THINK INDIA JOURNAL

ISSN:0971-1260
Vol-22- Issue-14-December-2019

The Comparison Of Numerical And Coding Solution For Initial Value Problems

Sasikala J[1], Praveenkumar K[2], Shiva Shankar C[3], Pavankumar S[4], Swathi M.S[5]

[1]Assistant Professor , Department of Science and Humanities, Sri Sairam College of Engineering, Anekal .

[2345]UG Scholars, Department of Computer Science & Engineering, Sri Sairam College of Engineering, Bengaluru.

ABSTRACT

In this paper presents about the comparison of Numerical and coding method for three method Taylors, Euler's modified method, Milne's and Adams Bash fourth method of solving initial value problems, In that we can conclude the better solution of Initial value problems.

Keywords: Coding, Taylors, Euler's modified method, Milne's and Adams Bash fourth method.

I. INTRODUCTION

Earlier, when we have the finite function and finite data to solve the initial value problem many numerical methods has followed. In this paper we like to discuss four methods of solving numerical methods with some example in numerical and coding, and we can get the comparison of the all the four results.

II. EXAMPLE

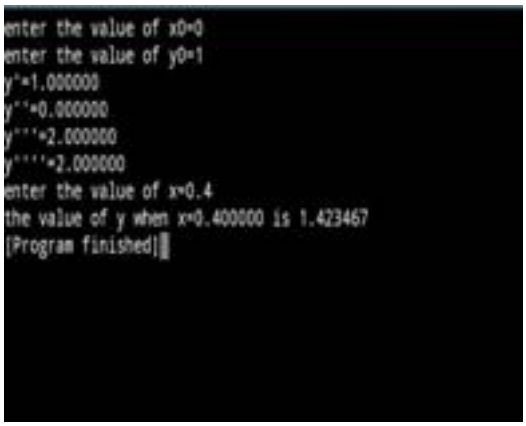
1. Using Taylor's series method, find the third order approximate solution at x = 0.4of the problem $\frac{dy}{dx} = x^2 y + 1$, with $y(0) = 0$ consider terms up to fourth degree.

Taylor series Method:

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
void main()
{
float x0,y0,x,y1,y2,y3,y4,y,h;
clrscr();
printf("enterthe value of x0=");
scanf("%f",&x0);
printf("enterthe value of y0=");
scanf("%f",&y0);
y1=x0*x0*y0+1;
printf("y'=%f\n",y1);
y2=x0*x0*y1+2*x0*y0;
printf("y"=%f\n",y2);
y3=2*x0*y1+2*x0*x0*y2+2*x0*y1+2*y1;
printf("y'"=%f\n",y3);
y4=2*x0*y2+2*y1+x0*x0*y3+y2*2*x0+2*x0*
y2+2*y2;
printf("y""=%f\n",y4);
printf("enterthe value of x=");
scanf("%f",&x);
h=x-x0;
y=y0+(y1*h)+(y2*pow(h,2))/2+(y3*pow(h,3))/6
+(y4*pow(h,4))/24;
printf("the value of y when x=%f is %f",x,y);
getch();
}
```

Output:

ISSN:0971-1260
Vol-22- Issue-14-December-2019

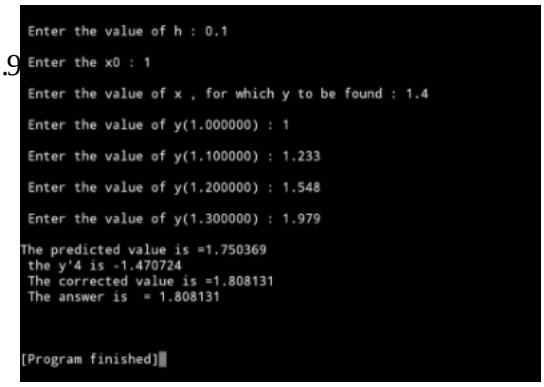


2. Given that $\frac{dy}{dx} = x^2(1-y)$ $and\ y(1) = 1; y(1.1) = 1.233; y(1.2) = 1.548; y(1.3) = 1.9$ Find y at x = 1.4 using Adams –Bash forth predictor and corrector formula. Compute y (0.8)

Adams Bash fourth Method:

```
#include<stdio.h>
int main()
{
float h=0,y[10],x[10],fx=0,hd=0,yd[10];
int i=0;
printf("\n Enter the value of h: ");
scanf("%f",&h);
printf("\n Enter the x0 : ");
scanf("%f",&hd);
printf("\n Enter the value of x, for which y to be
found : ");
scanf("%f",&fx);
x[0]=hd;
while(hd<fx){
hd=hd+h;
i++;
}
for(int j=1;j<=i;j++)
x[j]=x[j-1]+h;
for(int j=0;j<i;j++)
{
printf("\n Enter the value of y(%f) : ",x[j]);
scanf("%f",&y[j]);
}
for(int j=0;j<i;j++)
yd[j]=(x[j]*x[j]-x[j]*x[j]*y[j]);
y[4]=y[3]+((h/24)*((55*yd[3])-
(59*yd[2])+(37*yd[1])-(9*yd[0])));
printf("\nThe predicted value is =%f",y[4]);
yd[4]=(x[4]*x[4])*(1+y[4]);
y[4]=y[3]+((h/24)*((9*yd[4])+(19*yd[3])-
(5*yd[2])+yd[1]));
printf("\nThe corrected value is =%f",y[4]);
printf("\n The answer is  = %f ",y[4]);
printf("\n\n\n");
}
```

Output:



3. Given $\frac{dy}{dx} = xy + y^2$, y(0) =1, y(0.1) =1.1169,y(0.2) = 1.2773 ,y(0.3) = 1.5049 find y(0.4) correct to three decimal places, using Milne's Predictor corrector method.

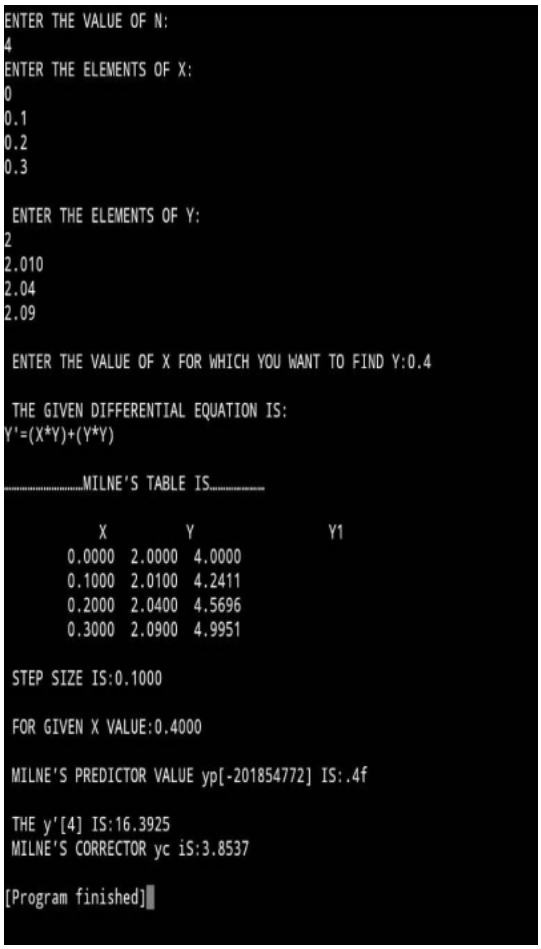
Milne's Method:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int i=1,n;
float f,h;
float x[100],y[100];
float y1[100],yp[100],yc[100];
printf("ENTER THE VALUE OF N:\n");
scanf("%d", &n);
printf("ENTER THE ELEMENTS OF X:\n");
```

ISSN:0971-1260
Vol-22- Issue-14-December-2019

```
for(i=1;i<=n;i++)
scanf("%f",&x[i]);
printf("\n ENTER THE ELEMENTS OF Y:\n");
for(i=1;i<=n;i++)
{
scanf("%f",&y[i]);
}
h=x[2]-x[1];
printf("\n ENTER THE VALUE OF X FOR
WHICH YOU WANT TO FIND Y:");
scanf("%f", &f);
printf("\n THE GIVEN DIFFERENTIAL
EQUATION IS:\n");
printf("Y'=(X*Y)+(Y*Y)\n");
for(i=1;i<=n;i++)
{
y1[i]=((x[i]*y[i])+(y[i]*y[i]));
}
printf("\n……..……..……..……..…….MILNE'S
TABLE IS…..……..……..……\n");
printf("\n\t  X\t    Y\t    Y1\n");
for(i=1;i<=n;i++)
{
printf("\t%.4f\t%.4f\t%.4f\n",x[i],y[i],y1[i]);
}
printf("\n STEP SIZE IS:%.4f\n",h);
printf("\n FOR GIVEN X VALUE:%.4f\n",f);
printf("\n MILNE'S PREDICTOR VALUE
yp[%d] IS:",yp);
for(i=1;i<=n;i++)
{
yp[n+1]=y[n-3]+((4*h)/3)*((2*y1[n-2])-(y1[n-
1])+(2*y1[n]));
printf(".4f\n",yp[n+1]);
break;
}
for(i=1;i<=n;i++)
{
y1[n+1]=yp[n+1]*(f+yp[n+1]);
printf("\n THE y'[%d] IS:%.4f\n",n,y1[n+1]);
break;
}
printf(" MILNE'S CORRECTOR yc iS:",yc);
for(i=1;i<=n;i++)
{
yc[n+1]=y[n-1]+((h/3)*((y1[n-
1])+(4*y1[n])+(y1[n+1])));
printf("%.4f\n",yp[n+1]);
break;
}
}
```

