Aim: To solve a differential equation using Euler's Method.

Algorithm:

1. Enter the initial values of x and y(x_0 and y_0).
2. Enter the value of x, for which y is to be determined.
3. Enter the width of the interval, 'h'.
4. Do:

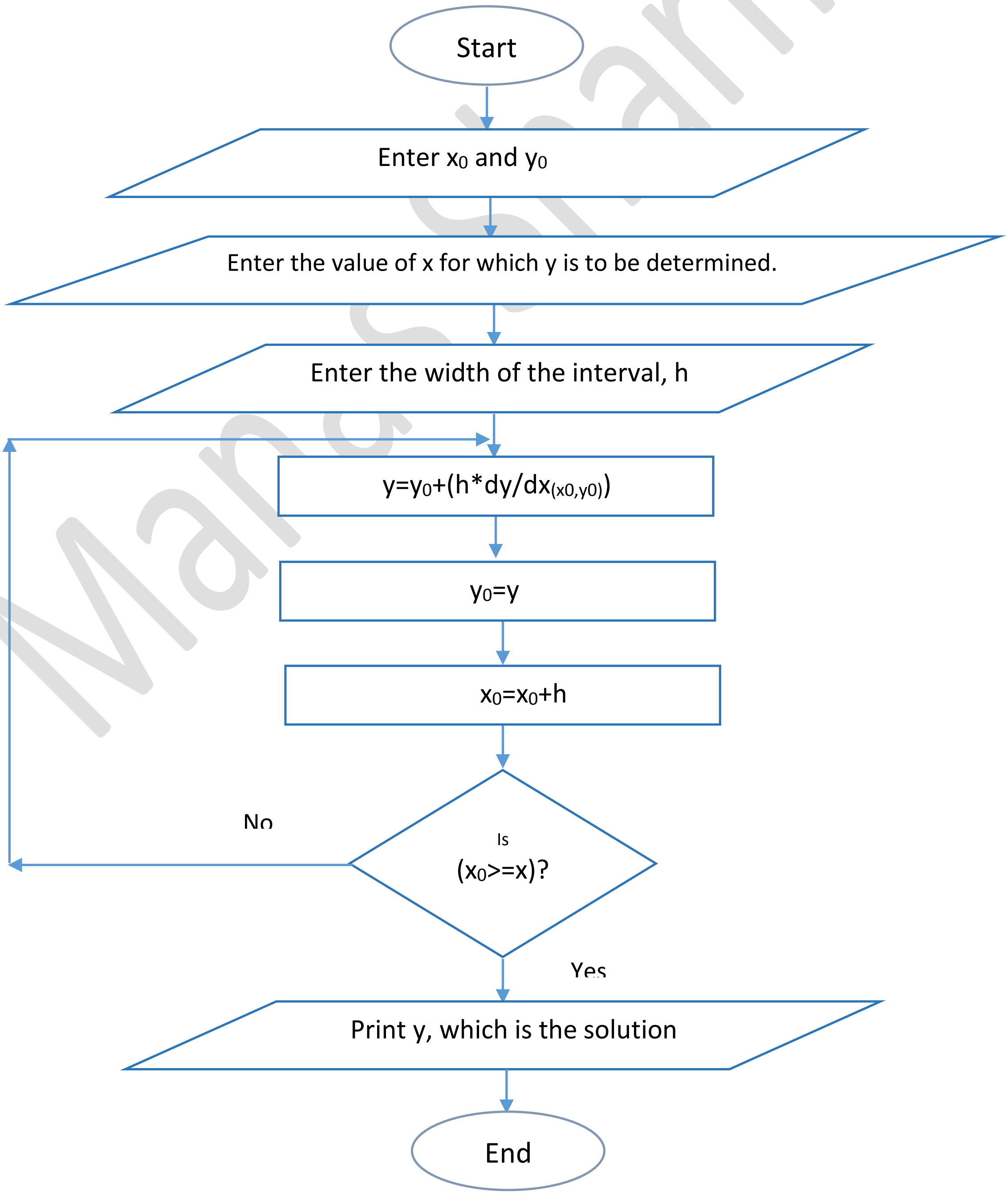
   $y=y_0+(h*dy/dx_{(x0,y0)})$

   $y_0=y$.

   $x_0=x_0+h$

   Until (x_0>=x)
5. Print y, which is the solution.

Flow Chart:



Program:

```
//Eulers Method to solve a differential equation
#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;
double df(double x, double y)            //function for defining dy/dx
{
    double a=x+y;                //dy/dx=x+y
    return a;
}
int main()
{
    int n;
    double x0,y0,x,y,h;            //for initial values, width, etc.
    cout.precision(5);            //for precision
    cout.setf(ios::fixed);
    cout<<"\nEnter the initial values of x and y respectively:\n";
//Initial values
    cin>>x0>>y0;
    cout<<"\nFor what value of x do you want to find the value of y\n";
    cin>>x;
    cout<<"\nEnter the width of the sub-interval:\n";            //input
width
    cin>>h;
    cout<<"x"<<setw(19)<<"y"<<setw(19)<<"dy/dx"<<setw(16)<<"y_new\n";
    cout<<"----------------------------------------------------------\n";
    while(fabs(x-x0)>0.0000001)        //I couldn't just write
"while(x0<x)" as they both are floating point nos. It is dangerous to
compare two floating point nos. as they are not the same in binary as they
are in decimal. For instance, a computer cannot exactly represent 0.1 or
0.7 in binary just like decimal can't represent 1/3 exactly without
recurring digits.
    {
        y=y0+(h*df(x0,y0));            //calculate new y, which is
y0+h*dy/dx
        cout<<x0<<setw(16)<<y0<<setw(16)<<df(x0,y0)<<setw(16)<<y<<endl;
        y0=y;                    //pass this new y as y0 in the next
iteration.
        x0=x0+h;                //calculate new x.
    }
    cout<<x0<<setw(16)<<y<<endl;
    cout<<"The approximate value of y at x=0 is "<<y<<endl;        //print
the solution.
    return 0;
}
```

Output:

For dy/dx=-2x-y

```
Enter the initial values of x and y respectively:
0       -1

For what value of x do you want to find the value of y
.4

Enter the width of the sub-interval:
.1
x                  y              dy/dx          y_new
----------------------------------------------------------
0.00000        -1.00000         1.00000        -0.90000
0.10000        -0.90000         0.70000        -0.83000
0.20000        -0.83000         0.43000        -0.78700
0.30000        -0.78700         0.18700        -0.76830
0.40000        -0.76830
The approximate value of y at x=0 is -0.76830
```

For dy/dx=x+y:

```
Enter the initial values of x and y respectively:
0       1

For what value of x do you want to find the value of y
1

Enter the width of the sub-interval:
0.1
x                  y              dy/dx          y_new
----------------------------------------------------------
0.00000         1.00000         1.00000         1.10000
0.10000         1.10000         1.20000         1.22000
0.20000         1.22000         1.42000         1.36200
0.30000         1.36200         1.66200         1.52820
0.40000         1.52820         1.92820         1.72102
0.50000         1.72102         2.22102         1.94312
0.60000         1.94312         2.54312         2.19743
0.70000         2.19743         2.89743         2.48718
0.80000         2.48718         3.28718         2.81590
0.90000         2.81590         3.71590         3.18748
1.00000         3.18748
The approximate value of y at x=0 is 3.18748
```