НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ
“КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ ИМЕНИ ИГОРЯ СИКОРСКОГО”
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КОМПЬЮТЕРНЫЙ ПРАКТИКУМ №6

Дисциплина: «Методы вычислений»

Тема: «Решениние задачи Коши методами Рунгк-Кутта и Адамса»

Выполнила

студентка 3 курса группы ФИ-41

Лавягина Ольга Алексеевна

Проверила

Стёпочкина Ирина Валерьевна

Киев — 2017 года

ОГЛАВЛЕНИЕ

1 ЗНАЧЕНИЕ ТОЧНОЙ ФУНКЦИИ РЕШЕНИЯ $Y\left(X\right)$

Уравнение имеет вид $y' = \left(1 - x^2\right) y + F\left(x\right)$. Необходимо взять

$$h = 0.1,\; x\left(0\right) = 0.$$

Пусть решение известно и определяется $y = e^x$ (вариант 8). Начальное значение $y\left(x\left(0\right)\right) = y\left(0\right) = e^0 = 1$.

Необходимо подставить решение в уравнение и определить в правой части $F\left(x\right)$.

Подставляем $e^x = \left(1 - x^2\right) e^x + F\left(x\right)$, откуда $F\left(x\right) = e^x - e^x + x^2 e^x = x^2 e^x$.

Таким образом, известный вид уравнения $y' = \left(1 - x^2\right) y + x^2 e^x$ и его точное решение, с помощью численных методов дальше строим приближённое решение.

2 ЗНАЧЕНИЯ ПРИБЛИЖЁННОГО РЕШЕНИЯ $Y(X)$ В ТЕХ ЖЕ ТОЧКАХ, ПОЛУЧЕННЫЕ ОБОИМИ МЕТОДАМИ

Используем формулы Рунге-Кутта четвёртого порядка. Они имеют вид

$$y_{i+1} = y_i + \frac{1}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4),$$

где

$$\begin{cases} k_1 = h \cdot f(x_i, y_i), \\ k_2 = h \cdot f\left(x_i + \frac{1}{2} \cdot h,\ y_i + \frac{1}{2} \cdot k_1\right), \\ k_3 = h \cdot f\left(x_i + \frac{1}{2} \cdot h,\ y_i + \frac{1}{2} \cdot k_2\right), \\ k_4 = h \cdot f(x_i + h, y_i + k_3), \end{cases}$$

где $f(x_i, y_i) = \left(1 - x^2\right) y + x^2 e^x$.

Приближённое решение методом Адамса-Башфорта находится по формуле

$$y_{k+1} = y_k + \frac{h}{24} (55 f_k - 59 f_{k-1} + 37 f_{k-2} - 9 f_{k-3}).$$

Метод обеспечиваем четвёртый порядок точности.

3 ГРАФИКИ ТОЧНОГО РЕШЕНИЯ И ОБОИХ ПРИБЛИЖЁННЫХ

Графики точного решения и обоих приближённых изображены на рисунке 3.1.

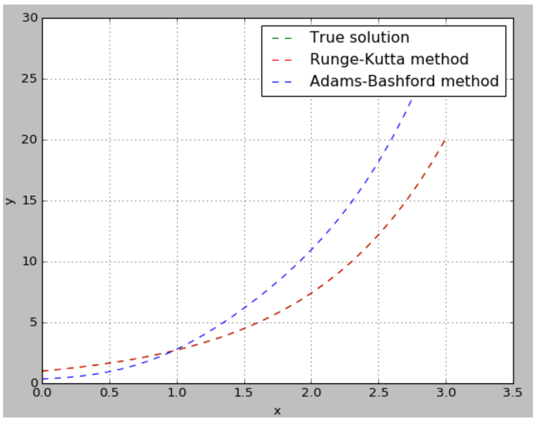

Рисунок 3.1 — Графики точного решения и обоих приближённых

Видно, что точное решение и решение методом Рунге-Кутта совпадают.

4 ОШИБКИ МЕТОДОВ

Для метода Рунге-Кутта была найдена теоретическая ошибка по формуле

$$e = \max_i \frac{\left|y_i^{(h)} - y_{2i}^{\frac{h}{2}}\right|}{15}.$$

Она равна 1.13914289957.

Для обоих методов построены графики ошибок (рис. 4.1), найденных по формуле $|trueSolution - methodSolution|$.

