



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Факультет Систем управления и робототехники

Лабораторная работа №1

по дисциплине

«Имитационное моделирование робототехнических систем»

Выполнил:

Студент группы R4135с
Дуль Ян Сергеевич

Преподаватель:

Ракшин Егор Александрович

Санкт-Петербург
2025

СОДЕРЖАНИЕ

1	Введение	3
2	Ход работы	4
2.1	Аналитическое решение ДУ	4
2.2	Моделирование	4
3	Выводы	11

1 Введение

В данной лабораторной работе будут рассмотрены численные методы решения дифференциальных уравнений - явный и неявный Эйлер, метод Рунге-Кутты. Такие методы являются альтернативой аналитическому методу, так как он часто может быть дорогим вычислительно. Проведем расчет, промоделируем с помощью Python и сравним вышеупомянутые методы с аналитическим решением ДУ.

2 Ход работы

2.1 Аналитическое решение ДУ

$$8.5\ddot{x} + 7.19\dot{x} - 6.68x = 6.36 \quad (2.1)$$

$$\ddot{x} + 0.846\dot{x} - 0.786x = 0.748 \quad (2.2)$$

Найдем общее решение ДУ:

$$\lambda^2 + 0.846\lambda - 0.786 = 0 \quad (2.3)$$

$$\lambda_1 = -1.405; \lambda_2 = 0.559 \quad (2.4)$$

$$x_o(t) = c_1 e^{-1.405t} + c_2 e^{0.559t} \quad (2.5)$$

Найдем частное решение ДУ:

$$x_p = A \quad (2.6)$$

$$- 6.68A = 6.36 \quad (2.7)$$

$$x_p = A = -0.952 \quad (2.8)$$

Следовательно,

$$x(t) = c_1 e^{-1.405t} + c_2 e^{0.559t} - 0.952 \quad (2.9)$$

2.2 Моделирование

Напишем функцию, генерирующую массивы значений для переменных вектора состояния $[x \ \dot{x}]^T$ для аналитического решения ДУ:

```

def analytical_method(x0, Tf, h):
    xp = -0.952
    l1 = -1.405
    l2 = 0.559

    A = np.array([[1, 1],
                  [l1, l2]])
    b = np.array([x0[0] - xp, x0[1]])

    A_inv = np.linalg.inv(A)
    c1, c2 = A_inv @ b

    t = np.arange(0, Tf + h, h)
    x_hist = np.zeros((len(x0), len(t)))
    x_hist[:, 0] = x0

    for k in range(len(t) - 1):
        x_hist[0, k + 1] = c1 * np.exp(l1 * t[k+1]) + c2 * np.exp(l2 * t[k+1]) + xp # x
        x_hist[1, k + 1] = c1 * l1 * np.exp(l1 * t[k+1]) + c2 * l2 * np.exp(l2 * t[k+1]) # x'

    return x_hist, t

```

Рисунок 1 — Листинг функции генерации значений вектора состояния

Система неустойчива, поэтому ожидается экспоненциальный рост x .
 Промоделируем систему для 4х методов и проанализируем графики.

