МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Нижегородский государственный университет им. Н.И. Лобачевского»
Национальный исследовательский университет

Институт информационных технологий, математики и механики
Кафедра математического обеспечения и суперкомпьютерных технологий

ЛАБОРАТОРНАЯ РАБОТА
«Численное решение задачи Коши для ОДУ»

Выполнил:
Студент группы 381706-2
Гущин Александр Владимирович
___________ Подпись

Руководитель:
Ассистент кафедры ДУМЧА
Морозов Кирилл Евгеньевич
___________ Подпись

Нижний Новгород
2019

Оглавление

Введение

Дифференциальное уравнение — уравнение, в которое входят производные функции, и может входить сама функция, независимая переменная и параметры. Порядок входящих в уравнение производных может быть различен (формально он ничем не ограничен). Производные, функции, независимые переменные и параметры могут входить в уравнение в различных комбинациях или могут отсутствовать вовсе, кроме хотя бы одной производной.

В отличие от алгебраических уравнений, в результате решения которых ищется число (несколько чисел), при решении дифференциальных уравнений ищется функция (семейство функций).

Дифференциальное уравнение порядка выше первого можно преобразовать в систему уравнений первого порядка, в которой число уравнений равно порядку исходного дифференциального уравнения.

Выбор модели

Для этой лабораторной работы я выбрал уравнение маятника с диссипацией y'' = -a * y' – sin(y).

Выбор языка

В этой лабораторной работе я использовал C# с Forms для облегчения визуального представления.

Выбор метода

Я использовал метод Рунге-Кутта для обыкновенного дифференциального уравнения второго порядка. Рассмотрим уравнение второго порядка, разрешенное относительно второй производной: $y'' = f(x, y, y')$ на отрезке [a, b] с начальными условиями $y(a) = y_0,\ y'(a) = y_0^1$. Это уравнение легко свести к системе уравнений первого порядка с помощью замены переменных: $y' = z$. Рассмотрим систему 2 уравнений. 1-го порядка

$$\begin{cases} y' = f_1(x, y, z) \\ z' = f_2(x, y, z) \end{cases}$$

на отрезке [a, b] с начальными условиями $y(a) = y_0$, $z(a) = z_0$. Напишем формулы метода Рунге – Кутта вначале для системы двух уравнений:

$$y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)/6$$

$$z_{i+1} = z_i + (l_1 + 2l_2 + 2l_3 + l_4)/6,$$

Где

$$k_1 = h \cdot f_1(x_i, y_i, z_i)\ ,\ l_1 = h \cdot f_2(x_i, y_i, z_i),$$
$$k_2 = h \cdot f(x_i + h/2, y_i + k_1/2, z_i + l_1/2),$$
$$l_2 = h \cdot f(x_i + h/2, y_i + k_1/2, z_i + l_1/2),$$
$$k_3 = h \cdot f_1(x_i + h/2, y_i + k_2/2, z_i + l_2/2)\ ,$$
$$l_3 = h \cdot f_2(x_i + h/2, y_i + k_2/2, z_i + l_2/2),$$
$$k_4 = h \cdot f_1(x_i + h, y_i + k_3, z_i + l_3)\ ,$$
$$l_4 = h \cdot f(x_i + h, y_i + k_3, z_i + l_3).$$

Вычисляются на каждом шаге.

