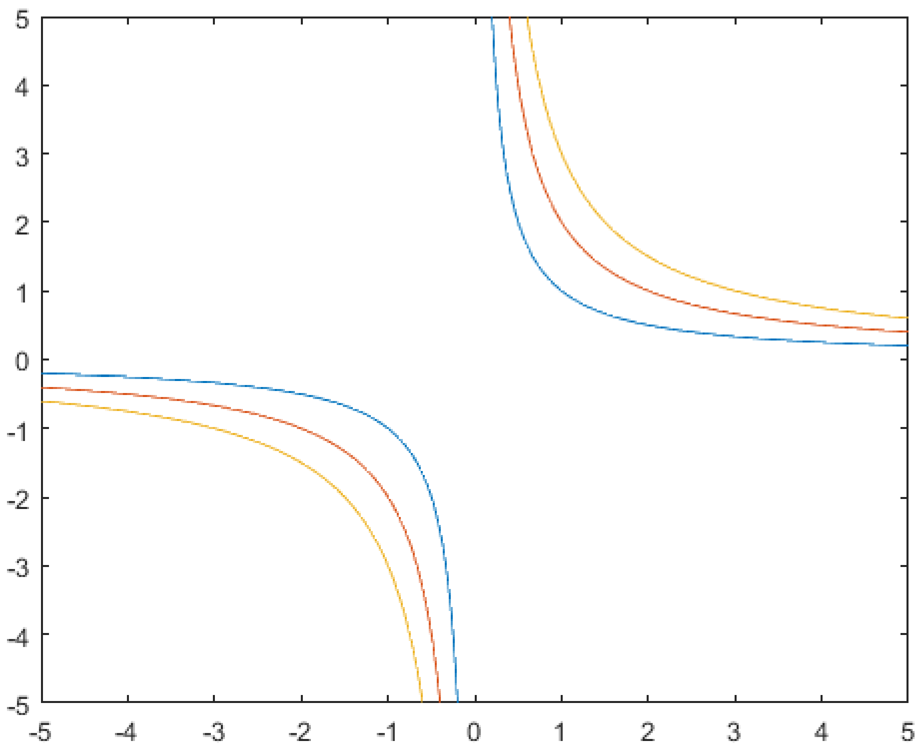
Intersection of two Cardiods

```
th=0:0.01:2*pi;
a=2;
r=a*(1+cos(th));
polarplot(th,r)
hold on
r=a*(1-cos(th));
polarplot(th,r)
hold off
```



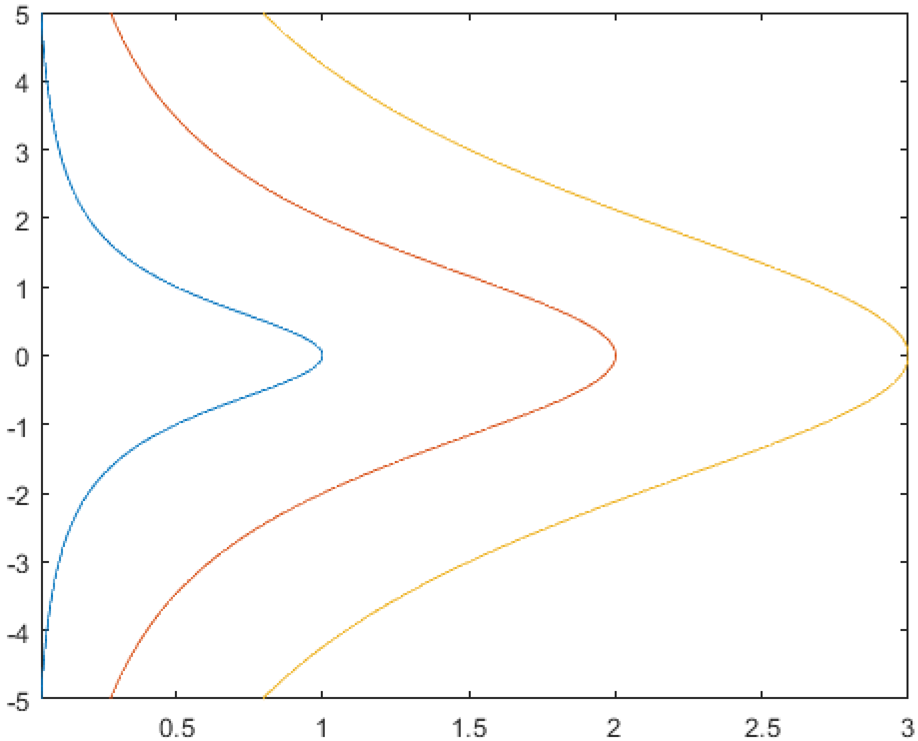
Plot rectangular hyperbola xy=c

```
syms x y
for c=[1 2 3];
fimplicit(x*y == c)
hold on
end
hold off
```



Plot the curve $y^2x = a^2(a-x)$

```
syms x y
for a=[1 2 3];
fimplicit(x*y^2 == a^2*(a-x))
hold on
end
hold off
```



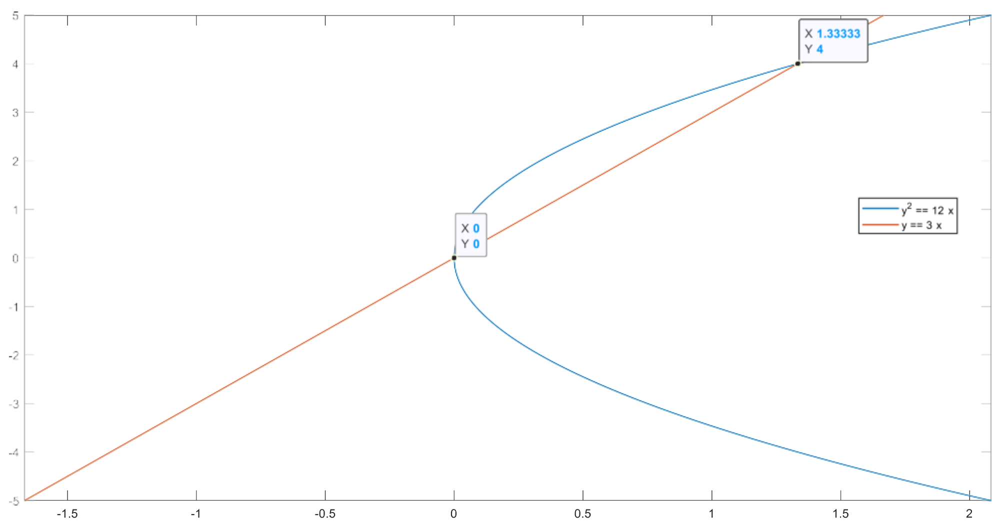
Assignment:

1) Plot a parabola and a straight line in xy-plane. Locate the points of intersection.
2) Draw a 6-leaved rose.

Assignment:

1) Plot a parabola and a straight line in xy-plane. Locate the points of intersection.

```
%Plot the curve
syms x y
a=3;
fimplicit(y^2 == 4*a*x)
hold on
fimplicit(y == a*x)
hold off
```



2) Draw a 6-leaved rose.

Cannot be plotted.

Laboratory - 2

ANGLE BETWEEN TWO POLAR CURVES, RADIUS OF CURVATURE AND CURVATURE

Aim: To find i) the angle between two polar curves

ii) the radius of curvature of Cartesian and polar curves

iii) the curvature.

Formulae:

1) Angle between two polar curves= $\phi = |\phi_1 - \phi_2| = \left|\tan^{-1}\left(r_1\frac{d\theta}{dr_1}\right) - \tan^{-1}\left(r_2\frac{d\theta}{dr_2}\right)\right|$

2) Radius of curvature in Cartesian form is $\rho = \frac{\left[1+{y_1}^2\right]^{\frac{3}{2}}}{y_2}$.

3) Radius of curvature in polar form is $\rho = \frac{\left(r^2+{r_1}^2\right)^{\frac{3}{2}}}{r^2+2{r_1}^2-rr_2}$.

4) Curvature = reciprocal of radius of curvature.

Matlab commands: solve, subs, abs, diff

<u>diff computes the derivative.</u>

Df = diff(f,var,n) computes the nth derivative of f with respect to var.

<u>Solve - Symbolic solution of algebraic equations</u>.

S = solve(eqn1,eqn2,...,eqnM,var1,var2,...,varN)

<u>subs - Symbolic substitution. Also used to evaluate expressions numerically.</u>

subs(S,OLD,NEW) replaces OLD with NEW in the symbolic expression S.

<u>abs Absolute value.</u>

abs(X) is the absolute value of the elements of X.

1. Find the angle between two polar curves $r = a(1+\sin\theta)$ and $r = a(1-\sin\theta)$. Hence show graphically the point of intersection.

