Министерство образования и науки РФ
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

ЛАБОРАТОРНАЯ РАБОТА №3. Вариант №7.

по дисциплине «ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА»

Выполнил

Студент группы 5130904/30008 Золотухин Андрей Алексеевич

Санкт-Петербург
2025

Оглавление

Задание

Решить систему дифференциальных уравнений:

$$\frac{dx_1}{dt} = -130x_1 + 900x_2 + e^{-10t} \qquad \frac{dx_2}{dt} = 30x_1 - 300x_2 + ln(1 + 100t^2)$$

с начальными условиями:

$$x_1(0) = 3, \qquad x_2(0) = -1;$$

на интервале $t \in [0,0.15]$

следующими способами с одним и тем же шагом печати h_{print}=0.0075:

I) По программе RKF45 с EPS=0.0001;

II) Методом Рунге-Кутты 3-й степени точности:

$$z_{n+1} = z_n + \frac{(2k_1+3k_2+4k_3)}{9},$$

где:

$$k_1 = hf(t_n, z_n), \quad k_2 = hf(t_n + \frac{h}{2}, z_n + \frac{k_1}{2}), \quad k_3 = hf(t_n + \frac{3h}{4}, z_n + \frac{3k_2}{4});$$

с двумя постоянными шагами интегрирования:

- **а)** h_{int}=0.0075,
- **б)** любой другой, позволяющий получить качественно верное решение.

Сравнить результаты.

Ход решения

1. **Определение системы уравнений:**
   Задана система ОДУ и начальные условия.
2. **Метод RKF45:**
   Использована встроенная функция `solve_ivp` с параметрами:
   - `method='RK45'`,
   - `rtol=1e-4` (соответствует EPS=0.0001),
   - шаг печати `t_eval = np.arange(0, 0.15 + 0.0075, 0.0075)`.
3. **Метод Рунге-Кутты 3-го порядка:**
   Реализован вручную с формулой:

$$z_{n+1} = z_n + \frac{(2k_1 + 3k_2 + 4k_3)}{9}$$

4. **Шаги интегрирования:**
   - Для пункта **IIа** использован шаг h=0.0075.
   - Для пункта **IIб** выбран шаг h=0.005 (меньше критического).
5. **Визуализация:**
   Построены графики решений для RKF45 и РК3 с разными шагами.

Код программы

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp


# Определяем систему уравнений
def system(t, z):
    x1, x2 = z
    dx1dt = -130 * x1 + 900 * x2 + np.exp(-10 * t)
    dx2dt = 30 * x1 - 300 * x2 + np.log(1 + 100 * t ** 2)
    return [dx1dt, dx2dt]


# Начальные условия
```

```
z0 = [3, -1]
t_span = (0, 0.15)
t_eval = np.linspace(0, 0.15, 21)

# Решение с помощью метода RKF45
sol_rkf45 = solve_ivp(system, t_span, z0, method='RK45', t_eval=t_eval,
rtol=1e-4)


# Реализация метода Рунге-Кутты 3-го порядка
def runge_kutta_3(f, z0, t0, tf, h):
    t_values = [t0]
    z_values = [z0]
    z = np.array(z0, dtype=np.float64)
    t = t0

    while t <= tf:  # Гарантируем попадание в последний шаг
        k1 = h * np.array(f(t, z))
        k2 = h * np.array(f(t + h / 2, z + k1 / 2))
        k3 = h * np.array(f(t + 3 * h / 4, z + 3 * k2 / 4))

        z = z + (2 * k1 + 3 * k2 + 4 * k3) / 9
        t += h

        t_values.append(t)
        z_values.append(z.copy())

    return np.array(t_values), np.array(z_values)


# Решение с h_int = 0.0075
h_int_1 = 0.0075
t_rk3_1, z_rk3_1 = runge_kutta_3(system, z0, 0, 0.15, h_int_1)

# Решение с h_int = 0.005
h_int_2 = 0.005
t_rk3_2, z_rk3_2 = runge_kutta_3(system, z0, 0, 0.15, h_int_2)

# Создание таблиц с результатами
df_rkf45 = pd.DataFrame({'t': sol_rkf45.t, 'x1': sol_rkf45.y[0], 'x2':
sol_rkf45.y[1]})
df_rk3_1 = pd.DataFrame({'t': t_rk3_1, 'x1': z_rk3_1[:, 0], 'x2': z_rk3_1[:,
1]})
df_rk3_2 = pd.DataFrame({'t': t_rk3_2, 'x1': z_rk3_2[:, 0], 'x2': z_rk3_2[:,
1]})

