USN :- 1BM17TE060 III SEM .. B SECTION.

MATHS ASSIGNMENT

1] Rank of a matrix-

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

A = 
    1   2   3   4
    5   6   7   8

rank(A) = 2
```

2] LU - Decomposition method -

```
>> A = [5 4 8 ; 1 6 8 ; 2 4 9]
>> [L1, U] = lu(A)

L1 = 1.0000    0         0
     0.2000    1.0000    0
     0.4000    0.4615    1.000

U =  5.0000    4.0000    8.0000
     0         5.2000    6.4000
     0         0         2.8462

>> [L2, U, P] = lu(A)

L2 = 1.0000    0         0
     0.2000    1.0000    0
     0.4000    0.4615    1.0000

U =  5.0000    4.0000    8.0000
     0         5.2000    6.4000
     0         0         2.8462

P =  1    0    0
     0    1    0
     0    0    1
```

```
>> x = U/(L\B)

x =  2
    -1
     3
```

3. Gauss - Seidel method.

```
GaussSeidel.m
function y = GaussSeidel(a, b, x, NoOfIterations)
D = diag(a);
a = a - diag(D);
D = 1./D;
n = length(x);
x = x(:);
y = zeros(n, NoOfIterations);
for j = 1: NoOfIterations
    for k = 1: n
        x(k) = (b(k) - a(k,:)*x)*D(k);
    end
    y(:, j) = x;
end.
```

```
>> A = [27  6  -1;  6  15  2;  1  1  54]

A =
    27    6   -1
     6   15    2
     1    1   54

>> B = [110;  85;  72]

B =  110
      85
      72
```

```
x0 = [0;  0;  0]
x0 =
      0
      0
      0
>> GaussSeidel(A, B, x0, 4)
ans =   3.1481    2.4321    2.4257    2.4255
        3.5407    3.5721    3.5729    3.5730
        1.9132    1.9259    1.9259    1.9259
```

Gauss - elimination method -

```
function x = gaussele(A, b);
  [m, n] = size(A);
   if, m ≅ n.
    end
    nb = n+1;
    Avg = [A, b];
  % forward elimination
    for k = 1 : n = 1
     for i = k+1:n
    factor = Avg(i, k) / Avg(k, k);
   (i, k:n, b) = Avg(i, k: n, b) - factor * Avg(k, k: nb);
      end
   end
 % backward Subtitution
   x = zeroes(n: 1);
  x(n) = Avg(n, nb) / Avg(n, n);
   for i = n-1 : 1
  x(i) = (Avg(i, nb) - Avg(i, i+1:n) * x(i+1: n) / Av
   end
```

```
x(i);
end.
output:
>> A = [5 1 1 1; 1 7 1 1; 1 1 6 1; 1 1 1 4]

A =  5  1  1  1
     1  7  1  1
     1  1  6  1
     1  1  1  4

>> B = [4; 12; -5; -6]

B =  4
     12
     -5
     -6

>> Gauss ele (A, b)
   =  1
      2
     -1
     -2
```

Eigen values and Eigen vectors.

```
>> A = [-2 2 -3; 2 1 -6; -1 -2 0]

A =
   -2   2   -3
    2   1   -6
   -1  -2    0

>> [v, W] = eig (A)

V =
   -0.9526    0.4082   -0.0230
    0.2722    0.8165    0.8353
   -0.1361   -0.4082    0.5492
```

-3.0000

0 0 0

0 5.0000 0

0 0 -3.0000

Rayleigh Power method

```
A = [6 -2 2; -2 3 -1; 2 -1 3];
i = 0;
x = [1; 1; 1];
while (i< 5)
X = (A*x)/max((A*x));
disp (x);
i = i+1;
end
disp(x);
t = X.'*x;
display (t);
u = ((A*x).')*x;
eivalue = u/t;
disp(eivalue);
```

Ans:

1

1

1

1

1

1

1

1

1

Scanned by CamScanner

1

1

t =

1.6667

6.0000

Newton Raphson -

```
Syms x;
fun = input("enter the function as a variable of x");
f = inline(fun);
z = diff(f(x));
f1 = inline(z);
x0 = input('Enter the initial value of interval');
x = x0;
for i=0; inf
 y = x;
 x = y-(f(x)/f1(x));
 if x==y
    break
   end
end
fPrintf('The total number of iterations are:');
;
x
```

Answer -

Enter the function as a variable of x exp(x) * ~~3~~ - 2

enter the initial value of interval 0.5

ans =

The total number of iterations are

i = 4

0.8532

Newton Raphson forward interpolation -

```
x = [8  6  8  5  15  18];
fx = [20  20  12  7  6  6];
dt = zeros (6, 10);
for i = 1:6 dt(i, 1) = x(i);
dt (i, 2) = fx(i); % calling function
end
n = 5,
for j = 3:10
for i = 1:n
dt(i, j) = dt(i+1, j-1) - dt(i, j-1)
end
n = n-1;
end
h = x(2) - x(1)
xp = 1.5;
for i = 1:5
q = (xp - x(i)) / h;
if (q>0 && q<1)
P = q
end
end
P
l = xp - (P* h)
for i = 1:5
if (i = = x(i))
r = i;
end % calculating different value of y
```

```
f0 = fx(r);
f01 = dt(r, 3);
f02 = dt(r, (3+1));
f03 = dt((r), (3+2));
f04 = dt((r), (3+3));
fp = (f0 + ((p * f01) + (p* (p-1) * f02) / (2)) + ((p * (p-1) * (p-2) * f03
     / (6)) + ((p * (p-1) * (p-2) * (p-3) * f04) / (24)) *
```
~~(((p * (p-1) (p-2~~

Answer

dt =

8	20	0	-8	11	-10	6	0	0	0
6	20	-8	3	1	-4	0	0	0	0
8	12	-5	4	-3	0	0	0	0	0
5	7	-1	1	0	0	0	0	0	0
15	6	0	0	0	0	0	0	0	0
18	6	0	0	0	0	0	0	0	0

Newton Raphson Backward interpolation -

```
x = [8   8   25   16   39   49];
fx = [14.621   11.843   9.870   8.418   7.305   6.413]
dt = zeros(6, 7);
for i = 1:6
dt(i, 1) = x(i);
dt (i, 2) = fx(i);
end
n = 5
for j = 3:7
for i = 1:n
```

```
t(i,j) = dt(i+1, j-1) - dt(i, j-1)
end
n = n-1;
end
h = x(2) - x(1)
xp = 27;
for i = 1:6
q = (xp - x(i)) /h;
if (q>0 && q<1)
p = q;
end
end
p
l = xp - (p * h)
for i = 1:6
if (l = = x(i))
r = i
end
end
f0 = fx(r);
f01 = dt((r-1), 3);
f02 = dt((r-2), (3+1));
f03 = dt((r-3), (3+2));
f04 = dt((r-4), (3+3));
fp = (f(0) + ((p * f01) + (p * (p+1) * f02)/(2)) + ((p * (p+1) * (p+
      * f03)/(6))
```

Scanned by CamScanner

1

Answer

dt = 8.0000 14.6210 -2.7780 0.8050 -0.2840
0.1020 -0.0380

8.0000 11.8430 -1.9730 0.5210 -0.1820
0.0640 0

25.0000 9.8700 -1.4520 0.3390 -0.1180
0 0

16.0000 8.4180 -1.1130 0.2210 0
0 0

39.0000 7.3050 -0.8920 0 0
0 0

49.0000 6.4130 0 0 0
0 0