Index

Experiment - 1

Write a program to find the all the solutions of the equation $dX = AX$ using Eigenvalue-Eigen vector method.

CODE

```
function[sols]=linear_DE_SYSTEM_SOLVER(~)
disp("--------------------")
disp("Experiment - 1")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")
syms t lambda
%A=[3 4 5;0 4 1;2 9 1];
A = [3 1;-2 1]
n = length(A);
[V,D] = eig(A);
eigenvalues = diag(D)
consts = reshape(sym('c%d',[1 n]),n,1);
unique_eigenvalues = unique(eigenvalues);
mults = histc(eigenvalues,unique_eigenvalues);
sols = sym('x%d',[1 n]);
if length(unique_eigenvalues) ~= length(eigenvalues)
%For repeating eigenvalues
i = 1;
ch_mat = A-lambda*eye(n);
%variable precision arithmetic
V = vpa(V);
while i<=n
[pos] = find(unique_eigenvalues == eigenvalues(i));
if mults(pos)>1
e_vector = V(:,i);
a_mat = subs(ch_mat,eigenvalues(i));
for j=1:mults(pos)
V(:,i) = V(:,i).*(t^(j-1));
P = inv(a_mat^(j-1))*e_vector;
V(:,i) = V(:,i)+P;
i = i+1;
```

```
    end
    else
    i = i+1;
    end
    end
    end
    for i=1:n
    sols(i) = round((V(i,:).*exp(eigenvalues'*i))*consts);
    end
```

OUTPUT

```
>> linear_DE_SYSTEM_SOLVER
------------------
Experiment - 1
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
------------------

A =

     3     4     5
     0     4     1
     2     9     1


eigenvalues =

   -2.1517
    3.0957
    7.0560


ans =

[- (6039247423656819*c2)/281474976710656 - (8449779702245897*c3)/8796093022208 + (1341151961004509*c1)/18014398509481984, - (8053174460133677*c3)/34359738368
fx >>


(3062972863788853*c2)/35184372088832 + (7686692808048327*c1)/4611686018427387904, - (5498785589460841*c1)/4611686018427387904 - (1913156019224553*c2)/10995116277776

- (6968309919990955*c3)/8388608]
```

Experiment - 2

Write a program to solve the initial value problem using Eigen values and Eigen vector method. $dX = AX, X(B) = (C)$

CODE

```
function[sols,vals]=Linear_DE_SYSTEM_SOLVERS_2(A,B,C)
disp("--------------------------")
disp("Name -> M K Lino Roshaan ")
disp("Roll No -> 2K22/MC/87 ")
disp("Experiment - 2 ")
disp("--------------------------")

syms t lambda
A = [3 1 -1;1 3 -1;3 3 -1];
B = pi/2;
C = [0,1,0];
n = length(A);
[V,D] = eig(A);
eigenvalues = diag(D);
const = reshape(sym('c%d',[1 n]),n,1);
unique_eigenvalues = unique(eigenvalues);
mults = histcounts(eigenvalues,unique_eigenvalues);
sols = sym('x%d',[1 n]);
if length(unique_eigenvalues) ~= length(eigenvalues)
%For repeating eigenvalues
```

```
if length(unique_eigenvalues) ~= length(eigenvalues)
%For repeating eigenvalues
i = 1;
ch_mat = A-lambda*eye(n);
%variable precision arithmetic
V = vpa(V);
while i<=n
[pos] = find(unique_eigenvalues == eigenvalues(i));
if mults(pos)>1
e_vector = V(:,i);
a_mat = subs(ch_mat,eigenvalues(i));
for j=1:mults(pos)
V(:,i) = V(:,i).*(t^(j-1));
P = inv(a_mat^(j-1))*e_vector;
V(:,i) = V(:,i)+P;
i = i+1;
end
else
i = i+1;
```

```
    end
    end
    end
    for i=1:n
    sols(i)=(V(i,:)*exp(eigenvalues.*t))*const(i);
    end
    vals=vpasolve(subs(sols,t,B)==C);
    constnames=fieldnames(vals);
    % The final solution is
    for i=1:n
    sols=subs(sols,const(i),vals.(constnames{i}));
    end
```