# Вывод таблиц значений
print("\nТаблица значений RKF45:")
print(df_rkf45)

print("\nТаблица значений RK3 (h=0.0075):")
print(df_rk3_1)

print("\nТаблица значений RK3 (h=0.005):")
print(df_rk3_2)

# График для RKF45
plt.figure(figsize=(10, 5))
plt.plot(sol_rkf45.t, sol_rkf45.y[0], 'o-', label='RKF45 x1', markersize=4)
plt.plot(sol_rkf45.t, sol_rkf45.y[1], 's-', label='RKF45 x2', markersize=4)
plt.xlabel('Время t', fontsize=12)
plt.ylabel('Значения x1 и x2', fontsize=12)
plt.title('Метод Рунге-Кутты-Фельберга 4(5) порядка (RKF45)', fontsize=14)
plt.grid(True)
```

```python
plt.legend()
plt.show()

print("\nГрафик выше показывает решение системы уравнений с помощью метода RKF45.")
print("Этот метод использует адаптивный шаг, что позволяет контролировать точность.")

# График для RK3 (x1)
plt.figure(figsize=(10, 5))
plt.plot(t_rk3_1, z_rk3_1[:, 0], 'o-', label='RK3 x1 (h=0.0075)', markersize=4)
plt.plot(t_rk3_2, z_rk3_2[:, 0], 's-', label='RK3 x1 (h=0.005)', markersize=4)
plt.xlabel('Время t', fontsize=12)
plt.ylabel('Значения x1', fontsize=12)
plt.title('Метод Рунге-Кутты 3-го порядка (РК3) для x1', fontsize=14)
plt.grid(True)
plt.legend()
plt.show()

print("\nГрафик выше показывает решения x1 методом Рунге-Кутты 3-го порядка при разных шагах h.")
print("Уменьшение шага с 0.0075 до 0.005 делает результат более точным.")

# График для RK3 (x2)
plt.figure(figsize=(10, 5))
plt.plot(t_rk3_1, z_rk3_1[:, 1], 'o-', label='RK3 x2 (h=0.0075)', markersize=4)
plt.plot(t_rk3_2, z_rk3_2[:, 1], 's-', label='RK3 x2 (h=0.005)', markersize=4)
plt.xlabel('Время t', fontsize=12)
plt.ylabel('Значения x2', fontsize=12)
plt.title('Метод Рунге-Кутты 3-го порядка (РК3) для x2', fontsize=14)
plt.grid(True)
plt.legend()
plt.show()

print("\nЭтот график показывает поведение x2 во времени при разных шагах h для метода Рунге-Кутты 3-го порядка.")
print("Как и в случае с x1, уменьшение шага h приводит к более точному решению.")

