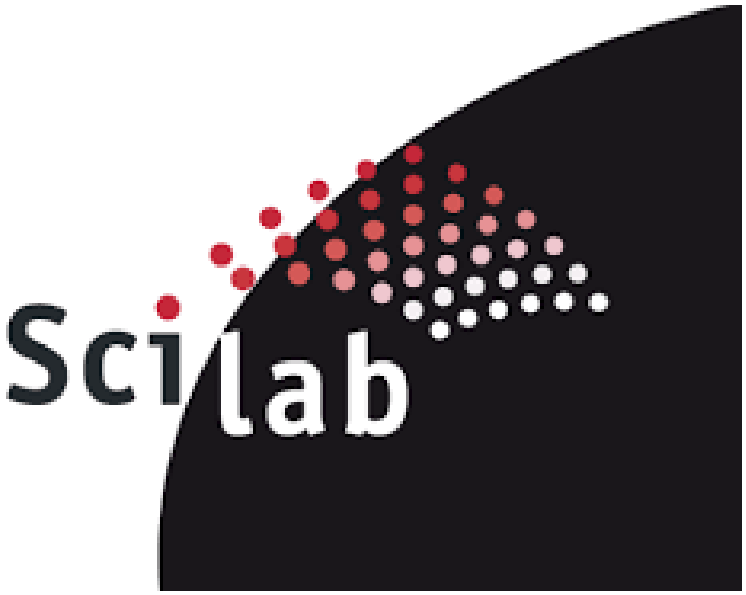
BHASKARACHARYA COLLEGE OF APPLIED SCIENCES
(University of Delhi)
Accredited **'A'** Grade by **NAAC**
'Star College Status' by **DBT**
2021 NIRF College Ranking - 32

MATHEMATICAL PHYSICS 3

PRACTICAL FILE



SUBMITTED BY
Roll No - 2107017
Name - Nabdeep Patel
Department - Bsc Physics(Hons.)
2st Year 4st Semester

SUBMITTED TO
Subject Teachers
Dr. Partha Pal
Dr. Harendra Kumar

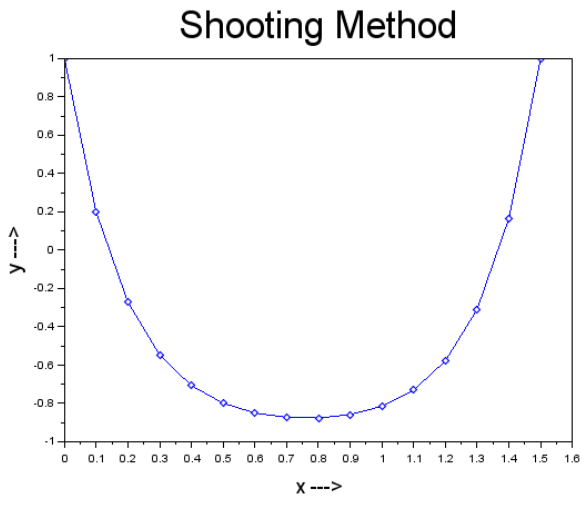
SHOOTING METHOD

Program

```
funcprot(1);clear;clc;clf;
xi=0;xf=%pi/2;h=0.1;x=xi:h:xf;n=length(x);yi=1;yf=1;
function[g]=f(x,y,z)
    g=20*y +32;
endfunction

function [y,z]=RK2(yi,zi)
    y(1)=yi;z(1)=zi;
    for i=1:n-1
        s1=z(i);p1=f(x(i),y(i),z(i));
        s2=z(i)+h*p1;p2=f(x(i)+h,y(i)+s1*h,z(i)+p1*h);
        y(i+1)=y(i)+(h/2)*(s1+s2);
        z(i+1)=z(i)+(h/2)*(p1+p2);

    end
endfunction

for i=1:2
    zt(i)=input("Enter guess"+string(i)+"for initial slope (dy/dx): ");
    [y,z] = RK2(yi,zt(i));
    yn(i)=y(n,1);
end
zc=zt(1)+(zt(2)-zt(1))*(yf-yn(1))/(yn(2)-yn(1));
[y,z]=RK2(yi,zc);
disp(" x          y           z");disp([x' y z]);
plot(x',y,'d-');
xlabel("x --->",'fontsize',4);
ylabel("y --->",'fontsize',4);
title("Shooting Method",'fontsize',6);
```

