

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
Московский государственный технический университет имени Н.Э. Баумана

Лабораторная работа №5
по курсу «Численные методы»
«Решение дифференциальных уравнений.
Метод Рунге-Кутты 4 порядка»

Выполнил:
студент группы ИУ9-62
Иванов Георгий

Проверила:
Домрачева А.Б.

Москва, 2017

Цель:

Анализ метода Рунге-Кутты 4 порядка для решения дифференциальных уравнений, обобщение метода для решения дифференциальных уравнений высших порядков (2 и выше).

Решение задачи Коши (частного решения) для дифференциального уравнения 2 порядка с помощью этого метода.

Постановка задачи:**Дано:**

1) Неоднородное дифференциальное уравнение 2 порядка с постоянными коэффициентами:

$$y''(x) + p(x)y'(x) + q(x)y(x) = f(x)$$

где $p(x)$ и $q(x)$ - постоянные функции, а $f(x)$ - функция, непрерывна на интервале интегрирования $[a; b]$.

2) Начальные условия задачи Коши:

$$y(a) = A, \quad y'(a) = B, \quad x \in [a; b]$$

Найти: Численное решение данного неоднородного линейного дифференциального уравнения 2 порядка при начальных условиях задачи Коши методом Рунге-Кутты.

Тестовый пример:

В нашем случае, выберем следующие значения:

$$y(x) = e^x, \quad p(x) = 1, \quad q(x) = -1$$

Дифференциальное уравнение примет вид:

$$y''(x) + y'(x) - y(x) = e^x$$

Определим начальные условия задачи Коши:

$$y(0) = e^0 = 1, \quad y'(0) = e^0 = 1, \quad x \in [0; 1]$$

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:

Методом Рунге-Кутты четвертого порядка точности называют одношаговый метод, относящийся к широкому классу методов Рунге-Кутты решения задачи Коши для обыкновенных дифференциальных уравнений и их систем.

Этот метод используют для точного расчета стандартных моделей достаточно часто, так как при небольшом объеме вычислений он обладает точностью метода $O^4(h)$, в отличие от стандартного метода Эйлера. Метод является явным, так как y_{n+1} находится по ранее найденным значениям. В зависимости от степени аппроксимации решение одной и той же задачи позволяет получать более точное решение при более крупном шаге, и приводит к снижению требуемых ресурсов ЭВМ при меньшем количестве шагов. Широко распространён и реализуем классический

метод Рунге-Кутты 4 порядка. Можно также повысить порядок вычисления, но это грозит к большим вычислительным трудностям.

Стоит отметить, что данный метод применим для дифференциальных уравнений 1 порядка, поэтому необходимо обобщить данный метод для уравнений высших порядков при помощи понижения порядка уравнения и переходом к системе дифференциальных уравнений 1 порядка.

Это можно сделать посредством замены переменных $y' = z$. Тогда получим систему:

$$\begin{cases} g(x, y, z) = y' = z \\ f(x, y, z) = z' = e^x + y'(x) - y(x) \end{cases}$$

с начальными условиями $y(a) = A, z(a) = B$ для первого и второго уравнения системы.

Описание алгоритма:

Приближенное значение, используемое в методе Рунге-Кутты 4 порядка, в следующих точках вычисляется по итерационной формуле:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

где $n = 0, \dots, N$ - число разбиений отрезка, y_n - старое значение, y_{n+1} - новое значение, а коэффициенты k_1, k_2, k_3, k_4 вычисляются следующим способом:

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 &= f(x_n + h, y_n + hk_3) \end{aligned}$$

где $f(x_n, y_n)$ - значение дифференциального уравнения 1 порядка в точке x_n , $h = \frac{b-a}{N}$ - длина шага.

ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ:

Листинг 1. Метод Рунге-Кутты 4 порядка для решения диф.уравнения

```
#!/python
# -*- coding: utf-8 -*-

import abc
import math

class RungeKutta(object):
    __metaclass__ = abc.ABCMeta
```

```

def __init__(self, t0, y0):
    self.t = t0
    self.y = y0
    self.yy = [0,0]
    self.y1 = [0,0]
    self.y2 = [0,0]
    self.y3 = [0,0]
    self.y4 = [0,0]

@abc.abstractclassmethod
def f(self, t, y):
    pass

def nextStep(self, dt):
    if dt < 0:
        return None

    self.y1 = self.f(self.t, self.y)
    for i in range(2):
        self.yy[i] = self.y[i] + self.y1[i] * (dt / 2.0)

    self.y2 = self.f(self.t + dt / 2.0, self.yy)
    for i in range(2):
        self.yy[i] = self.y[i] + self.y2[i] * (dt / 2.0)

    self.y3 = self.f(self.t + dt / 2.0, self.yy)
    for i in range(2):
        self.yy[i] = self.y[i] + self.y3[i] * dt

    self.y4 = self.f(self.t + dt, self.yy)
    for i in range(2):
        self.y[i] = self.y[i] + dt / 6.0 * \
(self.y1[i] + 2.0 * self.y2[i] + 2.0 * self.y3[i] + self.y4[i])

    self.t += dt

class RungeKuttaImpl(RungeKutta):
    def f(self, t, y):
        fy = [0,0]

