KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)

Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

OFFICE OF THE DEAN - FRESHMAN ENGINEERING

DEPARTMENT OF MATHEMATICS

211MAT1301

(Linear Algebra and Calculus)

MATLAB

Academic Year (2021-2022) / ODD Semester

MATLAB RECORD

Student Name :

Register Number :

Year /Semester :

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

OFFICE OF THE DEAN - FRESHMAN ENGINEERING

DEPARTMENT OF MATHEMATICS

BONAFIDE CERTIFICATE

Certified that this is the bonafide record of work done by Ms/Mr. ________________________________

________________________ Register No. ________________________ of First year for the course

211MAT1301 - Linear Algebra and Calculus during First Semester in the academic year

2021 - 2022

Staff In-charge **Dean/FE**

MARK SUMMARY

S. No.	Topic	Marks	Faculty Signature
1	Cayley Hamilton Theorem		
2	Diagonalization of Matrices		
3	Linearly Independent / Dependent vectors		
4	Rolle's Theorem		
5	Mean Value Theorem		
6	Ordinary Differential Equations		
7	Maxima and Minima		
8	Limit of a function		
9	Partial Derivatives		
10	Beta and Gamma Functions		

Cayley Hamilton Theorem

Aim:

To write a MATLAB program to verify Cayley Hamilton theorem for the matrix $A = \begin{bmatrix} 7 & 2 & -2 \\ -6 & -1 & 2 \\ 6 & 2 & -1 \end{bmatrix}$. Also compute A^{-1} and A^4 using Cayley Hamilton Theorem.

Main Commands:

syms - to define symbolic variables and fynctions

charpoly - to create the characteristic polynomial

format rational - to create numbers in the rational form

Source Code:

```
syms A s
A=[7 2 -2; -6 -1 2; 6 2 -1];
I=[1 0 0; 0 1 0; 0 0 1];
charpoly(A,s)==0
```

ans = $s^3 - 5\,s^2 + 7\,s - 3 = 0$

```
verification=A^3-5*A^2+7*A-3*I
```

```
verification = 3×3
     0     0     0
     0     0     0
     0     0     0
```

```
format rational
inv_A=(1/3)*(A^2-5*A+7*I)
```

```
inv_A =
      -1             -2/3            2/3
       2              5/3           -2/3
      -2             -2/3            5/3
```

```
A_power4=5*A^3-7*A^2+3*A
```

```
A_power4 =
     241            80           -80
    -240           -79            80
     240            80           -79
```

Aim:

To write a MATLAB program to verify Cayley Hamilton theorem for the matrix $A = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 2 \end{bmatrix}$. Also compute A^{-1} and A^4 using Cayley Hamilton Theorem.

Main Commands:

syms - to define symbolic variables and fynctions

charpoly - to create the characteristic polynomial

format rational - to create numbers in the rational form

Source Code:

```
syms A s
A=[2 1 1; 0 1 0; 1 1 2];
I=[1 0 0; 0 1 0; 0 0 1];
charpoly(A,s)==0
```

ans = $s^3 - 5\,s^2 + 7\,s - 3 = 0$

```
verification=A^3-5*A^2+7*A-3*I
```

```
verification =
       0              0              0
       0              0              0
       0              0              0
```

```
format rational
inv_A=(1/3)*(A^2-5*A+7*I)
```

```
inv_A =
       2/3           -1/3           -1/3
       0              1              0
      -1/3           -1/3            2/3
```

```
A_power4=5*A^3-7*A^2+3*A
```

```
A_power4 =
      41             40             40
       0              1              0
      40             40             41
```

Aim:

To write a MATLAB program to verify Cayley Hamilton theorem for the matrix $A = \begin{bmatrix} -2 & 2 & -3 \\ 2 & 1 & -6 \\ -1 & -2 & 0 \end{bmatrix}$. Also compute A^{-1} and A^4 using Cayley Hamilton Theorem.