Output

```
Enter guess1for initial slope (dy/dx): 0
Enter guess2for initial slope (dy/dx): 9

 x         y          z

   0.     1.         - 11.313922
   0.1    0.1986078  - 6.6372891
   0.2  - 0.2713578  - 3.8903618
   0.3  - 0.5465248  - 2.2743399
   0.4  - 0.7068680  - 1.3191621
   0.5  - 0.7989518  - 0.7467711
   0.6  - 0.8494507  - 0.3901584
   0.7  - 0.8728732  - 0.1446178
   0.8  - 0.8757234    0.0630283
   0.9  - 0.8582936    0.2962834
   1.   - 0.8145752    0.6284534
   1.1  - 0.7302076    1.1657349
   1.2  - 0.5777694    2.081204
   1.3  - 0.3078698    3.6705927
   1.4    0.1668516    6.4478362
   1.5    1.           11.311264
```



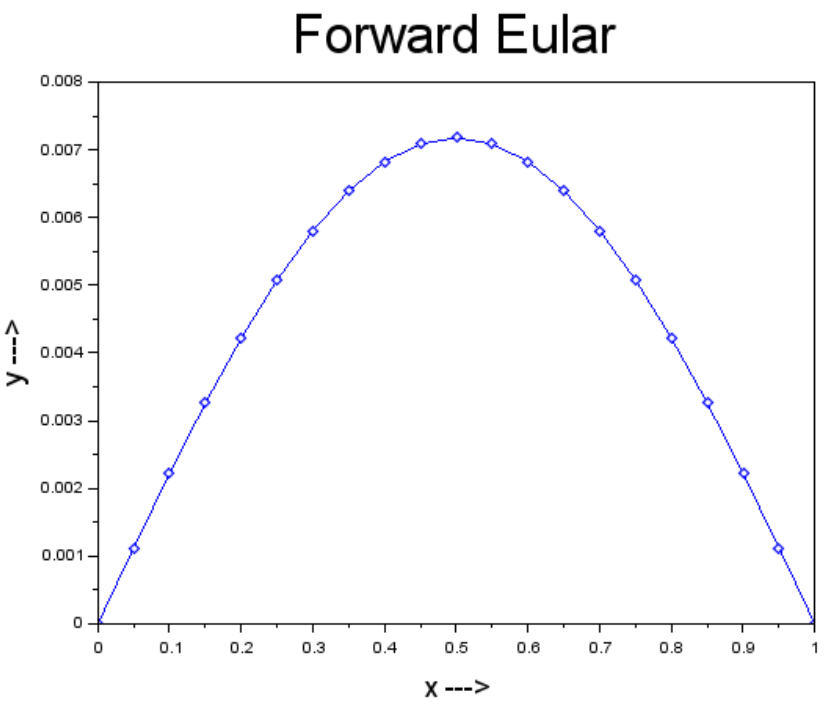
FORWARD EULAR METHOD

Program

```
clear;clc;clf;
function u=f(x,t)
    u=sin(%pi*x)
endfunction
function g=ex(x,t)
    g=exp((-%pi^2)*t)*sin(%pi*x)
endfunction
xi=0
xf=1
ti=0
tf=0.5
h=0.05
k=0.0005
alp=1
lambda=((alp^2)/(h^2))*k
n=floor((xf-xi)/h)
m=floor((tf-ti)/k)
u(1)=0;u(n+1)=0
x(1)=xi;x(n+1)=xf
for i=1:n-1
    x(i+1)=x(i)+h
end
for i=1:n-1
    w(i)=f(x(i+1))
end
for i=1:n-1
    for j=1:n-1
        if i==j then
            A(i,j)=(1-(2*lambda))
        elseif i==j+1 then
            A(i,j)=lambda
        elseif i==j-1 then
            A(i,j)=lambda
        else
            A(i,j)=0
        end
    end
end
for t=1:m
    w=A*w
end
for i=1:n-1
    u(i+1)=w(i)
end
disp("x          y");disp([x u]);
plot(x,u,'d-')
xlabel("x --->",'fontsize',4);
ylabel("y --->",'fontsize',4);
title("Forward Eular",'fontsize',6);
```

Output

x	y
0.	0.
0.05	0.0011228
0.1	0.0022179
0.15	0.0032584
0.2	0.0042187
0.25	0.0050751
0.3	0.0058066
0.35	0.0063950
0.4	0.0068260
0.45	0.0070890
0.5	0.0071773
0.55	0.0070890
0.6	0.0068260
0.65	0.0063950
0.7	0.0058066
0.75	0.0050751
0.8	0.0042187
0.85	0.0032584
0.9	0.0022179
0.95	0.0011228
1.	0.