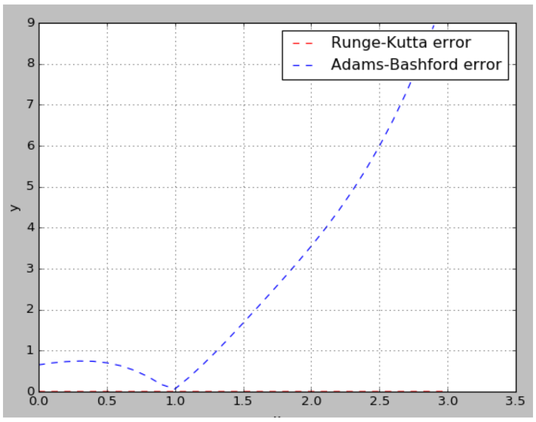

Рисунок 4.1 — Графики ошибок

5 ЛИСТИНГ ПРОГРАММЫ

Листинг файла __main__.py

```
import matplotlib.pyplot as plt
import numpy as np
import math


from function import f
from runge_kutta import runge_kutta
from adams_bashford import adams_bashford
from error import runge_kutta_theoretical_error, error

if __name__ == '__main__':
    h = 0.1
    left = 0
    right = 3
    x, runge_kutta_result = runge_kutta(h, left, right)
    x_adams_bashford, adams_bashford_result = adams_bashford(h, left, right)
    y = []
    for element in x:
        y.append(math.exp(element))
    plt.xlabel('x')
    plt.ylabel('y')
    plt.plot(x, y, 'g--', label = 'True solution')
    plt.plot(x, runge_kutta_result, 'r--', label = 'Runge-Kutta method')
    plt.plot(x_adams_bashford, adams_bashford_result, 'b--', label = 'Adams-Bashford method')
    plt.legend()
    plt.grid(True)
    plt.show()

    theoretical_error = runge_kutta_theoretical_error(h, left, right, runge_kutta_result)
    print(theoretical_error)

    new_y = []
    for element in x_adams_bashford:
        new_y.append(math.exp(element))
    runge_kutta_error = error(runge_kutta_result, y)
    adams_bashford_error = error(adams_bashford_result, new_y)
    plt.xlabel('x')
    plt.ylabel('y')
    plt.plot(x, runge_kutta_error, 'r--', label = 'Runge-Kutta error')
    plt.plot(x_adams_bashford, adams_bashford_error, 'b--', label = 'Adams-Bashford error')
    plt.legend()
    plt.grid(True)
    plt.show()
```

Листинг файла function.py

```
import math


def f(x, y):
    return (1 - x**2) * y + x**2 * math.exp(x)
```

Листинг файла runge_kutta.py

```
from function import f

def runge_kutta(h, left, right):
  x, y = 0, 1
  result = []
  x_axis = []
  while right >= left:
      result.append(y)
      x_axis.append(x)
      left += h
      k1 = h * f(x, y)
      k2 = h * f(x + 0.5 * h, y + 0.5 * k1)
      k3 = h * f(x + 0.5 * h, y + 0.5 * k2)
      k4 = h * f(x + h, y + k3)
      y += (k1 + 2 * k2 + 2 * k3 + k4) / 6
      x = left
  result.append(y)
  x_axis.append(x)
  return x_axis, result
```

Листинг файла adams_bashford.py

```
from runge_kutta import runge_kutta
from function import f

def adams_bashford(h, left, right):
    tmp = runge_kutta(h, left, right)[0]
    result = tmp[0:4]
    x = left
    i = 3
    x_axis = []
    while right > x:
        result.append(result[i] + h *
                   (55*f(x + 3 * h, result[i]) -
                    59*f(x + 2 * h, result[i-1]) +
                    37*f(x + h, result[i-2]) -
                    9*f(x, result[i-3])) / 24)
        x_axis.append(x)
        x += h
        i += 1
    result = result[4:len(result)]
```

```
    return x_axis, result
```

Листинг файла error.py

```
import scipy


from runge_kutta import runge_kutta
from adams_bashford import adams_bashford

def runge_kutta_theoretical_error(h, left, right, result):
    new_result = runge_kutta(0.5 * h, left, right)[0]
    new_result = new_result[::2]
    result = scipy.array(result)
    new_result = scipy.array(new_result)
    error = abs(result - new_result) / 15.0
    return max(error)

def error(result, solution):
    result = scipy.array(result)
    solution = scipy.array(solution)
    return abs(result - solution)
```

ВЫВОДЫ

Было найдено решение уравнения вида $y' = f(x, y)$ с помощью методов Рунке-Кутта и Адамса-Батфорта. Построены графики ошибок, из которых видно, что решение, полученное с помощью метода Рунке-Кутта, совпадает с точным.