Experiment No. (1)
Working with Matrices

1-1 Entering Matrix

The best way for you to get started with MATLAB is to learn how to handle matrices. You only have to follow a few basic conventions:

- Separate the elements of a row with blanks or commas.
- Use a semicolon (;) to indicate the end of each row.
- Surround the entire list of elements with square brackets, [ ].

For Example

```
>> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB displays the matrix you just entered.

```
A =
    16   3    2    13
    5    10   11   8
    9    6    7    12
    4    15   14   1
```

Once you have entered the matrix, it is automatically remembered in the MATLAB workspace. You can refer to it simply as **A**. Also you can enter and change the values of matrix elements by using **workspace** window.

1-2 Subscripts

The element in row **i** and column **j** of **A** is denoted by **`A(i,j)`**. For example, **`A(4,2)`** is the number in the fourth row and second column. For the above matrix, **`A(4,2)`** is **15**. So to compute the sum of the elements in the fourth column of **A**, type

```
>> A(1,4) + A(2,4) + A(3,4) + A(4,4)
ans =
34
```

You can do the above summation, in simple way by using **sum** command.

If you try to use the value of an element outside of the matrix, it is an error.

```
>> t = A(4,5)
??? Index exceeds matrix dimensions.
```

On the other hand, if you store a value in an element outside of the matrix, the size increases to accommodate the newcomer. The initial values of other new elements are zeros.

```
>> X = A;
>> X(4,5) = 17
X =
16   3    2    13   0
5    10   11   8    0
9    6    7    12   0
4    15   14   1    17
```

1-3 Colon Operator

The colon " : " is one of the most important MATLAB operators. It occurs in several different forms. The expression

```
>> 1:10
```

is a row vector containing the integers from 1 to 10

```
1 2 3 4 5 6 7 8 9 10
```

To obtain nonunit spacing, specify an increment. For example,

```
>> 100:-7:50
100 93 86 79 72 65 58 51
```

Subscript expressions involving colons refer to portions of a matrix.

```
>>A(1:k,j)
```

is the first k elements of the jth column of A.

The colon by itself refers to *all* the elements in a row or column of a matrix and the keyword **end** refers to the *last* row or column. So

```
>> A(4,:)            or    >> A(4,1:end)
```
give the same action

```
ans =
4   15   14   1

>> A(2,end)
ans =
8
```

1-4 Basic Matrix Functions

Command	Description

sum(x)
`>> x=[1 2 3`
`      4 5 6];`
`>> sum(x)`
`ans =`
`       5     7     9`
	The sum of the elements of x. For matrices, sum(x) is a row vector with the sum over each column.

`>> sum(x,2)`
`ans=`
`       6`
`       15`
	sum (x,dim) sums along the dimension dim.

`>>sum(sum(x))`
`ans =`
`    21`
	In order to find the sum of elements that are stored in matrix with *n* dimensions, you must use **sum** command *n* times in cascade form, this is also applicable for **max, min, prod, mean, median** commands.

Command	Description

mean(x)
`x=[1 2 3; 4 5 6];`
`>> mean(x)`
`ans =`
`    2.5   3.5    4.5`
 	
`>> mean(x,2)`
`ans =`
`         2`
`         5`
`>>mean(mean(x))`
`ans =`
`          3.5000`
	The average of the elements of x. For matrices, mean(x) is a row vector with the average over each column.
	mean (x,dim) averages along the dimension dim.

zeros(N)
zeros(N,M)
`>> zeros(2,3)`
`ans =`
`         0     0     0`
`         0     0     0`
	Produce N by N matrix of zeros.
	Produce N by M matrix of zeros.

ones(N)
ones(N,M)
`>> ones(2,3)`
`ans =`
`         1     1     1`
`         1     1     1`
	Produce N by N matrix of ones.
	Produce N by M matrix of ones.

Command	Description

`size(x)`
`>> x=[1 2 3`
`      4 5 6];`
`>> size(x)`
`ans =`
`        2     3`
	return the size (dimensions) of matrix x.

`length(v)`
`>> v=[1 2 3];`
`>> length(v)`
`ans =`
`        3`
	return the length (number of elements) of vector v.

`numel(x)`
 	
`>> v =[55 63 34];`
`>> numel(v)`
`ans =`
`     3`
 	
`>> x=[1 2`
`      4 5`
`      7 8 ];`
`>> numel(x)`
`ans =`
`     6`
	returns the number of elements in array x.

Command	**Description**

`single quote ( ' )`
 	
`>> x=[1 2 3`
`      4 5 6`
`      7 8 9];`
`>> x'`
`ans =`
`     1     4     7`
`     2     5     8`
`     3     6     9`
`>> v=[1 2 3];`
`>> v'`
`ans =`
`     1`
`     2`
`     3`
	Matrix transpose. It flips a matrix about its main diagonal and it turns a row vector into a column vector.

`max (x)`
`>> x=[1 2 3`
`      4 5 6];`
`>> max (x)`
`ans =`
`        4     5     6`
`>> max(max(x))`
`ans =`
`     6`
	Find the largest element in a matrix or a vector.

Command	**Description**

min (x)
>> x=[1 2 3
      4 5 6];
>> min (x)
ans =
        1   2   3
>> min(min(x))
ans =
        1
	Find the smallest element in a matrix or a vector.

magic(N)
>> magic(3)
ans =
        8   1   6
        3   5   7
        4   9   2
	produce N Magic square. This command produces valid magic squares for all N>0 except N=2.

inv(x)
>> x=[1 4;
      5 8];
>> inv(x)
ans =
   -0.6667   0.3333
    0.4167   -0.0833
	produce the inverse of matrix x.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>diag(x)
>> x=[1 2 3
4 5 6
7 8 9];
>> diag(x)
ans =
1
5
9
>> v=[1 2 3];
>> diag(v)
ans =
1 0 0
0 2 0
0 0 3</td><td>Return the diagonal of matrix x. if x is a vector then this command produce a diagonal matrix with diagonal x.</td></tr>
<tr><td>prod(x)
>> x=[1 2 3
4 5 6];
>> prod(x)
ans = 4 10 18
>> prod(prod(x))
ans =
720</td><td>Product of the elements of x. For matrices, Prod(x) is a row vector with the product over each column.</td></tr>
</table>

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>median(x)
x=[4 6 8
10 9 1
8 2 5];
>> median(x)
ans =
8 6 5
>> median(x,2)
ans =
6
9
5
>> median(median(x))
ans =
6</td><td>The median value of the elements of x. For matrices, median (x) is a row vector with the median value for each column.

median(x,dim) takes the median along the dimension dim of x.</td></tr>
<tr><td>sort(x,DIM,MODE)
>> x = [3 7 5
0 4 2];
>> sort(x,1)
ans =
0 4 2
3 7 5
>> sort(x,2)
ans =
3 5 7
0 2 4
>> sort(x,2,'descend')
ans =
7 5 3
4 2 0</td><td>Sort in ascending or descending order.
- For vectors, sort(x) sorts the elements of x in ascending order.
For matrices, sort(x) sorts each column of x in ascending order.

DIM= 1 by default
MODE= 'ascend' by default</td></tr>
</table>

Command	**Description**

`det(x)`
`>> x=[5 1 8`
`      4 7 3`
`      2 5 6];`
`>> det(x)`
`ans =`
`   165`
	Det is the determinant of the square matrix x.

`tril(x)`
`>> x=[5 1 8`
`      4 7 3`
`      2 5 6];`
`>> tril(x)`
`ans =`
`     5     0     0`
`     4     7     0`
`     2     5     6`
	Extract lower triangular part of matrix x.

`triu(x)`
`>> x=[5 1 8`
`      4 7 3`
`      2 5 6];`
`>> triu(x)`
`ans =`
`     5     1     8`
`     0     7     3`
`     0     0     6`
	Extract upper triangular part of matrix x.

Note

When we are taken away from the world of linear algebra, matrices become two-dimensional numeric arrays. Arithmetic operations on arrays are done element-by-element. This means that addition and subtraction are the same for arrays and matrices, but **that multiplicative operations are different**. MATLAB uses a **dot (.), or decimal point, as part of the notation for multiplicative array operations**.

Example: Find the factorial of 5

```
>> x=2:5;
>> prod(x)
```

Example: if x = [1,5,7,9,13,20,6,7,8], then

a) replace the first five elements of vector x with its maximum value.
b) reshape this vector into a 3 x 3 matrix.

solution

```
a)
 >> x(1:5)=max(x)

b)
 >> y(1,:)=x(1:3);
 >> y(2,:)=x(4:6);
 >> y(3,:)=x(7:9);
 >> y
```

Example: Generate the following row vector b=[1, 2, 3, 4, 5, ... 9,10], then transpose it to column vector.

solution

```
>> b=1:10
b =
    1    2    3    4    5    6    7    8    9    10
>> b=b';
```

Experiment No. (2)
LAGRANGE'S MEAN VALUE THEOREM

AIM

To find the maximum and minimum function (extreme values) for given function $y = f(x)$ using MATLAB

MATLAB CODE