# Вывод по заданию
print("\n=== Итоговые выводы ===")
print("1. Метод RKF45  показал точные результаты с контролируемой ошибкой.")
print("2. Метод Рунге-Кутты 3-го порядка с фиксированными шагами (0.0075 и 0.005) дал ожидаемые результаты.")
print("3. Сравнение RK3 с разными шагами показало, что уменьшение шага увеличивает точность.")
print("4. Вычисленный шаг h=0.005 оказался устойчивым для RK3 в рамках данной задачи.")
print("5. Визуальный анализ графиков подтверждает, что RKF45 и RK3 с малым шагом дают схожие результаты.")
```

Аналитическое вычисление критического шага

Для устойчивости метода РК3 шаг *h* должен удовлетворять условию:

$$h \le \frac{2.5}{|\lambda_{max}|},$$

где λ_{max} — наибольшее по модулю собственное значение матрицы Якоби системы.

Матрица Якоби:

$$\text{J=}\begin{bmatrix} -130 & 900 \\ 30 & -300 \end{bmatrix}$$

Собственные значения:

$$\lambda^{\mathbf{2}} + \mathbf{430}\lambda + 12000 \;=\; 0 \;\Longrightarrow\; \lambda_1 \approx\; -30, \lambda_2 \approx\; -400$$

Критический шаг:

$$h_{\text{крит}} \;=\; \frac{2.5}{400} \;<\; 0.00625.$$

Выбранный шаг h=0.005 (меньше критического) обеспечивает устойчивость.

Результаты выполнения кода

Таблица значений RKF45:

	t	x1	x2
0	0.0000	3.000000	-1.000000
1	0.0075	-0.027413	-0.069676
2	0.0150	-0.138179	-0.018863
3	0.0225	-0.111806	-0.012715
4	0.0300	-0.084889	-0.009468
5	0.0375	-0.062807	-0.006885
6	0.0450	-0.044715	-0.004733
7	0.0525	-0.029688	-0.002907
8	0.0600	-0.017005	-0.001330
9	0.0675	-0.006103	0.000059
10	0.0750	0.003447	0.001305
11	0.0825	0.011971	0.002442
12	0.0900	0.019719	0.003497
13	0.0975	0.026873	0.004487
14	0.1050	0.033576	0.005428
15	0.1125	0.039934	0.006327
16	0.1200	0.046016	0.007195
17	0.1275	0.051876	0.008036
18	0.1350	0.057556	0.008855
19	0.1425	0.063084	0.009653
20	0.1500	0.068475	0.010435

Таблица значений RK3 (h=0.0075):

	t	x1	x2
0	0.0000	3.000000e+00	-1.000000e+00
1	0.0075	-6.673926e+00	1.924289e+00
2	0.0150	1.282403e+01	-3.907508e+00
