Mechi Multiple Campus

(Tribhuvan University)

Bhadrapur, Jhapa

Lab Report of Numerical Methods (CACS-252)

Faculty of Humanities and Social Sciences

Tribhuvan University

Kathmandu, Nepal

Submitted By

Name: Rajeev Kamat

Symbol no:18020022

Submitted to

Mechi Multiple Campus

Department of Bachelor in Computer

Bhadrapur Jhapa, Nepal.

ACKNOWLEDGEMENT

This project is specially designed to develop enhance the knowledge of student in web technology with the advanced computer technology. The assigned for the partial fulfillment of BCA (Bachelor in Computer Application) third semester student. In many school and collages, spite of advantage of computer technology. They lag behind in fair result processing. The project assigned was to solve the current problem of various academic institutions. We know that to complete this project successfully different have provided us great help. We would like to express our grateful to all of them who have provided direct and indirect help to complete this project.

We would heartily like to express our respect for **Numerical Methods** teacher Mr. Narayan Dhamala for his keen support and supervision to complete this project successfully.

We are grateful to our collage for their important suggestion and help. We are thankful to our collage management to provide us great support during this project

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the lab report entitled "Numerical Methods Lab sheet" is an academic work done by "Rajeev Kamat" submitted in the partial fulfillment of the requirement for the degree of Bachelor of Computer Application at Faculty of Humanities and social science, Tribhuvan University under my guidance and supervision. To the best of my knowledge, the worked performed by him in the lab report is his own creation.

Name:
Designation:
Date:

Signature of supervisor

Bisection Method

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#define EPS 0.000001
#define F(x) log(x)-cos(x)
void bim(float*a, float*b, float*root, float*s, int*count); void
main ()
{ int count;
   float a, b, root, s;
   cout<<"SOLUTION BY BISECTION METHOD:"<<endl;
   cout<<"input starting values:"<<endl;
   cin>>a>>b;
   bim(&a, &b, &root, &s, &count); if
   (s==0)
   {  cout<<"starting points do not bracket any root"<<endl;
      cout<<"check whether they bracket EVEN roots."<<endl;
   }
   else
   { cout<<"root="<<root<<endl;
      float fun = F(root);
      cout<<"f(root)="<<fun<<endl;

      cout<<"Iterations="<<count<<endl;
   }
   getch();
   clrscr();
}
void bim(float*a, float*b, float*root, float*s, int*count)
```

```
{ float x1, x2, x0, f0, f1, f2;
   x1=*a;
   x2=*b;
   f1=F(x1);
   f2=F(x2);
   if(f1*f2> 0)
   { *s=0;
      return;
   }
   else
   { *count=0; begin:
      x0=(x1+x2)/2.0;
      f0=F(x0);
      if(f0==0)
      { *s=1; *root=x0; return; }
      if(f1*f0<0)
      {  x2=x0;
      }
      else
       {  x1=x0;
            f1=f0;
      }
      if(fabs((x2-x1)/x2) < EPS)
      { *s=1;
         *root=(x1+x2)/2.0;
         return;
      }
      else
```

```
    { *count=*count+1;
      goto begin;

    }

  }

}
```

Output of Bisection Method

Newton-Raphson Method

```
#include<iostream.h>

#include<conio.h>

#include<math.h>

#define EPS 0.000001

#define MAXIT 20

#define F(x) (x)*(x)*(x)+(x)*(x)-3*(x)-3

#define FD(x) 3*(x)*(x)+2*(x)-3

void main()

{ clrscr(); int

  count;