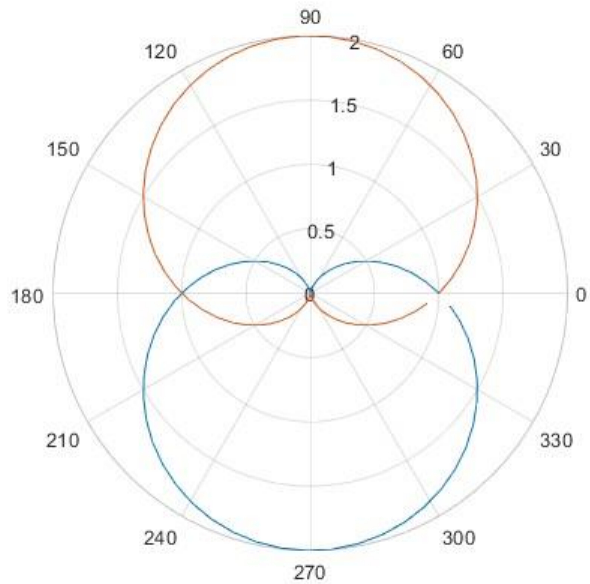
```
syms r1 th r2 a
r1=a*(1-sin(th));
ph1=r1/diff(r1);
r2=a*(1+sin(th));
ph2=r2/diff(r2);
it=solve(r1-r2==0,th);
% simplify(ph1*ph2)  % to show product of slope as -1
Ang=abs(atan(ph1)-atan(ph2));
subs(Ang,it)
```

ans =

$\frac{\pi}{2}$

To plot these polar curves to identify the point of intersection

```
clear
a=1;
t=0:0.1:2*pi;
r1=a*(1-sin(t));
polarplot(t,r1)
hold on
r2=a*(1+sin(t));
polarplot(t,r2)
hold off
```



2. Find the angle between two polar curves $r = ae^{\theta}$ and $re^{\theta} = a$.

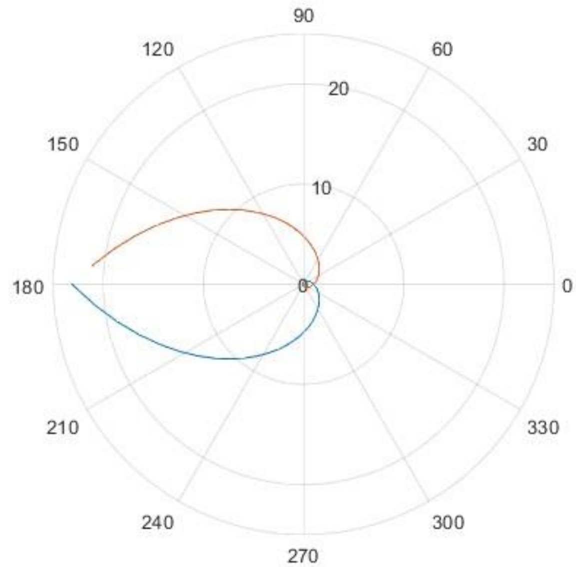
```
clear
syms r1 th r2 a
r1=a*exp(th);
ph1=r1/diff(r1);
r2=a*exp(-th);
ph2=r2/diff(r2);
it=solve(r1-r2==0,th);
Ang=abs(atan(ph1)-atan(ph2));
subs(Ang,it)
```

ans =

$\frac{\pi}{2}$

Plotting exercise

```
clear
a=1;b=1;
t=-pi:0.1:pi;
r1=a*exp(-t);
polarplot(t,r1)
hold on
r2=b*exp(-t);
polarplot(t,r2)
hold off
```



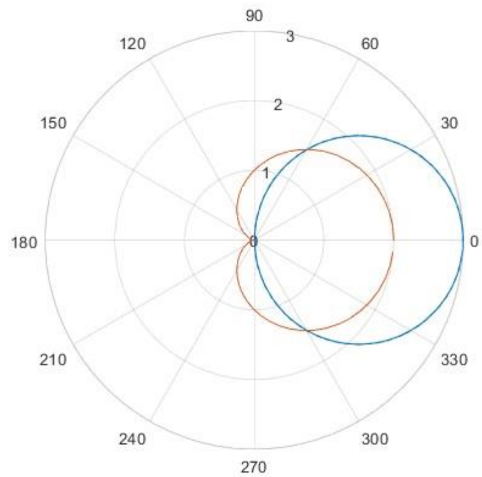
3. Find the angle between two polar curves $r = 3\cos\theta$ and $r = 1 + \cos\theta$

```
syms r1 th r2
r1=3*cos(th);
ph1=r1/diff(r1);
r2=1+cos(th);
ph2=r2/diff(r2);
it=solve(r1-r2==0,th);
Ang=abs(atan(ph1)-atan(ph2));
subs(Ang,it)
```

ans = $\begin{pmatrix} \frac{\pi}{6} \\ \frac{\pi}{6} \end{pmatrix}$

Plotting exercise

```
clear
syms r1 th r2
th=0:0.1:2*pi;
r1=3*cos(th);
polarplot(th,r1);
hold on
r2=1+cos(th);
polarplot(th,r2);
hold off
```



4. Find the radius of curvature of y=4sinx-sin2x at $x=\frac{\pi}{2}$.

```
clear
syms x y
y=4*sin(x)-sin(2*x);
d1=diff(y,1);
d2=diff(y,2);
rc=(1+d1^2)^(3/2)/d2;
rcv=abs(subs(rc,{x},{pi/2}))
```

rcv =

$$\frac{5\sqrt{5}}{4}$$

5. Find the radius of curvature of $y = ax^2 + \mathrm{bx} + c$ at $x = \frac{\sqrt{a^2-1}-b}{2a}$.

```
clear
syms x y a b c
y=a*x^2+b*x+c;
d1=diff(y,1);
d2=diff(y,2);
rc=(1+d1^2)^(3/2)/d2;
rcv=subs(rc,{x},{(sqrt(a^2-1)-b)/(2*a)})
```

rcv = $\frac{(a^2)^{3/2}}{2\,a}$

6. Find the radius of curvature (0,a) for the catenary $y = a\cosh\left(\frac{x}{a}\right)$.

```
clear
syms x y a
y=a*cosh(x/a);
d1=diff(y,1);
d2=diff(y,2);
rc=(1+d1^2)^(3/2)/d2;
rcv=subs(rc,{x},{a})
```

rcv = $\frac{a\,(\sinh(1)^2+1)^{3/2}}{\cosh(1)}$

```
vpa(rcv)
```

ans = $2.3810978455418157297811067388869\,a$

7. Find the radius of curvature (a,0) for the curve $y = x^3(\,x - a)$.

```
clear
syms x y a
y=x^3*(x-a);
d1=diff(y,1);
d2=diff(y,2);
rc=(1+d1^2)^(3/2)/d2;
rcv=subs(rc,{x},{a})
```

rcv =
$\frac{(a^6+1)^{3/2}}{6\,a^2}$

8. Show that radius of curvature for $r = a(1 - \cos\theta)$ varies as $\sqrt{r}$ using polar form of the formula.

```
clear
syms r th a
r=a*(1-cos(th));
r1=diff(r,th);
r2=diff(r,th,2);
rc=(r^2+r1^2)^(3/2)/(r^2+2*r1^2-r*r2)
```

ans =

$$\frac{2\sqrt{2}\sqrt{-a^2\,(\cos(\mathrm{th})-1)}}{3}$$

Assignment:

Compute the curvature of the above Cartesian and polar curves.

Laboratory - 3

PARTIAL DERIVATIVES AND JACOBIAN

Aim: To find i) the partial derivatives
ii) Jacobian using the partial derivatives
iii) Jacobian using the built-in function
iv) Visualization of Jacobian.

.

Formulae:

1) $\frac{\partial f}{\partial x} \rightarrow$ Differentiate ***f*** w.r.t x by treating all other independent variables as constant

2) Jacobian of (u,v) w.r.t (x,y) is $J = \frac{\partial(u,v)}{\partial(x,y)} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$

Note : 1) $\frac{\partial^2 f}{\partial x \partial y}$ and $\frac{\partial^2 f}{\partial y \partial x}$ are called mixed derivatives and are equal.

2) Jacobian is a consequence of geometric distortion due to transformation

Matlab commands: diff, simplify, disp, if, fimplicit3, jacobian

diff - computes the derivative

Df = diff(f,var,n) computes the nth partial derivative of f with respect to var.

simplify - Algebraic simplification

S = simplify(expr) performs algebraic simplification of expr.

If expr is a symbolic vector or matrix, this function simplifies each element of expr

disp - Display array

disp(X) displays array X

If X is a string or character array, the text is displayed

if - Conditionally execute statements

```
if expression
      statements
    ELSEIF expression
    statements
    ELSE
      statements
  END
```

fimplicit3 Plot implicit surface

fimplicit3(FUN) plots the surface where FUN(X,Y,Z)==0 between the axes limits, with a default range of [-5 5].

jacobian Jacobian matrix.

jacobian(f,x) computes the Jacobian of the scalar or vector f with respect to the vector x.

Activities

1. Find all possible partial derivatives up to second order for the function f(x,y)=xy
2. Find all possible partial derivatives up to second order for f(x,y,z)=xyz
3. If $u = x^2 \tan^{-1}\left(\frac{y}{x}\right) - y^2\tan^{-1}\left(\frac{x}{y}\right)$ then show that $\frac{\partial^2 u}{\partial x\,\partial y} = \frac{x^2 - y^2}{x^2 + y^2}$.
4. If $u = x^y$ then show that $\frac{\partial^2 u}{\partial x\,\partial y} = \frac{\partial^2 u}{\partial y\,\partial x}$
5. If u=log(tanx+tany+tanz) then show that $\sin 2x\, u_x + \sin 2y\, u_y + \sin 2z\, u_z = 2$
6. If $Z = \frac{x^2 + y^2}{x + y}$ then show that $\left(\frac{\partial z}{\partial x} - \frac{\partial z}{\partial y}\right)^2 = 4\left(1 - \frac{\partial z}{\partial x} - \frac{\partial z}{\partial y}\right)$
7. Find the Jacobian of $x = r\cos\theta$ and $x = r\sin\theta$
8. Find the Jacobian of $u = x\sin y$ and $v = y\sin x$
9. Find the Jacobian of $x = u^2 - v^2$ and $y = 2uv$
10. Find the Jacobian of $u = 5y, v = 4x^2 - 2\sin(yz)$ and $w = yz$

Assignment

Compute the Jacobian using built-in function and vice versa for the problem Nos.7 to 10.

