**Задание:**

1. Решить систему уравнений (по варианту в лр 6) 3 методами.
2. Выполнить оценку методов (количество итераций, невязка* (см. стр 3), погрешность).
3. Оценить вычислительную сложность каждого метода.

Система уравнений по варианту:

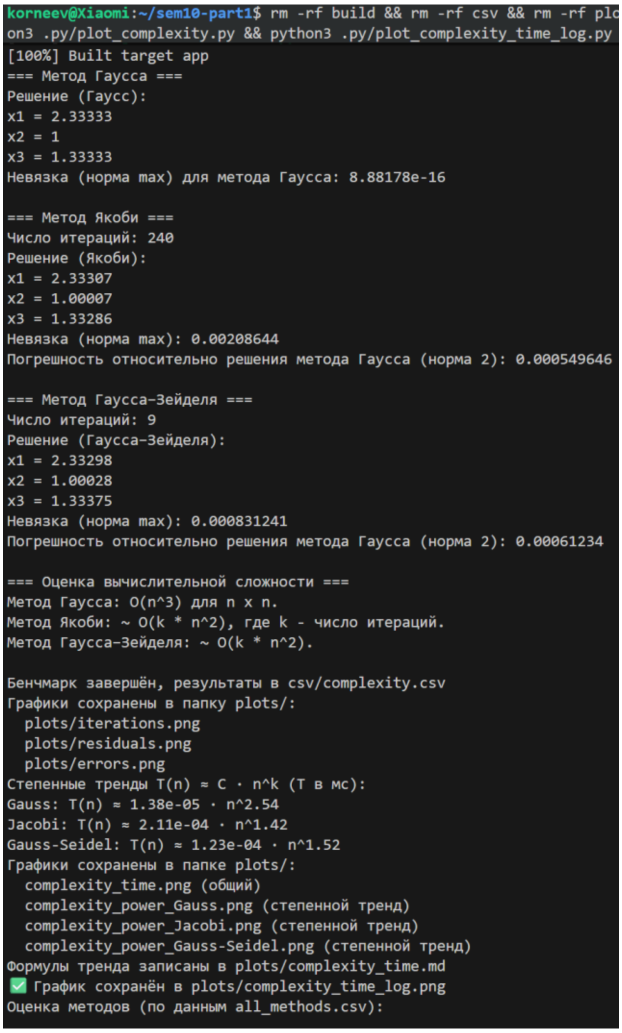
**Система 14**

$$\begin{cases} 4x - y + 2z = 11 \\ x + 3y - z = 4 \\ 2x + y + z = 7 \end{cases}$$

**Решение:** $x = \frac{7}{3}, y = 1, z = \frac{4}{3}$

## Выполнение работы

Вывод в терминале VS Code представлен на рисунке 1:



```
korneev@Xiaomi:~/sem10-part1$ rm -rf build && rm -rf csv && rm -rf plo
on3 .py/plot_complexity.py && python3 .py/plot_complexity_time_log.py
[100%] Built target app
=== Метод Гаусса ===
Решение (Гаусс):
x1 = 2.33333
x2 = 1
x3 = 1.33333
Невязка (норма max) для метода Гаусса: 8.88178e-16

=== Метод Якоби ===
Число итераций: 240
Решение (Якоби):
x1 = 2.33307
x2 = 1.00007
x3 = 1.33286
Невязка (норма max): 0.00208644
Погрешность относительно решения метода Гаусса (норма 2): 0.000549646

=== Метод Гаусса-Зейделя ===
Число итераций: 9
Решение (Гаусса-Зейделя):
x1 = 2.33298
x2 = 1.00028
x3 = 1.33375
Невязка (норма max): 0.000831241
Погрешность относительно решения метода Гаусса (норма 2): 0.00061234

=== Оценка вычислительной сложности ===
Метод Гаусса: O(n^3) для n x n.
Метод Якоби: ~ O(k * n^2), где k - число итераций.
Метод Гаусса-Зейделя: ~ O(k * n^2).

Бенчмарк завершён, результаты в csv/complexity.csv
Графики сохранены в папку plots/:
  plots/iterations.png
  plots/residuals.png
  plots/errors.png
Степенные тренды T(n) ≈ C · n^k (T в мс):
Gauss: T(n) ≈ 1.38e-05 · n^2.54
Jacobi: T(n) ≈ 2.11e-04 · n^1.42
Gauss-Seidel: T(n) ≈ 1.23e-04 · n^1.52
Графики сохранены в папке plots/:
  complexity_time.png (общий)
  complexity_power_Gauss.png (степенной тренд)
  complexity_power_Jacobi.png (степенной тренд)
  complexity_power_Gauss-Seidel.png (степенной тренд)
Формулы тренда записаны в plots/complexity_time.md
✅ График сохранён в plots/complexity_time_log.png
Оценка методов (по данным all_methods.csv):
```