OUTPUT

```
>> Linear_DE_SYSTEM_SOLVERS_2
--------------------------
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
Experiment - 2
--------------------------

ans =

[0, 0.079013825539440955995903921963 9*exp(2*t) + 0.015908134992895448728021472198971*11^(1/2)*exp(t)

- 0.012499248922989281143445442442049*14^(1/2)*exp(2*t), 0]
```

Experiment - 3

Write a program to find the eigenvalues and eigenvectors of the Sturm-Liouville problem: $d\text{^}2\,X + \lambda X = 0, dX(0) = 0, dX(L) = 0$

CODE

```
function [e_value, e_function] = sturm_liouville(L)
disp("--------------------")
disp("Experiment - 3")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")
syms y(x)
syms lambda n
%Equation is, y" + (lambda)y = 0
L = 1;
%L = pi/4;
%sprintf stores the string in buffer
sprintf('Solving for various conditions...')
sprintf('When lambda > 0')
assume(lambda > 0);
%dsolve computes symbolic solutions to ordinary differential equations
solution = dsolve(diff(y, 2) + lambda * y == 0);
e_function = solution
diff_sol = diff(solution, x);
vals = solve(subs(diff_sol, x,0) == 0, subs(diff_sol, x,L) == 0);
non_zero = vals;
sprintf('Non-zero values in the solution')
vals;
%Eigen Value for this solution
e_value = ((n * pi) / L) .^ 2;
sprintf('When lambda = 0')
try
solution = dsolve(subs(diff(y, 2) + lambda * y == 0), lambda, 0);
catch
sprintf('No non-trivial solution')
end
sprintf('When lambda < 0')
assume(lambda < 0);
solution = dsolve(diff(y, 2) + lambda * y == 0);
diff_sol = diff(solution, x);
%No Explicit non trivial solution exists
vals = solve(subs(diff_sol, 0) == 0, subs(diff_sol, L) == 0);
```

OUTPUT

```
>> sturm_liouville
-------------------
Experiment - 3
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
-------------------

ans =

    'Solving for various conditions...'


ans =

    'When lambda > 0'


e_function =

- sin(lambda^(1/2)*x)*C2 + cos(lambda^(1/2)*x)*C1


ans =

    'Non-zero values in the solution'


ans =

    'When lambda = 0'


ans =

    'No non-trivial solution'
ans =

    'When lambda < 0'

Warning: Unable to find explicit solution. For options, see help.
> In sym/solve (line 317)
In sturm_liouville (line 37)

ans =

n^2*pi^2

>>
```

Experiment - 4

Write a program to solve Lagrange's equation using Lagrange's method.

CODE (a)

```
function [] = Lagrange() %Function Name
disp("--------------------")
disp("Experiment - 4(a)")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")
syms x y z p q c1 c2
lhs=((y^2)*z/x)*p+(x*z)*q; %LHS Side
rhs=(y^2); %RHS Side
C=coeffs(lhs,[q p]); %Coefficients of equation
P=C(1); %Extracting value of P
Q=C(2); %Extracting value of Q
R=rhs; %Extracting value of R
%int() is an integrating function
%1st and 2nd expression
P=P*x/z;
Q=Q*x/z;
U = int(P,y) == int(Q,x) + c1; %Solving for U
S = int(P,y) - int(Q,x);
U = simplify(S)
%1st and last expression
P=y^2*x/P;
R=y^2*z/R;
V = int(P,x) == int(R,z) + c2; %Solving for V
T = int(P,x) - int(R,z);
V = simplify(T)
disp("Solution is given as f(U,V): ")
disp('f(')
disp(U)
disp(', ')
disp(V)
disp(') == 0')
```

