**Rajshahi University of Engineering & Technology**

CSE 2104: Sessional Based on CSE 2103

Lab Report 10

Dated: 25.06.18

Submitted to

Shyla Afroge

Instructor, CSE 2104

Assistant Professor

Dept. of Computer Science & Engineering

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of Computer Science & Engineering

Problem#01: Numerical Solution of Ordinary Differential Equation by Runge Kutta 2nd order Method

**Theory:** Runge-Kutta methods achieve the accuracy of a Taylor series approach without requiring the calculation of higher derivatives. The formulas that are used listed here.

**k1 = h \* f(x0, y0)**

**k2 = h \* f(x0 + h, y0 + k1)**

**Solution:**

#include <iostream>

using namespace std;

double dFunc(double x, double y) {

return x - y \* y;

}

int main() {

double Xo, Yo, Y1, h, X;

double k1, k2;

cin >> Xo >> Yo >> X;

while(1) {

cin >> h;

while(Xo != X) {

k1 = h \* dFunc(Xo, Yo);

k2 = h \* dFunc(Xo + h, Yo + k1);

Xo = Xo + h;

Y1 = Yo + (k1 + k2) / 2.0;

}

cout << Y1 << endl;

}

}

OUTPUT:

0 1 .1

.01

0.9145

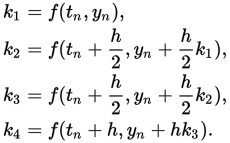
.0001

0.9021

**Discussion:** As the value of h decreased, the result becomes more accurate. The program gets the value of h and calculates the result until it is killed manually.

Problem#02: Numerical Solution of Ordinary Differential Equation by Runge Kutta 4th order Method

**Theory:** Runge-Kutta methods achieve the accuracy of a Taylor series approach without requiring the calculation of higher derivatives. The formulas that are used listed here.



**Solution:**

#include <iostream>

using namespace std;

double dFunc(double x, double y) {

return x - y \* y;

}

int main() {

double Xo, Yo, Y1, h, X;

double k1, k2, k3, k4;

cin >> Xo >> Yo >> X;

while(1) {

cin >> h;

while(Xo != X) {

k1 = h \* dFunc(Xo, Yo);

k2 = h \* dFunc((Xo + h/2.0), (Yo + k1/2.0));

k3 = h \* dFunc((Xo + h/2.0), (Yo + k2/2.0));

k4 = h \* dFunc((Xo + h), (Yo + k3));

Xo = Xo + h;

Y1 = Yo + (k1 + 2.0 \* k2 + 2.0 \* k3 + k4) / 6.0;

}

cout << Y1 << endl;

}

}

OUTPUT:

0 1 .1

.1

0.913795

.0001

0.902066

**Discussion:** As the value of h decreased, the result becomes more accurate. The program gets the value of h and calculates the result until it is killed manually.