Name : Deepankar Sharma

Course: BCA

University Roll No: 2092014

Student Id : 20041299

Semester: 3

Date: January 7, 2022

**Practical 15: Runge-Kutta Method (**Fourth Order**)**

**Objective:** To find the value of y=f(x) at a given value of x for given differential equation for a given value of y0(=f(x0)), and h(size of interval).

**2. Algorithm:**

1. Start
2. Define the function f(x)
3. Input y0.
4. Input h, size of intervals.
5. Input x\_unk, Unknown x
6. Calculate n=(x\_unk- x0)/n
7. For i=0 to n:

y[i+1]= y[i]+ k(i)

x[i+1]= x[i]+ h

1. Print y[n-1]
2. **f1(n):**

k1 = f(x[n], y[n])

k1\*=h

Return k1

1. **f2(n):**

k2 = f(x[n]+h/2, y[n]+k1/2)

k2\*=h

Return k2

1. **f3(n):**

k3 = f(x[n]+h/2, y[n]+k2/2)

k3\*=h

Return k3

1. **f4(n):**

k4 = f(x[n]+h, y[n]+k3)

k4\*=h

Return k4

1. **k(n):**

k = k1+2k2+2k3+k4

k\*=(1/6)

Return k

1. Stop.

**Code:**

#include <iostream>

#include<math.h>

#define f(x, y) (2\*y) + (3\*exp(x))

// #define f(x, y) x+ (y\*y)

// float y0, x0=0;

float h; // gap in intervals

static float x[10]= {0.0}, y[10]= {0.0};

using namespace std;

float k1(int n){

float k1 = f(x[n], y[n]);

k1\*=h;

// cout<<"h: "<<h<<endl;

// cout<<"xn: "<<x[n]<<endl;

// cout<<"yn: "<<y[n]<<endl;

// cout << "fn: " << f(x[n], y[n]) << endl;

// cout<<"k1: "<<k1<<endl;

return k1;

}

float k2(int n){

float a= x[n]+ (h/2);

float b= y[n]+ (k1(n)/2);

float k2 = f(a, b);

k2 \*= h;

// cout << "h: " << h << endl;

// cout << "a: " << a << endl;

// cout << "b: " << b << endl;

// cout << "fn: " << f(a, b) << endl;

// cout << "k2: " << k2 << endl;

return k2;

}

float k3(int n){

float a= x[n]+ (h/2);

float b= y[n]+ (k2(n)/2);

float k3 = f(a, b);

k3 = h \* k3;

// cout << "h: " << h << endl;

// cout << "a: " << a << endl;

// cout << "b: " << b << endl;

// cout << "fn: " << f(a, b) << endl;

// cout << "k2: " << k3 << endl;

return k3;

}

float k4(int n){

float a= x[n]+ (h);

float b= y[n]+ (k3(n));

float k4= f(a, b);

k4= k4\*h;

// cout << "h: " << h << endl;

// cout << "a: " << a << endl;

// cout << "b: " << b << endl;

// cout << "fn: " << f(a, b) << endl;

// cout << "k4: " << k4 << endl;

return k4;

}

float k(int n){

float k= k1(n)+ 2\*k2(n)+ 2\*k3(n)+ k4(n);

k/=6;

printf("k[%d]: %.6f\n", n, k);

// cout << "k: " << k << endl;

return k;

}

int main()

{

int n=0; // number of iterations

float x\_unk, y\_unk=0; // xUknown, yUknown

cout << "Enter the y(x0): ";

cin >> y[0];

// y[0]= 1;

cout << "Enter the gap in intervals (h): ";

cin >> h;

// h= .1;

cout << "Enter the value of x at which you want to approximate y(x): ";

cin >> x\_unk;

// x\_unk= .2;

n= (x\_unk-x[0])/h;

// k(0);

for (int i = 0; i < n; i++)

{

/\* code \*/

printf("at n=%d, \n", i);

// cout<< k(n)<<endl;

y[i+1]= y[i]+ k(i);

x[i+1]= x[i]+ h;

printf("y[%d]: %.6f", i+1, y[i+1]);

y\_unk= y[i+1];

cout<<endl<<endl;

}

cout<< "The value of y(" << x\_unk << "): "<< y\_unk<<endl;

return 0;

}

**Output:**

Windows PowerShell

Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS E:\03 Semester\CBNST\Unit 04> cd "e:\03 Semester\CBNST\Unit 04\" ; if ($?) { g++ 15\_RungeKutta\_4th\_Order.cpp -o 15\_RungeKutta\_4th\_Order } ; if ($?) { .\15\_RungeKutta\_4th\_Order }

Enter the y(x0): 0

Enter the gap in intervals (h): 0.05

Enter the value of x at which you want to approximate y(x): 0.2

at n=0,

k[0]: 0.161699

y[1]: 0.161699

at n=1,

k[1]: 0.186996

y[2]: 0.348695

at n=2,

k[2]: 0.215378

y[3]: 0.564073

at n=3,

k[3]: 0.247192

y[4]: 0.811265

The value of y(0.2): 0.811265