Решение задачи Коши для системы ОДУ модифицированным методом Эйлера

Филиппов Денис Константинович (группа 5030102/10401)

24 мая 2024 г.

1 Постановка задачи

Требуется запрограммировать модифицированный метод Эйлера решения задачи Коши для системы ОДУ:

$$y' = f(t, y),\ t \in [t_0, T]$$
$$y(t_0) = y_0$$

Здесь $y = (y_1(t), \ldots, y_2(t))^T$, $f(t, y) = (f_1(t, y), \ldots, f_n(t, y))^T$. Программа должна работать для произвольной размерности системы уравнений. Функция правой части системы и начальное условие подаются на вход программе.

Вычисления должны производиться с пошаговым контролем точности по правилу Рунге: если на текущем шаге точность не достигается, то шаг уменьшается в 2 раза, если достигнутая погрешность меньше заданной в 64 раза, то шаг увеличивается в 2 раза.

2 Описание алгоритма

2.1 Процедура одного шага по методу Эйлера

Идея модифицированного метода Эйлера состоит в том, что производную вычисляют не в i-ой точке, а между двумя соседними точками: i и $i+1$. Данная процедура состоит из следующих шагов:

1). В точке i вычисляют значение производной, делают пол-шага и вычисляют значение функции на середине отрезка:

$$y_{i+\frac{1}{2}} = y_i + \frac{h}{2} \cdot f(t_i, y_i)$$

2). В точке $i+\frac{1}{2}$ вычисляют производную, делается полный шаг из точки i в точку $i+1$ по значению уточненной производной:

$$y_{i+1} = y_i + h \cdot f(t_{i+\frac{1}{2}}, y_{i+\frac{1}{2}});$$

Данный метод обладает точностью $O(h^2)$, то есть на порядок выше, чем метод Эйлера, при увеличении числа вычислений всего в 2 раза.

2.2 Оценка точности по правилу Рунге

Оценка точности по правилу Рунге - это метод определения точности численного метода путем сравнения результатов, полученных с разными размерами шага интегрирования. Формула для оценки точности выглядит следующим образом:

$$R = \frac{||y_1 - y_2||}{2^p - 1}$$

Здесь y_1 - значение функции в точке, вычисленное с использованием шага интегрирования h, y_2 - значение функции в этой же точке, вычисленное с использованием шага интегрирования $h/2$, $p = 2$ - порядок точности метода.

2.3 Описание входных и выходных данных

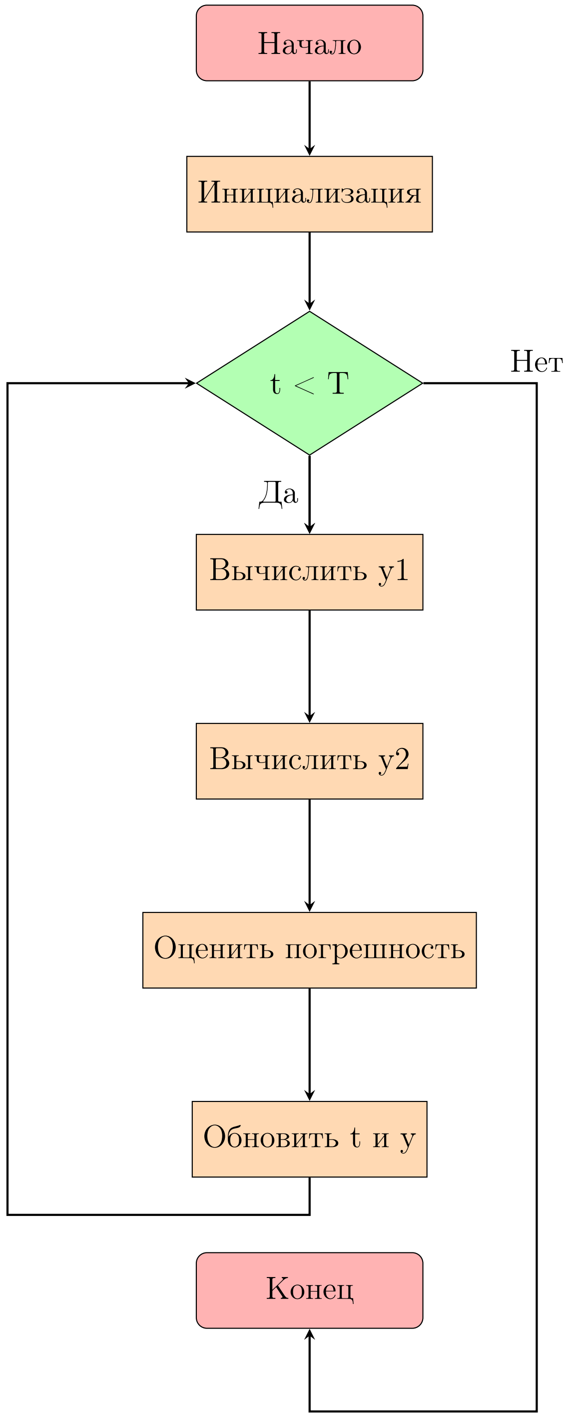
На вход подаются несколько строк, в которых:

- t_0 - начало промежутка
- T - конец промежутка
- h_0 - начальный шаг
- N_x - максимальное число вызовов функции правой части
- eps - желаемая точность
- n - число уравнений
- Следующие $n + 3$ строк определяют функцию правой части на Python
- Начальное условие

Программа печатает в консоль следующие столбцы, одна строчка соответствует одному шагу интегрирования:

1. Значение t
2. Значение шага h
3. Оценка Рунге R
4. Истраченое число вычислений правой части N
5. Значения функций решений

2.4 Блок-схема



3 Пример работы программы

Для запуска программы в файле main.py требуется библиотека Numpy. Пример выполнения программы в виде скриншота: входные данные и полный вывод.

```
main
/home/aphellay/PycharmProjects/Euler-Method-2/.venv/bin/python /home/aphellay/PycharmProjects/Euler-Method-2/main.py
1.5
2.5
0.1
10000
0.0001
3
def fs(t, v, kounter):
#
#
  A = np.array([[-0.4, 0.02, 0], [0, 0.8, -0.1], [0.003, 0, 1]])
  kounter[0] += 1
  return np.dot(A, v)
1 1 2
```

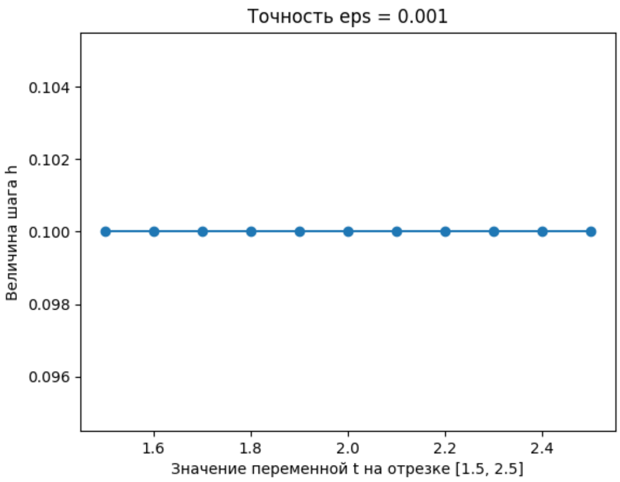
Рис. 1: Входные данные

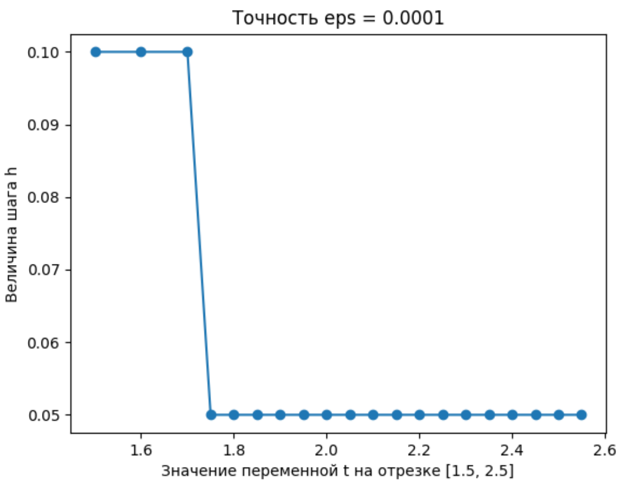
```
    1.500000     0.100000  0.00000e+00           0    1.000000     1.000000     2.000000
    1.600000     0.100000  8.45154e-05           6    0.962820     1.061398     2.210309
    1.700000     0.100000  9.33737e-05          12    0.927221     1.125613     2.442690
    1.750000     0.050000  1.28171e-05          24    0.909992     1.158775     2.568019
    1.800000     0.050000  1.34742e-05          30    0.893138     1.192634     2.699768