Find all possible partial derivatives up to second order for the function f(x,y)=xy

```
clear
syms x y f
f=x*y
dfx=diff(f)
dfy=diff(f,y)
dfxx=diff(f,2) % dfxx=diff(diff(f),x)
dfxy=diff(dfx,y)
dfyx=diff(dfy,x)
dfyy=diff(dfy,y)
```

f = $x\,y$ dfx = y dfy = x dfxx = 0 dfxy = 1
dfyx = 1 dfyy = 0

Find all possible partial derivatives up to second order for f(x,y,z)=xyz

```
clear
syms x y z f(x,y,z)
f=x*y*z
dfx=diff(f)
dfy=diff(f,y)
dfz=diff(f,z)
dfxx=diff(f,2) % dfxx=diff(diff(f),x)
dfxy=diff(dfx,y)
dfxz=diff(dfx,z)
dfyx=diff(dfy,x)
dfyy=diff(dfy,y)
dfyz=diff(dfy,z)
dfzx=diff(dfz,x)
dfzy=diff(dfz,y)
dfzz=diff(dfz,z)
```

f = $x\,y\,z$ dfx = $y\,z$ dfy = $x\,z$ dfz = $x\,y$
dfxx = 0 dfxy = z dfxz = y dfyx = z
dfyy = 0 dfyz = x dfzx = y dfzy = x
dfzz = 0

If $u = x^2 \tan^{-1}\left(\frac{y}{x}\right) - y^2 \tan^{-1}\left(\frac{x}{y}\right)$ then show that $\frac{\partial^2 u}{\partial x\,\partial y} = \frac{x^2 - y^2}{x^2 + y^2}$

```
clear
syms x y u
u=x^2*atan(y/x)-y^2*atan(x/y)
dux=diff(u);
dfxyL=diff(dux,y);
dfxy=simplify(dfxyL)
```

u = $x^2 \,\mathrm{atan}\left(\frac{y}{x}\right) - y^2 \,\mathrm{atan}\left(\frac{x}{y}\right)$

dfxy = $\frac{x^2 - y^2}{x^2 + y^2}$

If $u = x^y$ then show that $\frac{\partial^2 u}{\partial x\,\partial y} = \frac{\partial^2 u}{\partial y\,\partial x}$

```
clear
syms x y u
u=x^y
dux=diff(u);
dfxy=diff(dux,y);
dfxyL=simplify(dfxy)
duy=diff(u,y);
dfyx=diff(duy,x);
dfxyR=simplify(dfyx)
if (dfxyL==dfxyR)
    disp('Mixed Derivatives are Equal')
end
```

u = x^y

dfxyL = $x^{y-1}\,(y\log(x) + 1)$

dfxyR = $x^{y-1}\,(y\log(x) + 1)$

Mixed Derivatives are Equal

If u=log(tanx+tany+tanz) then show that $\sin 2x\, u_x + \sin 2y\, u_y + \sin 2z\, u_z = 2$

```
clear
syms x y z u
u=log(tan(x)+tan(y)+tan(z))
ux=diff(u,x)
uy=diff(u,y)
uz=diff(u,z)
LHS=sin(2*x)*ux+sin(2*y)*uy+sin(2*z)*uz
simplify(LHS)
```

u = $\log(\tan(x)+\tan(y)+\tan(z))$

ux = $\frac{\tan(x)^2+1}{\tan(x)+\tan(y)+\tan(z)}$

uy = $\frac{\tan(y)^2+1}{\tan(x)+\tan(y)+\tan(z)}$

uz = $\frac{\tan(z)^2+1}{\tan(x)+\tan(y)+\tan(z)}$

LHS = $\frac{\sin(2\,x)\,(\tan(x)^2+1)}{\tan(x)+\tan(y)+\tan(z)} + \frac{\sin(2\,y)\,(\tan(y)^2+1)}{\tan(x)+\tan(y)+\tan(z)} + \frac{\sin(2\,z)\,(\tan(z)^2+1)}{\tan(x)+\tan(y)+\tan(z)}$

ans =2

If $Z = \frac{x^2+y^2}{x+y}$ then show that $\left(\frac{\partial z}{\partial x} - \frac{\partial z}{\partial y}\right)^2 = 4\left(1 - \frac{\partial z}{\partial x} - \frac{\partial z}{\partial y}\right)$

```
clear
syms x y Z
Z=(x^2+y^2)/(x+y)
ux=diff(Z)
uy=diff(Z,y)
LHS=(ux-uy)^2
RHS=4*(1-ux-uy)
LHS1=simplify(LHS)
RHS1=simplify(RHS)
if (LHS1==RHS1)
     disp('Given relation is true')
end
```

Z = $\frac{x^2+y^2}{x+y}$

ux = $\frac{2\,x}{x+y} - \frac{x^2+y^2}{(x+y)^2}$

uy = $\frac{2\,y}{x+y} - \frac{x^2+y^2}{(x+y)^2}$

LHS = $\left(\frac{2\,x}{x+y} - \frac{2\,y}{x+y}\right)^2$

RHS = $\frac{8\,(x^2+y^2)}{(x+y)^2} - \frac{8\,x}{x+y} - \frac{8\,y}{x+y} + 4$

LHS1 = $\frac{4\,(x-y)^2}{(x+y)^2}$

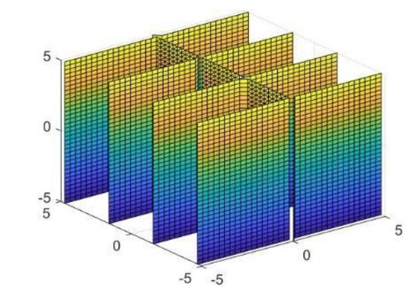
RHS1 = $\frac{4\,(x-y)^2}{(x+y)^2}$

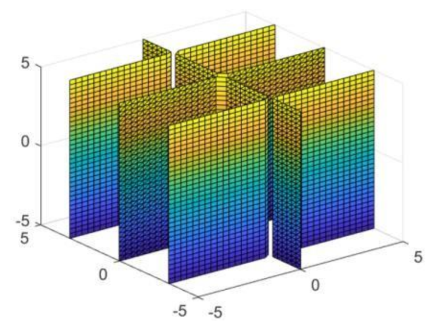
Given relation is true

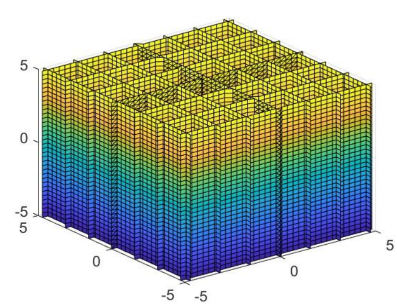
Find the Jacobian of $x = r\cos\theta$ and $x = r\sin\theta$

```
syms x y r th
x=r*cos(th);
y=r*sin(th);
J=[diff(x,r) diff(x,th);diff(y,r) diff(y,th)];
J1=simplify(det(J));
fimplicit3(x)
fimplicit3(y)
fimplicit3(J)
```

J1 = r



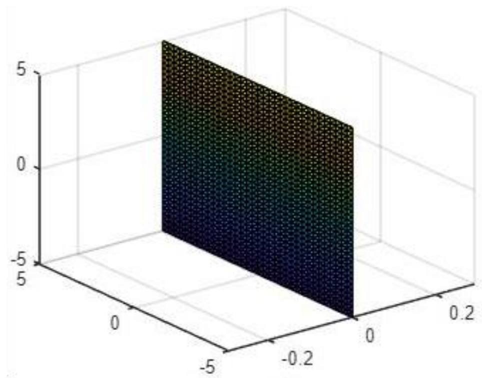


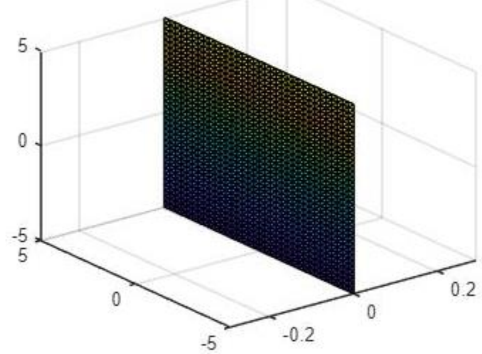


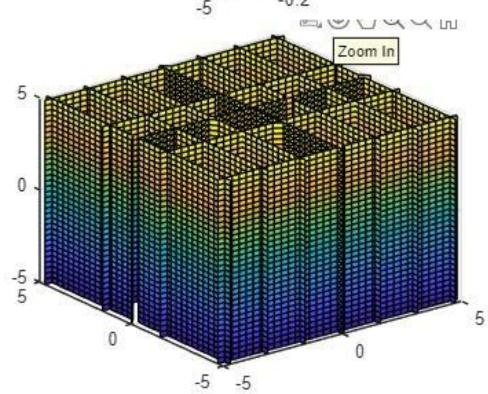
Find the Jacobian of $u = x\sin y$ and $v = y\sin x$

```
clear
syms x y u v
u=x*sin(y);
v=y*sin(x);
J=[diff(u,x) diff(u,y);diff(v,x) diff(v,y)];
J1=simplify(det(J))
fimplicit3(x)
fimplicit3(y)
fimplicit3(J)
```