$$h = 0.01, \quad x(0) = [0, 0.5]^T, \quad T_f = 10$$

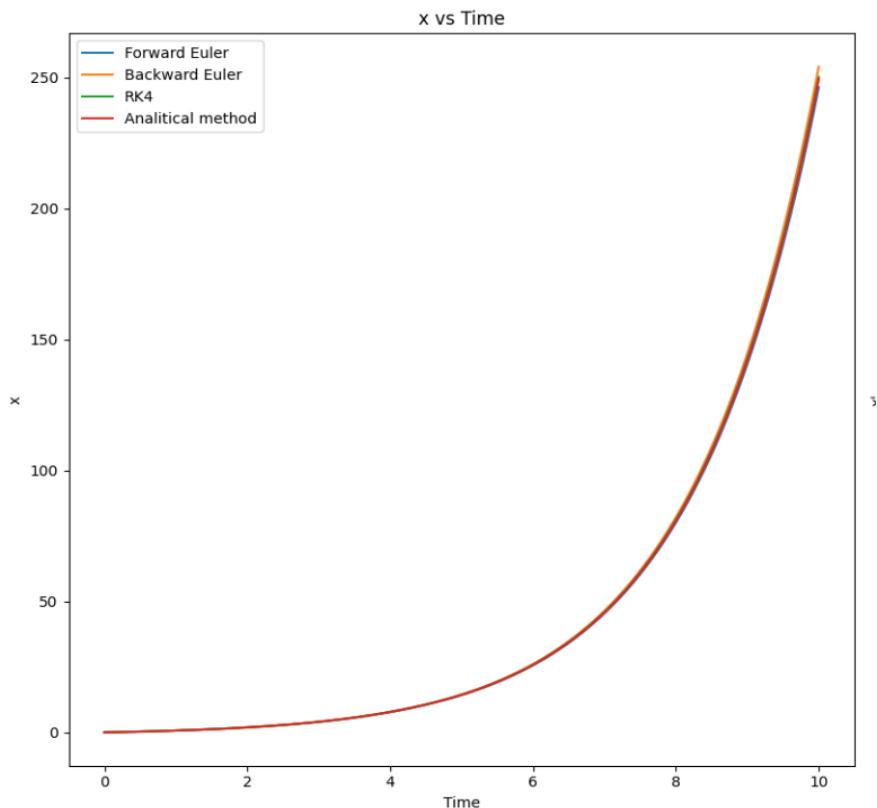


Рисунок 2 — График $x(t)$

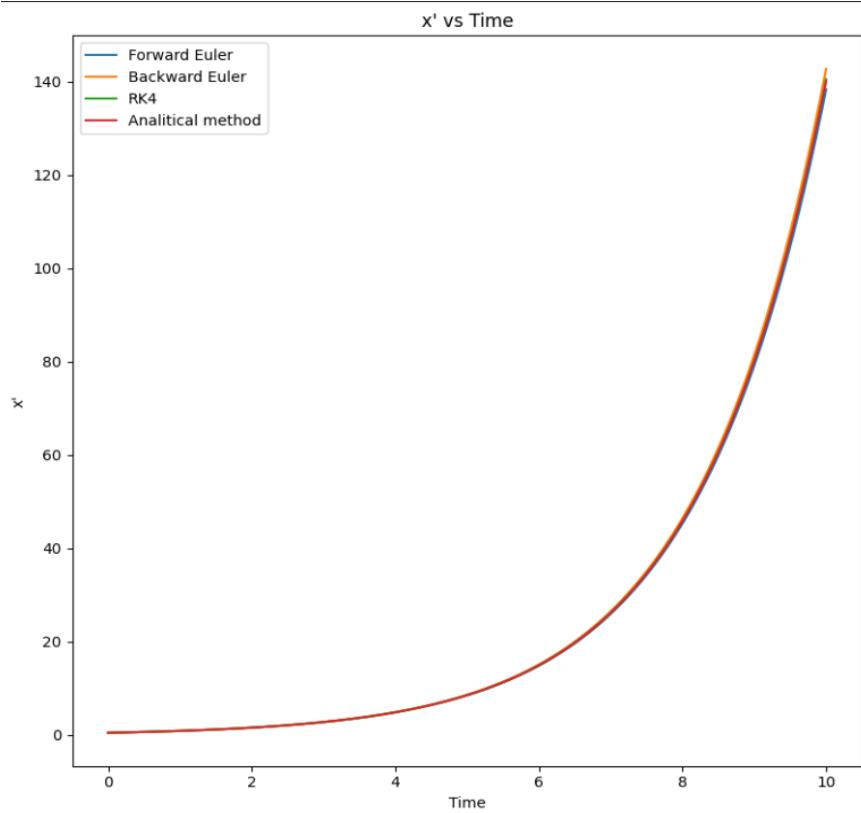