CRANK NICHOLSON

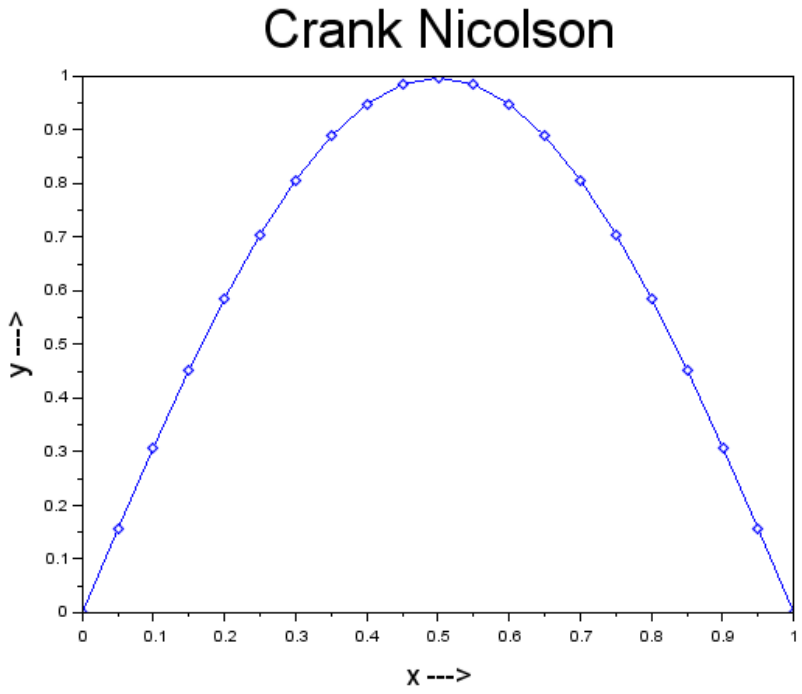
Program

```
clear;clc;clf;
function u=f(x,t)
    u=sin(%pi*x)
endfunction
function g=ex(x,t)
    g=exp((-%pi^2)*t)*sin(%pi*x)
endfunction
xi=0
xf=1
ti=0
tf=0.5
h=0.05
k=0.0005
alp=1
lambda=((alp^2)/(h^2))*k
n=floor((xf-xi)/h)
m=floor((tf-ti)/k)
u(1)=0;u(n+1)=0
x(1)=xi;x(n+1)=xf
for i=1:n-1
    x(i+1)=x(i)+h
end
for i=1:n-1
    w(i)=f(x(i+1))
end
for i=1:n-1
    for j=1:n-1
        if i==j then
            B(i,j)=(1-lambda)
            A(i,j)=(1+lambda)
        elseif i==j+1 then
            B(i,j)=lambda/2
            A(i,j)=-lambda/2
        elseif i==j-1 then
            B(i,j)=lambda/2
            A(i,j)=lambda/2
        else
            A(i,j)=0
            B(i,j)=0
        end
    end
end

disp(A,B)
for t=1:m
    h=A*w
    h=B*w
end
for i=1:n-1
```

```
    u(i+1)=h(i)
end
for i=1:n+1
    exf(i)=ex(x(i),0.5)
end
disp(x,u)
plot(x,u,'d-')
```

Output

```
    x         u

    0.        0.
    0.05      0.1560493
    0.1       0.3082561
    0.15      0.4528726
    0.2       0.5863379
    0.25      0.7053656
    0.3       0.8070249
    0.35      0.8888126
    0.4       0.9487147
    0.45      0.9852563
    0.5       0.9975377
    0.55      0.9852563
    0.6       0.9487147
    0.65      0.8888126
    0.7       0.8070249
    0.75      0.7053656
    0.8       0.5863379
    0.85      0.4528726
    0.9       0.3082561
    0.95      0.1560493
    1.        0.
```