J1 = $\begin{pmatrix} \sin(y) & x\cos(y) \\ y\cos(x) & \sin(x) \end{pmatrix}$



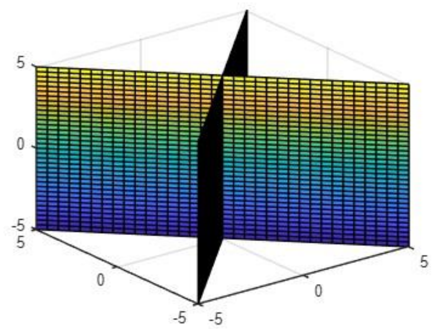


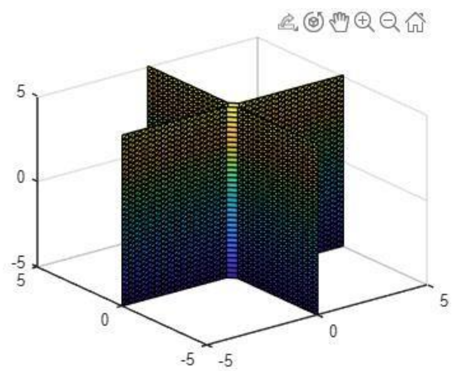


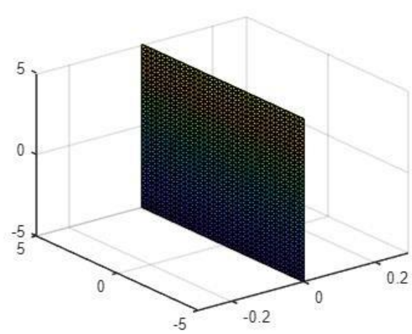
Find the Jacobian of $x = u^2 - v^2$ and $y = 2uv$

```
clear
syms x y u v
x=u^2-v^2;
y=2*u*v;
J=[diff(x,u) diff(x,v);diff(y,u) diff(y,v)];
J1=simplify(det(J))
fimplicit3(x)
fimplicit3(y)
fimplicit3(J)
```

J1 = $\begin{pmatrix} 2u & -2v \\ 2v & 2u \end{pmatrix}$



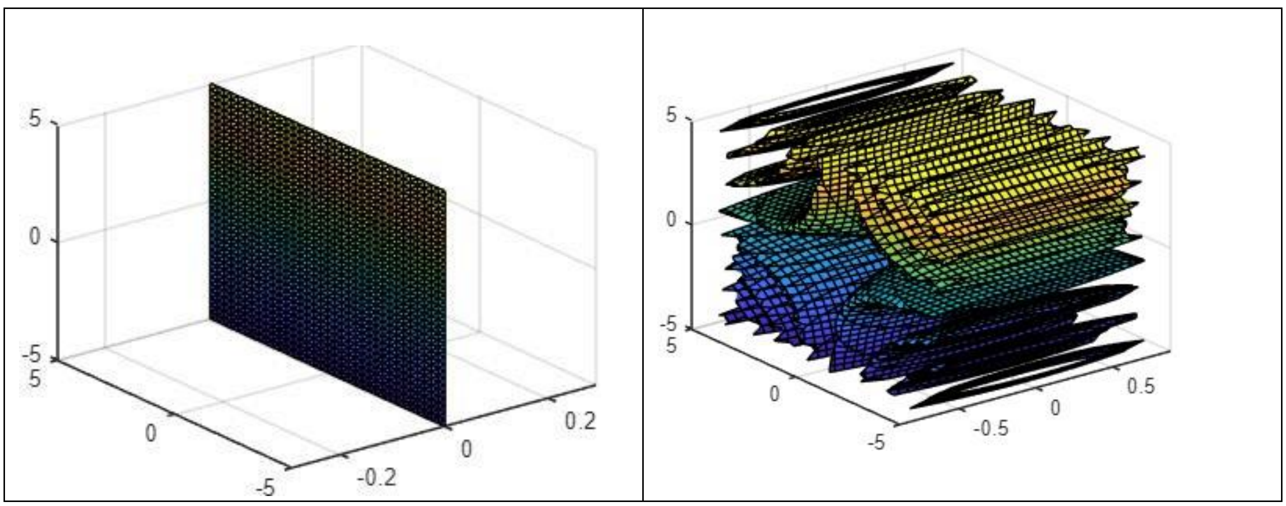


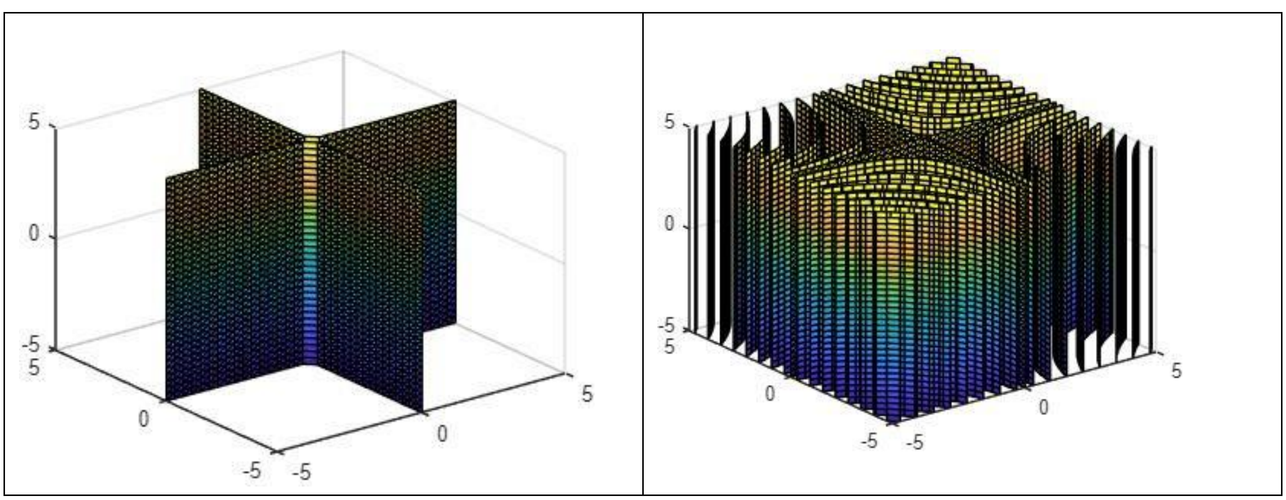


Find the Jacobian of $u = 5y, v = 4x^2 - 2\sin(\text{yz})$ and $w = \text{yz}$

```
 clear
syms x y z u v w
u=5*y;
v=4*x^2-2*sin(y*z);
w=y*z;
fimplicit3(u)
fimplicit3(v)
fimplicit3(w)
J=jacobian([u;v;w],[x y z])
fimplicit3(J)
```

$$J = \begin{pmatrix} 0 & 5 & 0 \\ 8\,x & -2\,z\cos(y\,z) & -2\,y\cos(y\,z) \\ 0 & z & y \end{pmatrix}$$





Laboratory - 4

FINDING EXTREME VALUES FOR A FUNCTION OF TWO VARIABLES

Aim: To find the extreme values of the function of two variables i.e. to find the maximum or minimum value of the function f(x,y).

Procedure: Procedure to find extreme values

1) Compute f_x and f_y the partial derivatives of f w.r.t. x and y

2) Solve $f_x = 0$ and $f_y = 0$ for x and y called critical points

3) Compute $A = f_{xx}, B = f_{xy}$ and $C = f_{yy}$ the partial derivatives of second order

4) Test the following conditions and hence draw the conclusion

a) If $AC - B^2 > 0$ and $A < 0$ then f has maximum at (x,y)

b) If $AC - B^2 > 0$ and $A > 0$ then f has minimum at (x,y)

c) If $AC - B^2 < 0$ then f has saddle points at (x,y)

d) If $AC - B^2 = 0$ then procedure is insufficient to find the nature.

Matlab commands: vpasolve, fsurf, fprintf, for

Syntax: ***vpasolve*** - Numerical solution of algebraic equations.

vpasolve([eq1,...,eqn],[x1,...,xm])

Solve the system of equations eq1,...,eqn in the variables x1,...,xm

fsurf - ***Plot 3-D surface***

fsurf(FUN) creates a surface plot of the function FUN(X,Y).

FUN is plotted over the axes size, with a default interval of -5 < X < 5, -5 < Y < 5.

fprintf - *Write formatted data to text file*

fprintf(FORMAT, A, ...) formats data and displays the results on the screen.

for - Repeat statements a specific number of times

for variable = expr, statement, ..., statement END

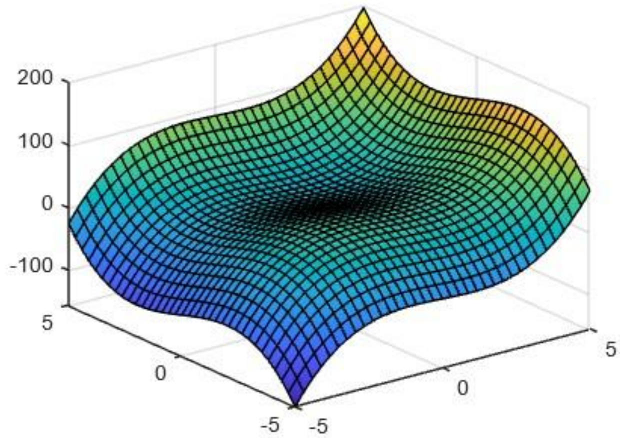
Activities

1. Find the extreme values of $f(x,y) = x^3 + y^3 - 3x - 12y + 20$
2. Find the extreme values of $f(x,y) = 2\text{xy} - 5x^2 - 2y^2 + 4x + 4y - 6$
3. Find the extreme values of $f(x,y) = x^3 + y^3 - 3\text{axy}$, where a=1