*Рисунок 1 – вывод в терминале*

Метод Гаусса представлен на рисунке 2.

```
=== Метод Гаусса ===
Решение (Гаусс):
x1 = 2.33333
x2 = 1
x3 = 1.33333
Невязка (норма max) для метода Гаусса: 8.88178e-16
```

*Рисунок 2 – метод Гаусса*

Вычислительная сложность данного подхода: O(n^3).

Метод Гаусса-Зейделя представлен на рисунке 3.

```
=== Метод Гаусса-Зейделя ===
Число итераций: 9
Решение (Гаусса-Зейделя):
x1 = 2.33298
x2 = 1.00028
x3 = 1.33375
Невязка (норма max): 0.000831241
Погрешность относительно решения метода Гаусса (норма 2): 0.00061234
```

*Рисунок 3 – метод Гаусса-Зейделя*

Вычислительная сложность: O(k * n^2), где k - число итераций.

Метод Якоби представлен на рисунке 4.

```
=== Метод Якоби ===
Число итераций: 240
Решение (Якоби):
x1 = 2.33307
x2 = 1.00007
x3 = 1.33286
Невязка (норма max): 0.00208644
```

*Рисунок 4 – метод Якоби*

Вычислительная сложность: O(n^2 × k).

На рисунке 5 представлены степенные тренды времени выполнения сгенерированные с помощью Python, а на рисунке 6 советующие формулы к ним.

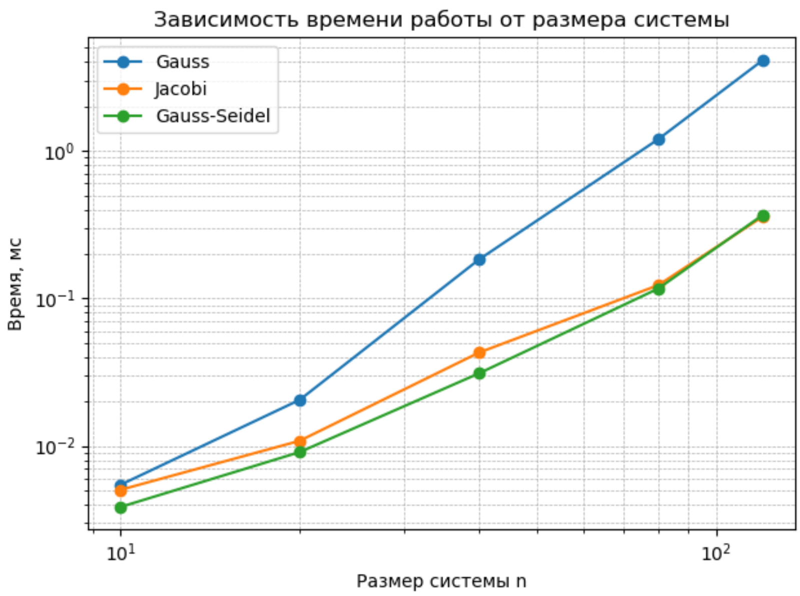
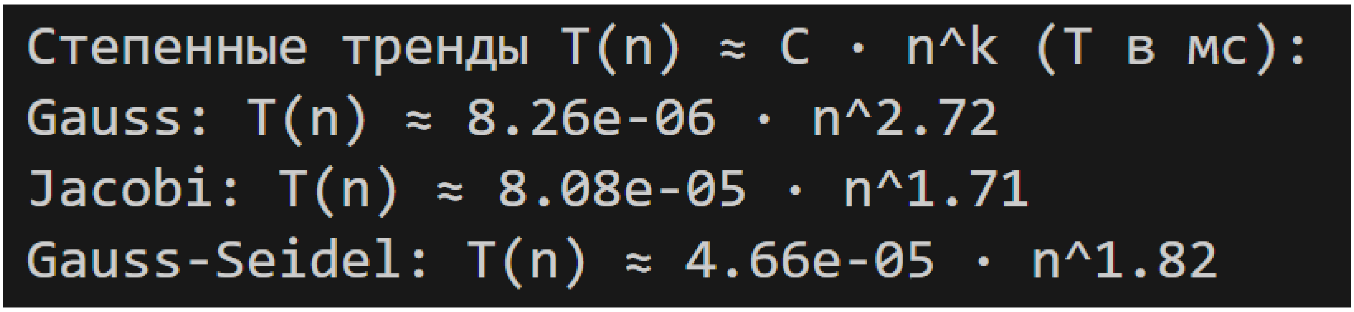