  float x0, xn, fx, fdx, fxn;
```

```
cout<<"SOLUTION BY NEWTON RAPHSON'S METHOD"<<endl;
cout<<"input initial value of x:"<<endl;
cin>>x0;
count=1;
begin:
fx=F(x0);
fdx=FD(x0);
xn=x0-fx/fdx;
if (fabs((xn-x0)/xn) < EPS) {
cout<<"root="<<xn<<endl;
   fxn =F(xn);
   cout<<"function value="<<fxn<<endl;
   cout<<"no. of iterations="<<count<<endl;
}
else
{ x0=xn;
   count=count+1;
   if(count<MAXIT)
   {  goto begin;
   }
   else
   { cout<<"SOLUTION DOESNOT CONVERGE."<<endl;
      cout<<"iterations="<<MAXIT<<endl;
   }
}
```

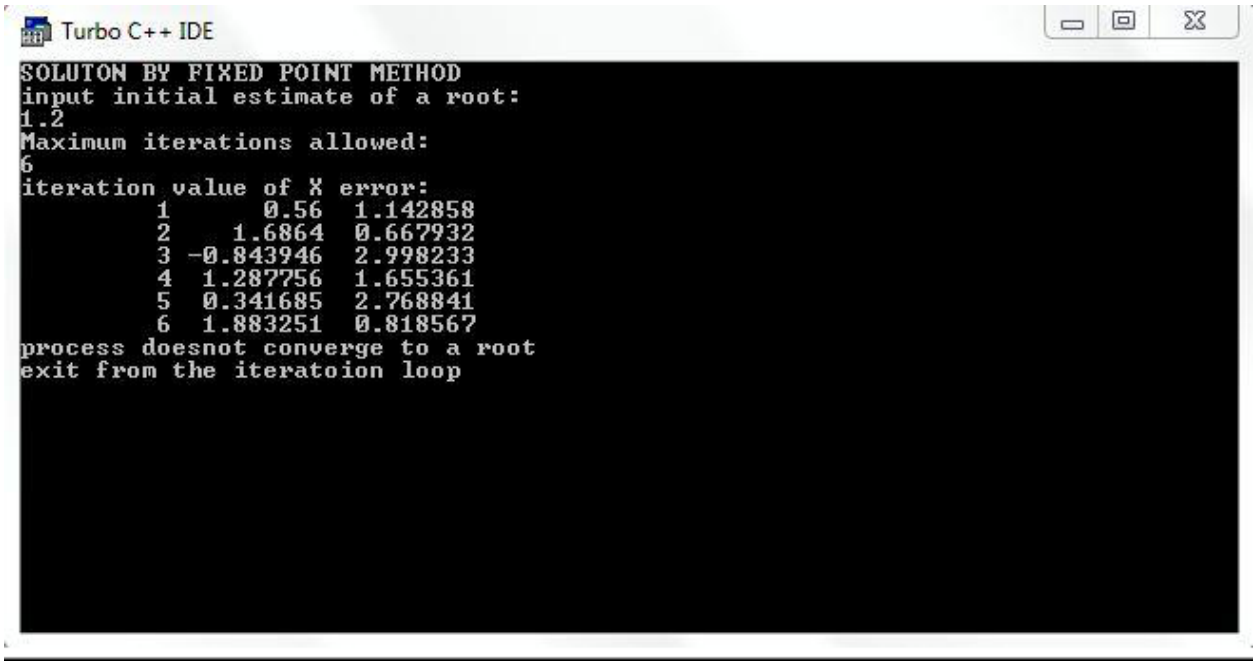
Output of Newton-Raphson Method

Fixed-point Iteration Method

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
#include<math.h>
#define EPS 0.000001
#define G(x) 2.0-(x)*(x)
void main()
{  clrscr();
   int MAXIT, i; float
   x0, x, error;
   cout<<"SOLUTON BY FIXED POINT METHOD"<<endl;
   cout<<"input initial estimate of a root:"<<endl; cin>>x0;
   cout<<"Maximum iterations allowed:"<<endl;
   cin>>MAXIT;
   cout<<"iteration value of X error:"<<endl;
   for(i=1; i<=MAXIT; i++)
```

```
  { x = G(x0); error=fabs((x-x0)/x);
     cout<<setw(10)<<i<<setw(10)<<x<<setw(10.8)<<error<<endl;
     if(error < EPS)
   goto end;

     else

     x0=x;

   }

   cout<<"process does not converge to a root"<<endl;
   cout<<"exit from the iteration loop"<<endl;
  end:
  getch();

}
```