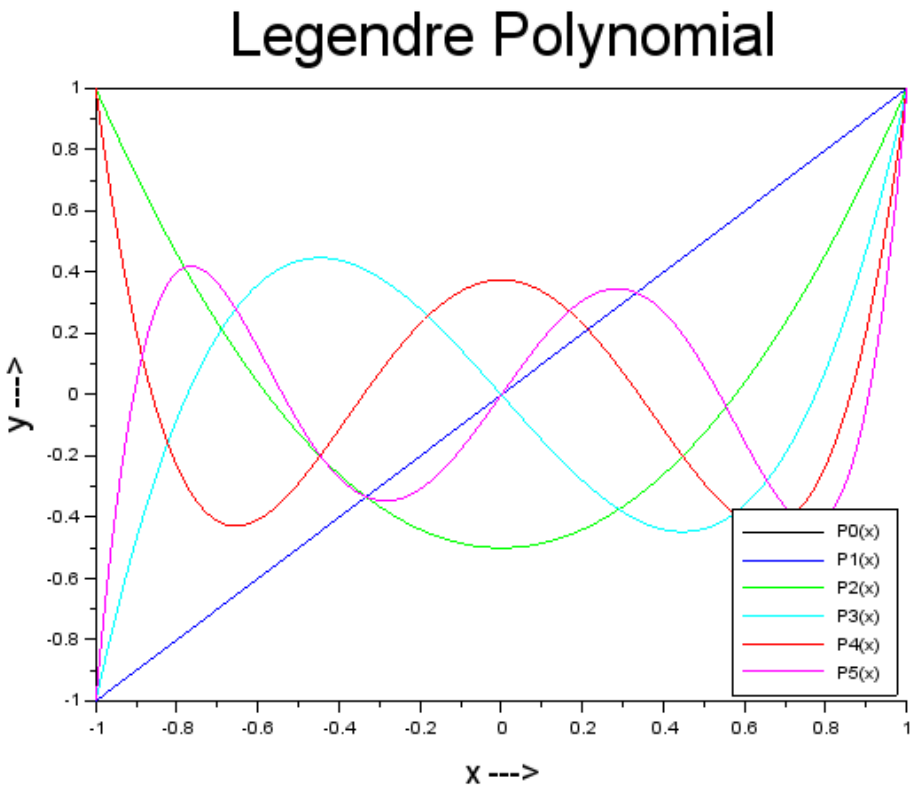


LEGENDRE POLYNOMIAL

Program

```
function pl=p(n,x)
    sum = 0
    for m=0:n/2
        den=factorial(m)*factorial(n-m)*(2^n)*factorial(n-2*m)
        sum=sum+((-1)^m)*factorial(2*n-2*m)*(x.^(n-2*m))/den
    end
    pl=sum
endfunction
disp(p(n,x))
clf;
x=-1:0.01:1
for n=0:5
    xlabel("x --->",'fontsize',4);
    ylabel("y --->",'fontsize',4);
    title("Legendre Polynomial",'fontsize',6);
    plot2d(x,p(n,x),n+1)
    hl=legend(['P0(x)';'P1(x)';'P2(x)';'P3(x)';'P4(x)';'P5(x)';],4);
end
```

Output



GAUSS QUADRATURE METHOD

Program

```
clear;clc;
function [l]=GQ(npt,w,x)
    l = 0
    f = (3/2)*((x.*x).*x)-(x/2)
endfunction
npt = 1
w(1) = 2
x(1) = 0
l = GQ(npt,w,x)
disp("Result from 1 point quadrature = ",l)
npt = 2
w(1) = 1
w(2) = 1
x(1) = -1/sqrt(3)
x(2) = 1/sqrt(3)
I = GQ(npt,w,x)
disp("Result from 2 point quadrature = ",l)
npt = 3
w(1) = 5/9
w(2) = 5/9
w(3) = 8/9
x(1) = -sqrt(3)/5
x(2) = sqrt(3)/5
x(3) = 0
I = GQ(npt,w,x)
disp("Result from 3 point quadrature = ",l)
```