*Рисунок 5 – степенные тренды времени выполнения*



```
Степенные тренды T(n) ≈ C · n^k (T в мс):
Gauss: T(n) ≈ 8.26e-06 · n^2.72
Jacobi: T(n) ≈ 8.08e-05 · n^1.71
Gauss-Seidel: T(n) ≈ 4.66e-05 · n^1.82
```

*Рисунок 6 – формулы степенных трендов времени выполнения*

Сравнение вычислительной сложности методов представлено на рисунке 7.

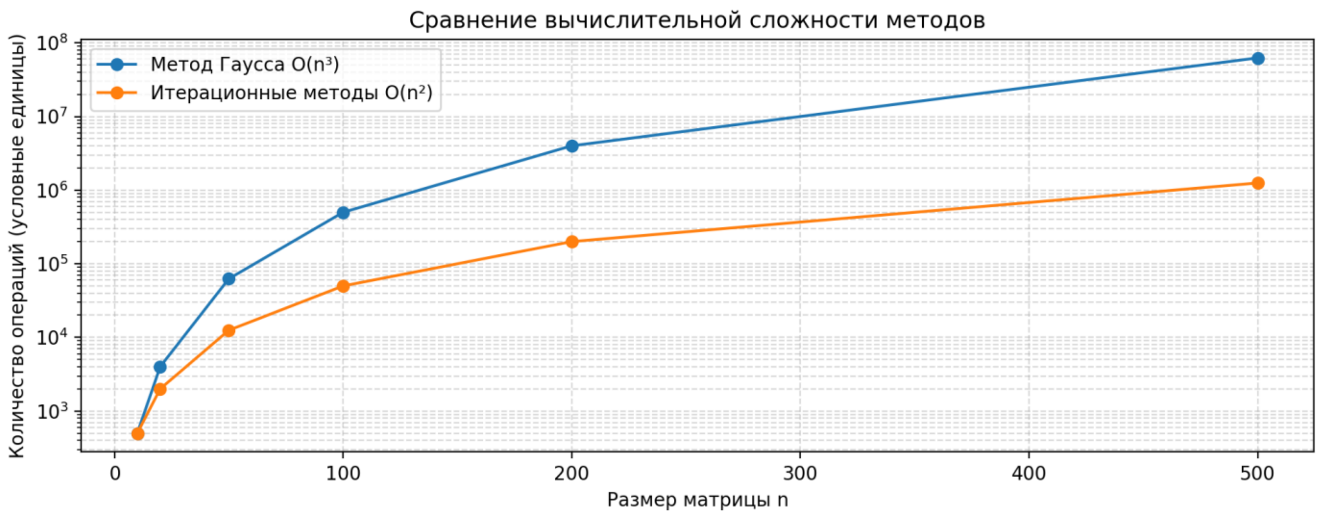



*Рисунок 7 – вычислительной сложности методов*

Логарифмическая шкала времени выполнения представлена на рисунке 8.

