Численные методы решения дифференциальных уравнений

Фролов А.А, Курылев Г.А., 2к, ИВТ-2

Ссылка на скринкаст:

https://rutube.ru/video/private/51227e213ee0fd529fa40d5286cd1d8c/?p=kyCJOOYaVdhV2fi_00sb3Q

Ссылка на программу:

https://github.com/grechixa/Numerical-Integration/tree/andrew

Примечание:

По счастливой случайности во время записи скринкаста в программе стояли неправильные начальные условия для контрольного примера 2, позже я заметил это и исправил, в отчете, который вы видите сейчас уже все верно.

Цель:

Разработать программы решения дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутта

Оборудование:

ПК, редактор кода VS Code, язык программирования Python.

Задача:

изучить численные методы Эйлера и Рунге-Кутта и предложенные варианты алгоритмов их реализации. Разработать программы решения дифференциальных уравнений, используя актуальный для студента язык программирования.

Программы для вычисления

Метод Эйлера:

diff_eq.py

```python
def euler(func, a, b, y0, n):

    h = (b - a) / n
    x = a
    y = np.array(y0, dtype=float)
    xs, ys = [x], [y.copy()]

    for i in range(n):
        y += h * func(x, y)
        x += h
        xs.append(x)
        ys.append(y.copy())

    return np.array(xs), np.array(ys)
```

Метод Рунге-Кутта:

diff_eq.py

```python
def runge(func, a, b, y0, n):
    h = (b - a) / n
```

```
    x = a
    y = np.array(y0, dtype=float)
    xs, ys = [x], [y.copy()]

    for i in range(n):
        k1 = h * func(x, y)
        k2 = h * func(x + h / 2, y + k1 / 2)
        k3 = h * func(x + h / 2, y + k2 / 2)
        k4 = h * func(x + h, y + k3)

        f_i = (k1 + 2 * k2 + 2 * k3 + k4) / 6

        y += f_i
        x += h

        ys.append(y.copy())
        xs.append(x)

    return np.array(xs), np.array(ys)
```

Примечание:

Оба метода реализованы для решения систем обыкновенных дифференциальных уравнений. Для этого в качестве начальных условий `y0` принимаются массивы (векторы), что позволяет решать как отдельные уравнения, так и системы уравнений любого порядка.