    1.850000     0.050000  1.41654e-05          36    0.876652     1.227187     2.838267
    1.900000     0.050000  1.48925e-05          42    0.860527     1.262426     2.983862
    1.950000     0.050000  1.56573e-05          48    0.844756     1.298342     3.136916
    2.000000     0.050000  1.64617e-05          54    0.829333     1.334924     3.297812
    2.050000     0.050000  1.73079e-05          60    0.814253     1.372157     3.466951
    2.100000     0.050000  1.81979e-05          66    0.799508     1.410026     3.644757
    2.150000     0.050000  1.91341e-05          72    0.785092     1.448511     3.831672
    2.200000     0.050000  2.01189e-05          78    0.771001     1.487590     4.028165
    2.250000     0.050000  2.11548e-05          84    0.757227     1.527236     4.234726
    2.300000     0.050000  2.22444e-05          90    0.743766     1.567420     4.451871
    2.350000     0.050000  2.33906e-05          96    0.730612     1.608110     4.680143
    2.400000     0.050000  2.45962e-05         102    0.717758     1.649267     4.920111
    2.450000     0.050000  2.58644e-05         108    0.705200     1.690849     5.172377
    2.500000     0.050000  2.71984e-05         114    0.692932     1.732810     5.437568
    2.550000     0.050000  2.86017e-05         120    0.680948     1.775097     5.716349

Process finished with exit code 0
```

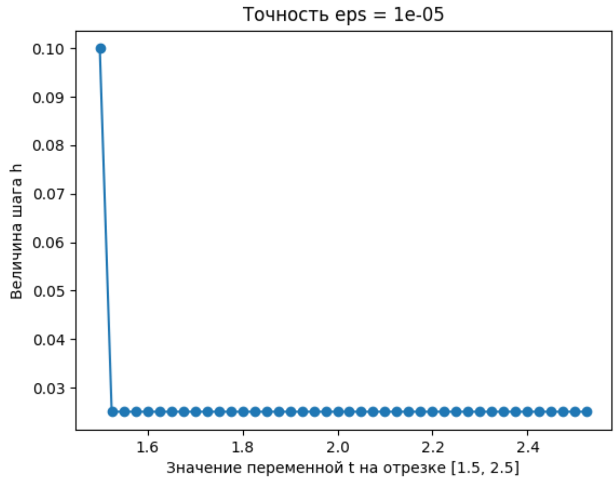
Рис. 2: Результат работы программы

4 Исследование полученных результатов

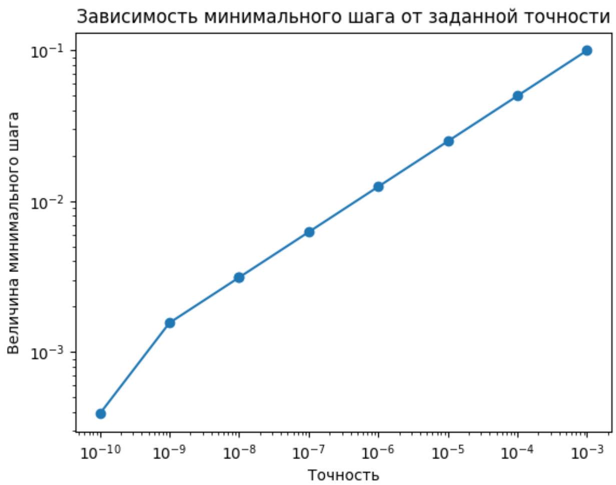
4.1 Изменение шага по отрезку для разных значений заданной точности



