Practical 1

A = [1 0 0 1 -1 ;

0 2 3 5 0;

-1 0 0 0 1;

6 8 1 2 -2;

1 1 1 1 1];

arr = eig(A);

sortedArr = sort(abs(arr));

disp(sortedArr(1));

disp(sortedArr(2));

Output:

Assignment\_1

0.4611

1.4643

Practical 2

To show the consistency and inconsistency of equations

Script File

A = input('Enter Matrix A : \n');

B = input('Enter Matrix B : \n');

Aug = [A B];

if(rank(A) == rank(B))

disp('The equation is consistent');

else

disp('The system of Equations is inconsistent');

end

Output:

Practical\_2

Enter Matrix A :

[1 0 ; 0 1]

Enter Matrix B :

[0 ; 0]

The system of Equations is inconsistent

Practical 3

To solve system of equations if consistent

Script:

A = input('Enter Matrix A : \n');

B = input('Enter Matrix B : \n');

Aug = [A B];

if(rank(A) == rank(Aug))

Ans = linsolve(A, B);

disp(Ans(1))

disp(Ans(2))

disp(Ans(3))

else

disp('The Equation is inconsistent')

end

Output:

Assignment\_2

Enter Matrix A :

[1 2 1 ;

2 3 5 ;

7 1 2]

Enter Matrix B :

[5 ; 7 ; 0 ]

-0.3636

2.7273

-0.0909

Practical 4:

Write a program to solve the following equations using ode23 and ode45 functions, and hence compare the values of y(3)

Dy/dx = y + x y(0) = 1

Script file:

function val = differentialEquation(y,t)

val = y + t;

on command window:-

[x, y1] = ode23('differentialEquation', [-5 5], 1);

>> plot(x, y1);

>> ans = y1(3)

ans =

0.4975

[x, y2] = ode45('differentialEquation', [-5 5], 1);

>> plot(x, y2)

ans = y2(3)

ans =

0.8986

Program 5:

Write a code to solve differential D2y + 4y = secx using method of variation of parameters

Script file:

syms t

g(t) = sec(t);

Yc = dsolve('D2y + 4\*y = 0');

y1 = cos(2\*t);

y2 = sin(2\*t);

uDot = diff(y1);

vDot = diff(y2);

w = [y1 y2 ; uDot vDot];

W = det(w);

w1 = [0 y2 ; g vDot];

W1 = det(w1);

w2 = [y1 0 ; uDot g];

W2 = det(w2);

uDot = W1 / W;

vDot = W2 / W;

u = int(uDot);

v = int(vDot);

Yp = u\*y1 + v\*y2;

Y(t) = Yc + Yp;

Output:

Practical\_5

>> Y(t)

ans =

cos(2\*t)\*cos(t) + C3\*cos(2\*t) + C4\*sin(2\*t) - sin(2\*t)\*(atanh(sin(t))/2 - sin(t))

Practical 6

Graphically compare the function sin(x) and it’s taylor series expansion upto degree of 10 in the neighbourhood of 1.

Script file:

syms g x;

g = taylor(sin(x), x, 1, 'order', 11);

h = ezplot(x,g);

set(h, 'color', 'r');

grid;

hold on;

y = sin(x);

k = ezplot(x, y);

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

title('Taylor Series');

hold off;

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

Practical 7:

To draw a tangent at point on a given curve y = 1 + x2, at the point (2,5) and also find the radius of curvature at that point.

Script File:

syms x t;

f(x) = 1 + x^2;

d(x) = diff(f(x));

g(x) = 5 + (x-2)\*d(2);

h = ezplot(f(x));

set(h, 'color', 'r');

grid;

hold on;

k = ezplot(g(x));

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

roc(x) =sqrt((1 + d(x)^2))^3;

r(x) = roc(x) / diff(d(x));

disp(r(2));

output:-

radius of curvature = 35.0464

Practical 8

Script file

syms x t;

f(x) = x^3 - 3\*x^2 + 3\*x;

g(x) = x^2;

plot1 = ezplot(f(x));

set(plot1, 'color', 'r');

grid;

hold on;

plot2 = ezplot(g(x));

set(plot2, 'color', 'b');

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

xlabel('X - Axis'); ylabel('Y - Axis');

title('Area Enclosed');

hold off;

H(x) = int(f(x) - g(x));

area1 = abs(H(1) - H(0));

area2 = abs(H(3) - H(1));

area = area1 + area2;

disp(area);

output

37/12

Practical 9

Script file:-

syms x y;

[x, y] = meshgrid(-2:0.1:2, -2:0.1:2);

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

figure(1)

mesh(x, y, f);

xlabel('X'); ylabel('Y');

grid;

figure(2)

contour(x, y, f);

xlabel('X'); ylabel('Y');

%finding the maximum value of f(x,y)

fmax = max(max(f));

kmax = find(f == fmax);

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

plot(x(kmax), y(kmax), '\*');

text(x(kmax), y(kmax), 'Maximum');

%finding the minimum value of f(x,y)

fmin = min(min(f));

%locating minimum value

kmin = find(f == fmin);

pos1 = [x(kmin), y(kmin)];

%Plotting the Minimum Values

plot(x(kmin), y(kmin), '\*');

text(x(kmin), y(kmin), 'Minimum');

hold off;

Output:-

 

Fmax = 0.0920

fmin = -0.0920

Practical 10

Write characteristic polynomial equations of a matrix by solving for P(lambda) and fitting it to n+1 points

Script:

function[co] = charPoly(A)

[m,n] = size(A);

if(m ~= n)

disp('Not agreable Matrix');

co = [];

return

end

for i=1 : (n+1)

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

y(i) = det(A - x(i) \* eye(n) );

end

co = polyfit(x, y, n);

function[a] = createFunction(A)

syms x;

a = 0;

for i =1 : length(A)

a = a + A(i) \* x^ (length(A)+1-i);

end

Output

ans = x^3 + 15x^2 + 18x – 3.27 \* 10-15