Руководство пользователя

Left border – левая граница

Right border – правая граница

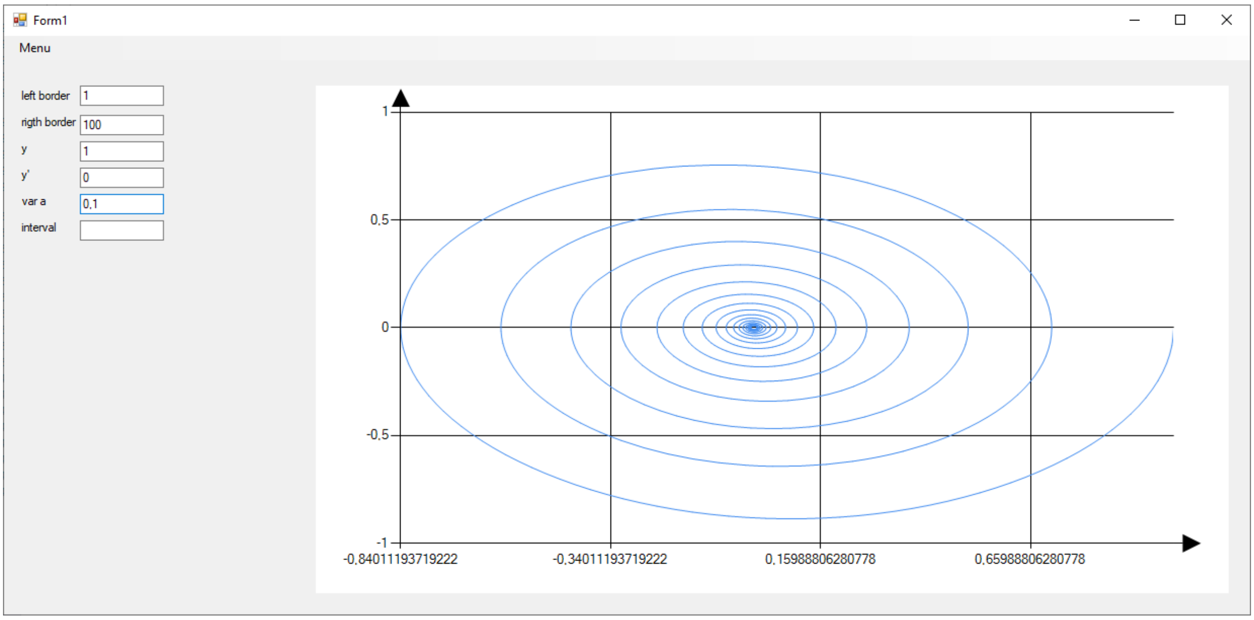
y и y’ для задачи Коши

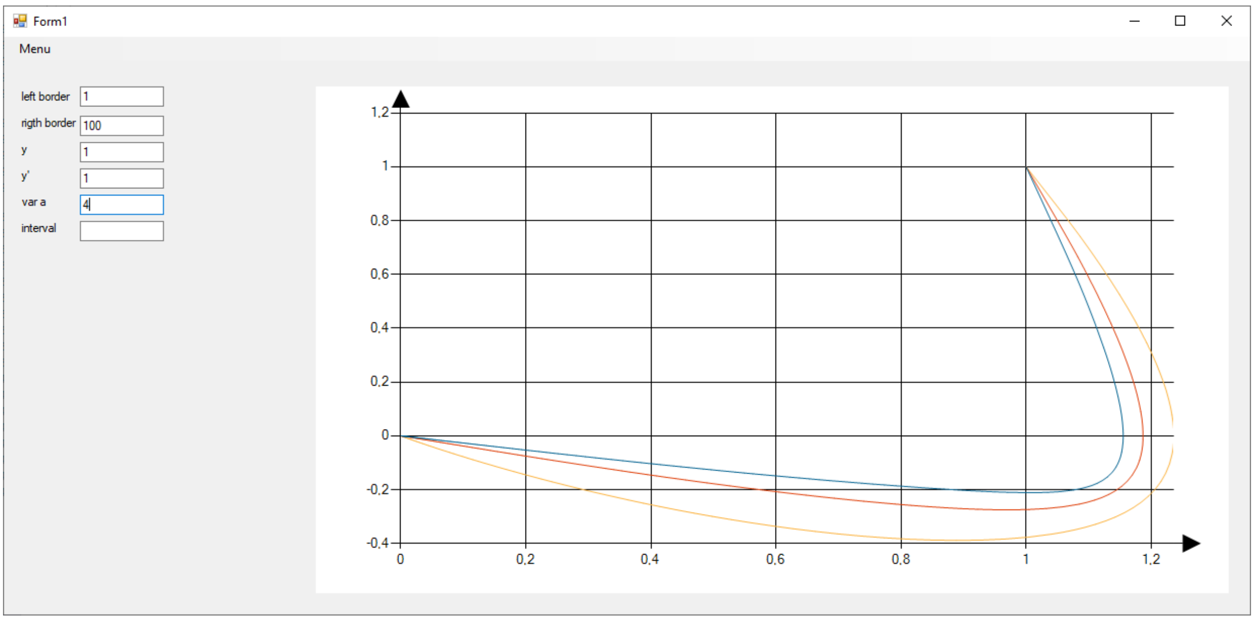
a – переменная

interval – интервал(по умолчанию стоит 0,1)

Функции

1. Generate – строит график
2. Clear – очищает график
3. Runge_Kutt(x[0], end_x, h) – для вычисления точек





Вывод

В ходе данной работы я укрепил свои знания дифференциальных уравнений и улучшил свое понимание C#.

Список использованной литературы

1. Википедия. https://en.wikipedia.org/wiki/Cauchy_problem
2. Теоретические сведения о методах численного решения обыкновенных дифференциальных уравнений https://vunivere.ru/work4559/page3

Приложение

```
void Runge_Kutt(double t0, double t1, double h) {
      int i = 0;
      while (t0 < t1 - h) {
        double k1 = h * z[i];
        double l1 = h * func(a, x[i], y[i], z[i]);
        double k2 = h * (z[i] + l1 / 2);
        double l2 = h * func(a, x[i] + h /2, y[i] + k1 / 2, z[i] + l1 / 2);
        double k3 = h * (z[i] + l2 / 2);
        double l3 = h * func(a, x[i] + h /2, y[i] + k2 / 2, z[i] + l2 / 2);
        double k4 = h * (z[i] + l3);
        double l4 = h * func(a, x[i] + h, y[i] + k3, z[i] + l3);
        x.Add(x[i] + h);
        y.Add(y[i] + (k1 + 2 * k2 + 2 * k3 + k4) / 6);
        z.Add(z[i] + (l1 + 2 * l2 + 2 * l3 + l4) / 6);
        ++i;
        t0 += h;
      }
    }
```