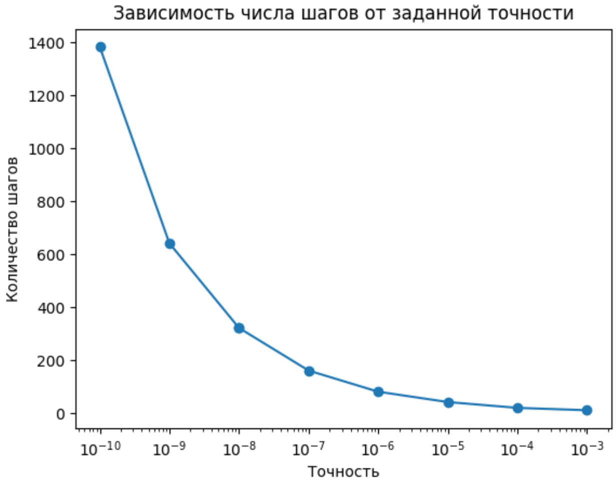




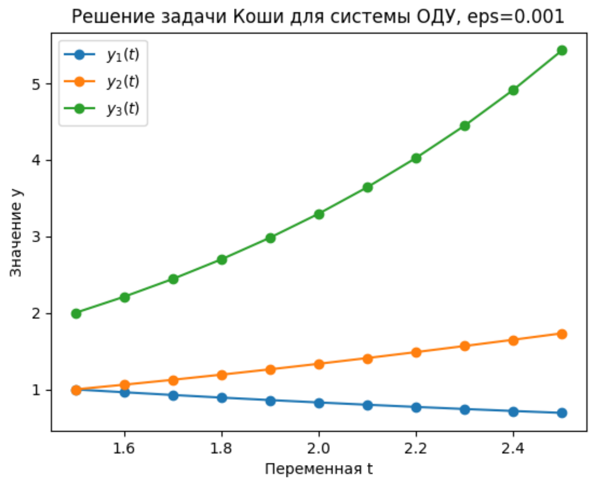
4.2 Зависимость минимального шага от заданной точности

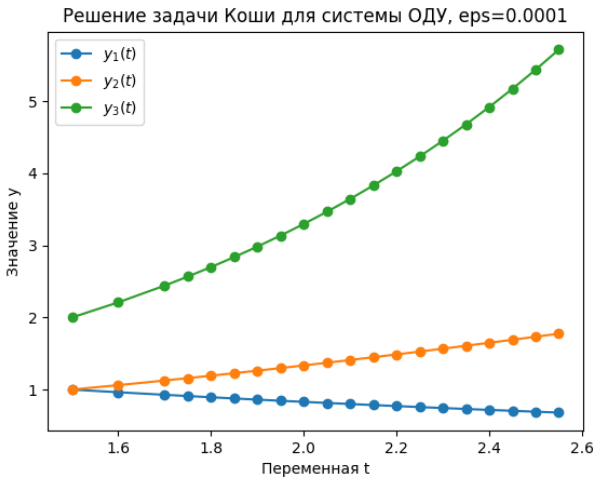


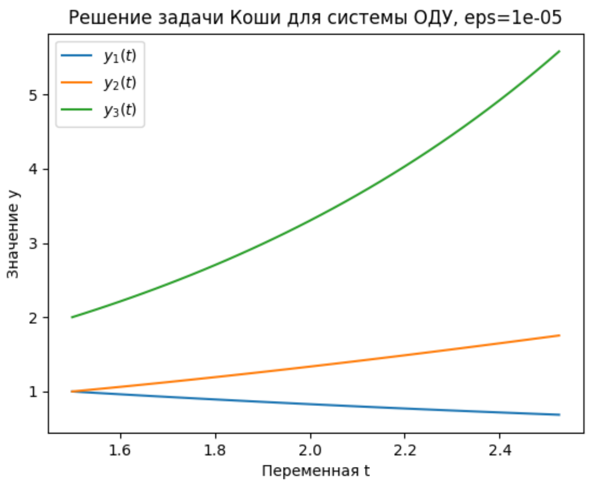
4.3 Зависимость числа шагов от заданной точности



4.4 Решение для разных значений заданной точности




Решение задачи Коши для системы ОДУ, eps=0.0001
$y_1(t)$
$y_2(t)$
$y_3(t)$
Значение y
Переменная t
1.6
1.8
2.0
2.2
2.4
2.6
1
2
3
4
5


Решение задачи Коши для системы ОДУ, eps=1e-05
$y_1(t)$
$y_2(t)$
$y_3(t)$
Значение y
Переменная t
1.6
1.8
2.0
2.2
2.4
1
2
3
4
5