Рисунок 3 — График $x'(t)$

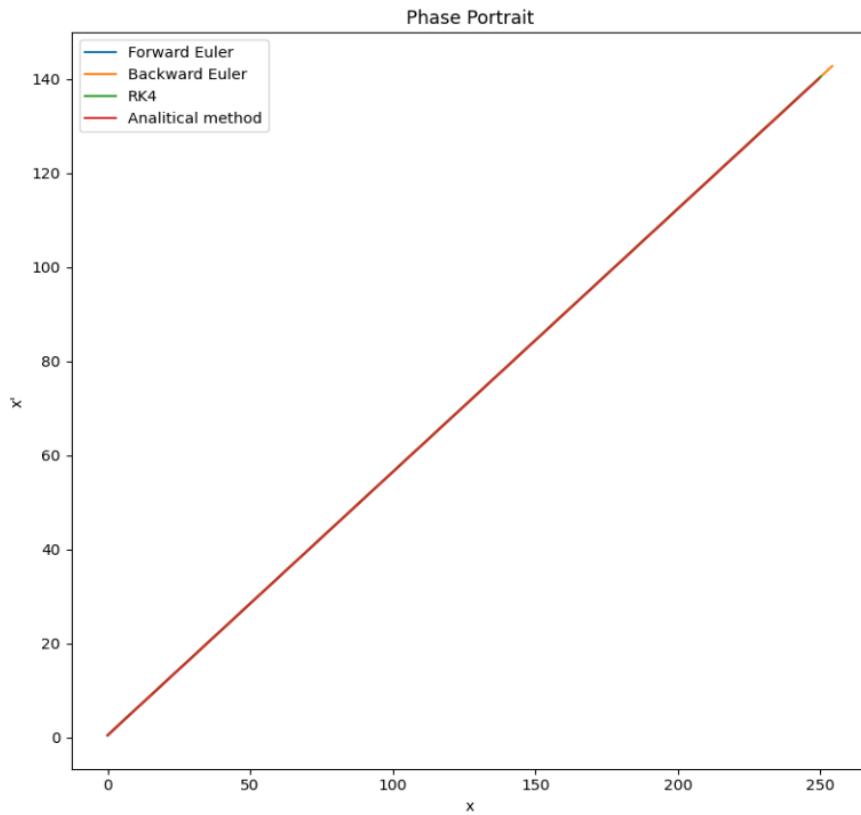


Рисунок 4 — Фазовый портрет

Промоделируем систему при большем периоде дискретизации ($h = 0.1$), чтобы увеличить ошибку:

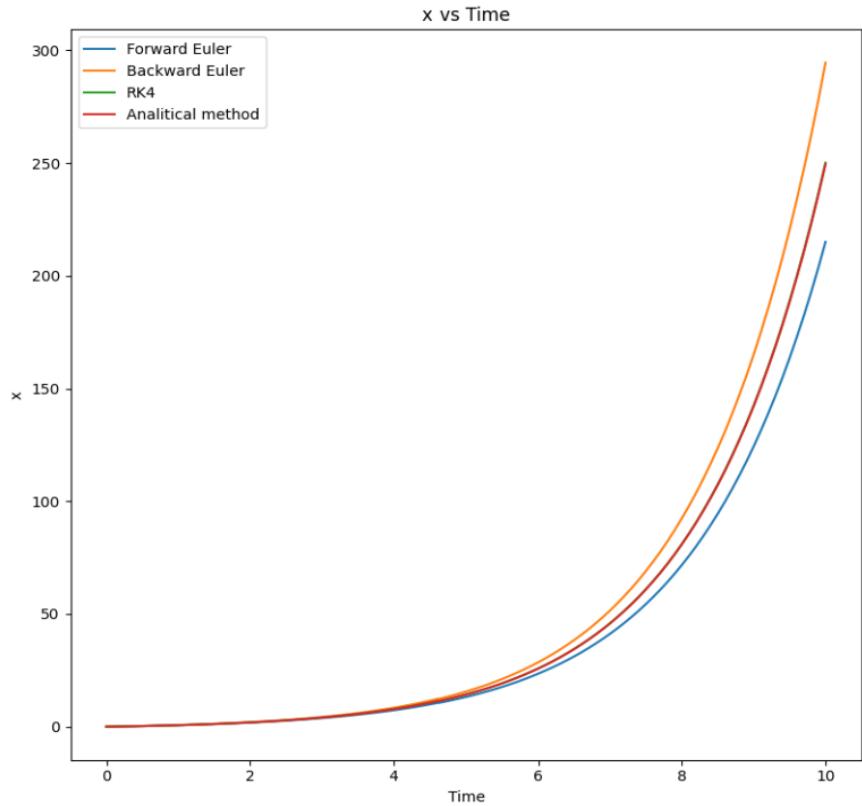


Рисунок 5 — График $x(t)$

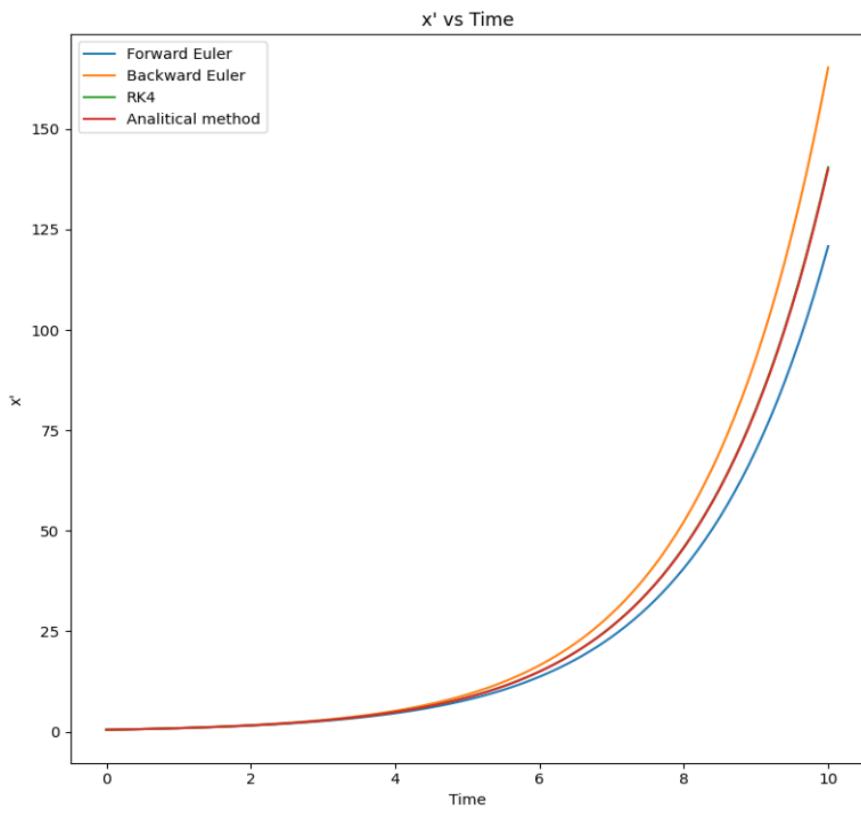


Рисунок 6 — График $x'(t)$

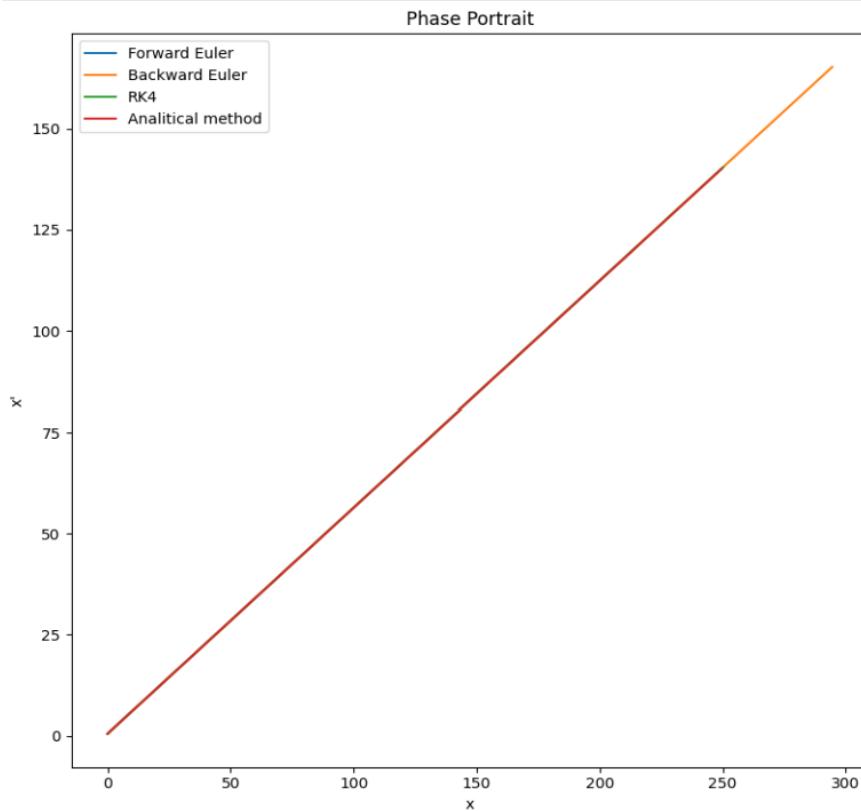


Рисунок 7 — Фазовый портрет

Промоделируем устойчивую систему с собственными числами $\lambda_1 = -6$ и $\lambda_2 = -5$:

$$h = 0.01, \quad x(0) = [0, 0.5]^T$$

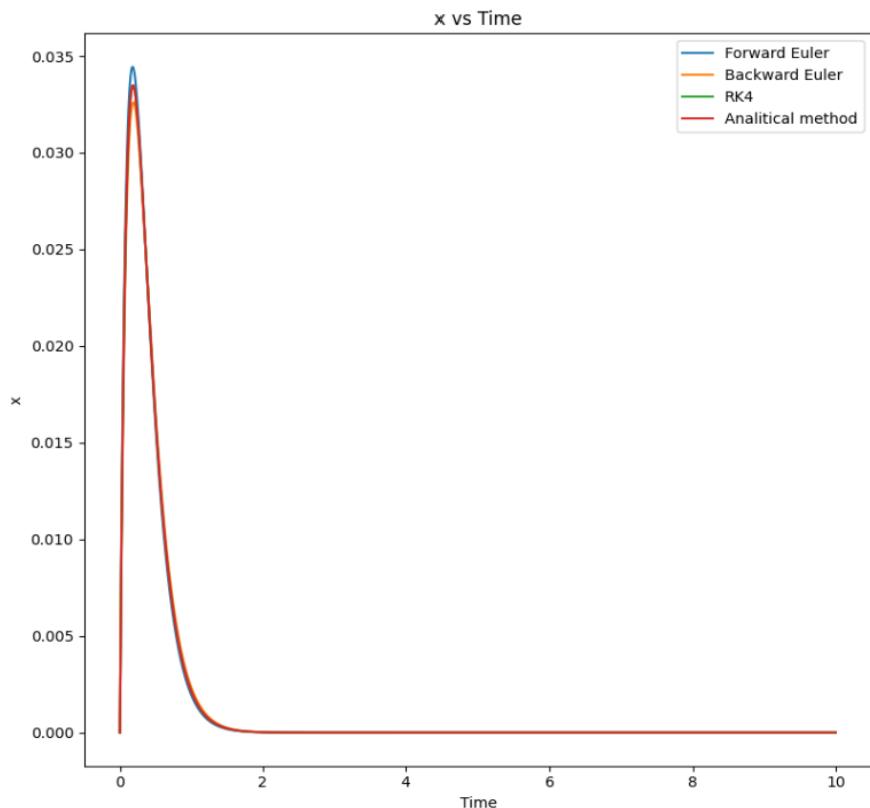


Рисунок 8 — График $x(t)$

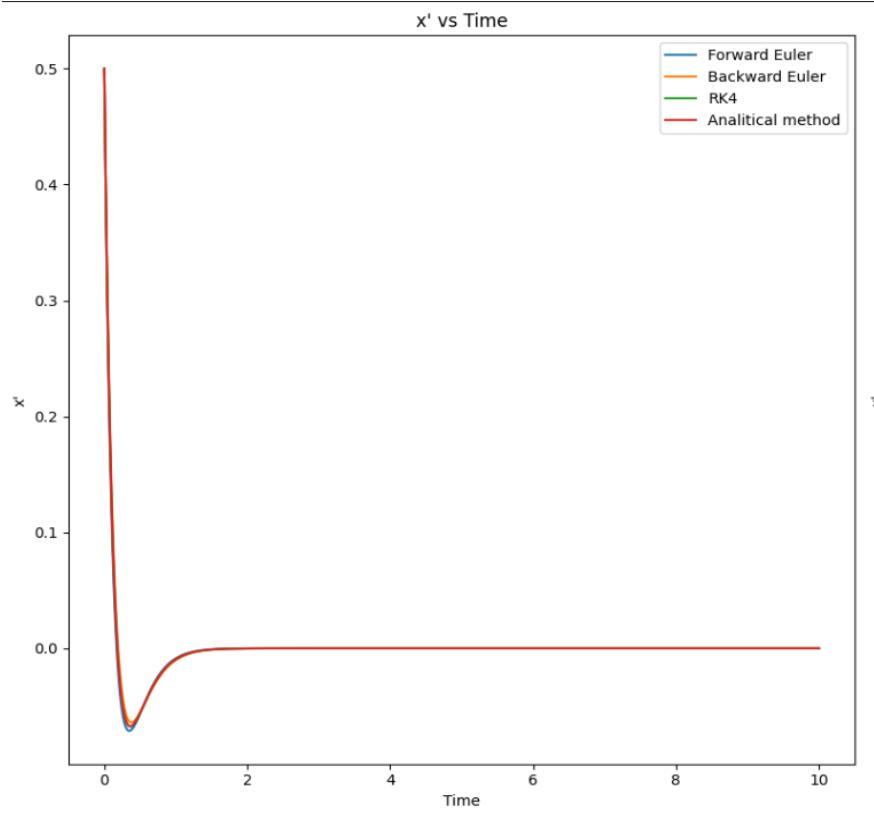


Рисунок 9 — График $x'(t)$

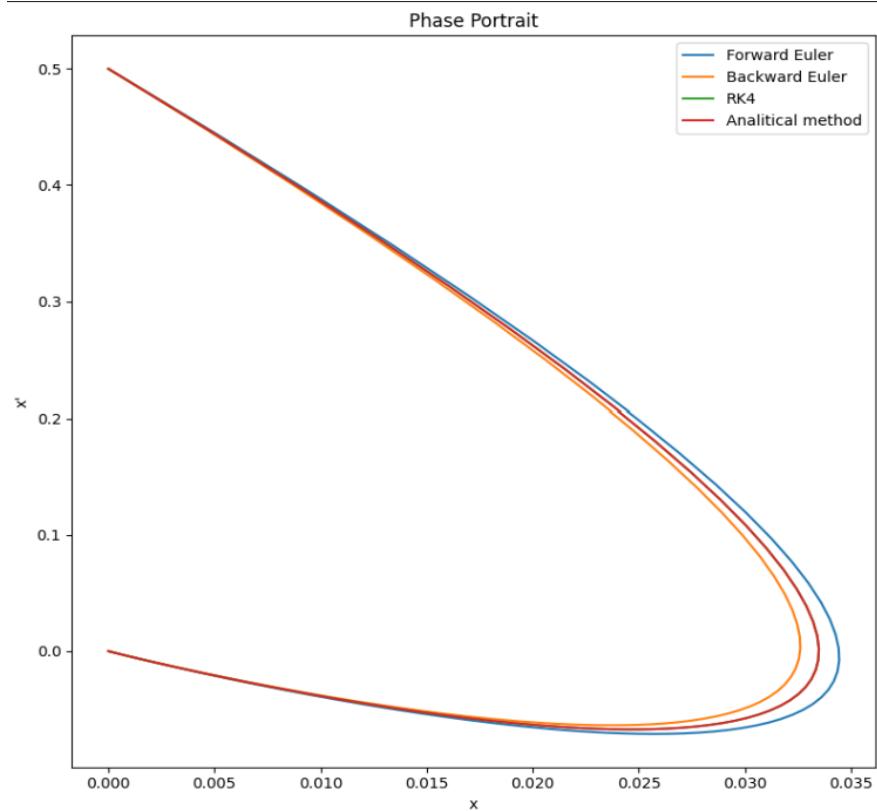


Рисунок 10 — Фазовый портрет

3 Выводы

Графики, полученные в результате моделирования неустойчивой системы, показывают, что при высокой частоте дискретизации методы генерируют значения с минимальным отличием друг от друга. Однако при увеличении шага дискретизации ближе всего к значениям, полученным аналитическим методом, находятся решения методом Рунге-Кутты. Так как аналитическое решение является наиболее близким к истине, то можно сделать вывод, что RK4 справился лучше остальных методов.

Промоделировав устойчивую систему, можно сказать, что явный Эйлер имеет чуть большую амплитуду значений в данном случае.

Метод Рунге-Кутты 4-го порядка оказался самым точным для аппроксимации из рассмотренных методов, однако он требует больше вычислений, чем, например, метод явного Эйлера. Для неустойчивой системы хуже всего справился неявный метод Эйлера, так как он имеет наибольшие расхождения с истинными значениями.