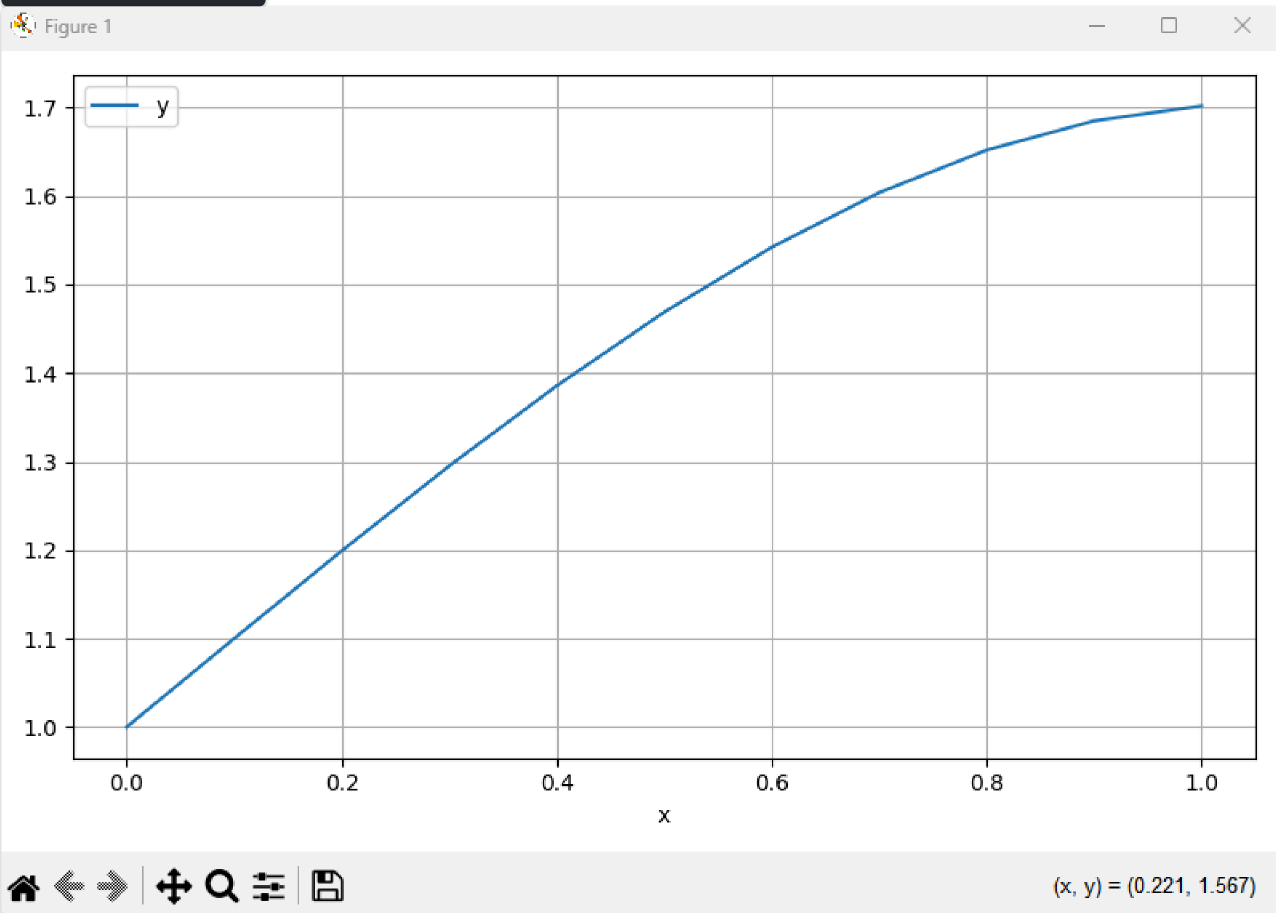
Контрольный пример 1

$$y` = y(1 - x)$$

$$x_0 = 0, y_0 = 1, n = 10$$

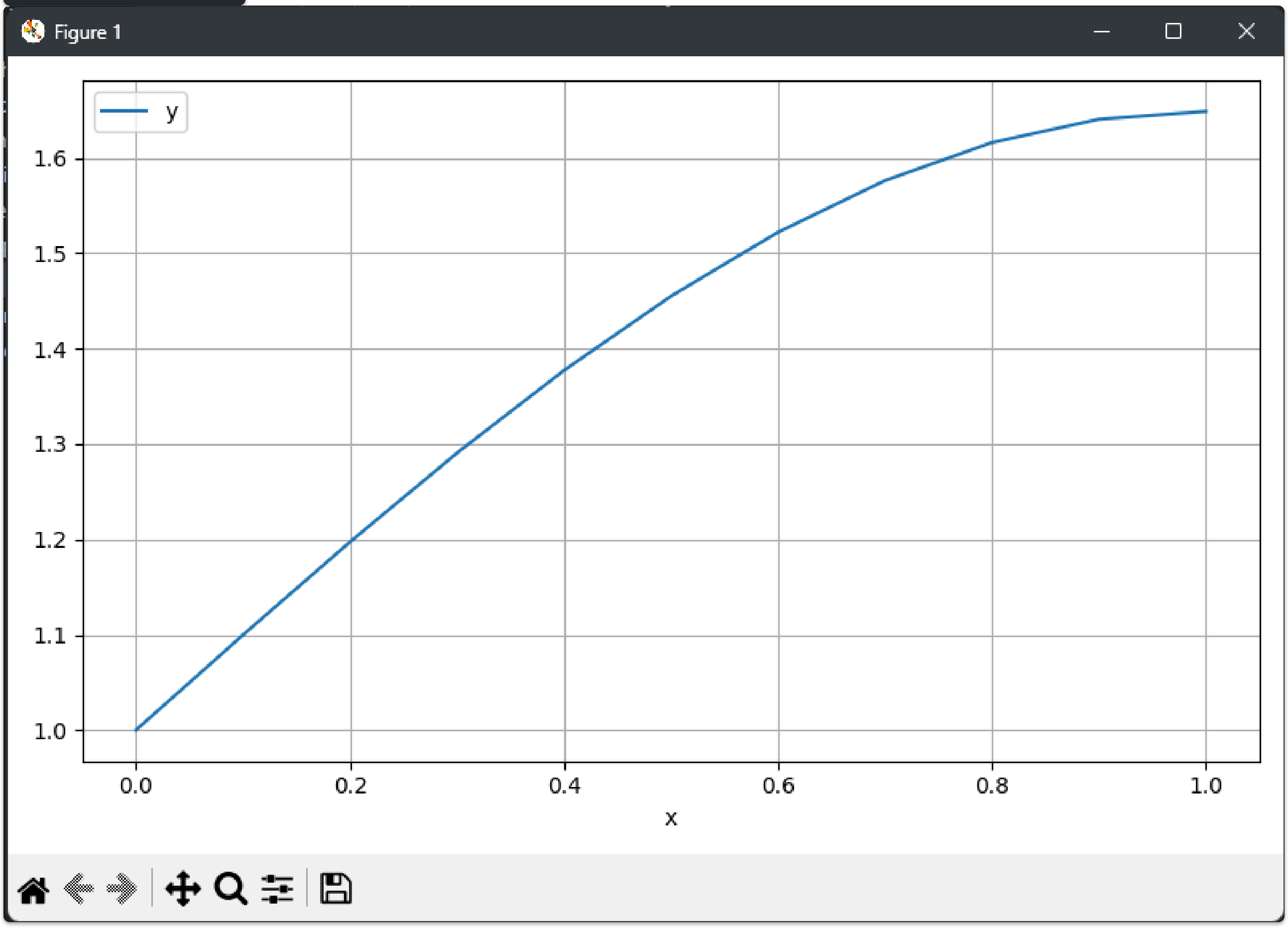
Результаты, полученные при решении методом Эйлера:

```
=== РЕЗУЛЬТАТ ===

+-----+---------+
|   x |       y |
+=====+=========+
| 0   | 1       |
+-----+---------+
| 0.1 | 1.1     |
+-----+---------+
| 0.2 | 1.199   |
+-----+---------+
| 0.3 | 1.29492 |
+-----+---------+
| 0.4 | 1.38556 |
+-----+---------+
| 0.5 | 1.4687  |
+-----+---------+
| 0.6 | 1.54213 |
+-----+---------+
| 0.7 | 1.60382 |
+-----+---------+
| 0.8 | 1.65193 |
+-----+---------+
| 0.9 | 1.68497 |
+-----+---------+
| 1   | 1.70182 |
+-----+---------+
```



Результаты, полученные при решении методом Рунге-Кутта:

```
=== РЕЗУЛЬТАТ ===

+-----+---------+
|   x |       y |
+=====+=========+
| 0   | 1       |
+-----+---------+
| 0.1 | 1.09966 |
+-----+---------+
| 0.2 | 1.19722 |
+-----+---------+
| 0.3 | 1.29046 |
+-----+---------+
| 0.4 | 1.37713 |
+-----+---------+
| 0.5 | 1.45499 |
+-----+---------+
| 0.6 | 1.52196 |
+-----+---------+
| 0.7 | 1.57617 |
+-----+---------+
| 0.8 | 1.61607 |
+-----+---------+
| 0.9 | 1.6405  |
+-----+---------+
| 1   | 1.64872 |
+-----+---------+
```



Вывод:

Метод Рунге-Кутта существенно точнее метода Эйлера для того же количества шагов. В инженерных и научных расчетах предпочтительнее использовать метод Рунге-Кутта 4-го порядка, так как он обеспечивает лучший баланс между точностью и вычислительной стоимостью. Метод Эйлера может быть полезен только для грубых оценок или когда вычислительные ресурсы сильно ограничены.

Для того чтобы получить близкий результат, нужно существенно увеличить количество шагов для метода Эйлера.

