LABORATORY MANUAL FOR MATHEMATICS PRACTICALS (WITH FOSS TOOLS) FOR 2ND SEMESTER B.Sc.

LIST OF PROGRAMS

Lab 1: Simple Programs

Lab 2: Plotting of curves

Lab 3: Tracing of curves (Cartesian form)

Lab 4: Tracing of curves (Polar form and Parametric form)

Lab 5: Surface Area

Lab 6: Volume of Revolution

Lab 7: Groups

Lab 8: Groups- Cayley Table

Lab 9: Differential Equations-1

Lab 10: Differential Equations-2

Lab 11: Differential Equations-3

Lab 12: Differential Equations-4

Lab 13: Revision

Lab 14: Preparatory Exam

NOTE:
In each lab one program has to be executed and relevant problems have to be solved manually.

Lab I :   Simple Programs

//1. Write a program to generate the sequence 1,4,7,10,13,…..,52

```
for i=1:3:52
    disp(i)
end
```

//2. Write a program to generate Prime numbers between 125 to 300.

```
p=primes(300);
k=length(p)
  for i=1:k
        if p(i)>125 & p(i)<300 then
          disp(p(i))
        end
  end
```

//3. Write a program to find the sum of first 100 natural numbers.

```
sum=0;
for i=1:100
   sum = i + sum;
end
printf("sum of natural numbers from 1 to 100 is %d", sum);
```

//4. Write a program to find the Largest number from the given set of 3 numbers

```
a=input("Enter the value for a :");
b=input("Enter the value for b  :");
c=input("Enter the value for c  :");
if a>b & a>c then
   mprintf("%d is the largest element\n",a)
elseif b>c then
   mprintf("%d is the largest element\n",b)
else
   mprintf("%d is the largest element\n",c)
end
```

//5. Write a program to generate first 10 terms of Fibonacci Series

```
mprintf("Fibonacci series ");
n=input("enter the number  =   ")
      for i=1:n
    if i<3
     f(i)=1
    disp(f(i));
    else
      f(i)=f(i-1)+f(i-2)
        disp(f(i))
   end
   end
```

PRACTICE

//Write a program to verify whether the given number is a palindrome or not

```
num=input("Enter the number :");
rev=0;
temp=num;
while num>0
    rem=modulo(num,10);
    num=(num-rem)/10;
    rev=rev*10+rem
end
if temp==rev then
    mprintf("The given number is a palindrome")
else
    mprintf("The given number is not a palindrome")
end
```

//Write a program to generate the Prime numbers till 50

```
primes(50)
```

//Write a program to find the Smallest number from the given set of 3 numbers(use < in Program 4)

Lab II : Plotting of curves

// 1.Plot the curve $y = 2x^2 - 2x + 1$

```
x =[0:0.1:10]';
 y=[2*x^2-2*x+1];
plot2d(x,y)
title("curve of  y=2x^2-2x+1 ",'fontsize',4)
xlabel("x")
ylabel("y")
```

//2. Plot the curve $y = x^2 + 1$ and $y = x^2 + 2x + 2$

```
x =[0:0.1:10]';
y=[x^2+1,x^2+2*x+2];
plot2d(x,y)
title("curves  of  y=x^2+1 and  y=x^2+2x+2 ", 'fontsize',4);
```

// 3.Plot the curve $\sin(x)$ and $\cos(x)$

```
x=[0:%pi/16:2*%pi]'
y=[sin(x), cos(x)]
plot2d(x,y)
title(" curves of sin(x) and cos(x)",'fontsize',4)
```

PRACTICE

1. **Plot the curve y=e^x**
2. **Plot the curve y=log x**
3. **Plot the curve y=x^3**

Lab III : Tracing of curves (Cartesian form)

// 1. Plot the curve catenary in Cartesian form

```
x=[-5:0.01:5]';
a=2;
y=a*cosh(x/a);
plot2d(x,y)
title("Catenary ",'fontsize',4)
xlabel(“ x ”)
ylabel(“ y”)
```

PRACTICE

Plot the curve Cissoids in Cartesian form

```
x =[0:0.1:15.99]'
a=16;
y=abs(sqrt ((x^3)/(a-x))) ;
plot2d(x,y)
plot2d(x,-y)
title("Cissoids ",'fontsize',4)
```