OUTPUT

```
>> Lagrange
-------------------
Experiment - 4(a)
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
-------------------

U =

- x^3/3 + y^3/3


V =

- z^2/2 + x^2/2

Solution is given as f(U,V):
f(
- x^3/3 + y^3/3

,
- z^2/2 + x^2/2

) == 0
fx >>
```

CODE (b)

```
function [] = Lagrange_2()
disp("--------------------")
disp("Experiment - 4(b)")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("-------------------")
syms x y z p q
lhs= (y*z)*p + (z*x)*q; %LHS Side
rhs= x*y; %RHS Side
C=coeffs(lhs,[q p]); %Coefficients of equation
P=C(1); %Extracting value of P
Q=C(2); %Extracting value of Q
R=rhs; %Extracting value of R


%1st and 2nd expression
P1=P/z;
Q1=Q/z;
U = int(Q1,x) - int(P1,y); %Solving for U
U = U*2;
disp("U = ")
disp(U)
```

```
23
24    %2nd and 3rd expression
25    Q2=Q/x;
26    R2=R/x;
27    V = int(R2,y) - int(Q2,z); %Solving for V
28    V = V*2;
29    disp("V = ")
30    disp(V)
31
32
33    disp("Solution is given as f(U,V): ")
34    disp('f(')
35    disp(U)
36    disp(', ')
37    disp(V)
38    disp(') == 0')
39
40    end
```

OUTPUT

```
>> Lagrange_2
--------------------
Experiment - 4(b)
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
--------------------
U =
- y^2 + x^2

V =
- z^2 + y^2

Solution is given as f(U,V):
f(
- y^2 + x^2


,
- z^2 + y^2

) == 0
fx >>
```

Experiments - 5

Write a program to solve non-linear PDE using Charpit's Method: f(x, y, z, p, q) = 0

CODE (a)

```
disp("--------------------")
disp("Experiment - 5(a)")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")

% Equations
% 2zx - px^2 - 2qxy + pq = 0

syms x y z p q a t c1

fun = 2*z*x - p*(x^2) - 2*q*x*y + p*q;
f_x = diff(fun, x);
f_y = diff(fun, y);
f_z = diff(fun, z);
f_p = diff(fun, p);
f_q = diff(fun, q);
b_1 = f_x + p*f_z;
b_2 = f_y + q*f_z;
b_3 = -p*f_p - q*f_q;
b_4 = - f_p;
b_5 = - f_q;

% since b2 = dq = 0
% Let q = a

eqn = subs(fun, q, a);
p = solve(eqn, p);
q = a;

p = simplify(p/(z-a*y));

% let z - a*y = t
% => dz - a*dy = dt

expr = int(1/t,t) - int(p,x);

expr = subs(expr,t,z-a*y) - log(c1);

disp("p = ")
disp(p)
disp("q = ")
disp(q)
disp("Expression = ")
disp(expr)

```

OUTPUT

```
>>
>> charpit_method
--------------------
Experiment - 5(a)
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
--------------------
p =
-(2*x)/(a - x^2)

q =
a

Expression =
log(z - a*y) - log(- a + x^2) - log(c1)

fx >>
```

CODE (b)

```
disp("--------------------")
disp("Experiment - 5(b)")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")

% Equations
% z^2 - pxyz == 0

syms x y z p q a c1

fun = z^2 - p*x*y*z ;

f_x = diff(fun, x);
f_y = diff(fun, y);
f_z = diff(fun, z);
f_p = diff(fun, p);
f_q = diff(fun, q);

b_1 = f_x + p*f_z;
b_2 = f_y + q*f_z;
b_3 = -p*f_p - q*f_q;
b_4 = - f_p;
b_5 = - f_q;

% Since dy/-df_p == dy/0 => y = constant and dy = 0

p = solve(fun, p)

z = int(p,x) + 0 + c1 % because q*dy = 0 as dy = 0

```

OUTPUT

```
>> charpit_method_2
-------------------
Experiment - 5(b)
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
-------------------

p =

z/(x*y)


z =

c1 + (z*log(x))/y

fx >>
```