```

```

        fy[0] = y[1]
        fy[1] = math.exp(t) + y[0] - y[1]
        return fy

def func(x):
    return math.exp(x)

if __name__ == "__main__":
    eq = RungeKuttaImpl(0, [1, 1])
    dt = 0.1
    while eq.t <= 1.00:
        print("Solution y'(% .2f) =" %(eq.t) + str(eq.y[1]) +
              " y(% .2f) =" %(eq.t) + str(eq.y[0]))
        eq.nextStep(dt)

```

Класс RungeKutta является абстрактным, а значит для использования его для решения системы дифференциальных уравнений 1 порядка необходимо переопределить правые части системы. RungeKutta.y - массив решений, где $y[0]$ - само решение, $y[1]$ - первая производная решения. RungeKutta.t - значение x_n на текущем шаге. Чтобы задать начальные условия задачи Коши, необходимо создать объект класса RungeKuttaImpl, причём вторым параметром конструктора является массив начальных значений условий задачи Коши. RungeKutta.nextStep(dt) - вычисление следующего значения решения системы дифференциального уравнения при $x_{i+1} = x_i + dt$

РЕЗУЛЬТАТЫ:

Для тестирования полученной программы было выбрано уравнение:

$$y''(x) + y'(x) - y(x) = e^x$$

с начальными условиями задачи Коши:

$$y(0) = e^0 = 1, \quad y'(0) = e^0 = 1, \quad x \in [0; 1]$$

Чтобы проверить точность метода, вычислим погрешность:

$$\varepsilon = y_n^* - y_n$$

где y_n^* - численное решение данного уравнения, y_n - полученное решение посредством метода Рунге-Кутты 4 порядка.

Ниже приведена таблица результата полученной программы для вычисления погрешности (Листинг 1) на указанном методе:

Значение x_n	Численное решение y_n^*	Полученное решение y_n	Погрешность ε
0	1	1.0	0.0
0.1	1.1051707977298888	1.1051709180756477	1.2034575891384236e-07
0.2	1.2214025180676706	1.2214027581601699	2.4009249921519427e-07
0.3	1.349858446117391	1.3498588075760032	3.6145861215253206e-07
0.4	1.4918242110670952	1.4918246976412703	4.865741751736152e-07
0.5	1.6487206531791918	1.6487212707001282	6.175209363856737e-07
0.6	1.8221180440212896	1.8221188003905089	7.563692192569249e-07
0.7	2.0137518022579552	2.0137527074704766	9.052125213848683e-07
0.8	2.2255398622919795	2.2255409284924674	1.0662004878980724e-06
0.9	2.459601869586057	2.4596031111569494	1.2415708923185775e-06
1.0	2.7182803947778686	2.718281828459045	1.433681176443713e-06

А также приведена таблица результата полученной программы для вычисления значений системы дифференциальных уравнений с начальными условиями задачи Коши (Листинг 1) на указанном методе:

Значение x_n	Решение y_n	Решение y'_n
0	1	1
0.1	1.1051707977298888	1.1051710632544716
0.2	1.2214025180676706	1.221403039179291
0.3	1.349858446117391	1.3498592176047473
0.4	1.4918242110670952	1.4918252320873335
0.5	1.6487206531791918	1.6487219269906563
0.6	1.8221180440212896	1.8221195777972221
0.7	2.0137518022579552	2.0137536069733986
0.8	2.2255398622919795	2.225541952677753
0.9	2.459601869586057	2.4596042641450997
1.0	2.7182803947778686	2.7182831158605034

Выводы:

В ходе выполнения лабораторной работы был рассмотрен метод решения дифференциальных уравнений, а именно классический метод Рунге-Кутты 4 порядка. Была написана реализация данного метода на языке программирования Python.

Как выше было сказано, данный метод широко используют в решениях дифференциальных уравнений различных порядков. Чтобы повысить точность данного метода, можно:

1. Повысить порядок метода Рунге-Кутты
2. Увеличить количество шагов

Но повышение порядка метода приведёт к вычислительной трудности, заключающийся в вычислении на каждом этапе N промежуточных значений. При этом изменение шага для методов Рунге-Кутты сложности не представляет.

Метод Рунге-Кутты - довольно широко используемый метод для решения дифференциальных уравнений, так как при небольшом объеме вычислений он обладает высокой точностью метода, в отличие от стандартного метода Эйлера. Но при этом нельзя забывать, что существуют и другие типы методов (многошаговые методы), которые обладают еще более высокой точностью, но при этом более сложны в понимании и реализации.

Поэтому самым оптимальным методом для решения задачи Коши дифференциальных уравнений любых порядков является классический метод Рунге-Кутты (4 порядка).