1. Find the extreme values of $f(x, y) = x^3 + y^3 - 3x - 12y + 20$

```
clear
syms x y
f=x^3+y^3-3*x-12*y+20
fx = diff(f,x);
fy = diff(f,y);
S = vpasolve([fx==0, fy==0],[x y]);
ml = [S.x S.y];
sf=fsurf(f);
xsy=size(ml);
ed=xsy(:,1);

for i=1:ed
fprintf('Critical Points are ( %d, %d ) \n',ml(i,:))
end

for i=1:ed
A=diff(fx,x);
AV=subs(A,{x y},{ml(i,:)});
B=diff(fx,y);
C=diff(fy,y);
Con=A*C-B^2;
ConV=subs(Con,{x y},{ml(i,:)});
fV=subs(f,{x y},{ml(i,:)});
if (ConV>0 & AV<0)
    fprintf('Maximum at ( %d, %d ) and Max. Value= %d \n',ml(i,:),fV)
elseif (ConV>0 & AV>0)
    fprintf('Minimum at ( %d, %d ) and Min. Value= %d \n',ml(i,:),fV)
elseif (ConV<0)
    fprintf('Saddle Point at ( %d, %d ) \n',ml(i,:))
else
    fprintf('Fuurther Investigation required at ( %f, %f) \n',ml(i,:))
end
end
```

D = $x^3 - 3\,x + y^3 - 12\,y + 20$

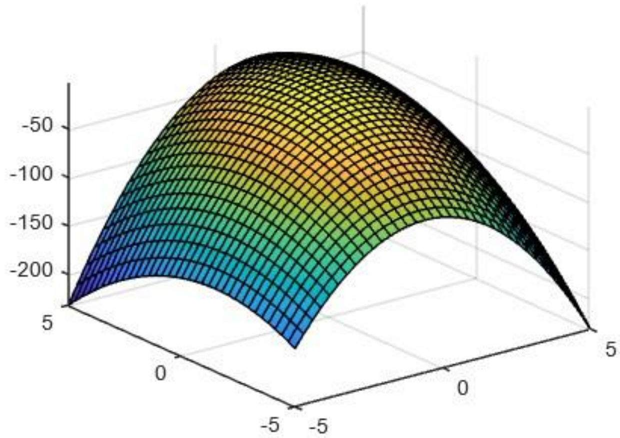



```
Critical Points are (-1,-2)
Critical Points are (1,-2)
Critical Points are (-1,2)
Critical Points are (1,2)

Maximum at ( -1, -2 ) and Max. Value= 38
Saddle Point at ( 1, -2 )
Saddle Point at ( -1, 2 )
Minimum at ( 1, 2 ) and Min. Value= 2
```

2. Find the extreme values of $f(x,y) = 2xy - 5x^2 - 2y^2 + 4x + 4y - 6$

```
clear
syms x y
f=2*x*y-5*x^2-2*y^2+4*x+4*y-6
fx = diff(f,x);
fy = diff(f,y);
S = vpasolve([fx==0, fy==0],[x y]);
fsurf(f)
ml = [S.x S.y];

xsy=size(ml);
ed=xsy(:,1);

for i=1:ed
fprintf('Critical Points are ( %f, %f ) \n',ml(i,:))
end

for i=1:ed
A=diff(fx,x);
AV=subs(A,{x y},{ml(i,:)});
B=diff(fx,y);
C=diff(fy,y);
Con=A*C-B^2;
ConV=subs(Con,{x y},{ml(i,:)});
fV=subs(f,{x y},{ml(i,:)});
if (ConV>0 & AV<0)
    fprintf('Maximum at ( %f, %f) and Max. Value= %f \n',ml(i,:),fV)
elseif (ConV>0 & AV>0)
    fprintf('Minimum at ( %f, %f) and Min. Value= %f \n',ml(i,:),fV)
elseif (ConV<0)
    fprintf('Saddle Point at ( %f, %f) \n',ml(i,:))
else
    fprintf('Fuurther Investigation required at ( %f, %f) \n',ml(i,:))
end
end
```

D = $-5\,x^2+2\,x\,y+4\,x-2\,y^2+4\,y-6$



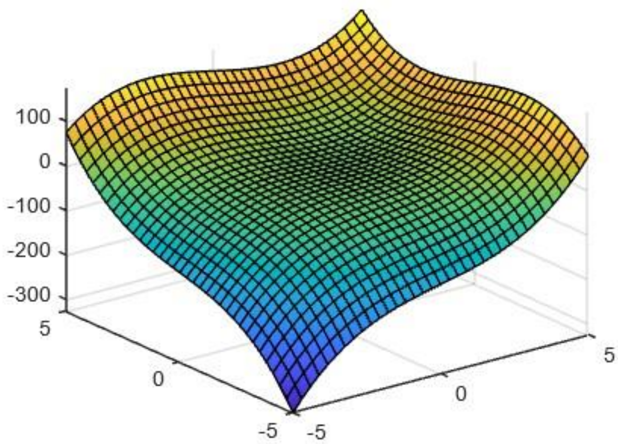

```
Critical Points are ( 0.6666674, 1.3333334)

Maximum at ( 0.666667, 1.333333) and Max. Value= -2.000000
```

3. Find the extreme values of $f(x, y) = x^3 + y^3 - 3axy$, where a=1

```
clear
syms x y a
a=1;
f=x^3+y^3-3*a*x*y
fx = diff(f,x);
fy = diff(f,y);
S = vpasolve([fx==0, fy==0],[x y]);
ml = [S.x S.y];
fprintf('Critical Points are ( %f, %f) \n',ml(1,:))
fprintf('Critical Points are ( %f, %f) \n',ml(4,:))
xsy=size(ml);
ed=xsy(:,1);
fsurf(f)
for i=[1 ed]
A=diff(fx,x);
AV=subs(A,{x y},{ml(i,:)});
B=diff(fx,y);
C=diff(fy,y);
Con=A*C-B^2;
ConV=subs(Con,{x y},{ml(i,:)});
fV=subs(f,{x y},{ml(i,:)});
if (ConV>0 & AV<0)
    fprintf('Maximum at ( %f, %f) and Max. Value= %f \n',ml(i,:),fV)
elseif (ConV>0 & AV>0)
    fprintf('Minimum at ( %f, %f) and Min. Value= %f \n',ml(i,:),fV)
elseif (ConV<0)
    fprintf('Saddle Point at ( %f, %f) \n',ml(i,:))
else
    fprintf('Fuurther Investigation required at ( %f, %f) \n',ml(i,:))

end
end
```

```
D = 
```
$x^3 - 3\,x\,y + y^3$

```
Critical Points are ( 0.000000, 0.000000)

Critical Points are ( 1.000000, 1.000000)
```




```
Saddle Point at ( 0.000000, 0.000000)

  Minimum at ( 1.000000, 1.000000) and Min. Value= -1.000000
```

Laboratory - 5

CONSISTENCY OF SYSTEM OF LINEAR EQUATIONS

Aim: To test whether the system of linear equations is consistent or not and hence to solve.

Procedure: 1) Construct the augmented matrix by appending the column matrix b into A i.e.[A:b]

2) Find the rank of A and [A:b]. If ranks are equal then declare system as "Consistent" otherwise "Inconsistent"

3) When system is consistent

– it will have unique solution if rank of A = rank of [A:b] = number of unknowns

– it will have infinite solutions if rank of A = rank of [A:b] < number of unknowns

4) Solve the system of equations using "solve" command.

Matlab commands: input, rank

Syntax: ***<u>input - Prompt for user input.</u>***

RESULT = input(PROMPT)

displays the PROMPT string on the screen, waits for input from the keyboard, evaluates any expressions in the input and returns the value in RESULT**.**

<u>rank - Matrix rank.</u>

rank(A)

provides an estimate of the number of linearly independent rows or columns of a matrix A

Activities

Test for consistency and hence solve the following systems.

1) $\begin{pmatrix} 2x+3y+5z=9 \\ 7x+3y-2z=8 \\ 2x+3y+5z=1 \end{pmatrix}$ 2) $\begin{pmatrix} 5x+3y+7z=4 \\ 3x+26y+2z=9 \\ 7x+2y+10z=5 \end{pmatrix}$ 3) $\begin{pmatrix} 27x+6y-z=85 \\ 6x+15y+2z=72 \\ x+y+54z=110 \end{pmatrix}$

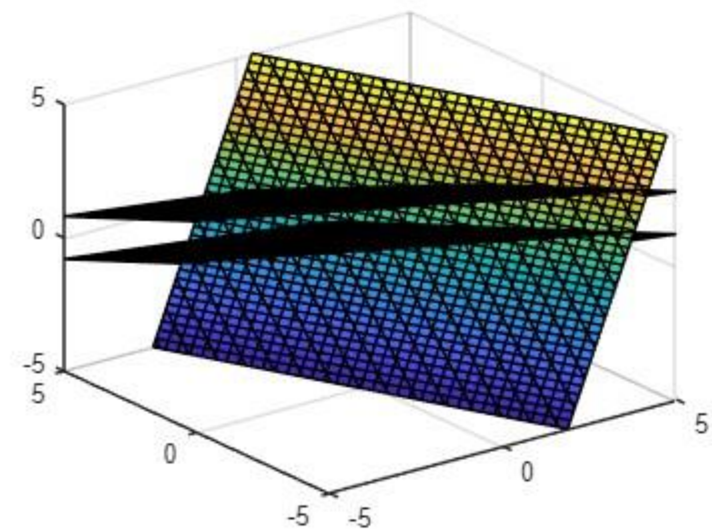
Program to test the consistency of a system and hence to solve

```
clear

A=input('Enter the Coefficient Matrix')
b=input('Enter the RHS Matrix')

% Construct the augmented marix Ag
Ag=[A b];
% Find the rank of coefficient matrix A and augmented marix Ag
ra=rank(A);
rg=rank(Ag);

        syms x y z
        D=[x; y; z];
        F=A*D;
        SS=F==b

if (ra==rg)
    disp('System is consistent')
    sz=size(A);
        S=solve([SS],{x y z});
            if (ra==sz(end))
            disp('System has Unique solution and is')
            else
            disp('System has infinite solutions and one particular solution
is')
            end
        disp(S)
else
    disp('System is Inconsistent')
end
% To visualization of planes
fimplicit3(SS)
```