```
clc;
syms x c;
f=input('Enter the function: ');
I=input('Enter the interval [a,b]: ');
df=diff(f,x);
dfc=subs(df,x,c);
rhs=((subs(f,x,I(2)))-(subs(f,x,I(1))))/(I(2)-I(1));
c=double(solve(dfc-rhs));
index=find(c>I(1)&c<I(2));
c=c(index);
for i=1:numel(c)
    disp(['The value of c is: ',num2str(c(i))])
    fc=double(subs(f,c(i)));
    m=double(subs(df,c(i)));
    b=double(subs(f,c(i))-subs(df,c(i))*c(i));
    tangent=m*x+c;
    disp('Tangent line is: ');
    disp(vpa(tangent,4));
    figure
    h=ezplot(tangent);
    set(h,'Color','black','LineWidth',1.5);
    hold on;
    h=ezplot(f,[I(1) I(2)]);
    set(h,'Color','red','LineWidth',1.5);
    plot([I(1) I(2)],[double(subs(f,I(1))) double(subs(f,I(2)))],'--
g','LineWidth',1,5);
    plot(c(i),fc,'o','MarkerEdgeColor','Blue','MarkerEdgeColor','blue');
    str=strcat('(',num2str(c(i)),',',num2str(fc),')');
    hold off;
    legend('Function','Tangent Line','Secant line',str,'Location','Best');
    title('Mean Value Theorem');
end
```

Output:

Enter functnx^2+1

f =

x^2 + 1

Enter interval [a,b][-2 2]

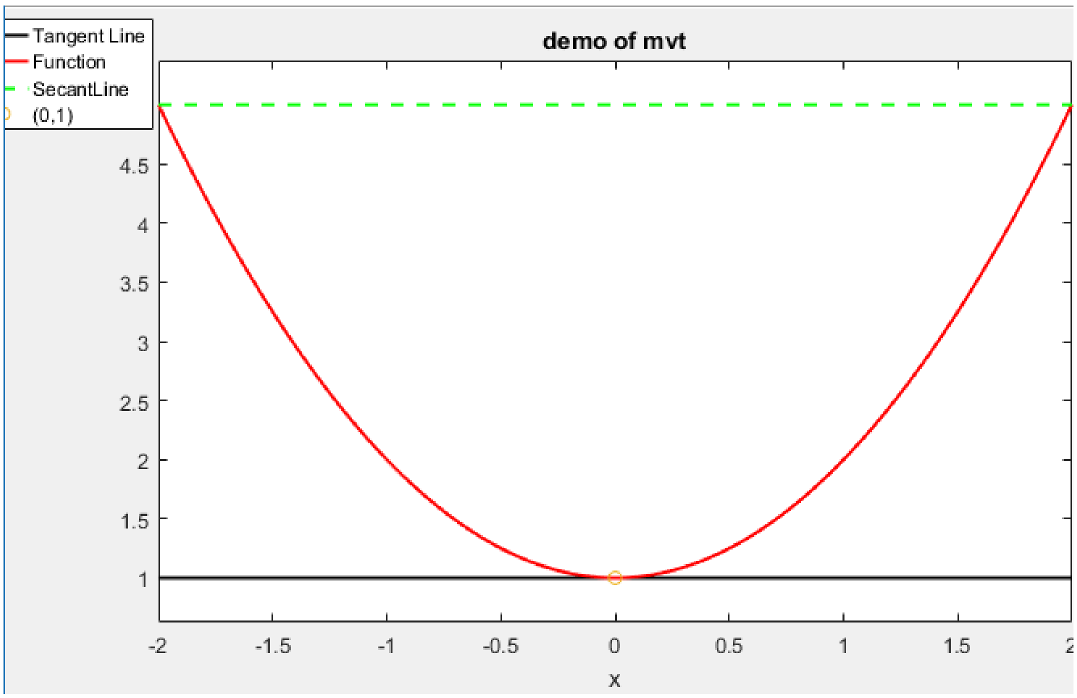
I =

-2 2

c =

0

The value of c is:0
Tangent Line is
1.0



3. INTEGRATION AS AREA UNDER THE CURVE

Aim:

- To write Matlab codes to evaluate the area under the curve
- To visualize the area enclosed between the coordinate axes and the curve y=f(x)
- A definite integral over an interval uses a limit process to assign measures to quantities such as area, volume, arc length, and mass. In this section, we will use a similar process to define the area enclosed by a curve between the lines through the coordinate axes.
- The goal is to find the area enclosed by the x-axis and the curve y=f(x) represented by the integration of the curve with a definite limits for x.

Mathematical form:

$Area = \int_a^b f(x)dx$ where a≤x≤b.

MATLAB Syntax used:

int(f, x, *a*, *b)*	Returns the definite integral of f with respect to x from *a* to *b*.
syms x	Symbolic variable x

Similarly, the above procedure can be extended to any number of integrals.

Example 1:

Evaluate the area enclosed between the curve $y = x^3 - 2x + 5$ and the ordinates $x = 1$and $x = 2$.

The area is given by $A = \int_1^2 (x^3 - 2x + 5)dx$

MATLAB Code:

```
syms x
f=x^3-2*x+5;
a=int(f,1,2)
disp('Area: '), disp(double(a));
```

Output:

```
a=
23/4
Area:
5.7500
```

Example 2:
Find the area under the curve $f(x) = x^2 cosx$ for $-4 \leq x \leq 9$.

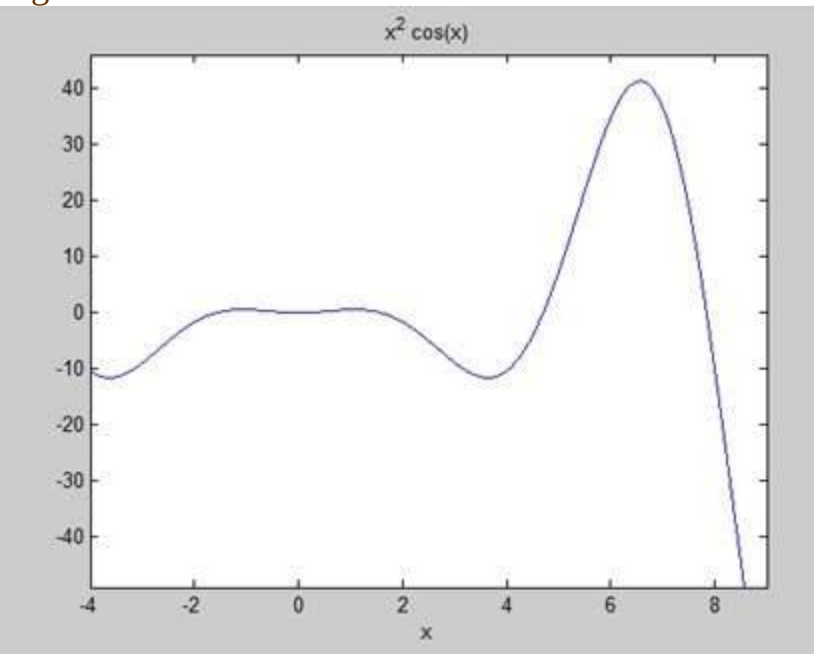
MATLAB Code:

```
syms x
f=x^2*cos(x);
ezplot(f,[-4,9])
a=int(f,-4,9)
disp('Area: '), disp(double(a));
```

Output:

```
a=
8*cos(4)+10*cos(9)+14*sin(4)+79*sin(9)
Area:
0.3326
```

Figure window:



4. MAXIMA-MINIMA OF A GIVEN FUNCTION

AIM:

To find the maximum and minimum function (extreme values) for given function $y = f(x)$ using MATLAB

PROCEDURE:

A critical value is any number $x = c$ such that $f'(c) = 0$ or $f'(c)$ does not exist

If $x = c$ is a critical value, then

1. $f''(c) > 0 \Rightarrow x = c$ gives a local minimum.
2. $f''(c) < 0 \Rightarrow x = c$ gives a local maximum.
3. $f''(c) = 0 \Rightarrow$ inconclusive (other methods needed)

SYNTAX:

Command	Description
Diff	Differentiate the function symbolically
Solve	Symbolic solution of algebraic equations, The input to solve can be either symbolic expressions or strings
Size	Dimensions of data and model objects and to access a specific size output.
Sprint	Format data into string. It applies the format to all elements of array A and any additional array arguments in column order, and returns the results to string *str*.
Double	Convert to double precision, double(x) returns the double-precision value for X. If X is already a double-precision array, double has no effect.

MATLAB CODE

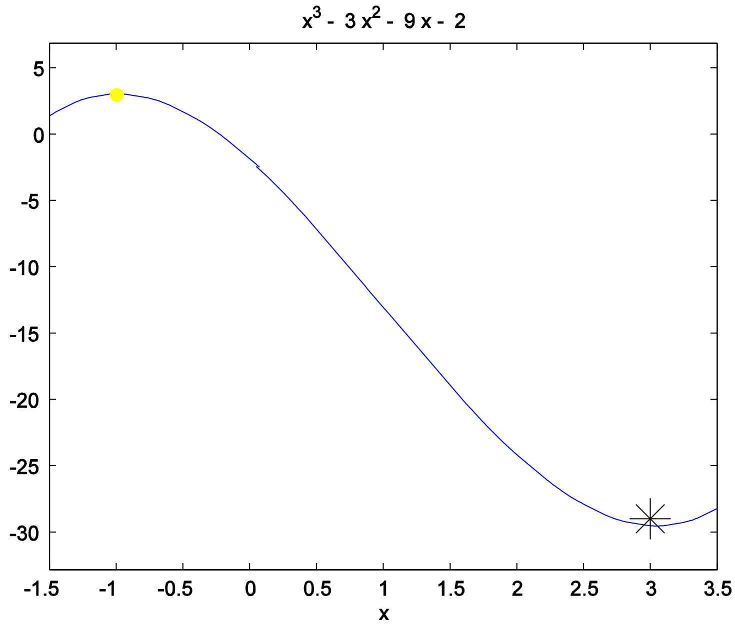
```
clc
clear all
syms x y real
f= input('Enter the function f(x):');
fx= diff(f,x);
ax=solve(fx)
fxx= diff(fx,x);
D=fxx;
figure
ezplot(f,[min(double(ax))-.5,max(double(ax))+.5]);
```

```
for i = 1:1:size(ax)
    T1=subs(D,x,ax(i));
    T2=subs(f,x,ax(i));

if(double(T1) == 0)
     sprintf('The point %d needs further investigation', double(ax(i)))

elseif (double(T1) < 0)
      sprintf('The maximum value of the function is %d at the point
%d',double(T2),double(ax(i)))
         st = 'y.';
else
  sprintf('The minimum value of the function is %d at the point
%d',double(T2),double(ax(i)))
            st = 'k*';
        end
    hold on
plot(double(ax(i)),double(T2),st,'markersize',20);
end
```

OUTPUT:

```
Enter the function f(x): x^3-3*x^2-9*x-2

ax =

 -1

  3

ans =

The maximum value of the function is 3 at the point -1

ans =

The minimum value of the function is -29 at the point 3
```

FIGURE WINDOW:



1. f(x) = x^3/3-9*x

5. LAPLACE TRANSFORMS

Evaluating Laplace transforms and inverse Laplace transforms of functions including impulse, Heaviside functions and applying convolution

Aim:

To evaluate Laplace transforms and inverse Laplace transforms of functions.

The Laplace Transform of a function $f(t)$ is defined as $F(s) = L[f(t)] = \int_0^{\infty} e^{-st} f(t)dt$, provided the integral exists.

List of Commands and Purposes

Command	Purpose
`laplace(f)`	To find the Laplace transform of a scalar symbol f with default independent variable t. The default return is a function of s.
`laplace(f,w)`	Returns the Laplace transform of fin symbol w instead of the default s.
`laplace(f,x,w)`	Assumes f as a function of the symbolic variable x and returns the Laplace transform as a function of w.
`ilaplace(F)`	To find the inverse Laplace transform of the scalar symbolic object F with default independent variable s. The default return is a function of t.
`ilaplace(F,x)`	Returns the inverse Laplace transform of the function F as a function of x instead of the default t.
`ilaplace(F,w,x)`	Assumes Fas a function of the symbolic variable w and returns the inverse Laplace transform of F as a function of x.
`heaviside(t-a)`	To input the heaviside's unit step function H(t−a).
`dirac(t-a)`	To input the dirac delta function δ(t−a).

Example 1. Write the MATLAB code to find the Laplace transform of t^2

MATLAB Code:
```
syms t s
f=input('Enter the function in terms of  t:');
F=laplace(f)
```

Input:
```
Enter the function in terms of  t:t^2
```

Output:
```
F =
2/s^3
```

Example 2. Write the MATLAB code to find the Laplace transform of $\sin(t)$ in terms of w.

MATLAB Code:
```
syms t w
f=input('Enter the function in terms of t:');
F=laplace(f,w)
```

Input:
```
Enter the function in terms of t: sin(t)
```

Output:
```
F =
1/(w^2 + 1)
```

Example 3. Write the MATLAB code to find the Laplace transform of x^3e^{-3x} in terms of w.

MATLAB Code:
```
clear all
clc
syms x w
f=input('Enter the function in terms of  x:');
F=laplace(f,x,w)
```

Input:
```
Enter the function in terms of x: x^3*exp(-3*x)
```

Output:
```
F =
 6/(w + 3)^4
```

Example 8. Write MATLAB commands to find (i) $L^{-1}\left[\frac{s}{s-a}\right]$ (ii) $L^{-1}\left[\frac{se^{-s}+ae^{-2s}}{s^2+a^2}\right]$

(i)