При 500 шагах:

```
Слишком много шагов. Показано только последнее значение:
+-----+---------+
|   x |       y |
+=====+=========+
|   1 | 1.64982 |
+-----+---------+
```

Контрольный пример 2

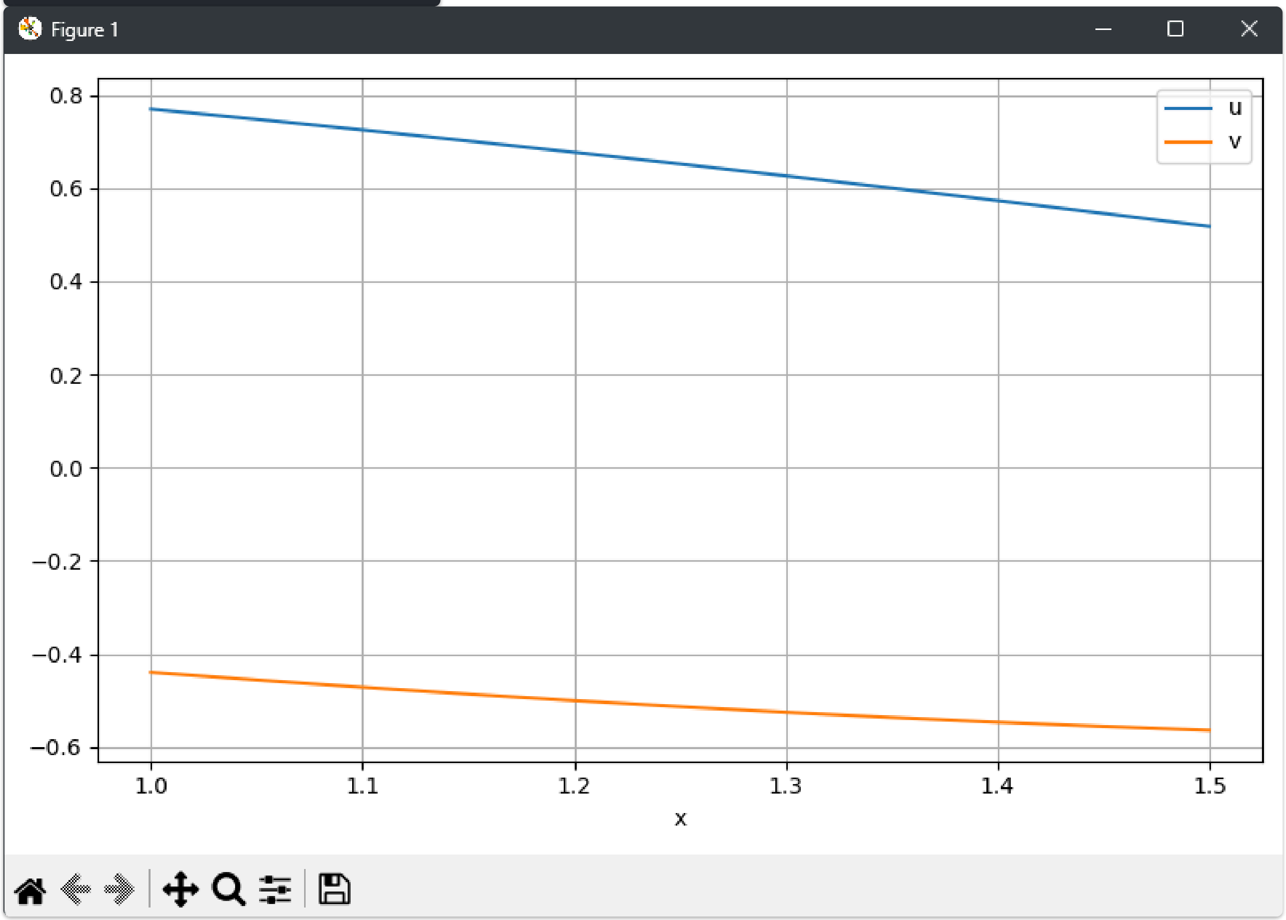
$$y^{``} + \frac{y^{`}}{x} + y = 0$$

$$a = 1, b = 1.5, y_0 = 0.77, y_0^{`} = -0.44$$

Результаты, полученные при решении методом Эйлера:

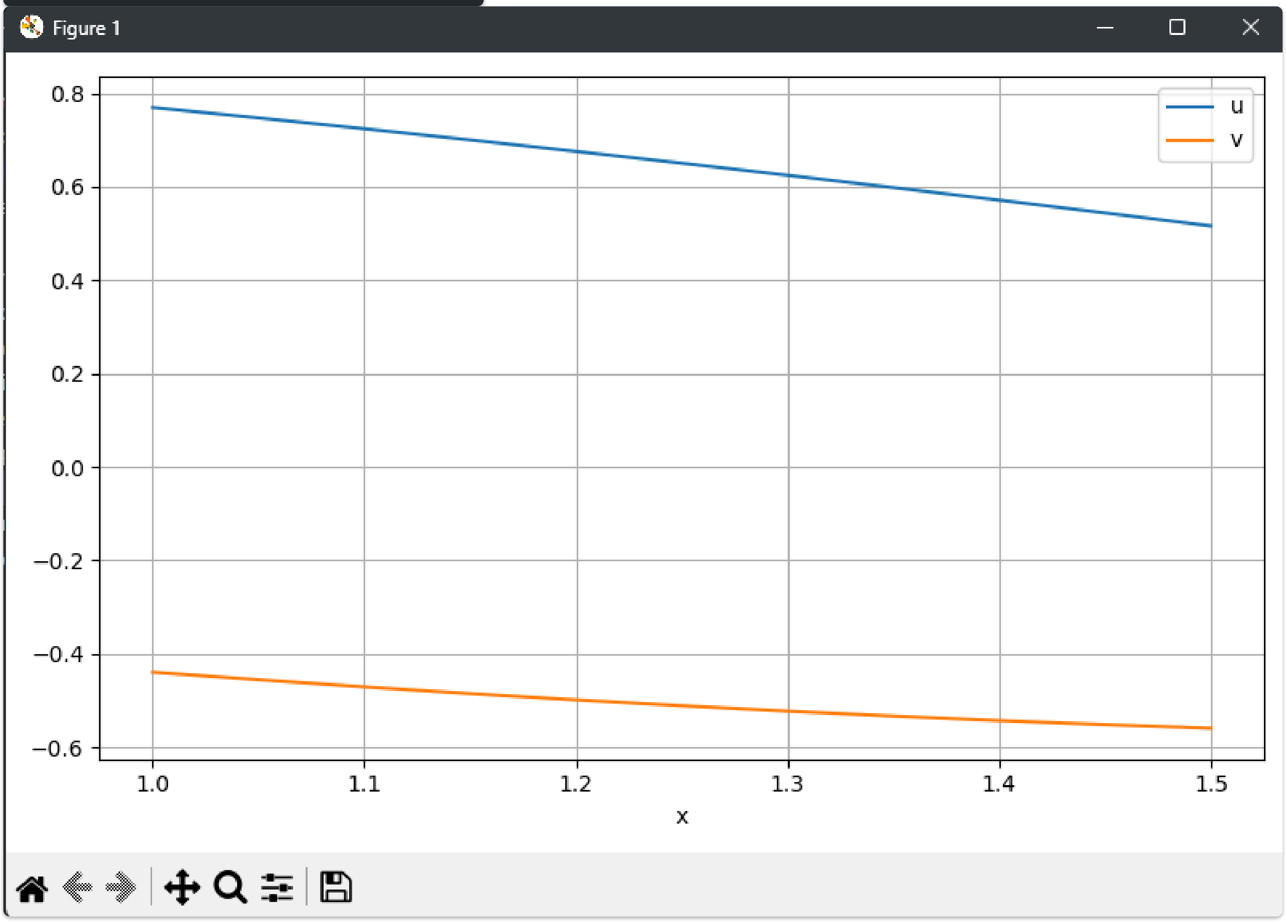
=== РЕЗУЛЬТАТ ===

x	u	v
1	0.77	-0.44
1.05	0.748	-0.4565
1.1	0.725175	-0.472162
1.15	0.701567	-0.486959
1.2	0.677219	-0.500865
1.25	0.652176	-0.513857
1.3	0.626483	-0.525911
1.35	0.600187	-0.537008
1.4	0.573337	-0.547128
1.45	0.545981	-0.556255
1.5	0.518168	-0.564372