Main Commands:

syms - to define symbolic variables and fynctions

charpoly - to create the characteristic polynomial

format rational - to create numbers in the rational form

Source Code:

```
syms A s
A=[-2 2 -3; 2 1 -6; -1 -2 0];
I=[1 0 0; 0 1 0; 0 0 1];
charpoly(A,s)==0
```

ans = $s^3 + s^2 - 21\,s - 45 = 0$

```
verification=A^3+A^2-21*A-45*I
```

```
verification =
       0              0              0
       0              0              0
       0              0              0
```

```
format rational
inv_A=(1/45)*(A^2+A-21*I)
```

```
inv_A =
      -4/15          2/15          -1/5
       2/15         -1/15          -2/5
      -1/15         -2/15          -2/15
```

```
A_power4=-A^3+21*A^2+45*A
```

```
A_power4 =
     149            136            -204
     136            353            -408
     -68           -136             285
```

Aim:

To write a MATLAB program to verify Cayley Hamilton theorem for the matrix $A = \begin{bmatrix} 1 & 1 & 3 \\ 1 & 3 & -3 \\ 2 & -4 & -4 \end{bmatrix}$. Also compute A^{-1} and A^4 using Cayley Hamilton Theorem.

Main Commands:

syms - to define symbolic variables and fynctions

charpoly - to create the characteristic polynomial

format rational - to create numbers in the rational form

Source Code:

```
syms s
A=[1 1 3; 1 3 -3; 2 -4 -4];
I=[1 0 0; 0 1 0; 0 0 1];
charpoly(A,s)==0
```

ans = $s^3 - 32\,s + 56 = 0$

```
verification=A^3-32*A+56*I
```

```
verification =
       0              0              0
       0              0              0
       0              0              0
```

```
format rational
inv_A=(1/56)*(-A^2+32*I)
```

```
inv_A =
       3/7            1/7            3/14
       1/28           5/28          -3/28
       5/28          -3/28          -1/28
```

```
A_power4=32*A^2-56*A
```

```
A_power4 =
     200           -312           -552
    -120            536            360
    -432            416           1312
```

Diagonalization of Matrices

Aim:

To write a MATLAB program to diagonalize the matrix $A = \begin{bmatrix} 1 & 1 & 3 \\ 1 & 5 & 1 \\ 3 & 1 & 1 \end{bmatrix}$ using orthogonal transformation.

Main Commands:

syms - to define symbolic variables and fynctions

eig - to get the eigen values

charpoly - to create the characteristic polynomial

Source Code:

```
syms s
A =[1 1 3;1 5 1;3 1 1]
```

```
A =
       1              1              3
       1              5              1
       3              1              1
```

```
charpoly(A,s)==0
```

ans = $s^3 - 7s^2 + 36 = 0$

```
eig_A = eig(A)
```

```
ans =
      -2
       3
       6
```

```
[N D]= eig (A);
M(: ,1)=N(: ,1)* sqrt (2) ;
M(: ,2)=N(: ,2)* sqrt (3) ;
M(: ,3)=N(: ,3)* sqrt (6)
```

```
M =
      -1              1              1
       *             -1              2
       1              1              1
```

```
[N D]= eig (A)
```

```
N =
    -985/1393       780/1351       881/2158
        *          -780/1351       881/1079
     985/1393       780/1351       881/2158
D =
      -2              0              0
       0              3              0
       0              0              6
```

Aim:

To write a MATLAB program to diagonalize the matrix $A = \begin{bmatrix} 2 & 1 & -1 \\ 1 & 1 & -2 \\ -1 & -2 & 1 \end{bmatrix}$ using orthogonal transformation.

Main Commands:

syms - to define symbolic variables and fynctions

eig - to get the eigen values

charpoly - to create the characteristic polynomial

Source Code:

```
syms s
A =[2 1 -1; 1 1 -2; -1 -2 1]
```

```
A =
       2              1             -1
       1              1             -2
      -1             -2              1
```

```
charpoly(A,s)==0
```

ans = $s^3 - 4\,s^2 - s + 4 = 0$

```
eig_A = eig(A)
```

```
ans =
      -1
       1
       4
```

```
[N D]= eig (A);
M(: ,1)=N(: ,1)* sqrt (2) ;
M(: ,2)=N(: ,2)* sqrt (6) ;
M(: ,3)=N(: ,3)* sqrt (3)
```

```
M =
       *              2             -1
       1             -1             -1
       1              1              1
```

```
[N D]= eig (A)
```

```
N =
       *           881/1079      -780/1351
     985/1393     -881/2158      -780/1351
     985/1393      881/2158       780/1351
D =
      -1              0              0
       0              1              0
       0              0              4
```

Aim:

To write a MATLAB program to diagonalize the matrix $A = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 5 & -1 \\ 1 & -1 & 3 \end{bmatrix}$ using orthogonal transformation.