```
symss a
f=ilaplace(s/(s-a))
```

Output

```
f=
dirac(t) + a*exp(a*t)
```

(ii)

```
symss a
f=ilaplace((s*exp(-s)+a*exp(-2*s))/(s^2+a^2))
```

Output

```
f=
heaviside(t-1)*cos((a^2)^(1/2)*(t-1))+(a*heaviside(t-2)*
sin((a^2)^(1/2)*(t-2)))/(a^2)^(1/2)
```

Example 9. Write MATLAB code to find $f(t)=L^{-1}\left[\frac{e^{-s}(e^{-s}-1)^2}{s^2}\right]$ and hence plot the curve $f(t)$.

MATLAB code

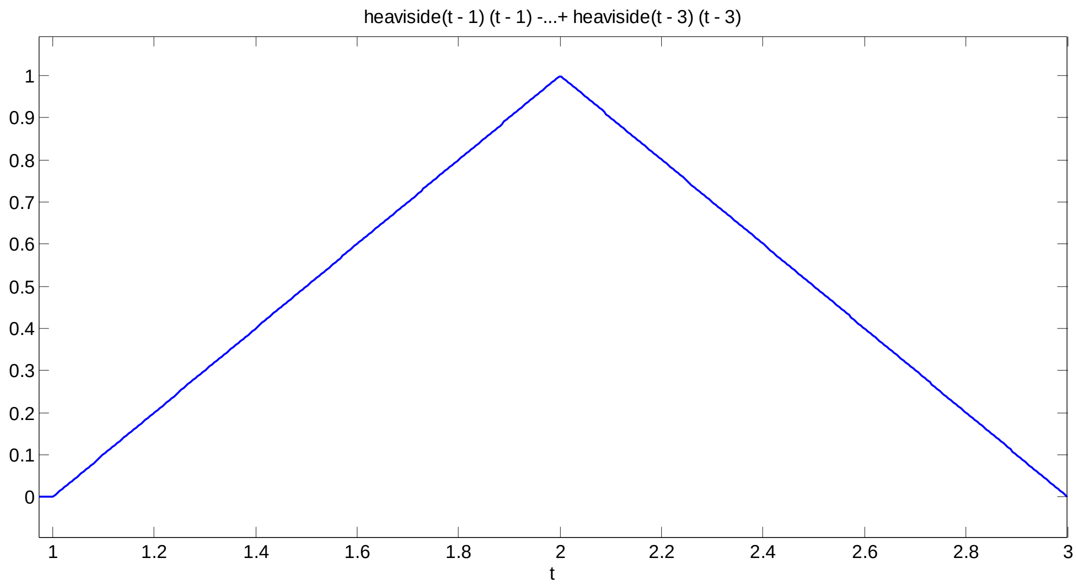
```
clear all
clc
syms s
F=input('Enter the function F(s):');
f=ilaplace(F)
ezplot(f)
```

Input

```
Enter the function F(s):(exp(-s)*(exp(-s)-1)^2)/(s^2)
```

Output

```
f =
heaviside(t-1)*(t-1)-2*heaviside(t-2)*(t-2)+heaviside(t-3)*(t-3)
```



Example 10. Write MATLAB code to find inverse Laplace transform of the function $F(s) = \dfrac{3s - 140}{s^2 + 400}$ and also find $L^{-1}[F(s+1)]$. Compare your answer by using first shifting theorem. Also plot the graphs of these functions in an interval [0,4] to observe the damping effect due to shifting.

MATLAB code

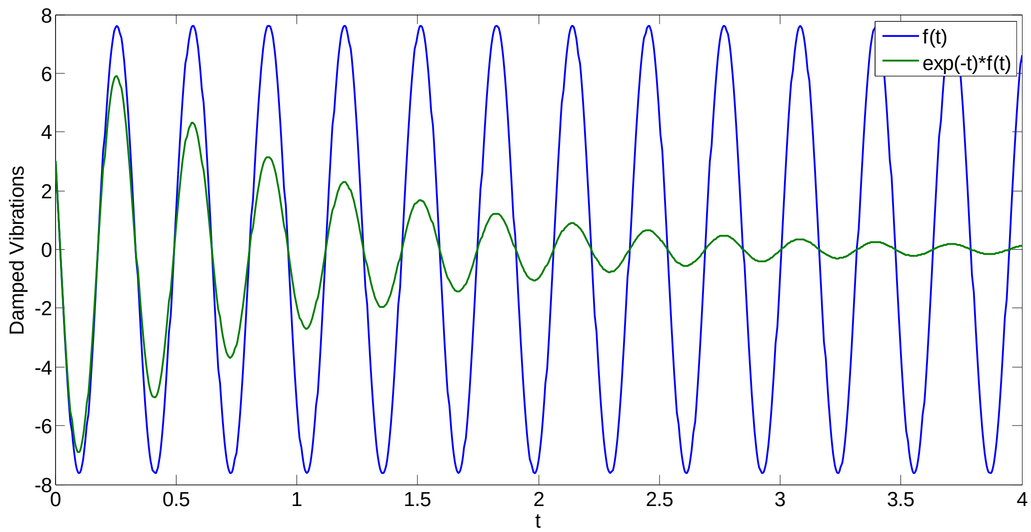
```
clear all
clc
syms s
F=input('Enter the function F(s):');
f=ilaplace(F)
F1=subs(F,s+1);
f1=ilaplace(F1)
x=linspace(0,4,1000);
y=subs(f,x);
y1=subs(f1,x);
plot(x,y,x,y1);
xlabel('t');
ylabel('Damped Vibrations');
legend('f(t)','exp(-t)*f(t)');
```

Input

```
enter the function F(s):(3*s-140)/(s^2+400)
```

Output

```
f =
3*cos(20*t)-7*sin(20*t)


f1 =

(3*(cos(20*t)-(7*sin(20*t))/3))/exp(t)
```



6 PLOTTING OF SURFACES AND VISUALIZING TANGENT PLANES

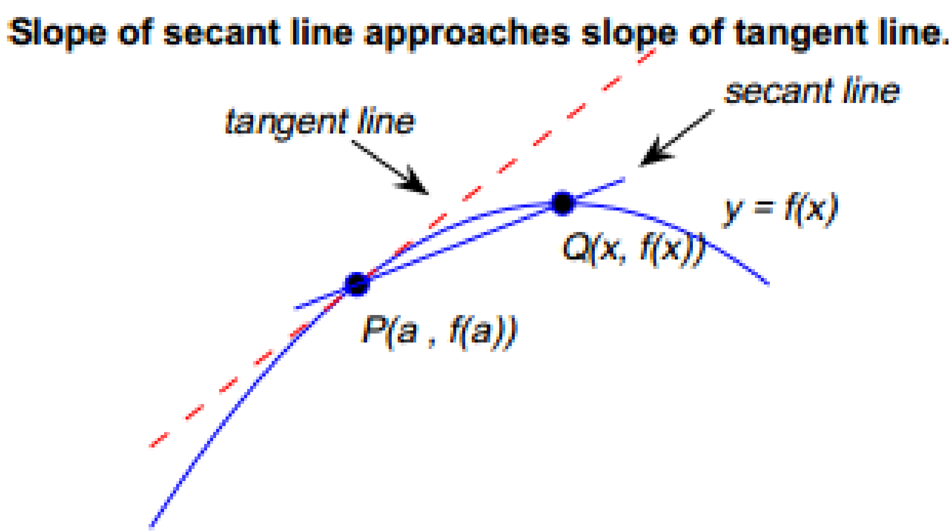
Aim:

- **Visualize the surface for the given function.**
- **Evaluate the derivative of a curve and visualize the tangent line at a point.**
- **Evaluate the equation of the tangent plane of a surface and visualize it.**

Relationship between tangent lines and secant lines:

$$m_{\tan} = \underset{\sec \to \tan}{lt} \; m_{\sec}$$

Geometrical Representation:



Command	Description
plot(y)	Plots the columns of Y versus the index of each value when Y is a real number. For complex Y, plot(Y) is equivalent to plot(real(Y),imag(Y))
plot3($X1$,$Y1$,$Z1$)	Displays a three-dimensional plot of a set of data points
surf(Z)	Creates a three-dimensional shaded surface from the z components in matrix Z, using x = 1:n and y = 1:m, wher e [m,n] = size(Z)
ezsurf(f)	Creates a graph of fun(x,y) using the surf function. fun is plotted over the default domain: $-2\pi < x < 2\pi$, $-2\pi < y < 2\pi$.
y = linspace(a,b,n)	Generates a row vector y of n points linearly spaced between and including a and b.
mesh(X,Y,Z)	Draws a wireframe mesh with color determined by Z so color is proportional to surface height
[X,Y] = meshgrid(x,y)	Transforms the domain specified by vectors x and y into arrays X and Y, which can be used to evaluate functions of two variables and three-dimensional mesh/surface plots.
subplot(m,n,p)	Breaks the figure window into an m-by-n matrix of small axes, selects the pth axes object for the current plot, and returns the axes handle.
colormap(*map*)	Sets the colormap to the matrix *map*. If any values in map are outside the interval [0 1], you receive the error Colormap must have values in [0, 1].

Program No.: 2

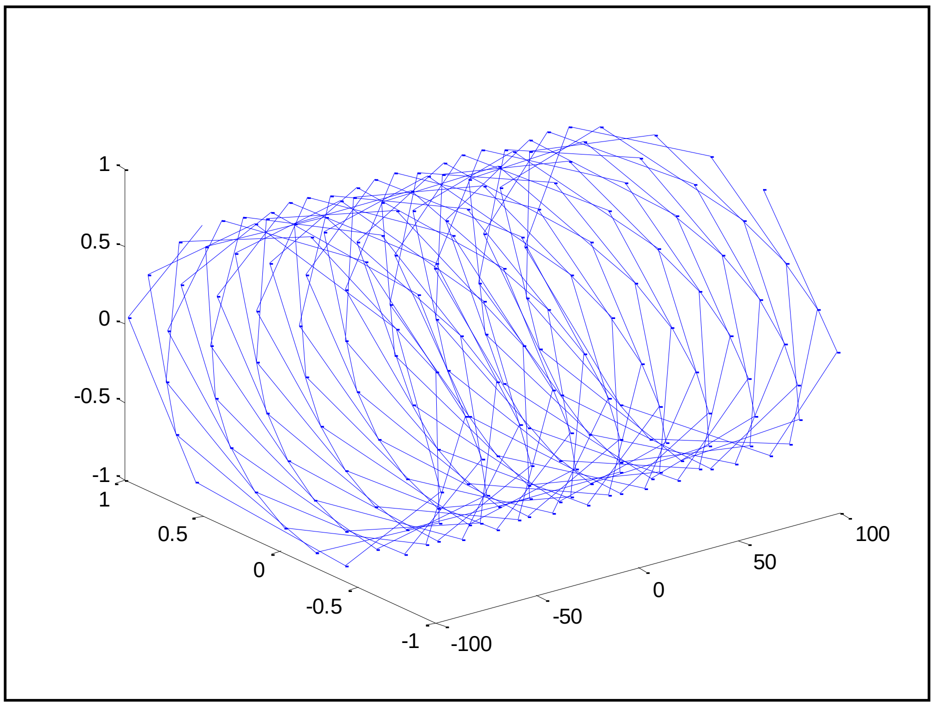
Draw the 3-D plot for the function $f(t)=(t,\ \sin t, \cos t)$, where $-100 \le t \le 100$.

MATLAB Code:

```
clc
clear all
syms t x y z
t=-100:1:100;
x=t;
y=sin(t);
z=cos(t);
plot3(x,y,z)
```

Output:

Figure Window:



Program No.: 3

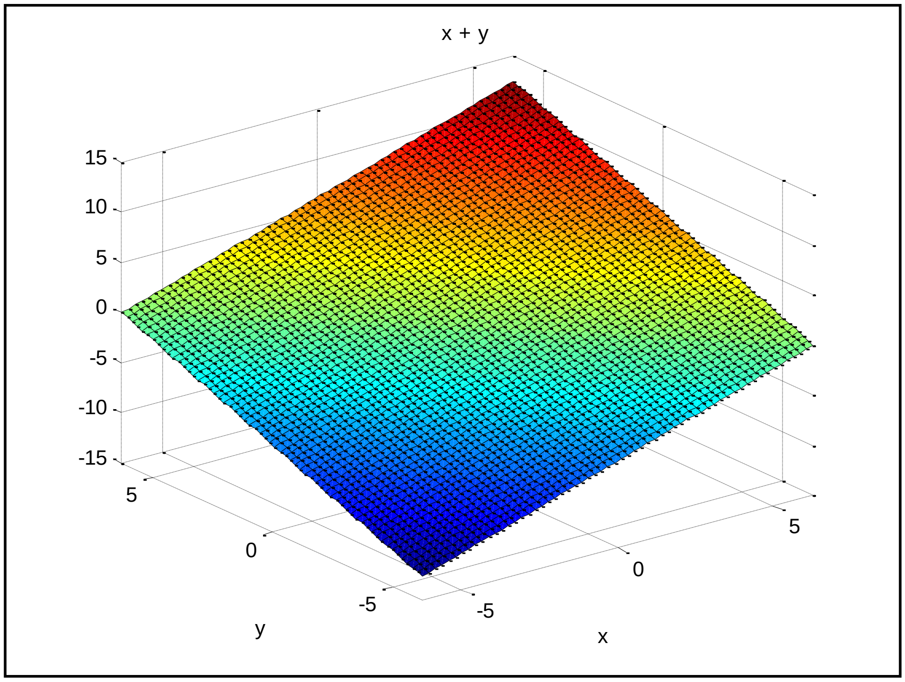
Draw the surface of the function $f(x,y)=x+y$ easily.

MATLAB Code:

```
clc
clear all
syms x y f
f=x+y;
ezsurf(f)
```

Output:

Figure Window:



Program No.: 6

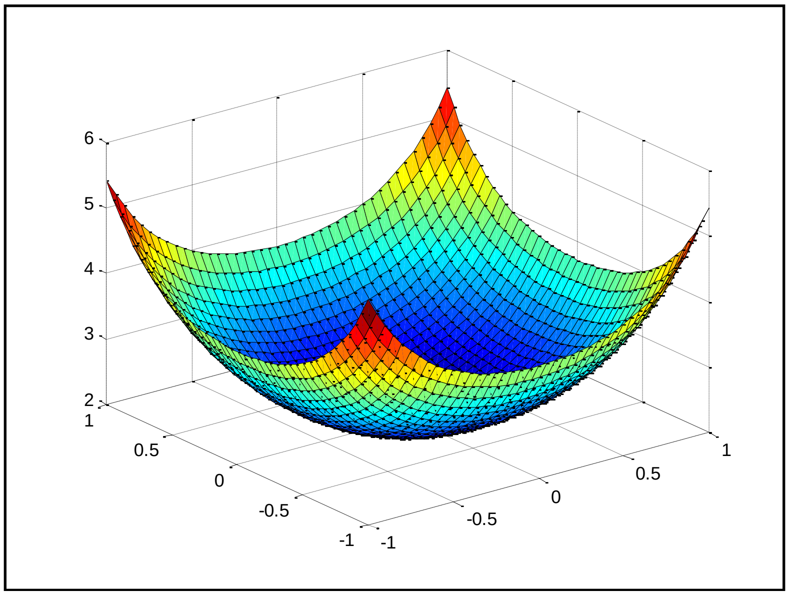
Draw the surface of the function $f(x,y)= x$+y, where $-1 \leq x \leq 1$ and $-1 \leq y \leq 1$.

MATLAB Code:

```
clc
clear all
syms x y f
x=-1:0.05:1;
y=-1:0.05:1;
[X,Y]=meshgrid(x,y);
Z=exp(X.^2)+exp(Y.^2);
surf(X,Y,Z)
```

Output:

<u>Figure Window</u>:



7. APPLICATIONS OF MULTIVARIABLE CALCULUS

Evaluating maxima and minima for functions of several variables

Aim:

To find the Maximum and Minimum values(Extreme values) for the given function$f(x,y)$ using MATLAB.

Mathematical form:

Let $z=f(x,y)$ be the given function. Critical points are points in the xy-plane where the tangent plane is horizontal. The tangent plane is horizontal, if its normal vector points in the z direction. Hence, critical points are solutions of the equations: $f_x(x,y)=0$ and $f_y(x,y)=0$.

MATLAB Syntax used:

diff	diff(expr) differentiates a symbolic expression expr with respect to its free variable as determined by symvar.
solve	Symbolic solution of algebraic equations, The input to solve can be either symbolic expressions or strings
size	Dimensions of data and model objects and to access a specific size output.
figure	Create figure graphics object, Figure objects are the individual windows on the screen in which the MATLAB software displays graphical output
double	Convert to double precision, double(x) returns the double-precision value for X. If X is already a double-precision array, double has no effect.
sprintf	Format data into string. It applies the *format* to all elements of array *A* and any additional array arguments in column order, and returns the results to string *str*.
ezsurf	Easy-to-use 3-D colored surface plotter, ezsurf(fun) creates a graph of fun(x,y) using the surf function. fun is plotted over the default domain: -2π <x< 2π, -2π <y< 2π.
plot3	The plot3 function displays a three-dimensional plot of a set of data points.

MATLAB Code:

- Input- Enter the function*f*(*x*, *y*).
- Visualizing function and its extreme values.

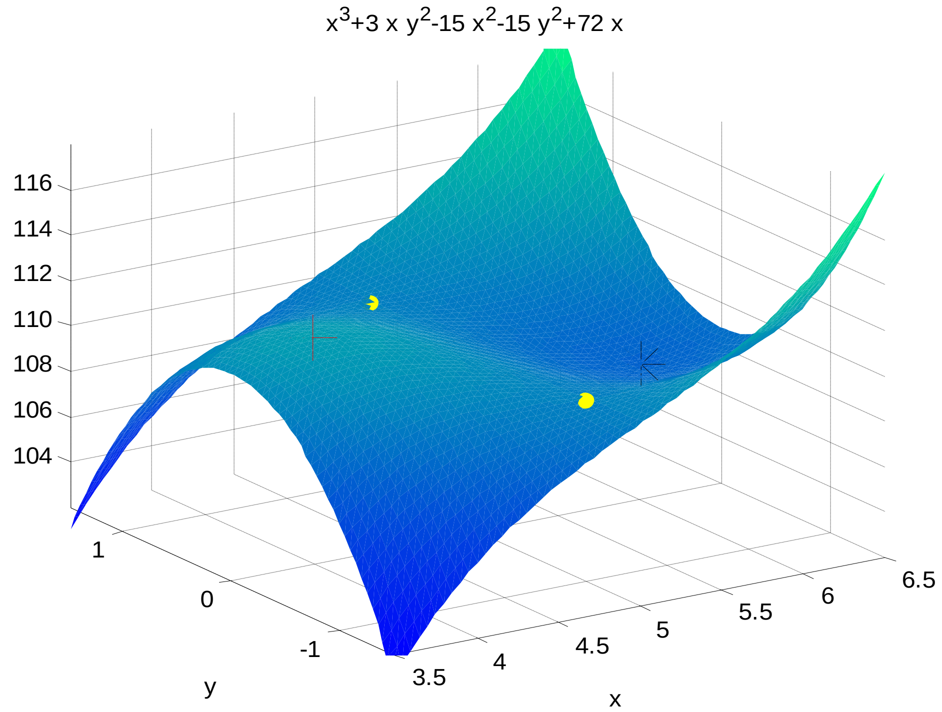
```
clc
clear all
syms xyreal
f= input('Enter the function f(x,y):');
fx= diff(f,x);
 fy=diff(f,y);
[ax ay] = solve(fx,fy)
fxx= diff(fx,x);
fxy=diff(fx,y);
fyy =diff(fy,y);
D=fxx*fyy - fxy^2;
figure
ezsurf(f,[min(double(ax))-.5,max(double(ax))+.5,min(double(ay)).5,max(double(ay))+.5]);
colormap winter
shading interp
for i = 1:1:size(ax)
    T1=subs(subs(D,x,ax(i)),y,ay(i));
    T2=subs(subs(fxx,x,ax(i)),y,ay(i));
    T3=subs(subs(f,x,ax(i)),y,ay(i));
if (double(T1) == 0)
     sprintf('The point (%d,%d) needs further investigation', double(ax(i)),double(ay(i)))
st='k+';
elseif (double(T1) < 0)
      sprintf('The point (%d,%d) is a saddle point',double(ax(i)),double(ay(i)))
      st = 'y.';
else
if (double(T2) < 0)
sprintf('The maximum value of the function is %d at the point %d,%d)',double(T3),double(ax(i)),double(ay(i)))
st = 'r+';
else
sprintf('The minimum value of the function is %d at the point (%d,%d)',double(T3),double(ax(i)),double(ay(i)))
            st = 'k*';
end
end
hold on
 plot3(double(ax(i)),double(ay(i)),double(T3),st,'markersize',20);
end
```

Output:

In Command window

Enter the function f(x,y):x^3+3*x*y^2-15*x^2-15*y^2+72*x
ax =
 6
 4
 5
 5
ay =
  0
  0
  1
 -1
ans =
The minimum value of the function is 108 at the point (6,0).
ans =
The maximum value of the function is 112 at the point (4,0).
ans =
The point (5,1) is a saddle point.
ans =
The point (5,-1) is a saddle point.

In Figure window



Example 2: Find the maximum and minimum value of the function $f(x,y) = 2(x^2 - y^2) - x^4 + y^4$.

Output:

Enter the function f(x,y):2*x^2-2*y^2-x^4+y^4

ax =

0

1

-1

0

0

1

1

-1

-1

ay =

0

0

0

1

-1

1

-1

1

-1

ans =The point (0,0) is a saddle point.

ans =The maximum value of the function is 1 at the point (1,0).

ans =The maximum value of the function is 1 at the point (-1,0).

ans =The minimum value of the function is -1 at the point (0,1).

ans =The minimum value of the function is -1 at the point (0,-1).

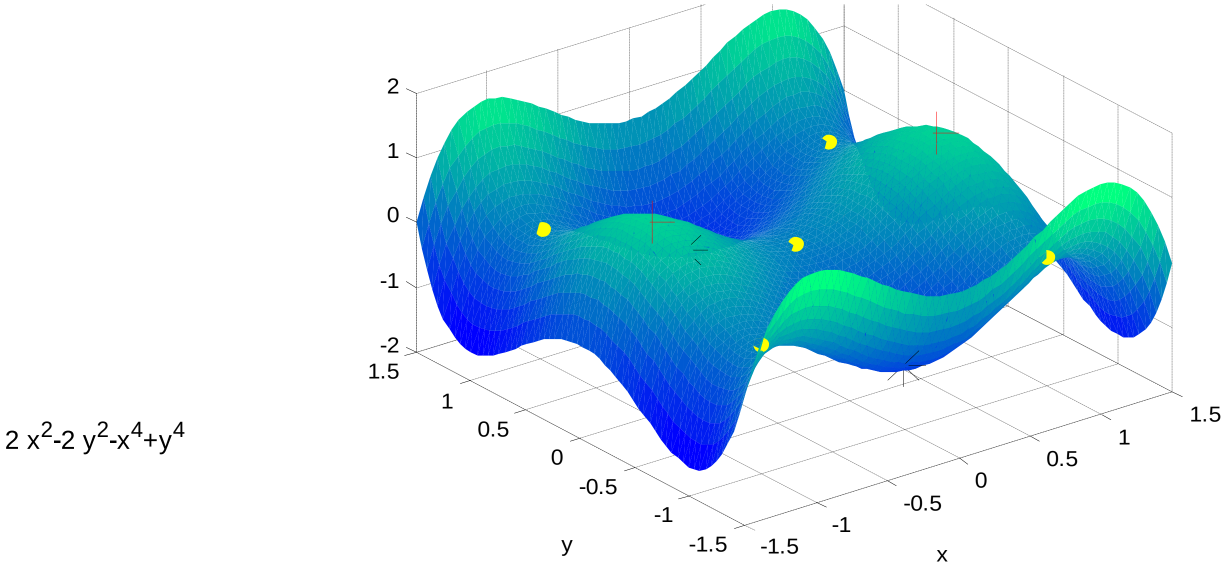
ans =The point (1,1) is a saddle point.

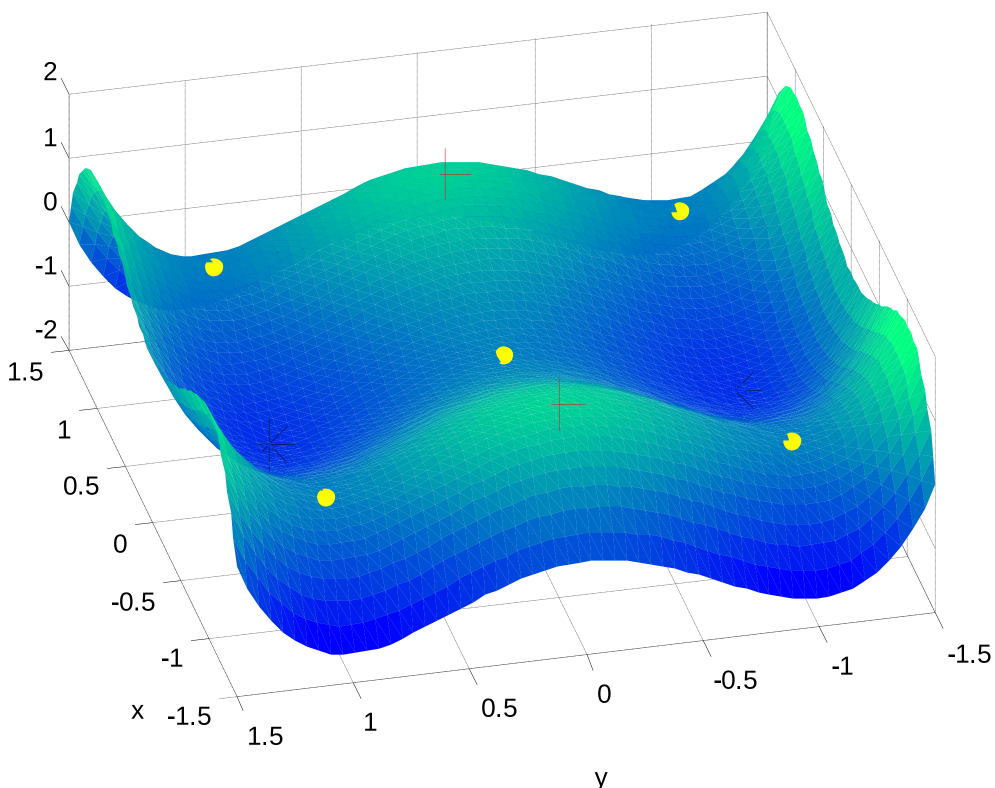
ans =The point (1,-1) is a saddle point.

ans =The point (-1,1) is a saddle point.

ans =The point (-1,-1) is a saddle point.

In Figure window





8. LAGRANGE's MULTIPLERS METHOD

Aim: To find the Maximum and Minimum values of $f(x,y,z)$ subject to the Constraint (assume that these extreme values exist) using **MATLAB**.

Description of the Problem & the method:

Here, we want to find the extreme values of $f(x,y,z)$ subject to a constraint of the form . Thus the point is restricted to lie on the level surface S with equation $f(x,y,z)=C$. In other words, if the maximum value of is $f(x_0,y_0,z_0)=C$, then the level surface is tangent to the level surface and so the corresponding gradient vectors are parallel.

MATLAB Syntax Used:

syms	short-cut for constructing symbolic variables.
input(' ')	displays the PROMPT string on the screen, waitsfor input from the keyboard
jacobian(f,v)	computes the Jacobian of the scalar or vector f with respect to the vector v. The (i,j)th entry of the result is df(i)/dv(j). Note that when f is scalar, the Jacobian of fis the gradient of f. Also, note that scalar v is allowed, although this is just diff(f,v).
solve(eqn1,eqn2,...,eqnM,var1,var2,...,varN)	symbolic solution of equations.
subs	symbolic substitution

MATLAB CODE: {Lagrange's Multiplier for THREE variables}

```
clc
clearall
symsxyzLreal
f=input('enter the objective function:');
g=input('enter the constraint function:');
h=f-(L*g);
gradh=jacobian(h,[x,y,z]);
[L,x1,y1,z1]=solve(g,gradh(1),gradh(2),gradh(3));
Z=[x1 y1 z1];
disp(['[x y z]='])
disp(Z)
```

MATLAB CODE: {Lagrange's Multiplier for TWO variables}

```
clc
clearall
symsxyL
f=input('enter the objective function:');
g=input('enter the constraint function:');
h=f-(L*g);
gradh=jacobian(h,[x,y]);
[L,x1,y1]=solve(g,gradh(1),gradh(2));
Z=[x1 y1];
a=length(x1);
fori=1:1:a


 G(i)=subs(f,[x,y],Z(i,:))
end
```

9- Evaluating Volume under Surfaces

Aim: To evaluate the volume under surface using double integral and to visualize the same using MatLab.

Solution Approach: We evaluate the double integrals by repeated application of the symbolic toolbox command for integration that, on applying twice, will read as**:**

$$\mathrm{volume} = \mathrm{int}\big(\mathrm{int}\big(\mathrm{f(x,y)}, y, y\ \ (x), y_2(x)\big), x, x\ \ , x_2\big).$$

The first function "viewSolid" is used to visualize the integrals in which the order of integration is as given in (1) and "viewSolidone" is for the integrals of the form (2).

What follows is the syntax for using "viewSolid" and "viewSolidone" commands:

viewSolid(z,0,f(x,y),y,y1(x),y2(x),x,x1,x2)

viewSolidone(z,0,f(x,y),x,x1(y),x2(y),y,y1,y2)

It should be observed that the "viewSolid" command is used when y1 and y2 are functions of x whereas x1 and x2 are constants. The "viewSolidone" command is used in the reverse case. Now we consider few examples for illustration of the approach mentioned above.

Example 1:
Set up a double integral to find the volume of a sphere of unit radius.

Solution:
Let the sphere be $x^2 + y^2 + z^2 = 1$. We know that due to the symmetry the volume of the sphere is 8 times its volume in the first octant. Thus we setup a double integral to find the volume below the surface of the sphere in the first octant only and write the total volume as:

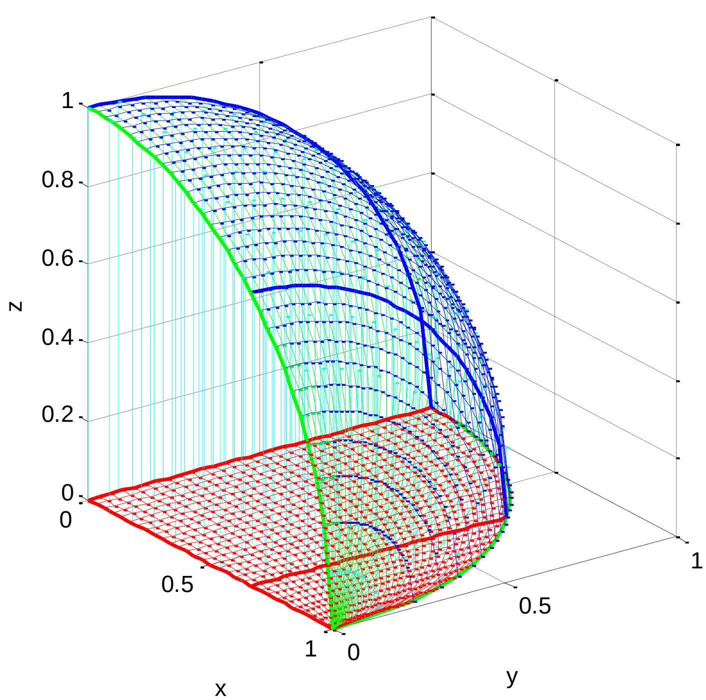
$$V = 8\int_{0}^{1}\int_{0}^{\sqrt{1-x^2}} \sqrt{1-x^2-y^2}\,dy\,dx\,.$$

MATLAB Code:

```
clc
clear all
syms x y z
vol=8*int(int(sqrt(1-x^2-y^2),y,0,sqrt(1-x^2)),x,0,1)
viewSolid(z,0+0*x*y,sqrt(1-x^2-y^2),y,0+0*x,sqrt(1-x^2),x,0,1);   axis   equal;
grid on;
```

Output:
vol =

(4*pi)/3



Example 2:

Find the volume of the solid that lies under the paraboloid $z = x\ \ + y\ \ $ and above the region D in the xy-plane bounded by the lines $y = 2x$ and the parabola $y = x$

Solution:

The double integral for this problem can be setup as:

$$\int_{0}^{4} \int_{y/2}^{\sqrt{y}} (x\ \ + y\ \ )\, dx\, dy$$

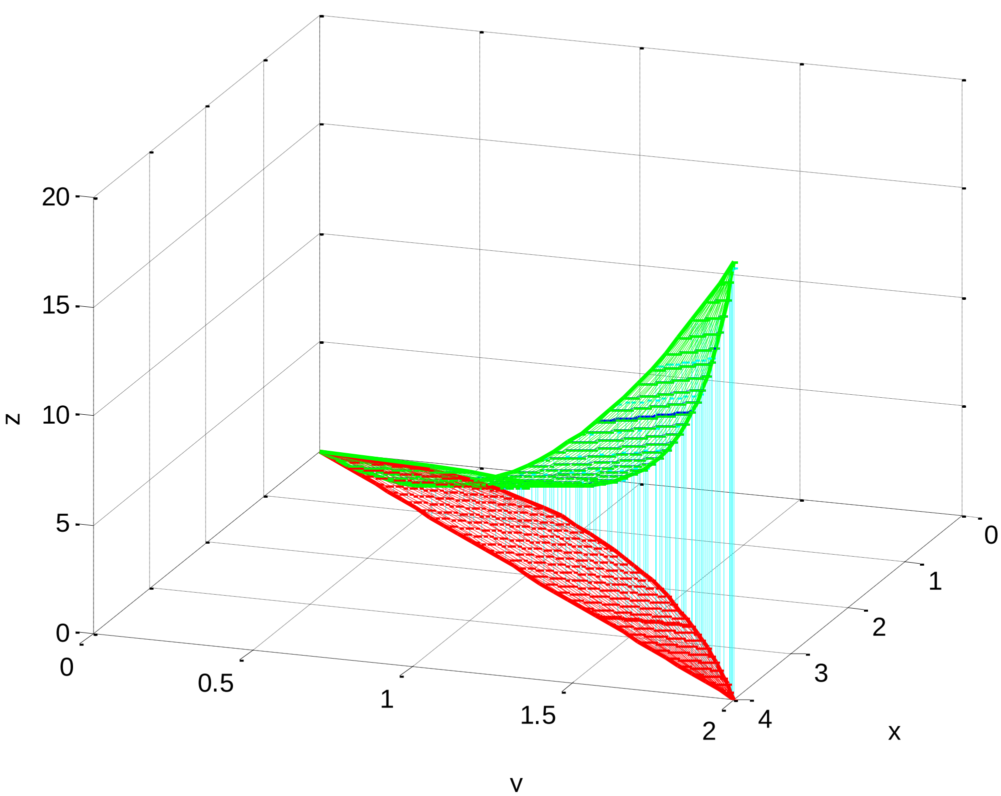
MATLAB Code:

```
clc
clear all
syms x y z
vol = int(int(x^2+y^2, x,y/2,sqrt(y)), y, 0, 4)
viewSolidone(z,0+0*x*y,x^2+y^2,x,y/2,sqrt(y),y,0,4); grid on
```

Output:

vol =

216/35



10 -**Triple integral for finding Volume of a solid region**

Aim: To find and visualize a volume of a solid region

Mathematical form:

Triple integral

${}_D f(x, y, z)dV$ where D={(x,y,z)|a≤x≤b, g_1(x)≤y≤g_2(x),h_1(x,y)≤z≤h_2(x,y)}or D={(x,y,z)|a≤z≤b, g_1(z)≤y≤g_2(z),h_1(y,z)≤x≤h_2(y,z)}

Syntax for triple integral:

int(int(int(f(x,y,z), z, lower limit of z, upper limit of z), y, lower limit of y, upper limit of y), x, lower limit of x, upper limit of x)

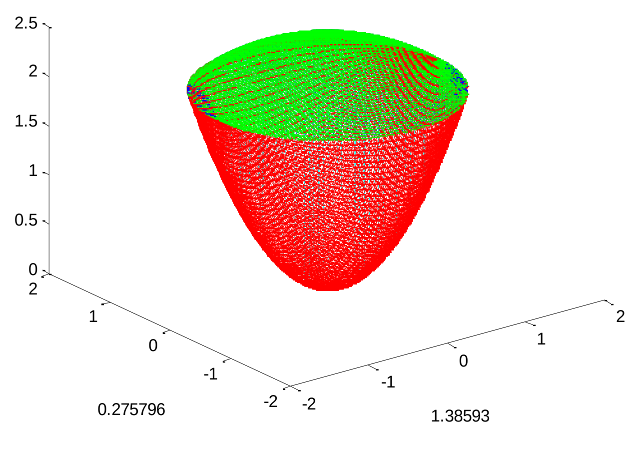
viewSolid(z, lower limit of z, upper limit of z, y, lower limit of y, upper limit of y, x, lower limit of x, upper limit of x)

Example 2:

Set up a triple integral for the volume of solid regionbounded below by the parabolid z=$x^2 + y^2$ and above the sphere $x^2 + y^2 + z^2 = 6$

MATLAB Code:

```
clc
clear all
syms x y z
viewSolid(z,x^2+y^2,sqrt(6-x^2-y^2), y, -sqrt(2-x^2), sqrt(2-x^2), x, -
sqrt(2),sqrt(2))
```



Example 4:

Find and view the volume of the sphere $x^2+y^2+z^2=1$ in the first octant.

Here $$0 \le z \le \sqrt{1-x^2-y^2}$$

$$0 \le x \le \sqrt{1-y^2}$$

$$0 \le y \le 1$$

Volume= $\iiint dzdxdy$

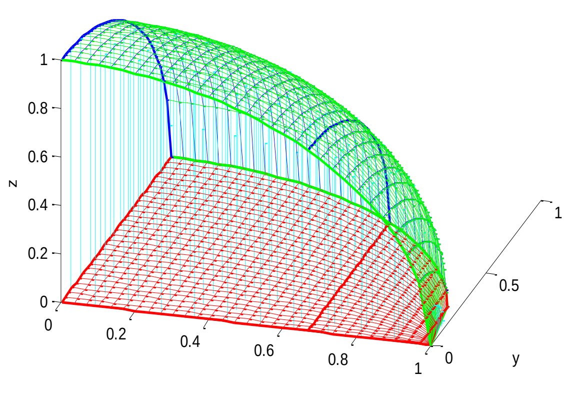
MATLAB CODE:

```
clc
clear all
syms x y z
int(int(int(1,z,0,sqrt(1-x^2-y^2)),x,0,sqrt(1-y^2)),y,0,1)
viewSolidone(z,0+0*y,sqrt(1-x^2-y^2),x,0+0*y,sqrt(1-y^2),y,0,1)
```

Output in the command Window:

ans = pi/6

Output in the figure Window:



11. Evaluating integrals using cylindrical and Spherical coordinates

Evaluate triple integrals using cylindrical coordinates

Cylindrical Coordinates:

In a cylindrical coordinates system, a point p in a space is represented by an ordered triple (r, θ, z) where (r, θ) is the polar representation of a point in the xy −plane.

Converting Cylindrical to rectangular

$$x = rcos\,\theta; y = rsin\,\theta; z = z$$

Rectangular to Cylindrical

$$r^2 = x^2 + y^2; \tan\theta = \frac{y}{x}; z = z$$

Theorem:

$$\iiint_V f(x, y, z)dV = \iiint f(rcos\,\theta, rsin\,\theta, z)rdzdrd\theta.$$

Evaluate triple integrals using spherical coordinates

In a spherical coordinates system, a point p in a space is represented by (ρ, θ, φ)

Converting Spherical to rectangular

$$x = \rho\sin\varphi\cos\theta; y = \rho\,sin\varphi\sin\theta; z = \rho cos\varphi$$

Rectangular to Spherical

$$\rho^2 = x^2 + y^2 + z^2, \theta = \cos^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right), \varphi = \tan^{-1}\left(\frac{y}{x}\right).$$

Theorem:

$$\iiint_V f(x, y, z)dV = \iiint f(\rho\sin\varphi\cos\theta, \rho\sin\varphi\sin\theta, \rho\cos\varphi)\rho^2\sin\varphi d\rho\, d\theta\, d\varphi\,.$$

The following MATLAB function files are used in this experiment:

Function Name	Syntax	Meaning
ezsurfvs	ezsurfvs(f, a, b, g_1, g_2)	Plot graph of $f(x, y)$ over vertically simple region $a \le x \le b$ and $g_1(x) \le y \le g_2(x)$
regionvs	regionvs(a, b, g_1, g_2)	Plot vertically simple region $a \le x \le b$ and $g_1(x) \le y \le g_2(x)$
ezsurfpol	ezsurfpol$(f, \alpha, \beta, h_1, h_2)$	Plot graph of $f(r, \theta)$ over region $\alpha \le \theta \le \beta$ and $h_1(\theta) \le r \le h_2(\theta)$
regionpol	regionpol$(\alpha, \beta, h_1, h_2)$	Plot region given by $\alpha \le \theta \le \beta$ and $h_1(\theta) \le r \le h_2(\theta)$

(These supporting function files can be downloaded from the link ftp://10.30.2.53/MATLAB/).

Triple integral (Cartesian coordinates: x, y, z)

Problem 1: The region D consistes of the points (x, y, z) with $x^2 + y^2 + z^2 \le 4$, $x^2 + y^2 \le 1$ and $z \ge 0$. Find the volume of this region.

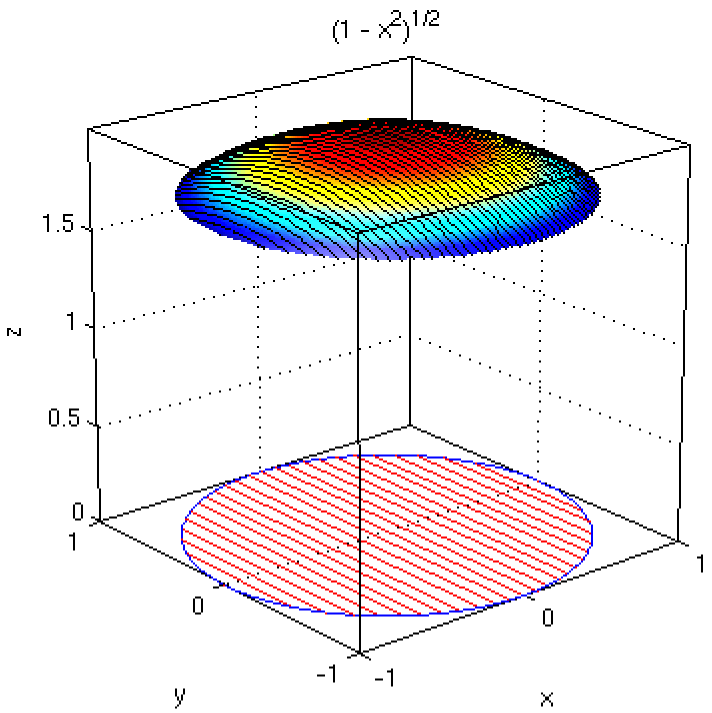
Note that $x^2 + y^2 + z^2 \le 4$ gives points inside of a sphere with radius 2, and $x^2 + y^2 \le 1$ gives points inside a cylinder of radius. We have $-1 \le x \le 1, -\sqrt{1 - x^2} \le y \le \sqrt{1 - x^2}, 0 \le z \le \sqrt{4 - x^2 - y^2}$.

MATLAB Code:

```
clear all
clc
syms x y z real
f = 1;
a = -1; b = 1;                          % limits for x
g1 = -sqrt(1-x^2); g2 = sqrt(1-x^2);    % limits for y
h1 = 0; h2 = sqrt(4-x^2-y^2);           % limits for z
I1 = int(f,z,h1,h2)                     % z integral: this is easy
I2 = int(I1,y,g1,g2)                    % y-integral
I = int(I2,x,a,b)                       % z-integral
Id = double(I)

ezsurfvs(h2,a,b,g1,g2); hold on         % draw upper surface
regionvs(a,b,g1,g2);                    % draw region R in xy-plane
```

Output:

```
I1 =
(- x^2 - y^2 + 4)^(1/2)

I2 =
3^(1/2)*(1 - x^2)^(1/2) - 2*asin((1 - x^2)^(1/2)/(4 - x^2)^(1/2))*(x^2/2 - 2)

I =
int(3^(1/2)*(1 - x^2)^(1/2) - 2*asin((1 - x^2)^(1/2)/(4 - x^2)^(1/2))*(x^2/2 - 2),
x == -1..1)

Id =
    5.8724
```



Triple integral (Cylindrical coordinates: r, θ, z)

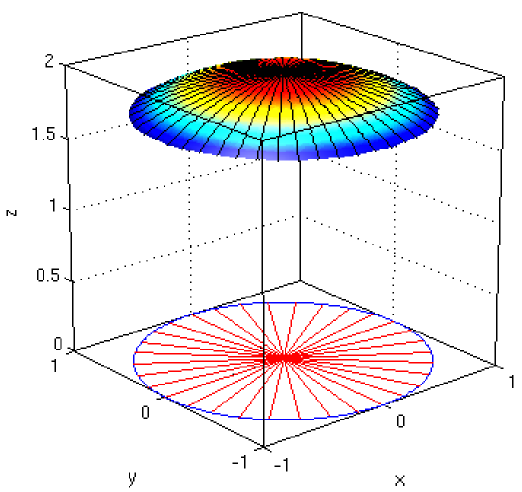
Consider the same example discussed above. Now our aim is to evaluate the integral using cylindrical coordinates. We have $0 \leq \theta \leq 2\pi, 0 \leq r \leq 1, 0 \leq z \leq \sqrt{4 - r^2}$

MATLAB Code:

```
clear all
clc
syms r theta z real
f = 1;                                % integral f*r*dr*dtheta*dz
a = 0; b = 2*pi;                      % limits for theta
G1 = 0; G2 = 1;                       % limits for r
H1 = 0; H2 = sqrt(4-r^2);             % limits for z
I1 = int(f*r,z,H1,H2)                 % z integral
I2 = int(I1,r,G1,G2)                  % r-integral
I = int(I2,theta,a,b)                 % theta-integral
Id = double(I)                        % we get same result as in previous example!

ezsurfpol(H2,a,b,G1,G2); hold on      % draw upper surface
regionpol(a,b,G1,G2); hold off        % draw region R in xy-plane
```

Output:

```
I1 =
r*(4 - r^2)^(1/2)

I2 =
8/3 - 3^(1/2)

I =
-2*pi*(3^(1/2) - 8/3)

Id =
    5.8724
```



Problem 2: Consider the region D of the points (x, y, z) with $x^2 + y^2 + z^2 \leq 4$ and $z \geq r\sqrt{3}$. Find the volume of this region.

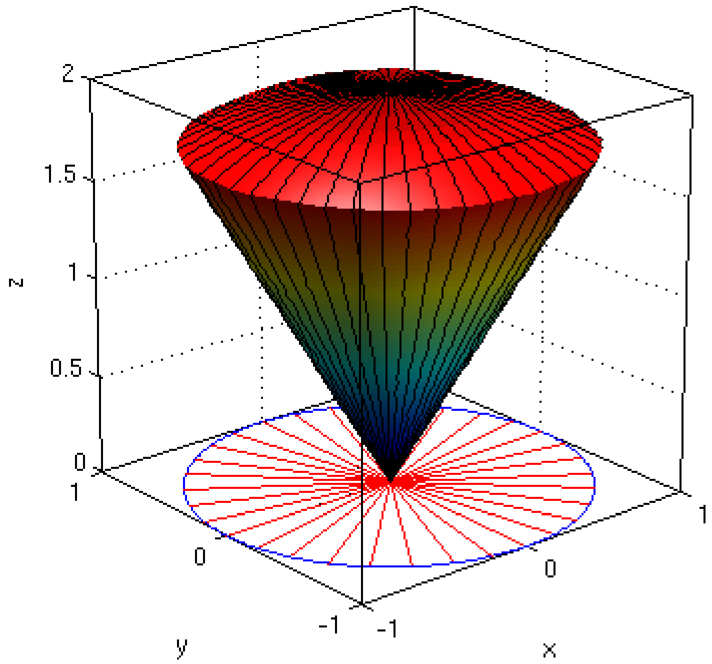
Note that $x^2 + y^2 + z^2 \leq 4$ gives points inside of sphere with radius 2 and $z \geq r\sqrt{3}$ gives points in a cone. We have $0 \leq \theta \leq 2\pi; 0 \leq r \leq 1; r\sqrt{3} \leq z \leq \sqrt{4 - r^2}$

MATLAB Code:

```
clear all
clc
syms r theta z real
f = 1;                              % integral f*r*dr*dtheta*dz
a = 0; b = 2*pi;                    % limits for theta
G1 = 0; G2 = 1;                     % limits for r
H1 = sqrt(3)*r; H2 = sqrt(4-r^2);   % limits for z
I1 = int(f*r,z,H1,H2)               % z integral
I2 = int(I1,r,G1,G2)                % r-integral
I = int(I2,theta,a,b)               % theta-integral
Id = double(I)

ezsurfpol(H1,a,b,G1,G2); hold on    % draw lower surface
ezsurfpol(H2,a,b,G1,G2);            % draw upper surface
regionpol(a,b,G1,G2); hold off      % draw region R in xy-plane
```

Output:

```
I1 =
-r*(3^(1/2)*r - (4 - r^2)^(1/2))

I2 =
8/3 - (4*3^(1/2))/3

I =
-(8*pi*(3^(1/2) - 2))/3

Id =
    2.2448
```



Triple integral (Spherical coordinates: ρ, φ, θ)

Consider the same example discussed above. Now our aim is to evaluate the integral using spherical coordinates. Note that on the boundary of the cone we have $z = r\sqrt{3}$. since $\frac{r}{z} = \tan\varphi$ we have $\varphi = \tan^{-1}\frac{1}{\sqrt{3}} = \frac{\pi}{6}$. Hence $0 \le \theta \le 2\pi, 0 \le \varphi \le \frac{\pi}{6}; 0 \le \rho \le 2$.

MATLAB Code:

```
clear all
clc
syms rho phi theta real

f = 1;                                  % integral f*rho^2*sin(phi)*drho*dphi*dtheta
a = 0; b = 2*pi;                        % limits for theta
h1 = 0; h2 = pi/6;                      % limits for phi
F1 = 0; F2 = 2;                         % limits for rho
I1 = int(f*rho^2*sin(phi),rho,F1,F2)    % z integral
I2 = int(I1,phi,h1,h2)                  % phi-integral
I = int(I2,theta,a,b)                   % theta-integral
Id = double(I)                          % we get same result as in previous problem 2 !
```

Output:

```
I1 =
(8*sin(phi))/3
```

```
I2 =
8/3 - (4*3^(1/2))/3

I =
-(8*pi*(3^(1/2) - 2))/3

Id =
    2.2448
```

12:- EVALUATING GRADIENT, CURL AND DIVERGENCE

Aim

- To visualize the 2-D and 3-D vector fields
- To find and visualize the gradient, divergence, and curl

- **MATLAB Syntax**

inline(expr)	**Constructs an inline function object from the MATLAB expression contained in the string expr.**
vectorize(fun)	Inserts a . before any ^, * or / in s. The result is a character string
quiver(x,y,u,v)	Displays velocity vectors as arrows with components (u,v) at the points (x,y)
quiver3(x,y,z,u,v,w)	Plots vectors with components (u,v,w) at the points (x,y,z))
vectarrow(p0,p1)	Plots a line vector with arrow pointing from point p0 to point p1. The function can plot both 2D and 3D vector with arrow depending on the dimension of the input

Some Important Commands

vectorize Command	inline Command
>> f = -(sin(x) + sin(y))^2; >> vectorize(f) ans = -(sin(x) + sin(y)).^2	>> f = inline('3*sin(2*x^2)') f= Inline function: f = f(x) = 3*sin(2*x^2) >> f = inline('sin(alpha*x)') f = Inline function: f(alpha,x) = sin(alpha*x)

Example1: Draw the two dimensional vector field for the vector $2x\vec{i} + 3y\vec{j}$

MATLAB Code

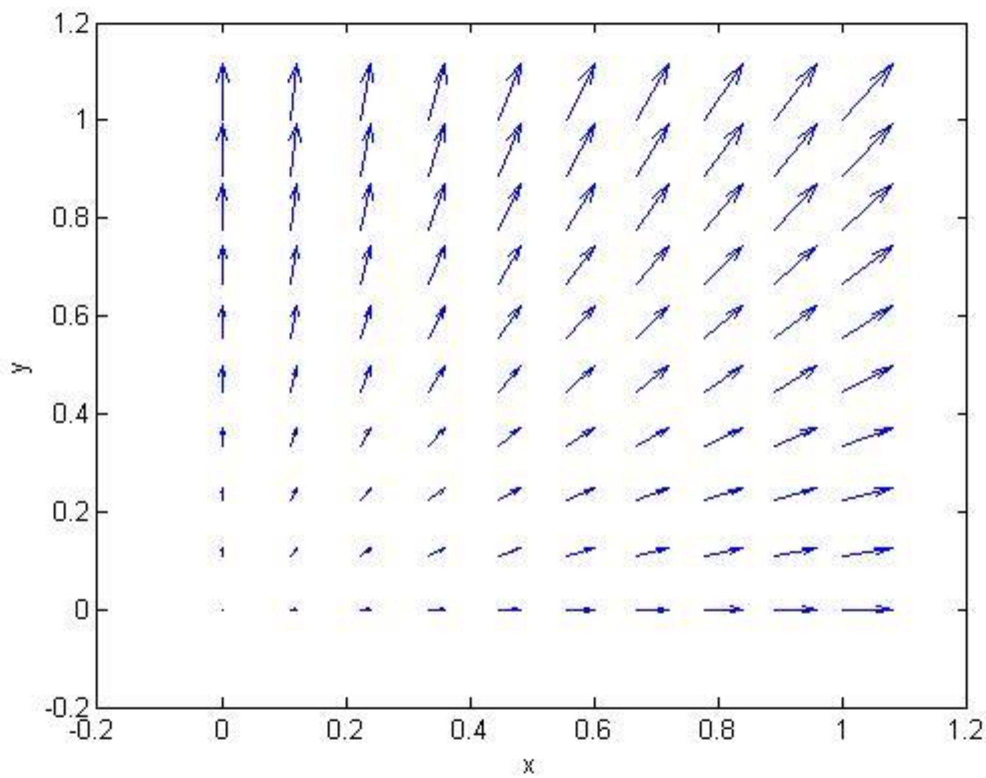
```
syms x   y
f=input('enter the vector as i,j order in vector form: ');
P = inline(vectorize(f(1)), 'x', 'y');
Q = inline(vectorize(f(2)), 'x', 'y');
x = linspace(0, 1, 10); y = x;
[X,Y] = meshgrid(x,y);
U = P(X,Y);
V = Q(X,Y);
quiver(X,Y,U,V,1)
axis on
xlabel('x')
ylabel('y')
```

Output

In the Command window:

Enter the vector as i, j order in vector form: [2*x   3*y]

In the Figure window:



Example 2:

Draw the three dimensional vector field for the vector $x\vec{i} + y\vec{j} + z\vec{k}$

MATLAB Code

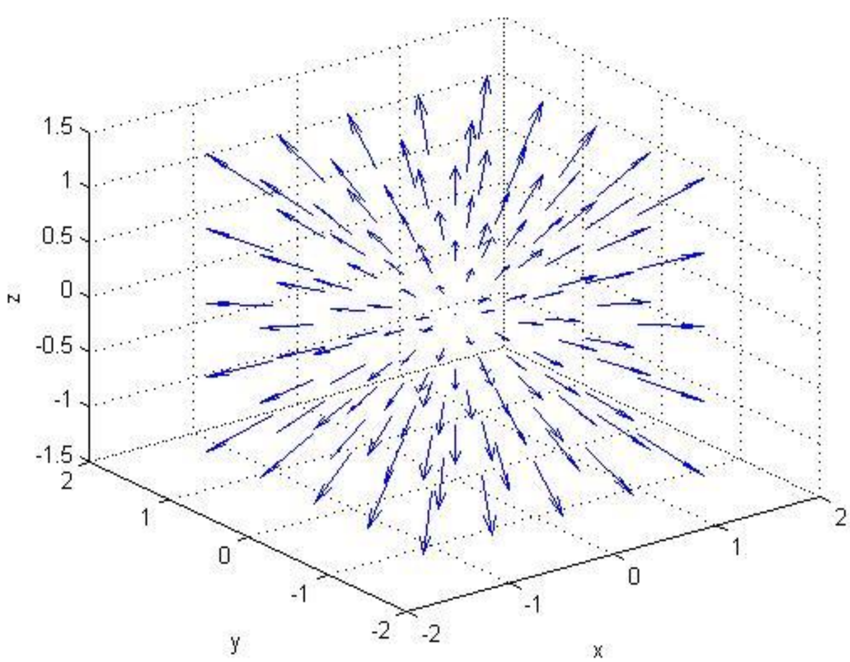
```
syms x   y   z
F=input('enter the vector as i,j and k order in vector form: ');
P = inline(vectorize(F(1)), 'x', 'y','z');
Q = inline(vectorize(F(2)), 'x', 'y','z');
R = inline(vectorize(F(3)), 'x', 'y','z');
x = linspace(-1 , 1, 5); y = x; z=x;
[X,Y,Z] = meshgrid(x,y,z);
U = P(X,Y,Z);
V = Q(X,Y,Z);
W = R(X,Y,Z);
quiver3(X,Y,Z,U,V,W,1.5);
axis on
xlabel('x')
ylabel('y')
zlabel('z')
```

Output

In the Command Window:

enter the vector as i, j and k order in vector form: [x   y   z]

In the figure window:



13. *Evaluating Line Integrals*

Aim:

Write a MATLAB code to find the work done by the force $\vec{\boldsymbol{F}}$ using line integral

Mathematical form:

Let the given function be $\vec{F} = F\ (x, y, z)\vec{\imath} + F\ (x, y, z)\vec{\jmath} + F\ (x, y, z)\vec{k}$.

Let the parametric form be $r = [r\ (t) \quad r\ (t) \quad r\ \ (t)]$ with $a \le t \le b$.

Then $\int_a^b \vec{F}.d\vec{r} = \int_a^b [F(r(t)).r'(t)]dt$

Example 1 Evaluate $\int_C \vec{F}\cdot d\vec{r}$ where $\vec{F}(x,y,z) = 8x^2yz\vec{i} + 5z\vec{j} - 4xy\vec{k}$ and C is the curve given by $\vec{r}(t) = t\vec{i} + t^2\vec{j} + t^3\vec{k}$, $0 \le t \le 1$.

Solution

Okay, we first need the vector field evaluated along the curve.

$$\vec{F}(\vec{r}(t)) = 8t^2(t^2)(t^3)\vec{i} + 5t^3\vec{j} - 4t(t^2)\vec{k} = 8t^7\vec{i} + 5t^3\vec{j} - 4t^3\vec{k}$$

Next we need the derivative of the parameterization.

$$\vec{r}'(t) = \vec{i} + 2t\vec{j} + 3t^2\vec{k}$$

Finally, let's get the dot product taken care of.

$$\vec{F}(\vec{r}(t))\cdot\vec{r}'(t) = 8t^7 + 10t^4 - 12t^5$$

The line integral is then,

$$\int_C \vec{F}\cdot d\vec{r} = \int_0^1 8t^7 + 10t^4 - 12t^5\,dt$$

$$= \left(t^8 + 2t^5 - 2t^6\right)\Big|_0^1$$

$$= 1$$

Example 2 Evaluate $\int_C \vec{F}\cdot d\vec{r}$ where $\vec{F}(x,y,z) = xz\vec{i} - yz\vec{k}$ and C is the line segment from $(-1,2,0)$ and $(3,0,1)$.

Solution

We'll first need the parameterization of the line segment.

$$\vec{r}(t) = (1-t)\langle -1,2,0\rangle + t\langle 3,0,1\rangle$$
$$= \langle 4t-1, 2-2t, t\rangle, \qquad 0 \le t \le 1$$

So, let's get the vector field evaluated along the curve.

$$\vec{F}(\vec{r}(t)) = (4t-1)(t)\vec{i} - (2-2t)(t)\vec{k}$$
$$= (4t^2 - t)\vec{i} - (2t - 2t^2)\vec{k}$$

Now we need the derivative of the parameterization.

$$\vec{r}'(t) = \langle 4, -2, 1\rangle$$

The dot product is then,

$$\vec{F}(\vec{r}(t)) \cdot \vec{r}'(t) = 4(4t^2 - t) - (2t - 2t^2) = 18t^2 - 6t$$

The line integral becomes,

$$\int_C \vec{F} \cdot d\vec{r} = \int_0^1 18t^2 - 6t\, dt$$
$$= (6t^3 - 3t^2)\Big|_0^1$$
$$= 3$$

MATLAB Code:

```
clc
clear all
syms x y z t
F =input('enter the i, j and k components of force in the vector form');
% for  two dimensions, k component is zero
T =input('enter the parametric form of x, y, and z  as a vector input ');
R=[x y z];
L=input('enter the parametric lower limit');
U=input('enter the parametric upper limit ');
R1=subs(R,[x,y,z],[T(1),T(2),T(3)]);
DR1=diff(R1,t);
F1=subs(F,[x,y,z],[T(1),T(2),T(3)])
NF=F1.*DR1;
NF1=NF(1)+NF(2)+NF(3);
I=int(NF1,t,L,U)
```

Example 1 Evaluate $\int_C \vec{F} \cdot d\vec{r}$ where $\vec{F}(x,y,z) = 8x^2yz\vec{i} + 5z\vec{j} - 4xy\vec{k}$ and C is the curve given by $\vec{r}(t) = t\vec{i} + t^2\vec{j} + t^3\vec{k}$, $0 \le t \le 1$.

OUTPUT:

In command window:

```
enter the i, j and k components of force in the vector form[8*x^2*y*z 5*z -4*x*y]
enter the parametric form of x, y, and z  as a vector input [t  t^2 t^3]
enter the parametric lower limit0
enter the parametric upper limit 1
F1 =
[ 8*t^7, 5*t^3, -4*t^3]
I =
 1
```

Example-2

Integrate f (x, y, z) = x **i** – 3y^2 **j**+ z **k** over the line segment C joining the origin to the point (1, 1, 1).

OUTPUT:

In command window:

```
enter the i, j and k components of force in the vector form[x -3*y^2 z]
enter the parametric form of x, y, and z  as a vector input [t t t]
enter the parametric lower limit0
enter the parametric upper limit 1
U =
     1
F1 =
[ t, -3*t^2, t]

I =
 0
```