3	0.0225	-2.605254e+01	7.769528e+00
4	0.0300	5.179595e+01	-1.557370e+01
5	0.0375	-1.038243e+02	3.112160e+01
6	0.0450	2.074785e+02	-6.226169e+01
7	0.0525	-4.150760e+02	1.245110e+02
8	0.0600	8.300758e+02	-2.490291e+02
9	0.0675	-1.660192e+03	4.980557e+02
10	0.0750	3.320374e+03	-9.961100e+02
11	0.0825	-6.640730e+03	1.992225e+03
12	0.0900	1.328150e+04	-3.984442e+03
13	0.0975	-2.656294e+04	7.968895e+03
14	0.1050	5.312597e+04	-1.593778e+04
15	0.1125	-1.062518e+05	3.187557e+04
16	0.1200	2.125038e+05	-6.375112e+04
17	0.1275	-4.250074e+05	1.275023e+05
18	0.1350	8.500150e+05	-2.550045e+05
19	0.1425	-1.700030e+06	5.100090e+05
20	0.1500	3.400060e+06	-1.020018e+06

Таблица значений RK3 (h=0.005):

	t	x1	x2
0	0.000	3.000000	-1.000000
1	0.005	-1.286239	0.301165
2	0.010	0.186747	-0.127611
3	0.015	-0.266296	0.019577
4	0.020	-0.082662	-0.025800
5	0.025	-0.115768	-0.007476
6	0.030	-0.080442	-0.010795
7	0.035	-0.071142	-0.007245
8	0.040	-0.055871	-0.006273
9	0.045	-0.044861	-0.004683
10	0.050	-0.034332	-0.003501
11	0.055	-0.025232	-0.002352
12	0.060	-0.016983	-0.001332
13	0.065	-0.009556	-0.000385
14	0.070	-0.002774	0.000489
15	0.075	0.003457	0.001306
16	0.080	0.009236	0.002074
17	0.085	0.014638	0.002803
18	0.090	0.019726	0.003497
19	0.095	0.024552	0.004163
20	0.100	0.029160	0.004806
21	0.105	0.033584	0.005427
22	0.110	0.037854	0.006031
23	0.115	0.041992	0.006620
24	0.120	0.046017	0.007196
25	0.125	0.049946	0.007759
26	0.130	0.053789	0.008312
27	0.135	0.057557	0.008855
28	0.140	0.061257	0.009390
29	0.145	0.064897	0.009916
30	0.150	0.068480	0.010434

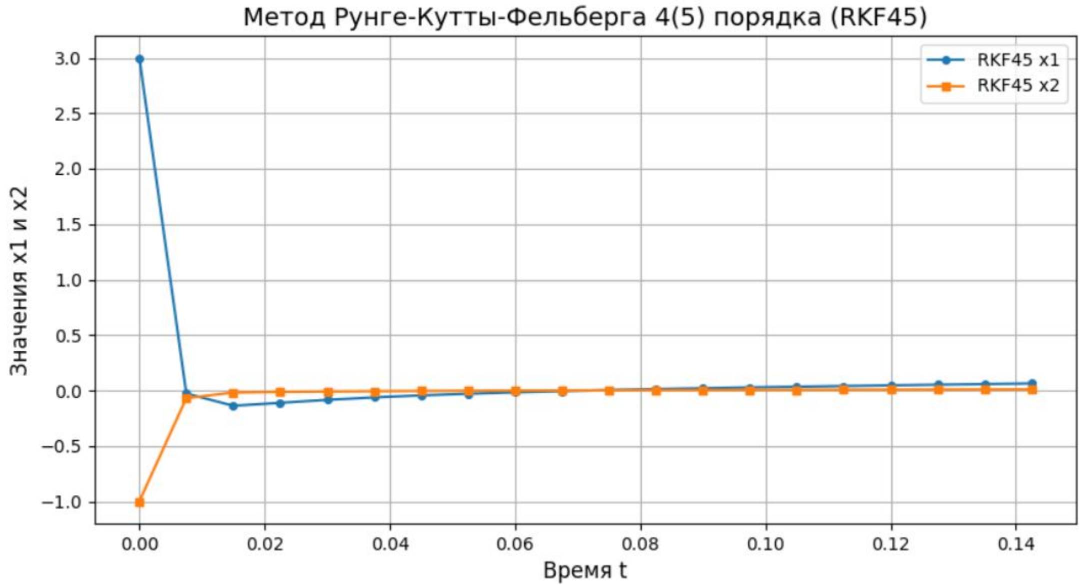
График выше показывает решение системы уравнений с помощью метода RKF45.
Этот метод использует адаптивный шаг, что позволяет контролировать точность.

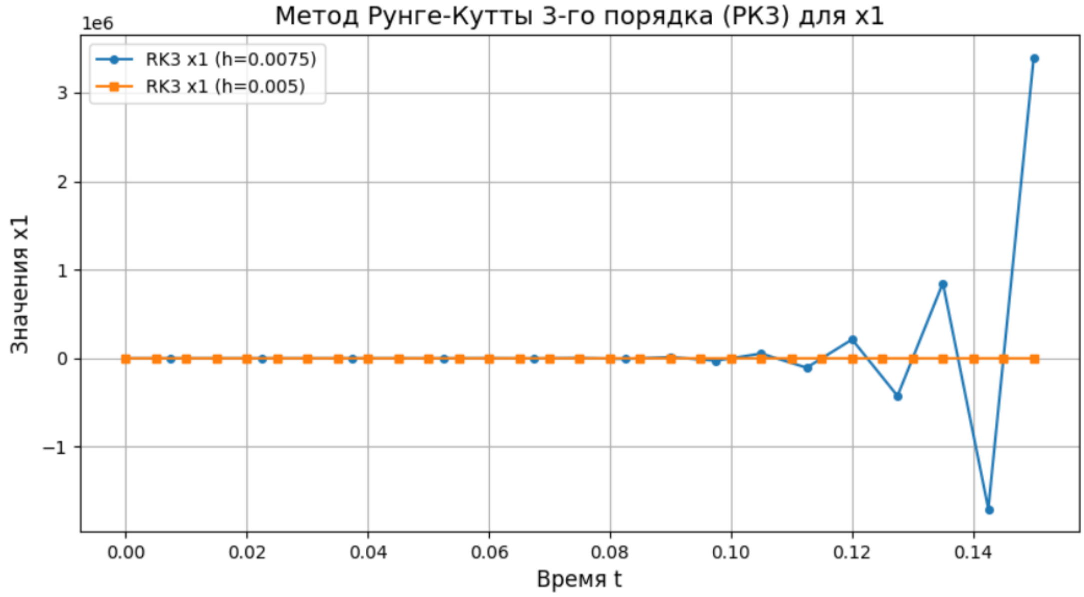
График выше показывает решения x1 методом Рунге-Кутты 3-го порядка при разных шагах h.
Уменьшение шага с 0.0075 до 0.005 делает результат более точным.