Main Commands:

syms - to define symbolic variables and fynctions

eig - to get the eigen values

charpoly - to create the characteristic polynomial

Source Code:

```
syms s
A =[3 -1 1; -1 5 -1; 1 -1 3]
```

```
A =
       3            -1             1
      -1             5            -1
       1            -1             3
```

```
charpoly(A,s)==0
```

ans = $s^3 - 11\,s^2 + 36\,s - 36 = 0$

```
eig_A = eig(A)
```

```
ans =
       2
       3
       6
```

```
[N D]= eig (A);
M(: ,1)=N(: ,1)* sqrt (2) ;
M(: ,2)=N(: ,2)* sqrt (3) ;
M(: ,3)=N(: ,3)* sqrt (6)
```

```
M =
       1            -1             1
       *            -1            -2
      -1            -1             1
```

```
[N D]= eig (A)
```

```
N =
     985/1393      -780/1351       881/2158
        *          -780/1351      -881/1079
    -985/1393      -780/1351       881/2158
D =
       2             0             0
       0             3             0
       0             0             6
```

Aim:

To write a MATLAB program to diagonalize the matrix $A = \begin{bmatrix} 2 & 0 & 4 \\ 0 & 6 & 0 \\ 4 & 0 & 2 \end{bmatrix}$ using orthogonal transformation.

Main Commands:

syms - to define symbolic variables and fynctions

eig - to get the eigen values

charpoly - to create the characteristic polynomial

Source Code:

```
syms s
A =[2 0 4; 0 6 0; 4 0 2]
```

```
A =
       2              0              4
       0              6              0
       4              0              2
```

```
charpoly(A,s)==0
```

ans = $s^3 - 10\,s^2 + 12\,s + 72 = 0$

```
eig_A = eig(A)
```

```
ans =
      -2
       6
       6
```

```
[N D]= eig (A);
M(: ,1)=N(: ,1)* sqrt (2) ;
M(: ,2)=N(: ,2)* sqrt (2) ;
M(: ,3)=N(: ,3)
```

```
M =
       1              1              0
       0              0             -1
      -1              1              0
```

```
[N D]= eig (A)
```

```
N =
     985/1393       985/1393         0
       0              0             -1
    -985/1393       985/1393         0
D =
      -2              0              0
       0              6              0
       0              0              6
```

Linearly Independant / Dependent Vectors

Aim:

To write a MATLAB program to test the vectors $(1,2,1),(2,1,0)$ and $(1,1,2)$ in $V_3(R)$ are linearly independient or not.

Main Commands:

syms - to define symbolic variables and fynctions

det - to find the determinant of the matrix

disp - to display the statements

Source Code:

```
A=[1 2 1; 2 1 1; 1 0 2]
```

```
A =
       1              2              1
       2              1              1
       1              0              2
```

```
det_A = det(A)
```

```
det_A =
      -5
```

```
if det(A)==0
    disp('The given vectors are dependent')
else
    disp('The given vectors are independent')
end
```

```
The given vectors are independent
```

Aim:

To write a MATLAB program to test the vectors $(1,4,\ 2),(-2,1,3)$ and $(4,11,\ 5)$ in $V_3(R)$ are linearly independient or not.

Main Commands:

syms - to define symbolic variables and fynctions

det - to find the determinant of the matrix

disp - to display the statements

Source Code:

```
A=[1 -2 4; 4 1 11; 2 3 5]
```

```
A =
       1             -2              4
       4              1             11
       2              3              5
```

```
det_A = det(A)
```

```
det_A =
      8
```

```
if det(A)==0
    disp('The given vectors are dependent')
else
    disp('The given vectors are independent')
end
```

```
The given vectors are independent
```

Aim:

To write a MATLAB program to test the vectors $(1,-2,-1,0),(2,-1,1,0),(2,\ 1,-1,1)$ and $(-1,-1,2,-1)$ in $V_4(R)$ are linearly independient or not.