1.

$$\begin{pmatrix} 2x+3y+5z=9 \\ 7x+3y-2z=8 \\ 2x+3y+5z=1 \end{pmatrix}$$

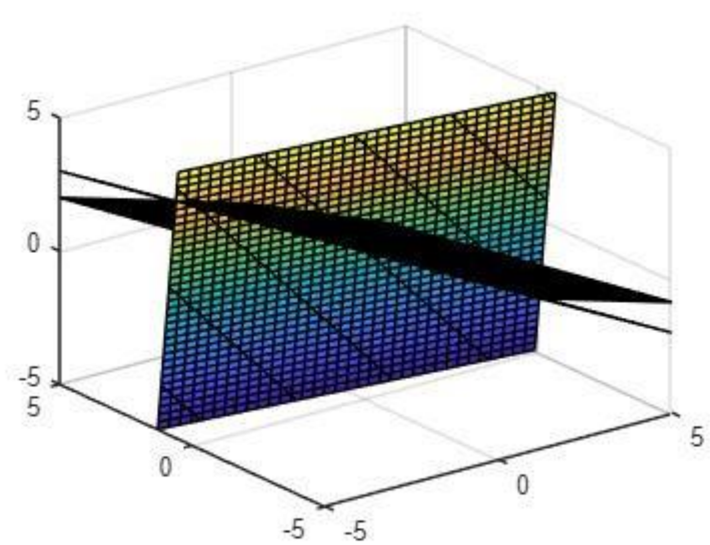
```
Enter the Coefficient Matrix[2 3 5;7 3 -2;2 3 5]
Enter the RHS Matrix[9;8;1]

System is Inconsistent
```



2.

$$\begin{pmatrix} 5\,x + 3\,y + 7\,z = 4 \\ 3\,x + 26\,y + 2\,z = 9 \\ 7\,x + 2\,y + 10\,z = 5 \end{pmatrix}$$

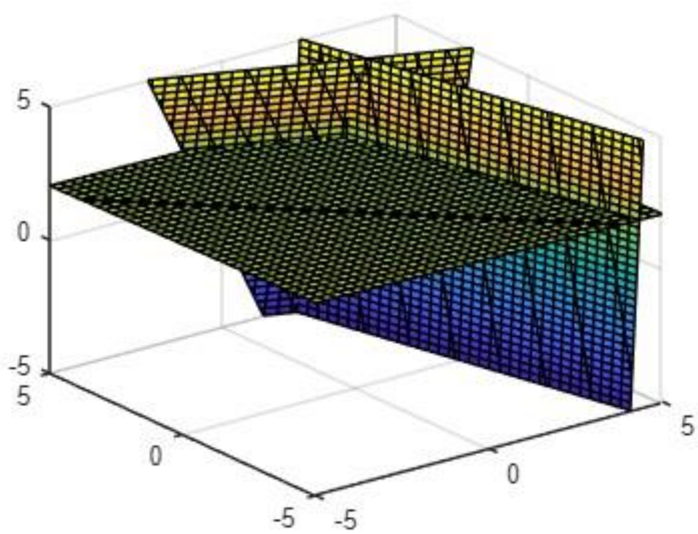
```
Enter the Coefficient Matrix[5 3 7;3 26 2;7 2 10]
Enter the RHS Matrix[4 9 5]'

System is consistent


System has infinite solutions and one particular solution is
    x: 7/11
    y: 3/11
    z: 0
```



3.

$$\begin{pmatrix} 27\,x+6\,y-z=85 \\ 6\,x+15\,y+2\,z=72 \\ x+y+54\,z=110 \end{pmatrix}$$

```
Enter the Coefficient Matrix[27 6 -1;6 15 2;1 1 54]
Enter the RHS Matrix[85 72 110]'

System is consistent


System has Unique solution and is


    x: 48250/19893
    y: 71078/19893
    z: 12771/6631
```



Laboratory - 6

GAUSS SEIDEL METHOD TO SOLVE A SYSTEM OF LINEAR EQUATIONS

Aim: To solve a system of linear equations using Gauss-Seidel method.

Given the system

$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$;

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$;

$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$

Working Rule:

Step1: Rewrite the above equations as follows:

$$x_1 = \frac{(b_1 - a_{12}x_2 - a_{13}x_3)}{a_{11}} \qquad (1)$$

$$x_2 = \frac{(b_2 - a_{21}x_1 - a_{23}x_3)}{a_{22}} \qquad (2)$$

$$x_3 = \frac{(b_3 - a_{31}x_1 - a_{32}x_2)}{a_{33}} \qquad (3)$$

Step2: Choose the initial approximation to the solution as $x_1 = 0$; $x_2 = 0$; $x_3 = 0$.

Step3: (i) Compute x_1 using the values of x_2 and x_3 from equation (1)

(ii) Compute x_2 using the values of x_1 obtained in step 3(i) and x_3 from equation (2)

(iii) Compute x_3 using the values of x_1 and x_2 obtained in step 3(i) and (ii) from equation (3)

Step4: Repeat the above procedure for the specified number of times

Activities

Solve the following systems using Gauss-Seidel Method.

1) $\begin{pmatrix} 10x + y + z = 12 \\ 2x + 10y + z = 13 \\ 2x + 2y + 10z = 14 \end{pmatrix}$ 2) $\begin{pmatrix} 5x - y = 9 \\ x - 5y + z = -4 \\ y - 5z = 6 \end{pmatrix}$ 3) $\begin{pmatrix} 27x + 6y - z = 85 \\ 6x + 15y + 2z = 72 \\ x + y + 54z = 110 \end{pmatrix}$

```
% program to solve a system of liner equations using Gauss-Seidel Method
a=input('Enter the Coefficient Matrix a:')
b=input('Enter the RHS Matrix b:')

itr=input('Enter the number of iterations:      ')
x1=0;x2=0;x3=0;
for i=1:itr
x1=[b(1,1)-a(1,2)*x2-a(1,3)*x3]/a(1,1);
x2=[b(2,1)-a(2,1)*x1-a(2,3)*x3]/a(2,2);
x3=[b(3,1)-a(3,1)*x1-a(3,2)*x2]/a(3,3);
fprintf('Iteration=%d\n',i)
fprintf('x1=%f, x2=%f,x3=%f \n',x1,x2,x3)
end
```

$$\begin{pmatrix} 10\,x + y + z = 12 \\ 2\,x + 10\,y + z = 13 \\ 2\,x + 2\,y + 10\,z = 14 \end{pmatrix}$$

```
Iteration=1
x1=1.200000, x2=1.060000,x3=0.948000
Iteration=2
x1=0.999200, x2=1.005360,x3=0.999088
Iteration=3
x1=0.999555, x2=1.000180,x3=1.000053
Iteration=4
x1=0.999977, x2=0.999999,x3=1.000005
```

$$\begin{pmatrix} 5\,x - y = 9 \\ x - 5\,y + z = -4 \\ y - 5\,z = 6 \end{pmatrix}$$

```
Iteration=1
x1=1.800000, x2=1.160000,x3=-0.968000
Iteration=2
x1=2.032000, x2=1.012800,x3=-0.997440
Iteration=3
x1=2.002560, x2=1.001024,x3=-0.999795
Iteration=4
x1=2.000205, x2=1.000082,x3=-0.999984
```

$$\begin{pmatrix} 27\,x + 6\,y - z = 85 \\ 6\,x + 15\,y + 2\,z = 72 \\ x + y + 54\,z = 110 \end{pmatrix}$$

```
Iteration=1
x1=3.148148, x2=3.540741,x3=1.913169
Iteration=2
x1=2.432175, x2=3.572041,x3=1.925848
Iteration=3
x1=2.425689, x2=3.572945,x3=1.925951
Iteration=4
x1=2.425492, x2=3.573010,x3=1.925954
```

Laboratory - 7

EIGNE VALUE AND EIGEN VECTOR

Aim: To find the eigen value and the corresponding eigen vector by power method

Given the square matrix A.

Let λ be a real or complex number and X be a non-zero column vector which satisfies the relation $AX=\lambda X$ then λ is the eigen value and X is the corresponding eigen vector.

WORKING RULE FOR POWER METHOD:

1) Let us choose X=[1 1 1]T as the initial guess for the eigen vector.

2) Compute AX and rewrite by taking the numerically largest value as common,

to get $AX= \lambda X_1$, where λ is the eigen value and X_1 is corresponding eigne vector

3) Repeat step-2 for n times.