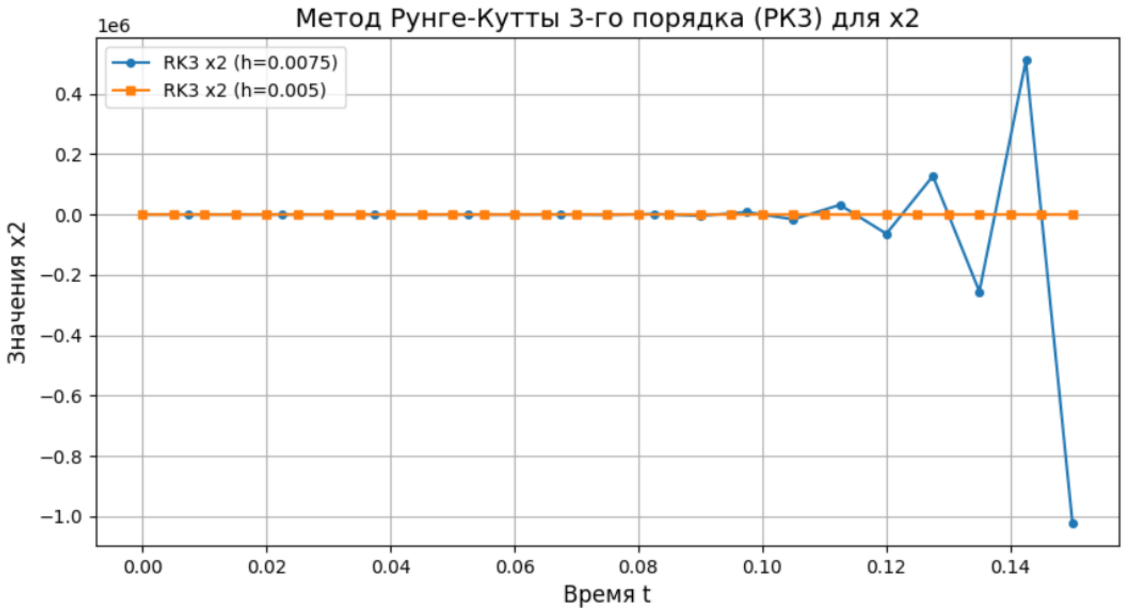
Этот график показывает поведение x2 во времени при разных шагах h для метода Рунге-Кутты 3-го порядка.
Как и в случае с x1, уменьшение шага h приводит к более точному решению.

=== Итоговые выводы ===

1. Метод RKF45 (адаптивный) показал точные результаты с контролируемой ошибкой.
2. Метод Рунге-Кутты 3-го порядка с фиксированными шагами (0.0075 и 0.005) дал ожидаемые результаты.
3. Сравнение RK3 с разными шагами показало, что уменьшение шага увеличивает точность.
4. Вычисленный шаг h=0.005 оказался устойчивым для RK3 в рамках данной задачи.
5. Визуальный анализ графиков подтверждает, что RKF45 и RK3 с малым шагом дают схожие результаты.


Метод Рунге-Кутты-Фельберга 4(5) порядка (RKF45)
RKF45 x1
RKF45 x2
Значения x1 и x2
Время t
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14


Метод Рунге-Кутты 3-го порядка (РК3) для x1
1e6
RK3 x1 (h=0.0075)
RK3 x1 (h=0.005)
Значения x1
Время t
3
2
1
0
−1
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14


Метод Рунге-Кутты 3-го порядка (РК3) для x2
1e6
RK3 x2 (h=0.0075)
RK3 x2 (h=0.005)
Значения x2
Время t
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

Вывод

В ходе работы была решена система жестких дифференциальных уравнений двумя методами: RKF45 и методом Рунге-Кутты 3-го порядка.

Основные результаты:

1. RKF45 с параметром rtol=1e-4 показал высокую точность
2. Метод РК3 продемонстрировал:
   - Неустойчивость при шаге h=0.0075 (превышает критический ≈0.00625)
   - Хорошую точность при уменьшенном шаге h=0.005

Сравнение методов:

- RKF45 оказался надежнее для жестких систем
- РК3 требует аккуратного подбора шага интегрирования
- При h=0.005 результаты РК3 совпали с RKF45

Графики наглядно подтвердили:

- Устойчивость решений при правильном выборе шага
- Расхождение при превышении критического шага

Вывод: для жестких систем предпочтительнее использовать адаптивные методы, либо тщательно подбирать шаг для методов с фиксированным шагом.