Main Commands:

syms - to define symbolic variables and fynctions

det - to find the determinant of the matrix

disp - to display the statements

Source Code:

```
A=[1 2 2 -1; -2 -1 1 -1; -1 1 -1 2; 0 0 1 -1]
```

```
A =
      1            2            2           -1
     -2           -1            1           -1
     -1            1           -1            2
      0            0            1           -1
```

```
det_A = det(A)
```

```
det_A =
      0
```

```
if det(A)==0
    disp('The given vectors are dependent')
else
    disp('The given vectors are independent')
end
```

```
The given vectors are dependent
```

Aim:

To write a MATLAB program to test the vectors $(1,1,\ 2,4),(2,-1,-5,2),(1,-1,-4,0)$ and $(2,1,1,6)$ in $V_4(R)$ are linearly independient or not.

Main Commands:

syms - to define symbolic variables and fynctions

det - to find the determinant of the matrix disp - to display the statements

Source Code:

```
A=[1 2 1 2; 1 -1 -1 1; 2 -5 -4 1; 4 2 0 6]
```

```
A =
       1           2           1           2
       1          -1          -1           1
       2          -5          -4           1
       4           2           0           6
```

```
det_A = det(A)
```

```
det_A =
       0
```

```
if det(A)==0
    disp('The given vectors are dependent')
else
    disp('The given vectors are independent')
end
```

```
The given vectors are dependent
```

Rolle's Theorem

Aim:

To write a MATLAB program to determine $c \epsilon (a, b)$ for $f(x) = (x-1)^2(x-3)$ in $(1, 3)$ using Rolle's theorem

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

solve - to solve the equation

format rational - to create numbers in the rational form

Source Code:

```
syms x
f(x)=(x-1)^2 *(x-3)
```

f(x) = $(x-1)^2\,(x-3)$

```
I=[1,3];
a=I(1);
b=I(2);
df=diff(f,x,1);
f_a=f(a)
```

f_a = 0

```
f_b=f(b)
```

f_b = 0

```
c=solve(df==0);
c=c(a<c & c<b);
disp(c)
```

$$\frac{7}{3}$$

Aim:

To write a MATLAB program to determine $c \epsilon (a, b)$ for $f(x) = (x+2)^3(x-3)^4$ in $(-2, 3)$ using Rolle's theorem

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

solve - to solve the equation

format rational - to create numbers in the rational form

Source Code:

```
syms x
f(x)=(x+2)^3*(x-3)^4
```

f(x) = $(x+2)^3\ (x-3)^4$

```
I=[-2,3];
a=I(1);
b=I(2);
df=diff(f,x,1);
f_a=f(a)
```

f_a = 0

```
f_b=f(b)
```

f_b = 0

```
c=solve(df==0);
c=c(a<c & c<b);
disp(c)
```

$\frac{1}{7}$

Mean Value Theorem

Aim:

To write a MATLAB program to determine $c \epsilon (a, b)$ for $f(x) = x + e^x$ in $(0, 1)$ using Lagrange's mean value theorem

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

solve - to solve the equation

format rational - to create numbers in the rational form

Source Code:

```
syms x
f(x)=x+exp(x)
```

f(x) = $x + e^x$

```
I=[0,1];
a=I(1);
b=I(2);
df=diff(f,x,1);
m=(f(b)-f(a))/(b-a);
c=solve(df==m,x);
c=c(a<c & c<b);
disp(c)
```

$\log(e - 1)$

Aim:

To write a MATLAB program to determine $c \epsilon (a, b)$ for $f(x) = x(x-1)(x-2)$ in $\left(0, \frac{1}{2}\right)$ using Lagrange's mean value theorem

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

solve - to solve the equation

format rational - to create numbers in the rational form

Source Code:

```
syms x
f(x)=x*(x-1)*(x-2)
```

f(x) = $x\ (x-1)\ (x-2)$

```
I=[0,0.5];
a=I(1);
b=I(2);
df=diff(f,x,1);
m=(f(b)-f(a))/(b-a);
c=solve(df==m,x);
c=c(a<c & c<b);
disp(c)
```

$1-\frac{\sqrt{21}}{6}$

Aim:

To write a MATLAB program to determine $c\,\epsilon\,(a,b)$ for $f(x)=(x+2)^3(x-3)^4 \text{ in } (-2,3)$ using Lagrange's mean value theorem

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

solve - to solve the equation

format rational - to create numbers in the rational form

Source Code:

```
syms x
f(x)=(x+2)^3*(x-3)^4
```

f(x) = $(x+2)^3\ (x-3)^4$

```
I=[-2,3];
a=I(1);
b=I(2);
df=diff(f,x,1);
m=(f(b)-f(a))/(b-a);
c=solve(df==m,x);
c=c(a<c & c<b);
disp(c)
```

$\frac{1}{7}$

Ordinary differential Equations

Aim

Solve $\frac{d^2y}{dx^2} + 4\frac{dy}{dx} - 5y = 0$

Main Commands

syms - Create symbolic variables and functions

diff - Differentiate symbolic expression or function

dsolve - Solve system of differential equations

Source Code

```
syms x y y(x)
y(x)=dsolve(diff(y,x,2)+4*diff(y,x,1)-5*y==0)
```

y(x) = $C_2\, e^{x} + C_1\, e^{-5x}$

Aim

Solve $\frac{d^2y}{dx^2} - 5\frac{dy}{dx} + 4y = 0$

```
syms x y y(x)
y(x)=dsolve(diff(y,x,2)-5*diff(y,x,1)+4*y==0)
```

y(x) = $C_1\, e^{x} + C_2\, e^{4x}$

Aim

Solve $\frac{d^2y}{dx^2} - 2\frac{dy}{dx} + 2y = 0$

```
syms x y y(x)
y(x)=dsolve(diff(y,x,2)-2*diff(y,x,1)+2*y==0)
```

y(x) = $C_1\, e^{x}\cos(x) - C_2\, e^{x}\sin(x)$

Aim

Solve $\frac{d^2y}{dx^2} - 4\frac{dy}{dx} + 4y = 0$

```
syms x y y(x)
y(x)=dsolve(diff(y,x,2)-4*diff(y,x,1)+4*y==0)
```

y(x) = $C_1\, e^{2x} + C_2\, x\, e^{2x}$

Aim

Solve $\frac{d^2y}{dx^2} - 5\frac{dy}{dx} + 4y = e^{5x}$

Main Commands

syms - Create symbolic variables and functions

diff - Differentiate symbolic expression or function

dsolve - Solve system of differential equations

Source Code

```
syms x y y(x)
y(x)=dsolve(diff(y,x,2)-5*diff(y,x,1)+4*y==exp(5*x))
```

```
y(x) =
```

$$\frac{e^{5x}}{4} + C_1 e^x + C_2 e^{4x}$$

Maxima and Minima

Aim:

To write a MATLAB program to find the maxima and minima of the function $f(x) = 3x^4 - 2x^3 - 6x^2 + 6x + 1$.

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

solve - to solve the equation

Source Code:

```
syms x
f(x)=3*x^4-2*x^3-6*x^2+6*x+1
```

f(x) = $3\,x^4 - 2\,x^3 - 6\,x^2 + 6\,x + 1$

```
r=solve(diff(f));
r1=r(1,1)
```

r1 = -1

```
second_derivative=subs(diff(f,2),r(1,1))
```

second_derivative(x) = 36

```
if second_derivative>0
    disp('The above point is minimum point')
else if second_derivative<0
    disp('The above point is maximum point')
else
    disp('Further investigation required')
end
end
```

The above point is minimum point

```
value = subs(f,r(1,1))
```

value(x) = -6

```
r2=r(2,1)
```

r2 =

$$\frac{1}{2}$$

```
second_derivative=subs(diff(f,2),r(2,1))
```

second_derivative(x) = -9

```
if second_derivative>0
    disp('The above point is minimum point')
else if second_derivative<0
    disp('The above point is maximum point')
else
    disp('Further investigation required')
end
end
```

```
The above point is maximum point
```

```
value = subs(f,r(2,1))
```

value(x) =

$$\frac{39}{16}$$

```
r3=r(3,1)
```

r3 = 1

```
second_derivative=subs(diff(f,2),r(3,1))
```

second_derivative(x) = 12

```
if second_derivative>0
    disp('The above point is minimum point')
else if second_derivative<0
    disp('The above point is maximum point')
else
    disp('Further investigation required')
end
end
```

```
The above point is minimum point
```

```
value = subs(f,r(3,1))
```

value(x) = 2

Aim:

To write a MATLAB program to find the maxima and minima of the function $f(x) = 3x^4 - 2x^3 - 6x^2 + 6x + 1$.