Output of Fixed-point Iteration Method

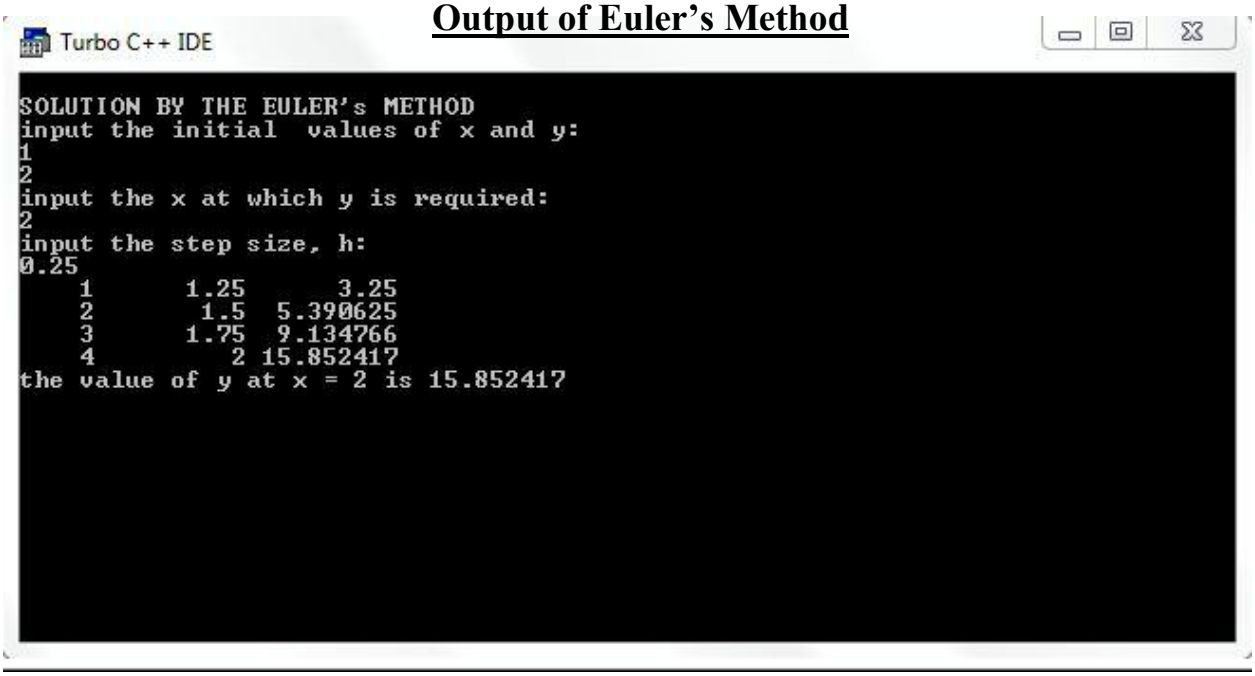


Euler's Method

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<iomanip.h>
float func(float x, float y);
void main()
{ clrscr();
   int i, n;
   float x, y, xp, h, dy; float
   func(float, float);
   cout<<endl;
   cout<<"SOLUTION BY THE EULER's METHOD"<<endl;
   cout<<"input the initial values of x and y:"<<endl;
   cin>>x>>y;
   cout<<"input the x at which y is required:"<<endl;
   cin>>xp;
   cout<<"input the step size, h:"<<endl;
   cin>>h;
   n = (int) ((xp-x)/h + 0.5);
   for(i=1; i<=n; i++)
   {  dy = h * func(x, y);
      x = x + h;
      y = y + dy;
      cout<<setw(5)<<i<<setw(10.6)<<x<<setw(10.6)<<y<<endl;
   }
   cout<<"the value of y at x = "<<x<<" is "<<y<<endl;
   getch();
   clrscr();
}
```