Experiment - 6

Write a program to solve $(aD^2 + bD'D + cD'^2)z = f(x, y)$

CODE (a)

```
function [sols] = linear_PDE_const_coeffs(eq)
disp("--------------------")
disp("Experiment - 6(a)")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")
syms D d m x y

disp("Give Equation is ->")

eq = D^3 - 6*(D^2)*d + 11*D*(d^2) - 6*d^3

ch_eq = subs(eq, D, m);
ch_eq = subs(ch_eq, d, 1);

vals = double(solve(ch_eq, m));
unique_vals = unique(vals, 'sorted');


n = length(vals);

f = sym('f%d',[1 n]);
k = sym('x%d',[1 n]);


n1 = length(unique_vals);

mults = unique(vals);
for i = 1:length(mults)
mults(i) = length(find(vals == mults(i)));
end

```

```
i = 1;

while i<=n1
    if mults(i) > 1
        current_m = unique_vals(i);
        for j = 1:mults(i)
            k(i) = f(i)*(y+(current_m*x))*(x^(j-1));
            i = i+1;
        end
    else
        k(i) = f(i)*(y+(unique_vals(i)*x));
        i = i + 1;
    end
end

sols = k;

disp("Solutions Are ->")

end
```

OUTPUT

```
>> linear_PDE_const_coeffs
--------------------
Experiment - 6(a)
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
--------------------
Give Equation is ->

eq =

D^3 - 6*D^2*d + 11*D*d^2 - 6*d^3

Solutions Are ->

ans =

[f1*(y + x), f2*(y + 2*x), f3*(y + 3*x)]

fx >>
```

CODE (a)

```
function [sols] = linear_PDE_const_coeffs(eq)
disp("--------------------")
disp("Experiment - 6(b)")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")
syms D d m x y

disp("Give Equation is ->")

eq = D^4 - d^4

ch_eq = subs(eq, D, m);
ch_eq = subs(ch_eq, d, 1);

vals = double(solve(ch_eq, m));
unique_vals = unique(vals, 'sorted');


n = length(vals);

f = sym('f%d',[1 n]);
k = sym('x%d',[1 n]);


n1 = length(unique_vals);

mults = unique(vals);
for i = 1:length(mults)
mults(i) = length(find(vals == mults(i)));
end

```

```
i = 1;

while i<=n1
    if mults(i) > 1
        current_m = unique_vals(i);
        for j = 1:mults(i)
            k(i) = f(i)*(y+(current_m*x))*(x^(j-1));
            i = i+1;
        end
    else
        k(i) = f(i)*(y+(unique_vals(i)*x));
        i = i + 1;
    end
end

sols = k;

disp("Solutions Are ->")

end
```

OUTPUT

```
>> linear_PDE_const_coeffs
--------------------
Experiment - 6(b)
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
--------------------
Give Equation is ->

eq =

D^4 - d^4

Solutions Are ->

ans =

[-f1*(- y + x*1i), f2*(y + x), f3*(y + x*1i), -f4*(- y + x)]

fx >>
```

Experiment - 7

Write a program to solve heat equation subject to boundary conditions and the initial conditions

CODE

```
disp("--------------------")
disp("Experiment - 7")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")
syms x T B Bn C C1 C2 lambda k t
%INCLUDE FUNCTION
%e_value is the p term which is equal to root(lambda)
[e_value, e_function] = sturm_liouville()
e_value = sqrt(e_value);
%SOLUTION
sol1 = subs(e_function,lambda,e_value)
%we can also use dsolve for this
sol = int(1/T,T)==int(-e_value*e_value*k,t);
sol2 = C*solve(sol,T)
sol1 = subs(sol1,C1,0)
Ux = sol1*sol2;
Ux = subs(Ux,-C*C2,Bn)
%Bn is half range fourier sine series