Output

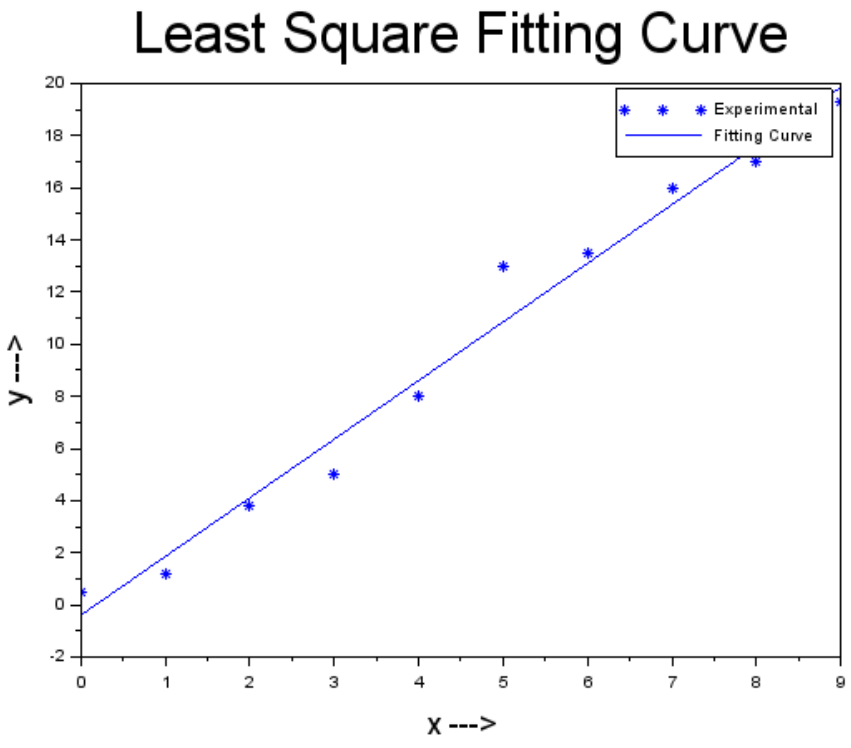
```
    0.

 Result from 1 point quadrature =

    0.

 Result from 2 point quadrature =

    0.

 Result from 3 point quadrature =
```

LEAST SQUARE FITTING CURVE

Program

```
clc;clear;clf;
x = [0,1,2,3,4,5,6,7,8,9]
y = [0.5,1.2,3.8,5,8,13,13.5,16,17,19.3]
n = length(x);
sumx = sum(x)
sumy = sum(y)
sum2x = sum(x.*x)
sum2y = sum(x.*y)
sumxy = sum(x.*y)
c = (sumx*sumxy-sumy*sum2x)/(sumx*sumx-n*sum2x)
m = (sumy-c*n)/sumx
x1 = 0:0.1:9
y1 = m*x1+c
disp(x,y,x1,y1)
plot(x,y,"*","linewidth",1)
plot(x1,y1,"-","linwidth",2)
xlabel("x --->",'fontsize',4);
ylabel("y --->",'fontsize',4);
title("Least Square Fitting Curve",'fontsize',6);

legend(["Experimental"],["Fitting Curve"])
```

Output



FOURIER COEFFICIENTS

Program

```
clc;clear;clf;
L=0
u=2*%pi
f='x'
a0=integrate(f,'x',L,u);
n=input("Enter the value of n")
for i = 1:n
    a(i)=integrate('x*cos(i*x)','x',L,u,1e-3)/(%pi)
    b(i)=integrate('x*sin(i*x)','x',L,u,1e-3)/(%pi)
end
function y=f(x)
    y=0
    for i =1:n
        y=y+a(i)*cos(i*x)+b(i)*sin(i*x)
    end
    y=y+a0/2
endfunction

x = -4*%pi:0.01:4*%pi
disp('a =',a0);
disp('a(n) =',a(n));
disp('b(n) =',b(n));
plot(x,f(x))
xlabel("x --->",'fontsize',4);
ylabel("y --->",'fontsize',4);
title("Fourier Coefficient",'fontsize',6);
```

Output

```
Enter the value of n2

    19.739209

 a =

  - 3.053D-16

 a(n) =

  - 1.

 b(n) =
```