Activities

Using power method find the dominant eigne value and the corresponding eigne vector

1) $\begin{bmatrix} 9 & 1 & 4 \\ 1 & 6 & 4 \\ 1 & 1 & 8 \end{bmatrix}$ 2) $\begin{bmatrix} 10 & 1 & 1 \\ 2 & 10 & 1 \\ 2 & 2 & 10 \end{bmatrix}$ 3) $\begin{bmatrix} 5 & -1 & 0 \\ 1 & -5 & 1 \\ 0 & 1 & -5 \end{bmatrix}$ 4) $\begin{bmatrix} 27 & 6 & -1 \\ 6 & 15 & 2 \\ 1 & 1 & 54 \end{bmatrix}$

```
clc
A=[9 1 4;1 6 1;1 1 8]
X=ones(3,1);
n=input('ENTER THE NUMBER OF ITERATIONS:\')
for i=1:n
Xn=A*X;
EVal=max(abs(Xn));
EVct=Xn/EVal;
X=EVct;
end

fprintf('Eigen Value is %f \n \n Corresponding Eigen Vector is',EVal)
disp(X)
[V,D]=eig(A)
```

```
A = 3×3
     9     1     4
     1     6     1
     1     1     8


n = 7


Eigen Value is 11.079731

 Corresponding Eigen Vector is

    1.0000
    0.2890
    0.4393


V = 3×3 complex
   0.8890 + 0.0000i   0.5774 + 0.0000i   0.5774 - 0.0000i
   0.2540 + 0.0000i   0.5774 + 0.0000i   0.5774 + 0.0000i
   0.3810 + 0.0000i  -0.5774 - 0.0000i  -0.5774 + 0.0000i

D = 3×3 complex
  11.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   6.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   6.0000 - 0.0000i
```

Laboratory - 8

SOLUTION OF FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS

Aim: To solve First Order Ordinary Differential Equation and to plot their solution curves

Matlab command: dsolve

Syntax: dsolve Symbolic solution of ordinary differential equations.

dsolve(eqn1,eqn2, ...) accepts symbolic equations representing ordinary differential equations and initial conditions.

By default, the independent variable is 't'. The independent variable may be changed from 't' to some other symbolic variable by including that variable as the last input argument.

Activities: Solve and plot the solution curves for the following

1) $\frac{\mathrm{dy}}{\mathrm{dx}} + y\cos\mathrm{x} = \cos\mathrm{x},\ y(0) = 0$

2) $\frac{\mathrm{dy}}{\mathrm{dx}} - y = e^{x}x,\ y(0) = 0$

3) $\frac{\mathrm{dy}}{\mathrm{dx}} + y = e^{-x},\ y(0) = 1$

4) $\frac{\mathrm{dy}}{\mathrm{dx}} + \sin\mathrm{x}\ y = \sin\mathrm{x},\ y(0) = 1$

5) $\frac{\mathrm{dy}}{\mathrm{dx}} + \cot\mathrm{x}\ y = \cos\mathrm{x}$

6) $\frac{\mathrm{dy}}{\mathrm{dx}} - \frac{2y}{1+x} = \frac{1}{1+x^3}$

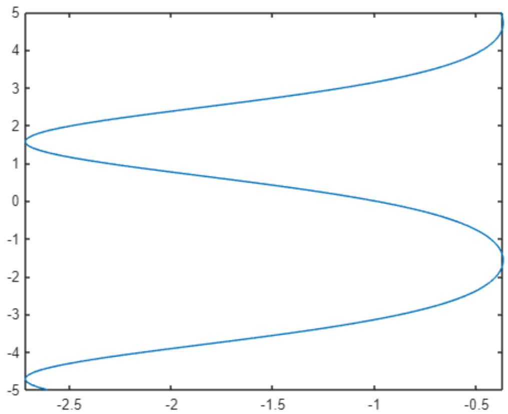
7) $\frac{\mathrm{dy}}{\mathrm{dx}} + y\cot\mathrm{x} = \frac{4x}{\sin\mathrm{x}}$

Solve $\frac{dy}{dx} + y\cos x = \cos x,\ y(0) = 0$

```
syms y(x)
Dy=diff(y);
f=dsolve(Dy+cos(x)*y==cos(x)) %General solution
fimplicit(f)
dsolve(Dy+cos(x)*y==cos(x),y(0)==0) % Particular solution
```

f = $C_1 e^{-\sin(x)} + 1$

ans = $1 - e^{-\sin(x)}$



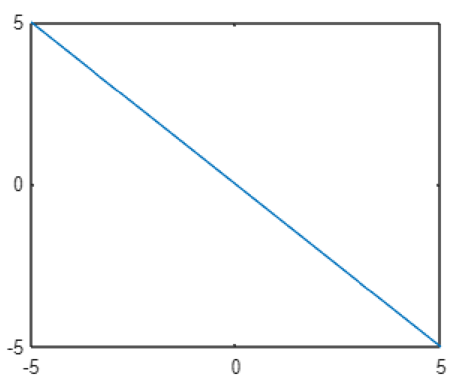
Solve $\frac{\mathrm{dy}}{\mathrm{dx}} - y = e^x x,\ y(0) = 0$

```
clear
syms y(x)
Dy=diff(y);
f=dsolve(Dy-y==x*exp(x)) %General solution
fimplicit(f)
dsolve(Dy-y==x*exp(x),y(0)==0) % Particular solution
```

Solve $\frac{\mathrm{dy}}{\mathrm{dx}} + y = e^{-x},\ y(0) = 1$

```
clear
syms y(x)
Dy=diff(y);
f=dsolve(Dy+y==exp(-x))
% plot the slution curve
fimplicit(f)
% solution with initial condition
dsolve(Dy+y==exp(-x), y(0) == 1)
```

f = $x\,\mathrm{e}^{-x} + C_1\,\mathrm{e}^{-x}$

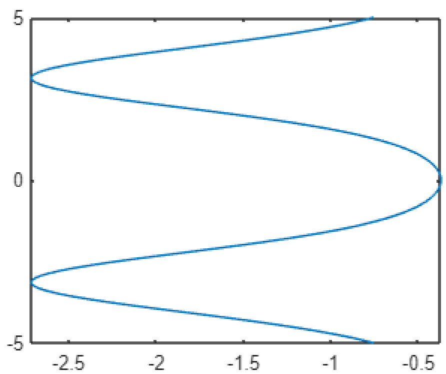


ans = $\mathrm{e}^{-x}\,(x+1)$

Solve $\frac{dy}{dx} + \sin x \, y = \sin x, \; y(0) = 1$

```
clear
syms y(x)
Dy=diff(y);
f=dsolve(Dy+sin(x)*y==sin(x))
% plot the slution curve
fimplicit(f)
% solution with initial condition
dsolve(Dy+sin(x)*y==sin(x), y(0) == 1)
```

f = $C_1 e^{\cos(x)} + 1$

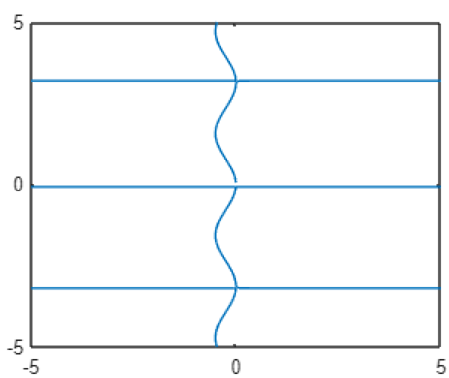


ans = 1

Solve $\frac{dy}{dx} + \cot x \, y = \cos x$

```
clear
syms y(x)
Dy=diff(y);
f=dsolve(Dy+cot(x)*y==cos(x))
```

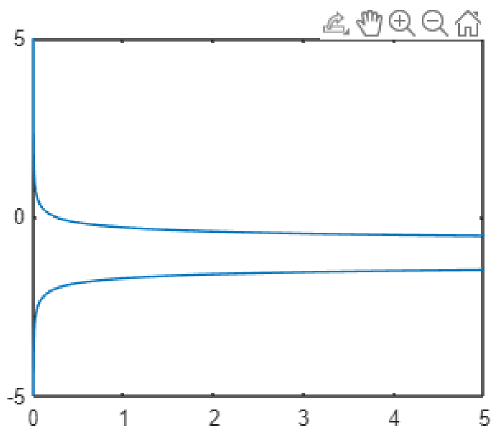
f = $\frac{\sin(x)}{2} + \frac{C_1}{\sin(x)}$



Solve Solve $\frac{\mathrm{dy}}{\mathrm{dx}} - \frac{2y}{1+x} = \frac{1}{(1+x)^3}$

```
clear
syms y(x)
Dy=diff(y);
f=dsolve(Dy-2*y/(1+x)==1/(x+1)^3)
% plot the slution curve
fimplicit(f)
```

f= $C_1\ (x+1)^2 - \frac{1}{4\ (x+1)^2}$

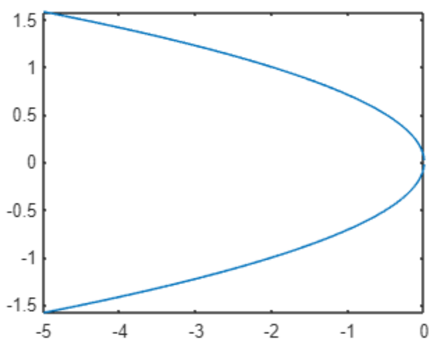


Solve $\frac{dy}{dx} + y\cot x = \frac{4x}{\sin x}$

```
clear
syms y(x)
Dy=diff(y);
f=dsolve(Dy+y*cot(x)==4*x/sin(x))
% plot the slution curve
fimplicit(f)
```

f =

$$\frac{2\,x^2}{\sin(x)} + \frac{C_1}{\sin(x)}$$



Laboratory - 9

EUCLID'S ALGORITHM

Aim: To find Greatest Common Divisor (gcd) for two integers using Euclid's algorithm

Matlab command: rem

Syntax: rem(x,y) Remainder after division.