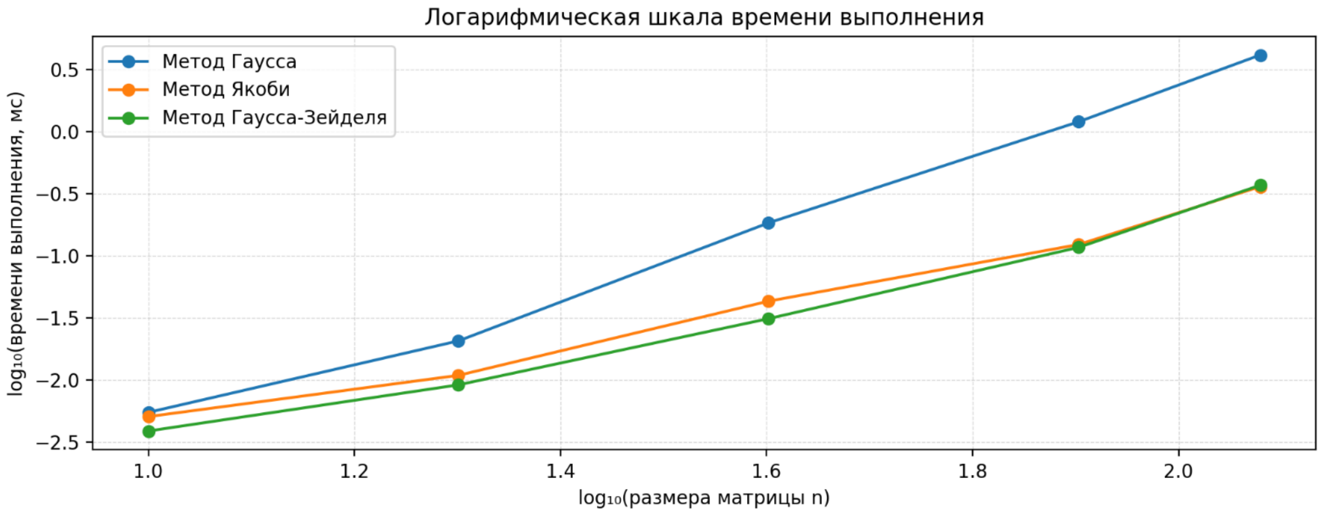



*Рисунок 8 – логарифмическая шкала времени выполнения*

На рисунке 9 представлена таблица оценки точности и сходимости методов решения системы сгенерированная .py скриптом из csv таблиц.

```
Оценка методов (по данным all_methods.csv):

Метод               Итерации          Невязка      Погрешность
--------------------------------------------------------------
Gauss                      0        8.882e-16        0.000e+00
Jacobi                   240        2.086e-03        5.496e-04
Gauss-Seidel               9        8.312e-04        6.123e-04
korneev@Xiaomi:~/sem10-part1$
```

*Рисунок 9 – Таблица оценки точности и сходимости методов решения систем.*

Таблица показывает, что метод Гаусса даёт минимальную невязку и точное решение, тогда как итерационные методы имеют большую ошибку. Метод Гаусса–Зейделя сходится быстрее Якоби, но оба зависят от свойств матрицы.

**Вывод:** В работе были реализованы три метода решения СЛАУ: Гаусса, Якоби и Гаусса–Зейделя. Эксперименты и логарифмические графики показали, что время работы метода Гаусса растёт кубически, тогда как итерационные методы увеличивают время значительно медленнее. Метод Гаусса дал точное решение с минимальной невязкой. Итерационные методы для данной системы не сошлись из-за отсутствия диагонального преобладания, что выразилось в росте ошибки. В итоге метод Гаусса оказался наиболее надёжным для текущей задачи, а итерационные методы требуют других свойств матрицы для корректной работы.