СПб НИУ ИТМО

кафедра ИПМ

Вычислительная математика

Лабораторная работа № 4

Решение обыкновенных дифференциальных уравнений методом Рунге – Кутта

Работу выполнил:

Студент II курса

Группы № 2120

Журавлев Виталий

Санкт-Петербург

2013 г.

**Цель работы:**

Составить подпрограмму для решения системы ОДУ порядка n (n<=5), используя метод Рунге – Кутта 4-го порядка.

Вычисление правых частей реализовать отдельной подпрограммой.

Найти решение заданной системы уравнений, получить оценку Рунге, используя двойной просчет, построить график решения.

**Описание метода:**

**Метод Рунге-Кутты 4-го порядка**

Метод Рунге — Кутты 4 порядка столь широко распространён, что его часто называют просто методом Рунге — Кутты.

Рассмотрим [задачу Коши](http://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%9A%D0%BE%D1%88%D0%B8)

\textbf{y}'=\textbf{f}(x,\textbf{y}),   \textbf{y}(x_0)=\textbf{y}_0.

Нам необходимо решить ее (интегрировать) на отрезке [ab](http://coll3ctor.ru/metod-ejlera-c/ab), методом Рунге-Кутта, соответственно.

Приближенное значение в последующих точках вычисляется по итерационной формуле:

 \textbf{y}_{n+1} = \textbf{y}_n + {1 \over 6} (\textbf{k}_1 + 2\textbf{k}_2 + 2\textbf{k}_3 + \textbf{k}_4) 

Вычисление нового значения проходит в четыре стадии:

 \textbf{k}_1 = h\ \textbf{f} \left( x_n, \textbf{y}_n \right), 

 \textbf{k}_2 = h\ \textbf{f} \left( x_n + {h \over 2}, \textbf{y}_n + {1 \over 2} \textbf{k}_1 \right), 

 \textbf{k}_3 = h\ \textbf{f} \left( x_n + {h \over 2}, \textbf{y}_n + {1 \over 2} \textbf{k}_2 \right), 

 \textbf{k}_4 = h\ \textbf{f} \left( x_n + h, \textbf{y}_n + \textbf{k}_3 \right). 

где h — величина шага сетки по х .

Этот метод имеет четвёртый порядок точности, то есть суммарная ошибка на конечном интервале интегрирования имеет порядок O(h^4) (ошибка на каждом шаге порядка O(h^5))

**Алгоритм:**

1.  Получаем исходные данные:

* Дифференциальное уравнение;
* Начальные условия;
* Отрезок, на котором будем решать;
* Шаг, с которым будем интегрировать.

2. Вычисляем количество точек:

Количество точек = длина отрезка / шаг: [countdots](http://coll3ctor.ru/metod-ejlera-c/countdots)

3. Создаем массив точек *X*. Для этого мы заполняем его в интервале от *a* до *b* с шагом в *h.*Первый элемент массива*X* - это начало отрезка (т.е.*X[0]=a*);

Далее, заполним массив [x_i](http://coll3ctor.ru/metod-ejlera-c/x_i)

4. Присваиваем первому элементу массива*Y*, начальное условие в ординате [y_0](http://coll3ctor.ru/metod-ejlera-c/y_0)

5. Решаем дифференциальное уравнение

6. После окончания заполнения массива *Y* у нас есть два готовых массива, по которым мы строим график, а следовательно получаем решение.

**Код метода вычисления:**

private void button\_solve\_Click(object sender, EventArgs e)

{

readVariables();

countDots = Convert.ToInt32(((b - a) / h));

X = new Massive(countDots);

Y = new Massive(countDots);

X[0] = a;

for (int i = 0; i < countDots - 1; ++i)

{

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

if (X[i] == b) break;

}

Y[0] = y0;

double k1 = 0.0, k2 = 0.0, k3 = 0.0, k4 = 0.0;

for (int i = 0; i < countDots - 1; ++i)

{

k1 = h \* f(X[i], Y[i]);

k2 = h \* f(X[i] + h / 2, Y[i] + 1 / 2 \* k1);

k3 = h \* f(X[i] + h / 2, Y[i] + 1 / 2 \* k2);

k4 = h \* f(X[i] + h, Y[i] + k3);

Y[i + 1] = Y[i] + (k1 + (2 \* k2) + (2 \* k3) + k4) / 6;

}

GraphPane pane = zedGraph.GraphPane;

pane.CurveList.Clear();

PointPairList list = new PointPairList();

for (int j = 0; j < countDots; ++j)

{

list.Add(X[j], Y[j]);

}

LineItem myCurve = pane.AddCurve("", list, Color.Green, SymbolType.None);

zedGraph.AxisChange();

zedGraph.Invalidate();

}

**Вывод:**

В процессе выполнения лабораторной работы был рассмотрен Метод Рунге-Кутта 4 порядка точности для решения обыкновенных дифференциальных уравнений.