Astroid and Strophoid in Cartesian form.

```
// Astroid : y=(a^(2/3)-x^(2/3))^(3/2);
a=8;
x=[0:0.01:8]';
y=(a^(2/3)-(x^2)^(1/3))^(3/2);
x1=[-8:0.01:0]';
y1=(a^(2/3)-(x1^2)^(1/3))^(3/2);
b=gca()
b.x_location="origin"
b.y_location="origin"
plot2d(x,y)
plot2d(x,-y)
plot2d(x1,y1)
plot2d(x1,-y1)
```

```
//Strophoid : y=x*((a+x)/(a-x))^(1/2)
a=16;
x=[-16:2:15.99]';
y1=x.*abs((a+x)./(a-x)).^(1/2);
b=gca()
b.x_location="origin"
b.y_location="origin"
y=[y1,-y1];
plot(x,y);
```

Lab IV : Tracing of curves (Polar form and Parametric form)

// 1. Plot the curve cardioid in Polar form

```
// Cardioid : r=a*(1+cos(t))
t=[0:0.1:2*%pi]';
a=1;
r=a*(1+cos(t));
polarplot(t,r)
title("Cardioid ",'fontsize',4)
```

// 2. Plot the curve Astroid in Parametric form

```
// Astroid : x=a*(cos(t))^3, y=a*(sin(t))^3
t=[-2*%pi:%pi/32:2*%pi]';
a=4;
x=a*(cos(t))^3
y=a*(sin(t))^3
b=gca()
b.x_location="origin"
b.y_location="origin"
plot2d(x,y)
title("Astroid ",'fontsize',4)
```

PRACTICE

// Plot the curve Three leaved rose in Polar form

```
// Three leaved rose : r=a*sin(3*t)
t=[0:0.1:2*%pi]';
a=1;
r=a*sin(3*t)
polarplot(t,r)
title("Three leaved rose ",'fontsize',4)
```

```
r=a*(1-cos(t))
r=a*cos(3*t)
r=a*cos(2*t)
```

LAB V: Surface Area

// 1. Find the surface area generated by revolving the arc of the catenary y=acosh(x/a)
//from x=0 to x=a about X-axis

```
a=1;
//funcprot (0)
function I=I(x)
      y=a*cosh(x/a)
     dy=sinh(x/a)
      I=2*%pi*y*sqrt((dy)^2+1)
endfunction
SA=intg(0,a,I);
disp(SA)
```

PRACTICE

// 2. Find the surface area generated by revolving the cardioid r=a(1+cosx) about the
//initial line

```
a=1
function I=I(x)
      r=a*(1+cos(x))
      y=r*sin(x)
     dr=-a*sin(x)
      I=2*%pi*y*sqrt((dr)^2+r^2)
endfunction
SA=intg(0,%pi,I);
mprintf("Surface area=",SA)
disp(SA)
```

LAB VI： Volume of Revolution

// 1. Find the volume of the solid generated by revolving the ellipse about the x-axis

```
a=2;
b=1;
function y=y(x)
    y=2*%pi*((b^2)*(1-(x/a)^2))
endfunction
v=intg(0,a,y)
disp(v,"volume=")
```

PRACTICE

//Find the volume of the solid obtained by revolving the cardioid r=a(1+cosθ)about the initial line

```
a=1;
function y=y(x)
y=(2/3)*%pi*(a*(1+cos(x)))^3*sin(x);
endfunction
I=intg(0,a,y)
disp(I)
```

//Find the volume of the solid obtained by revolving of one arc of the cycloid x=a(1+sint) and y=a(1+cost) about its base.

```
//cycloid x=a(t+sint), y=a(1+cost)
a=1;
function v=v(t)
   y=a*(1+cos(t))
   dx= a*(1+cos(t))
   v=2*%pi*y^2*dx
endfunction
vol=intg(0,%pi,v);
mprintf("The required volume is \n %f", vol)
```

LAB VII : Groups

// 1. To show that [1,-1] is a group under multiplication

```
H=[1,-1];
a=1;
b=-1;

// closure law

if a*b==a | a*b==b then
    printf("H is closed under multiplication\n");
else
    printf("H is not closed under multiplication and hence H is not a group\n");
    abort;
end

// Associative law

if a*(b*a)==(a*b)*a then
    printf("H is associative under multiplication\n");
else
    printf("H is not associative under multiplication and hence H is not a group\n");
    abort;
end

// Identity  law

if a*a==a & b*a==b then
    e=a;
    printf("e=%f is an unique identity element\n",a);

elseif a*b==a & b*b==b then
    printf("e=%f is an unique identity element\n",b);
else
    printf("No identity element exits and hence H not a group\n");
    abort;
end

// Inverse  exists
in1=e/a;
in2=e/b;
if in1==a | in1==b & in2==a | in2==b then
    mprintf("i=%f is an inverse element of %d\n",in1,a);
     mprintf("i=%f is an inverse element of %d\n",in2,b);
 else
     printf("No inverse element exists and hence H is not a group\n")
     abort;
end
printf("H is a Group\n");
```