```
float func(float x, float y)
{  float f;
   f = x + y + x*y;
   return (f);
}
```

Output of Euler's Method

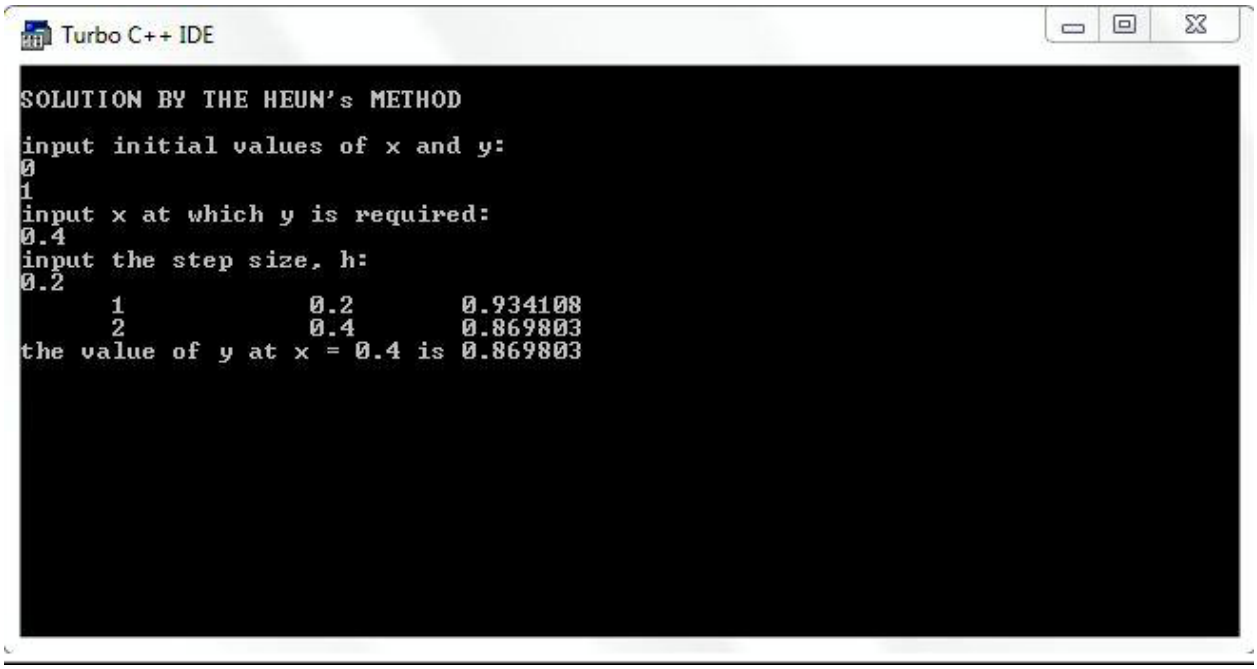


Heun's Method

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<iomanip.h>
float func(float x, float y);
void main()
{ clrscr();
   int i, n;
   float x, y, xp, h, m1, m2;
   float func(float, float);
   cout<<endl;
   cout<<"SOLUTION BY THE HEUN's METHOD"<<endl;
   cout<<endl;
   cout<<"input initial values of x and y:"<<endl;
   cin>>x>>y;
   cout<<"input x at which y is required:"<<endl;
   cin>>xp;
   cout<<"input the step size, h:"<<endl;
   cin>>h;
   n = (int) ((xp - x)/h + 0.5);
   for(i=1; i<=n; i++)
   {  m1 = func(x, y);
      m2 = func(x+h, y+m1*h);
      x  = x + h;
      y = y + 0.5 * h * (m1 + m2);
      cout<<setw(7)<<i<<setw(15.9)<<x<<setw(15.9)<<y<<endl;
   }
   cout<<"the value of y at x = "<<x<<" is "<<y;
   getch();
```

```
    clrscr();
}
float func(float x, float y)
{  float f;
    f = -y/(2*y + 1);
    return(f);
}
```

Output of Heun's Method



Runge-Kutta Method

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<iomanip.h>
float func(float x, float y);
void main()
{
clrscr();
   inti, n;
   float x, y, xp, h, m1, m2, m3, m4; float func(float, float); cout<<endl;
   cout<<"*******************************************"<<endl;
   cout<<"SOLUTION BY 4th ORDER RUNGE- KUTTA METHOD:"<<endl;
   cout<<"*******************************************"<<endl;
   cout<<"input initial values of x and y:"<<endl;
   cin>>x>>y;
   cout<<"input x at which y is required:"<<endl;
   cin>>xp;
   cout<<"input step size, h:"<<endl;
   cin>>h;
   n = (int) ((xp - x)/h + 0.5);
   cout<<endl;
   cout<<"-----------------------------------"<<endl;
   cout<<setw(5)<<"STEP"<<setw(15.9)<<"X"<<setw(15.9)<<"Y"<<endl;
   cout<<"-----------------------------------"<<endl;
   for(i=1; i<=n; i++)
   {  m1 = func(x, y);
      m2 = func(x + 0.5*h, y + 0.5*m1*h);
      m3 = func(x + 0.5*h, y + 0.5*m2*h);
      m4 = func(x+h, y + m3*h);
```

```
    x = x + h;
    y = y + (m1 + 2.0*m2 + 2.0*m3 + m4) * h/6.0;
    cout<<setw(5)<<i<<setw(15.9)<<x<<setw(15.9)<<y<<endl;
  }
  cout<<"the value of y at x = "<<x<<" is "<<y<<endl;
  getch();
  clrscr();
}
float func(float x, float y)
{  float f;
  f = y + sqrt(y);
  return(f);
}
```

Output of Runge-Kutta Method