```

OUTPUT

```
>> heat_eqn
--------------------
Experiment - 7
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
--------------------

ans =

    'Solving for various conditions...'


ans =

    'When lambda > 0'


e_function =

- sin(lambda^(1/2)*x)*C2 + cos(lambda^(1/2)*x)*C1


ans =

    'Non-zero values in the solution'


ans =

    'When lambda = 0'


ans =

    'No non-trivial solution'


ans =

    'When lambda < 0'

e_value =

n^2*pi^2

e_function =

- sin(lambda^(1/2)*x)*C2 + cos(lambda^(1/2)*x)*C1

sol1 =

- sin(x*(pi*(n^2)^(1/2))^(1/2))*C2 + cos(x*(pi*(n^2)^(1/2))^(1/2))*C1

sol2 =

C*exp(-k*n^2*t*pi^2)

sol1 =

-C2*sin(x*(pi*(n^2)^(1/2))^(1/2))

Ux =

Bn*exp(-k*n^2*t*pi^2)*sin(x*(pi*(n^2)^(1/2))^(1/2))

fx >>
```

Experiment - 8

Write a program to solve wave equation subject to boundary and the initial conditions

CODE

```
disp("--------------------")
disp("Experiment - 8")
disp("Name -> M K Lino Roshaan")
disp("Roll No -> 2K22/MC/87")
disp("--------------------")
syms u(x,t) x k c t f(x) g(x) X(x) T(t) A B Bn C D C1 C2 lambda
%INCLUDE FUNCTION FOR DISTANCE
[e_value, e_function] = sturm_liouville();
e_value = sqrt(e_value);
%SOLUTION FOR X
sol_X = simplify(subs(e_function,lambda^(1/2),k));
%INCLUDE FUNCTION FOR TIME
[e_value2, e_function2] = sturm_liouville();
e_value2 = sqrt(e_value2);
%SOLUTION FOR T
sol_T = subs(subs(e_function,x,t),lambda^(1/2),k*c);
X(x) = subs(subs(sol_X,C1,A),-C2,B)
T(t) = subs(subs(sol_T,C1,C),-C2,D)
%Applying Boundary condition and Initial condition
fprintf('\n\nAfter applying Boundary and Initial Conditions:\n')
%Bn is half range fourier sine series
X(x) = subs(subs(X(x),A,0),B,Bn)
%f(x) and g(x) are half range fourier sine series
T(t) = subs(subs(T(t),C,f(x)),D,g(x))

fprintf('\nAs we have assumed initially: u(x,t) = X(x)*T(t):\n')
u(x,t) = X(x)*T(t)
```

OUTPUT

```
>> wave_eqn
--------------------
Experiment - 8
Name -> M K Lino Roshaan
Roll No -> 2K22/MC/87
--------------------

ans =

    'Solving for various conditions...'


ans =

    'When lambda > 0'


e_function =

- sin(lambda^(1/2)*x)*C2 + cos(lambda^(1/2)*x)*C1


ans =

    'Non-zero values in the solution'


ans =

    'When lambda = 0'


ans =

    'No non-trivial solution'
```

fx

```
ans =

    'When lambda < 0'


ans =

    'Solving for various conditions...'


ans =

    'When lambda > 0'


e_function =

- sin(lambda^(1/2)*x)*C2 + cos(lambda^(1/2)*x)*C1


ans =

    'Non-zero values in the solution'


ans =

    'When lambda = 0'


ans =

    'No non-trivial solution'
```

```
ans =

    'When lambda < 0'

X(x) =

sin(k*x)*B + cos(k*x)*A

T(t) =

sin(c*k*t)*D + cos(c*k*t)*C


After applying Boundary and Initial Conditions:

X(x) =

Bn*sin(k*x)

T(t) =

cos(c*k*t)*f(x) + sin(c*k*t)*g(x)

As we have assumed initially: u(x,t) = X(x)*T(t):

u(x, t) =

Bn*sin(k*x)*(cos(c*k*t)*f(x) + sin(c*k*t)*g(x))

fx >>
```