Note : For generalisation take a=H(i), b=H(i+1)

LAB VIII: Groups- Cayley Table

// 1. Construct a Cayley Table for (Z4,+4)

```
function cayley(n)
    for i=0:n-1
        for j=0:n-1
            c=i+j;
            if c>=n then
                c=c-n;
            end
            mprintf("%d \t",c);
        end
        mprintf("\n");
    end
endfunction
```

// type Cayley(4) at the cursor ----> to get Cayley table of (Z4,+4)
// type Cayley(5) at the cursor ----> to get Cayley table of (Z5,+5)

PRACTICE

//Create Cayley table for (Z4,X4)
// Enter cayley(4) at the cursor

```
function cayley(n)
    for i=0:n-1
        for j=0:n-1
            c=i*j;
            if c>=n then
                c=modulo(c,n)
            end
            mprintf("%d \t",c);
        end
        mprintf("\n");
    end
endfunction
```

//Create Cayley table for (G,X 10) where G={2,4,6,8}
// Enter cayley(10) at the cursor

```
function cayley(n)
    for i=2:2:n-1
        for j=2:2:n-1
            c=i*j;
            if c>=n then
                c=modulo(c,n)
            end
            mprintf("%d \t",c);
        end
        mprintf("\n");
    end
endfunction
```

LAB IX : Differential Equations-1

// 1. Solve dy/dx= sec(x)^2 with y(0)=0 - Variable separable

```
function f=f(x, y)
f=sec(x)^2
endfunction
x0=0;
y0=0;
x=[0:0.1:1];
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

// 2. Solve (x^2+y^2)dx-2xy dy=0 – Homogenous equation

```
function f=f(x, y)
  f=(x^2+y^2)/(2*x*y)
endfunction
x0=1;
 y0=1;
x=[1:0.1:10]
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

PRACTICE

// Solve dy/dx=1/(x^2+1) - Variable separable

```
function f=f(x, y)
f=1/(x^2+1)
endfunction
x0=0;
 y0=0;
x=[0:0.1:10]
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

//Solve dy/dx=e^x - Variable separable

```
function f=f(x, y)
   f=%e^x
endfunction
x0=0
 y0=1;
x=[1:0.1:10]
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

LAB X: Differential Equations-2

// 1. Solve dy/dx = 2*y*tan(x)+sin(x) – Linear Equation

```
function f=f(x, y)
  f=-2*y*tan(x)+sin(x)
endfunction
x0=%pi/3;
 y0=0;
x=5;
x=[1:0.1:5]
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

PRACTICE

// Solve dy/dx = (-y+2*x^3)/(2*x) - Linear Equation

```
function f=f(x, y)
f=(-y+2*x^3)/(2*x)
endfunction
x0=0; y0=1;
x=[1:0.1:5]
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

//Solve dy/dx=y^2-y*sin(t)+cos(t) –Linear Equation

```
function f=f(t, y)
   f=y^2-y*sin(t)+cos(t)
endfunction
y0=0;
t0=0;
t=0:0.1:%pi;
y=ode(y0,t0,t,f);
plot2d(t,y)
```

LAB XI： Differential Equations-3

// 2. Solve xdy/dx +(1-x)y=x^2*y^2 –Bernoulli Equation

```
function f=f(x, y)
f=(x^2*y^2+(x-1)*y)/x
 endfunction
x0=-1;
 y0=1;
x=[4:0.1:15]
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

PRACTICE

// Solve dy/dx+ytan(x)=y^3*sec(x)- Bernoulli Equation

```
function f=f(x, y)
f=(y^3*sec(x))-(y*tan(x))
 endfunction
x0=%pi;
 y0=3;
x=[%pi:0.1:2*%pi]
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

LAB XII： Differential Equations-4

//1. Solve dy/dx = -((4*x+3*y+1)/(3*x+2*y+1)) –Exact

```
function f=f(x, y)
f=-((4*x+3*y+1)/(3*x+2*y+1))
endfunction
x0=1
y0=0;
x=[1:0.1:6]
a=ode(x0,y0,x,f)
disp(a)
plot2d(x,a)
```

PRACTICE

// Solve (x^2-y)dx + (y^2-x)dy =0- Exact

```
function f=f(x, y)
  f=(y-x^2)/(y^2-x)
endfunction
x0=10
 y0=1;
x=[3:0.1:10]
a=ode(x0,y0,x,f)
disp(a)
 plot2d(x,a)
```