Результаты, полученные при решении методом Рунге-Кутта:

```
=== РЕЗУЛЬТАТ ===

+------+----------+-----------+
|    x |        u |         v |
+======+==========+===========+
| 1    | 0.77     | -0.44     |
+------+----------+-----------+
| 1.05 | 0.747594 | -0.456083 |
+------+----------+-----------+
| 1.1  | 0.724406 | -0.471314 |
+------+----------+-----------+
| 1.15 | 0.700478 | -0.485669 |
+------+----------+-----------+
| 1.2  | 0.675854 | -0.499122 |
+------+----------+-----------+
| 1.25 | 0.650581 | -0.511653 |
+------+----------+-----------+
| 1.3  | 0.624704 | -0.523242 |
+------+----------+-----------+
| 1.35 | 0.598273 | -0.533869 |
+------+----------+-----------+
| 1.4  | 0.571334 | -0.543518 |
+------+----------+-----------+
| 1.45 | 0.543937 | -0.552175 |
+------+----------+-----------+
| 1.5  | 0.516133 | -0.559827 |
+------+----------+-----------+
```



Вывод:

Метод Рунге-Кутта показывает более детальную картину - решение не просто плавно уменьшается, а колеблется. Это похоже на правду для таких уравнений.

Метод Эйлера слишком упрощает - он показывает только общую тенденцию уменьшения, но пропускает важные детали поведения решения.

Вывод: Если нужен быстрый приблизительный ответ - подойдет метод Эйлера. Если важна точность и все особенности решения - лучше использовать метод Рунге-Кутта.

Для получения близкого результата методом Эйлера пришлось увеличить количество шагов в 10 раз:

```
+-------+----------+-----------+
| 1.5   | 0.516323 | -0.560278 |
+-------+----------+-----------+
```