Output:



4. Solve the following by Euler's modified method $\frac{dy}{dx} = \log(x+y) y(0) = 2$ to find $y(0.4)$ by taking $h=0.2$

Euler's Modified Method:

```
#include<stdio.h>
#include <math.h>
#define F(x,y) (log(x+y))
void main()
{
 double y0,x0,y1,x1,y1_0,a,n,h,f,f1;
```

ISSN:0971-1260
Vol-22- Issue-14-December-2019

```
int j,count,flag;
printf("\nEnter the value of x0: ");
scanf("%lf",&x0);
printf("\nEnter the value of y0: ");
scanf("%lf",&y0);
printf("\nEnter the value of h: ");
scanf("%lf",&h);
printf("\nEnter the value of last point: ");
scanf("%lf",&n);
for(x1=x0+h,j=1; x1<=n+h; x1=x1+h,j++)
{
count=0;
flag=0;
f=F(x0,y0);
y1_0 = y0 + (h * f);
printf("\n\n * * y%d_0 = %.3lf * *",j,y1_0);
do
{
count++;
f=F(x0,y0);
f1=F(x1,y1_0);
y1 =y0 + h/2 * ( f + f1);
printf("\n\n * * x = %.3lf => y%d_%d = %.3lf *
*",x1,j,count,y1);
if(fabs(y1-y1_0)<0.00001)
{
printf("\n\n\n\n * * * * y%d = %.3lf * * *
*\n\n",j,y1);
flag=1;
}
else
y1_0 = y1;
}while(flag!=1);
y0 =y1;
}
}
```

```
Enter the value of x0: 0

Enter the value of y0: 2

Enter the value of h: 0.2

Enter the value of last point: 0.4


* * y1_0 = 2.139 * *

* * x = 0.200 => y1_1 = 2.154 * *

* * x = 0.200 => y1_2 = 2.155 * *

* * x = 0.200 => y1_3 = 2.155 * *

* * x = 0.200 => y1_4 = 2.155 * *


* * * * y1 = 2.155 * * * *


* * y2_0 = 2.309 * *

* * x = 0.400 => y2_1 = 2.331 * *

* * x = 0.400 => y2_2 = 2.332 * *

* * x = 0.400 => y2_3 = 2.332 * *

* * x = 0.400 => y2_4 = 2.332 * *


* * * * y2 = 2.332 * * * *


* * y3_0 = 2.502 * *

* * x = 0.600 => y3_1 = 2.530 * *

* * x = 0.600 => y3_3 = 2.531 * *

* * x = 0.600 => y3_4 = 2.531 * *


* * * * y3 = 2.531 * * * *


[Program finished]
```

Output:

ISSN:0971-1260
Vol-22- Issue-14-December-2019

III. CONCLUSION

Comparison of all the four method of numerical results and coding results

S.No	Name of the Method	Solutions	
		Numerical Methods	Coding
1	Taylor's series	1.42151	1.423467
2	Adams Bash fourth	1.7940	1.808131
3	Milne Method	1.8346	1.83909
4	Euler's method	2.1414	2.5310

*** Numerical methods solved by using the respective formula.**

The conclusion of this paper, by using the coding method we can reduce the computing the cost and we will get the better approximate results than the numerical methods.

IV. REFERENCES

1. Numerical Methods for Initial Value Problems in Ordinary Differential Equations by Simeon Ola Fatunla

2. Numerical Methods for Ordinary Differential Equations: Initial Value Problems By David F. Griffiths, Desmond J. Higham

3. An Open Source C++ Library for Computational Physiology and Biology by Gary R. Mirams1, Christopher J. Arthurs1, Miguel O. Bernabeu2,3, Rafel Bordas1, Jonathan Cooper1, Alberto Corrias4, Yohan Davit5, Sara-Jane Dunn6, Alexander G. Fletcher7, Daniel G. Harvey1 , Megan E. Marsh8, James M. Osborne1, Pras Pathmanathan1,9, Joe Pitt-Francis1, James Southern10, Nejib Zemzemi11, David J. Gavaghan1, March 2013 | Volume 9 | Issue 3 |

4. A Book on C; Programming in C by Al I Kelley, Ira Pohl