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

solve - to solve the equation

Source Code:

```
syms x
f(x)=4*x^3-16*x^2+12*x
```

f(x) = $4\,x^3-16\,x^2+12\,x$

```
r=solve(diff(f));
r1=r(1,1)
```

r1 =

$$\frac{4}{3}-\frac{\sqrt{7}}{3}$$

```
second_derivative=subs(diff(f,2),r(1,1))
```

second_derivative(x) = $-8\,\sqrt{7}$

```
if second_derivative > 0
    disp('The above point is minimum point')
else if second_derivative < 0
    disp('The above point is maximum point')
else
    disp('Further investigation required')
end
end
```

The above point is maximum point

```
value = subs(f,r(1,1))
```

value(x) =

$$16-16\,\left(\frac{\sqrt{7}}{3}-\frac{4}{3}\right)^2-4\,\left(\frac{\sqrt{7}}{3}-\frac{4}{3}\right)^3-4\,\sqrt{7}$$

```
r2=r(2,1)
```

r2 =

$$\frac{\sqrt{7}}{3}+\frac{4}{3}$$

```
second_derivative=subs(diff(f,2),r(2,1))
```

second_derivative(x) = $8\,\sqrt{7}$

```
if second_derivative > 0
    disp('The above point is minimum point')
else if second_derivative < 0
    disp('The above point is maximum point')
else
    disp('Further investigation required')
end
end
```

The above point is minimum point

```
value = subs(f,r(2,1))
```

value(x) =

$$4\ \sqrt{7} - 16\ \left(\frac{\sqrt{7}}{3} + \frac{4}{3}\right)^2 + 4\ \left(\frac{\sqrt{7}}{3} + \frac{4}{3}\right)^3 + 16$$

Limit of a function

Aim:

To write a MATLAB program to find $\lim\limits_{\substack{x \longrightarrow 1 \\ y \longrightarrow 2}} (x^2 + y^2)$

Main Commands:

syms - to define symbolic variables and fynctions

limit - to find the limit of the function

Source Code:

```
syms x y f g
f=x^2+y^2
```

f = $x^2 + y^2$

```
%limit_value=limit(limit(f,x,1),y,2)
```

Aim:

To write a MATLAB program to find $\lim\limits_{\substack{x \longrightarrow 1 \\ y \longrightarrow 2}} \left(\frac{2x^2 y}{x^2 + y^2 + 1}\right)$

Main Commands:

syms - to define symbolic variables and fynctions

limit - to find the limit of the function

Source Code:

```
syms x y
f=(2*x^2*y)/(x^2+y^2+1)
```

f =

$$\frac{2\,x^2\,y}{x^2 + y^2 + 1}$$

```
%limit_value=limit(limit(f,x,1),y,2)
```

Derivatives of a function

Aim:

To write a MATLAB program to find the first and second order derivatives of the function $z = x^3 + y^3 - 3axy$.

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

Source Code:

```
syms x y a
z=x^3+y^3-3*a*x*y
```

z = $x^3 - 3\,a\,x\,y + y^3$

```
disp('First derivative with respect to x')
```

```
First derivative with respect to x
```

```
diff(z,x,1)
```

ans = $3\,x^2 - 3\,a\,y$

```
disp('First derivative with respect to y')
```

```
First derivative with respect to y
```

```
diff(z,y,1)
```

ans = $3\,y^2 - 3\,a\,x$

```
disp('Second derivative with respect to x')
```

```
Second derivative with respect to x
```

```
diff(z,x,2)
```

ans = $6\,x$

```
disp('Second derivative with respect to y')
```

```
Second derivative with respect to y
```

```
diff(z,y,2)
```

ans = $6\,y$

```
disp('Second derivative with respect to x & y')
```

```
Second derivative with respect to x & y
```

```
diff(diff(z,x,1),y,1)
```

ans = $-3\,a$

Aim:

To write a MATLAB program to find the first and second order derivatives of the function $z = x^4 + y^4 - 2x^2 - 2y^2 + 4xy.$

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

Source Code:

```
syms x y a
z=x^4+y^4-2*x^2-2*y^2+4*x*y
```

z = $x^4 - 2\,x^2 + 4\,x\,y + y^4 - 2\,y^2$

```
disp('First derivative with respect to x')
```

First derivative with respect to x

```
diff(z,x,1)
```

ans = $4\,x^3 - 4\,x + 4\,y$

```
disp('First derivative with respect to y')
```

First derivative with respect to y

```
diff(z,y,1)
```

ans = $4\,y^3 - 4\,y + 4\,x$

```
disp('Second derivative with respect to x')
```

Second derivative with respect to x

```
diff(z,x,2)
```

ans = $12\,x^2 - 4$

```
disp('Second derivative with respect to y')
```

Second derivative with respect to y

```
diff(z,y,2)
```

ans = $12\,y^2 - 4$

```
disp('Second derivative with respect to x & y')
```

Second derivative with respect to x & y

```
diff(diff(z,x,1),y,1)
```

ans = 4

Aim:

To write a MATLAB program to find the first and second order derivatives of the function $z = x^3 + 3xy^2 + 15x^2 - 15y^2 + 72x.$

Main Commands:

syms - to define symbolic variables and fynctions

diff - to find the derivative of the function

Source Code:

```
syms x y a
z=x^3+3*x*y^2+15*x^2-15*y^2+72*x
```

z = $x^3 + 15\,x^2 + 3\,x\,y^2 + 72\,x - 15\,y^2$

```
disp('First derivative with respect to x')
```

```
First derivative with respect to x
```

```
diff(z,x,1)
```

ans = $3\,x^2 + 30\,x + 3\,y^2 + 72$

```
disp('First derivative with respect to y')
```

```
First derivative with respect to y
```

```
diff(z,y,1)
```

ans = $6\,x\,y - 30\,y$

```
disp('Second derivative with respect to x')
```

```
Second derivative with respect to x
```

```
diff(z,x,2)
```

ans = $6\,x + 30$

```
disp('Second derivative with respect to y')
```

```
Second derivative with respect to y
```

```
diff(z,y,2)
```

ans = $6\,x - 30$

```
disp('Second derivative with respect to x & y')
```

```
Second derivative with respect to x & y
```

```
diff(diff(z,x,1),y,1)
```

```
ans = 6 y
```

Beta and Gamma functions

Aim:

To write a MATLAB program to find $\Gamma(1)$

Main Commands:

syms - to define symbolic variables and fynctions

gamma - to gamma function values

Source Code:

```
syms y
y = gamma(1)
```

```
y = 
      1
```

Aim:

To write a MATLAB program to find $\Gamma(10)$

Main Commands:

syms - to define symbolic variables and fynctions

gamma - to gamma function values

Source Code:

```
syms y
y = gamma(10)
```

```
y = 
  362880
```

Aim:

To write a MATLAB program to find $\Gamma(0.5)$

Main Commands:

syms - to define symbolic variables and fynctions

gamma - to gamma function values

Source Code:

```
syms y
y = gamma(0.5)
```

```
y = 
     296/167
```

Aim:

To write a MATLAB program to find $\beta(1,5)$

Main Commands:

syms - to define symbolic variables and fynctions

gamma - to gamma function values

Source Code:

```
syms x
x=beta(1, 5)
```

```
x =
      1/5
```

Aim:

To write a MATLAB program to find $\beta(1,5)$

Main Commands:

syms - to define symbolic variables and fynctions

gamma - to gamma function values

Source Code:

```
syms x
x=beta(3, sqrt(2))
```

```
x =
    204/1189
```

Aim:

To write a MATLAB program to find $\beta(1,5)$

Main Commands:

syms - to define symbolic variables and fynctions

gamma - to gamma function values

Source Code:

```
syms x
x=beta(pi, exp(1))
```

```
x =
    125/3299
```

Aim:

To write a MATLAB program to find $\beta(1,5)$

Main Commands:

syms - to define symbolic variables and fynctions

gamma - to gamma function values

Source Code:

```
syms x
x=beta(0, 1)
```

```
x =
       1/0
```