Procedure: Euclid's Algorithm

$a=b*q+r$

$b=r*q+r_1$

$r=r_1*q+r_2$

....

continue till remainder becomes ZERO

$r_n=r_{n-1}*q+0$

The non-zero remainder represents the gcd.

Activities: Find the gcd for the following pair of numbers

1) (21,36) 2) (-41,-17) 3) (12,-120) 4) (0,15)

Assignment : Using gcd find the lcm of the numbers

```
clear
clc
a=input('Enter the value of a:');
b=input('Enter the value of b:');

if (a~=0 & b~=0)
a1=a;b1=b;
if (a<0|b<0)
    a=abs(a);
    b=abs(b);
end
r=a;
while ( r~=0 )
r=rem(a,b);
a=b;
b=r;
end
gc=a;
fprintf ( 'gcd(%d,%d) = %d \n',a1,b1, gc);
else
    fprintf ( 'Enter the non-zero numbers');
end
```

```
1)
Enter the value of a:21
Enter the value of b:36

gcd(21,36) = 3

2)
Enter the value of a:-41
Enter the value of b:-17

gcd(-41,-17) = 1

3)
Enter the value of a:12
Enter the value of b:-120

gcd(12,-120) = 12

4)
Enter the value of a:0
Enter the value of b:15

Enter the non-zero numbers
```

Laboratory - 10

BETA AND GAMMA FUNCTIONS

Aim: Evaluate Gamma Function

Syntax: `Y = gamma(X)`

Y = gamma(X) returns the gamma function evaluated at the elements of X

Gamma Function

The `gamma` function is defined for real `x > 0` by the integral:

$$\Gamma(x) = \int e^{-t} t^{x-1} dt$$

The `gamma` function interpolates the `factorial` function. For integer `n`:
`gamma(n+1) = factorial(n) = prod(1:n)`.
The domain of the `gamma` function extends to negative real numbers by analytic continuation, with simple poles at the negative integers. This extension arises from repeated application of the recursion relation
Γn−1= Γn / n−1

Evaluate the gamma function with a scalar and a vector

Evaluate Γ(0.5), which is equal to $\sqrt{\pi} = 1.7724538851 \cong 1.7725$

```
y = gamma(0.5)              ans = 1.7725

gamma(1)                    ans = 1

gamma(3/2)                  ans =  0.8862

gamma(0)                    ans =  Inf

gamma(3)                    ans = 2

x = 1:5; y = gamma(x)
y =  1     1     2     6    24

x = 5:10;      y = gamma(x)
y =  24         120         720        5040       40320      362880

x = -3.5:3.5;         y = gamma(x)
y = 0.2701   -0.9453    2.3633   -3.5449    1.7725    0.8862    1.3293    3.3234
```

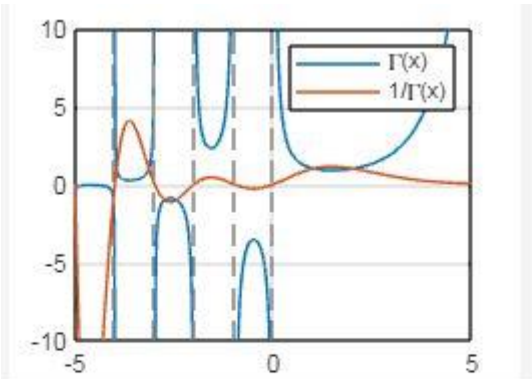
Plot Gamma Function

Plot the gamma function and its reciprocal.

Use `fplot` to plot the gamma function and its reciprocal. The gamma function increases quickly for positive arguments and has simple poles at all negative integer arguments (as well as 0). The function does not have any zeros. Conversely, the reciprocal gamma function has zeros at all negative integer arguments (as well as 0).

X — Input array: scalar | vector | matrix | multidimensional array
Input array, specified as a scalar, vector, matrix, or multidimensional array. The elements of `X` must be real.

```
fplot(@gamma)
hold on
fplot(@(x) 1./gamma(x))
ylim([-10 10])
legend('\Gamma(x)','1/\Gamma(x)')
hold off
grid on
```



Beta function

Aim: Evaluate Beta Function

Syntax: `B = beta(Z,W)`

`B = beta(Z,W)`returns the beta function evaluated at the elements of `Z` and `W`. Both `Z` and `W` must be real and nonnegative.

Beta Function

The beta function is defined by

$$B(z,w)=\int_0^1 t^{z-1}(1-t)^{w-1}\,dt =\Gamma(z)\,\Gamma(w)\,/\,\Gamma(z+w).$$

The Γz term is the gamma function $\Gamma(z) = \int_0^\infty t^{Z-1}e^{-z}\;dt$

Z — Input array scalar | vector | matrix | multidimensional array

Input array, specified as a scalar, vector, matrix, or multidimensional array. The elements of `Z` must be real and nonnegative. `Z` and `W` must be the same size, or else one of them must be a scalar.

W — Input array scalar | vector | matrix | multidimensional array

Input array, specified as a scalar, vector, matrix, or multidimensional array. The elements of `W` must be real and nonnegative. `Z` and `W` must be the same size, or else one of them must be a scalar.

If `Z` or `W` is equal to 0, the beta function returns `Inf`.

If `Z` and `W` are both 0, the beta function returns `NaN`

Evaluate the Beta function with a scalar and a vector

```
B = beta(3,5)  ANS B = 0.009   since 1/105=0.009523

B = beta(3/2,5/2)            B = 0.1963

B = beta(3/2,2)              B = 4/15 =  0.2667

B = beta(5,4)                B = 1/280 =  0.0036
```

Compute Beta Function for Integer Arguments

Compute the beta function for integer arguments w=3 and z=1,...,10. Based on the definition, the beta function can be calculated as

B(z,3)=Γz Γ3 / Γ(z+3)=(z−1)! 2! / (z+2)! = 2/ z(z+1)(z+2).

Set the output format to rational to show the results as ratios of integers.

```
format rat
B = beta((1:10)',3)

B = 1/3  1/12   1/30 1/60  1/105   1/168  1/252   1/360   1/495   1/660
```

Plot Beta Function

Calculate the beta function for z = 0.05, 0.1, 0.2, and 1 within the interval 0≤w≤10. Loop over values of z, evaluate the function at each one, and assign each result to a row of B.

```
Z = [0.05 0.1 0.2 1];
W = 0:0.05:10;
B = zeros(4,201);
for i = 1:4
B(i,:) = beta(Z(i),W);
end

>> B
ANS B =

Columns 1 through 9

1/0         3905/98       7117/239      3535/134    3131/127      2808/119      2402/105    5699/255
5047/230      1/0         7117/239     7945/403       1420/87     4628/317      7952/587       2348/183
4283/348      1643/138
1/0         3131/127      4628/317      4632/413    3164/333      2461/291      2557/330    5439/752
6715/982        1/0          20            10            20/3           5             4               10/3
20/7        5/2


Columns 10 through 18 …………..
```

Plot all of the beta functions in the same figure.

```
plot(W,B)
grid on
legend('$z = 0.05$', '$z = 0.1$','$z = 0.2$','$z = 1$','interpreter','latex')
title('Beta function for $z = 0.05, 0.1, 0.2$, and $1$','interpreter','latex')
xlabel('$w$','interpreter','latex')
ylabel('$B(z,w)$','interpreter','latex')
```

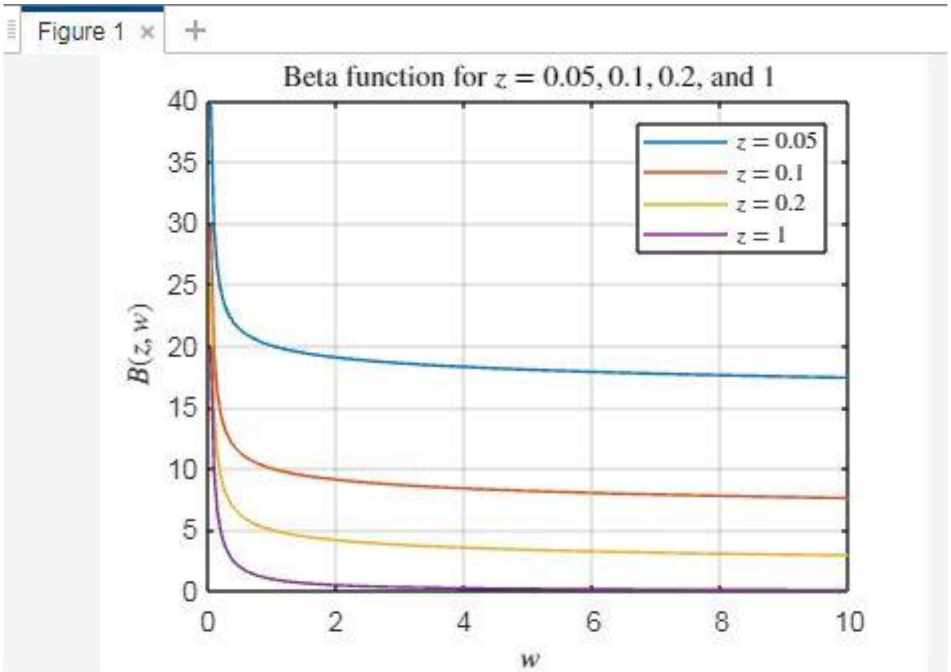


Figure 1
Beta function for $z = 0.05, 0.1, 0.2$, and 1
$z = 0.05$
$z = 0.1$
$z = 0.2$
$z = 1$
$B(z, w)$
w
0
2
4
6
8
10
5
15
20
25
30
35
40