Контрольный пример 3

$$\frac{dy}{dt} = -2x + 5z$$

$$\frac{dy}{dt} = \sin(t-1)x - y + 3z$$

$$\frac{dz}{dt} = -x + 2z$$

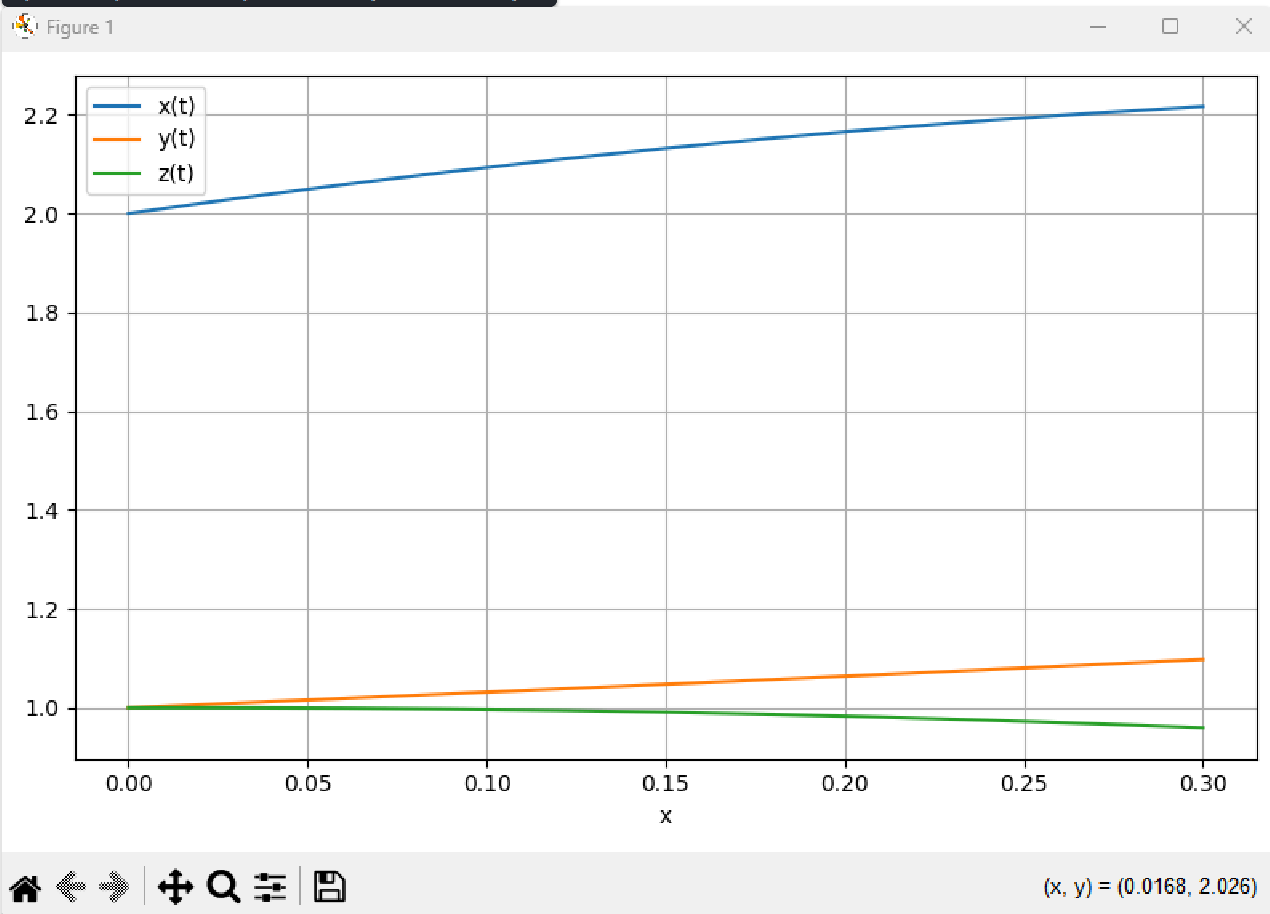
$$x_0 = 2, y_0 = 1, z_0 = 1$$

$$a = 0, b = 0.03$$

Результаты, полученные при решении методом Эйлера:

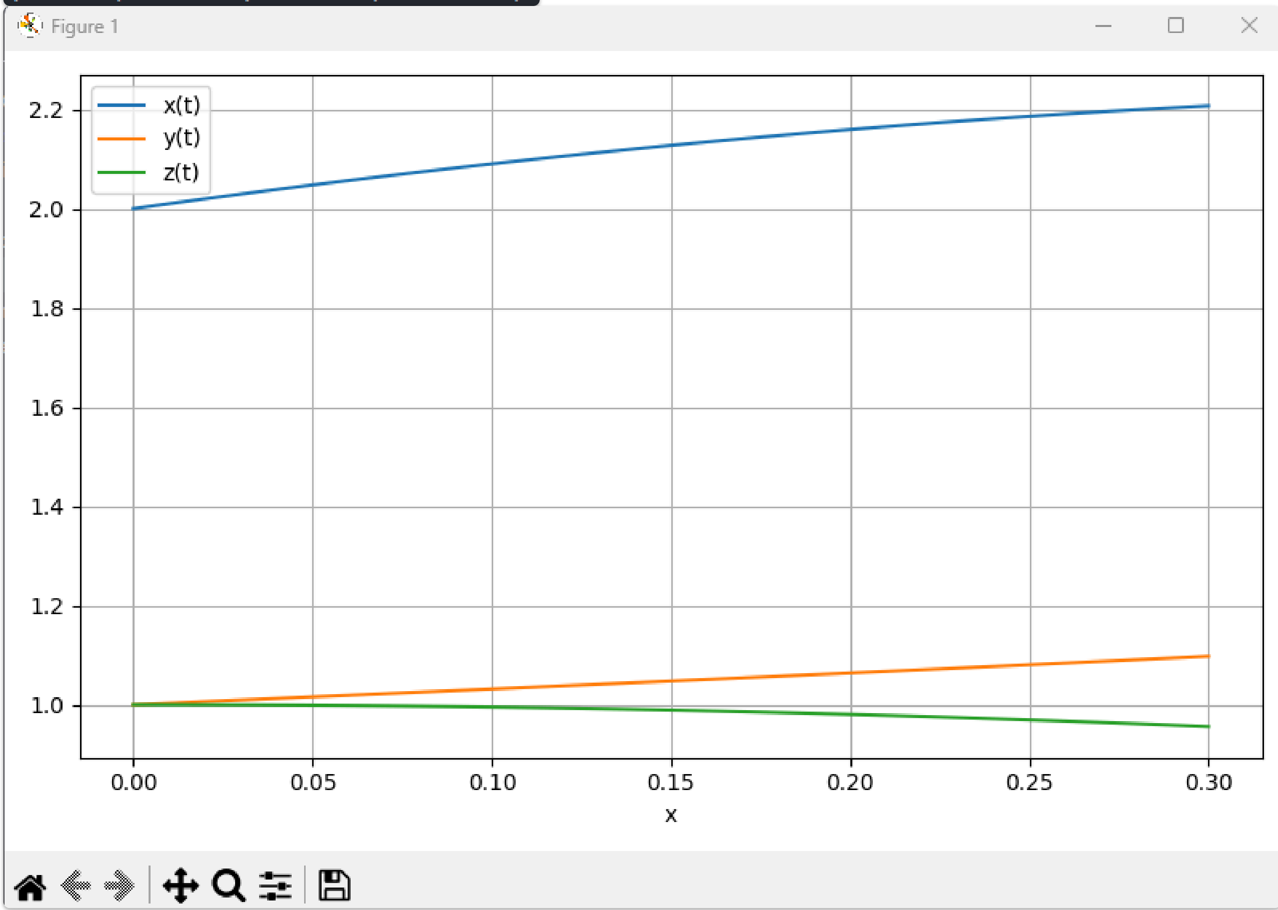
=== РЕЗУЛЬТАТ ===

x	x(t)	y(t)	z(t)
0	2	1	1
0.03	2.03	1.00951	1
0.06	2.0582	1.01899	0.9991
0.09	2.08457	1.02848	0.9973
0.12	2.10909	1.03801	0.994601
0.15	2.13174	1.04761	0.991004
0.18	2.15248	1.05733	0.986512
0.21	2.17131	1.06718	0.981128
0.24	2.1882	1.0772	0.974857
0.27	2.20314	1.08739	0.967702
0.3	2.21611	1.09779	0.95967



Результаты, полученные при решении методом Эйлера h/10:

0.258	2.1898	1.08303	0.967277
0.261	2.19117	1.08405	0.966511
0.264	2.19252	1.08507	0.965736
0.267	2.19385	1.08609	0.964953
0.27	2.19516	1.08712	0.964162
0.273	2.19645	1.08814	0.963361
0.276	2.19773	1.08916	0.962552
0.279	2.19898	1.09019	0.961734
0.282	2.20021	1.09122	0.960907
0.285	2.20142	1.09226	0.960072
0.288	2.20261	1.09329	0.959228
0.291	2.20379	1.09433	0.958376
0.294	2.20494	1.09536	0.957515
0.297	2.20607	1.0964	0.956645
0.3	2.20719	1.09745	0.955767



Вывод:

При уменьшении шага в 10 раз:

1. x(t) немного уменьшился: с 2.216 до 2.207
2. y(t) почти не изменился: 1.098 vs 1.097
3. z(t) немного уменьшился: с 0.960 до 0.956

Что это значит:

- Решение **стало точнее** при меньшем шаге
- Изменения небольшие, значит метод Эйлера с шагом 0.03 уже давал неплохое приближение
- **Меньший шаг = выше точность**, но больше вычислений

Вывод: Для этой системы уменьшение шага не сильно меняет результат - оба варианта дают близкие значения. Но в целом метод с меньшим шагом надежнее.