

Сравнение быстродействия алгоритмов точного и численного решения систем линейных алгебраических уравнений

И.И. Сергеев, И.Д. Фёдоров

научный руководитель А.В. Фаворская

МФТИ, 23 ноября 2016

Системы линейных алгебраических уравнений (СЛАУ)

Рассматриваемая система уравнений:

$$\begin{cases} a_{11}u_1 + \dots + a_{1n}u_n = f_1 \\ \dots \\ a_{n1}u_1 + \dots + a_{nn}u_n = f_n \end{cases}$$

Матричная запись:

$$Au = f$$

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$$

Метод Гаусса

- Прямой ход

С помощью вычитания строк матрица приводится к верхнетреугольному виду

- Обратный ход

По очереди разрешаются уравнения, начиная с последнего

- Выбор главного элемента \Rightarrow повышение точности

На i -ом шаге i -ая строка переставляется со строкой, содержащей наибольшее значение в i -ом столбце

Итерационные методы

Метод Якоби

$$u_i^{k+1} = \frac{1}{a_{ii}} \left(f_i - \sum_{j=1}^{i-1} a_{ij} u_j^k - \sum_{j=i+1}^n a_{ij} u_j^k \right)$$

Метод Гаусса-Зейделя

$$u_i^{k+1} = \frac{1}{a_{ii}} \left(f_i - \sum_{j=1}^{i-1} a_{ij} u_j^{k+1} - \sum_{j=i+1}^n a_{ij} u_j^k \right)$$

Метод релаксации

$$u_i^{k+1} = (1 - \tau) u_i^k + \frac{\tau}{a_{ii}} \left(f_i - \sum_{j=1}^{i-1} a_{ij} u_j^{k+1} - \sum_{j=i+1}^n a_{ij} u_j^k \right)$$

Цели

- Сравнение быстродействия методов
- Изучение зависимости скорости работы
 1. от параметра τ (для методов релаксации)
 2. от числа обусловленности μ

Генерирование матриц с числом обусловленности, точно равным заданному

- Число обусловленности по определению

$$\mu = \|A^{-1}\| \cdot \|A\|$$

- Норма матрицы

$$\|A\| = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$$

- Алгоритм генерации:

1. B – целиком случайная, ортогонализованная по Грамму-Шмидту
2. C – случайная диагональная, фиксированы минимум – 1, максимум -- μ
3. $A = B'C$

Генерирование матриц с числом обусловленности, точно равным заданному

- Преимущества:

1. Точное значение μ

- Недостатки:

1. Сложность процесса ортогонализации – $O(n^3)$
2. Отсутствие диагонального преобладания

Генерирование матриц с числом обусловленности, приблизительно равным заданному

- Оценочная формула:

$$\mu \geq \frac{\max_{1 \leq i \leq n} |a_{ii}|}{\min_{1 \leq i \leq n} |a_{ii}|}$$

- Алгоритм генерации:

1. Ниже диагонали – симметрично
2. По диагонали – фиксированы минимум 1, максимум μ
3. Выше диагонали – случайные

Генерирование матриц с числом обусловленности, приблизительно равным заданному

- Преимущества:

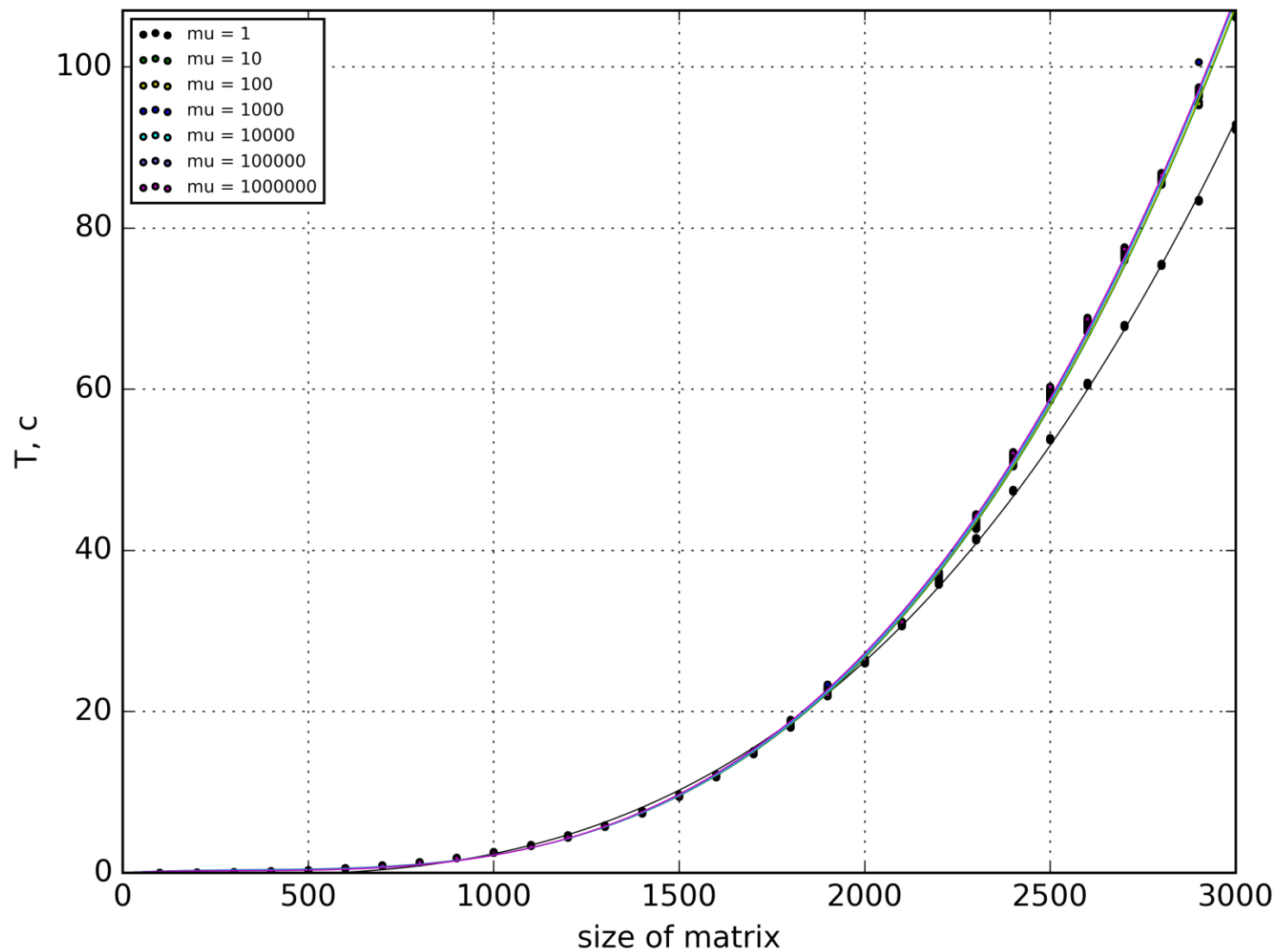
1. Сложность -- $O(n^2)$
2. Диагональное преобладание

- Недостатки

1. Узкий класс матриц
2. Неточное значение μ (отклонение $<0.3\%$)

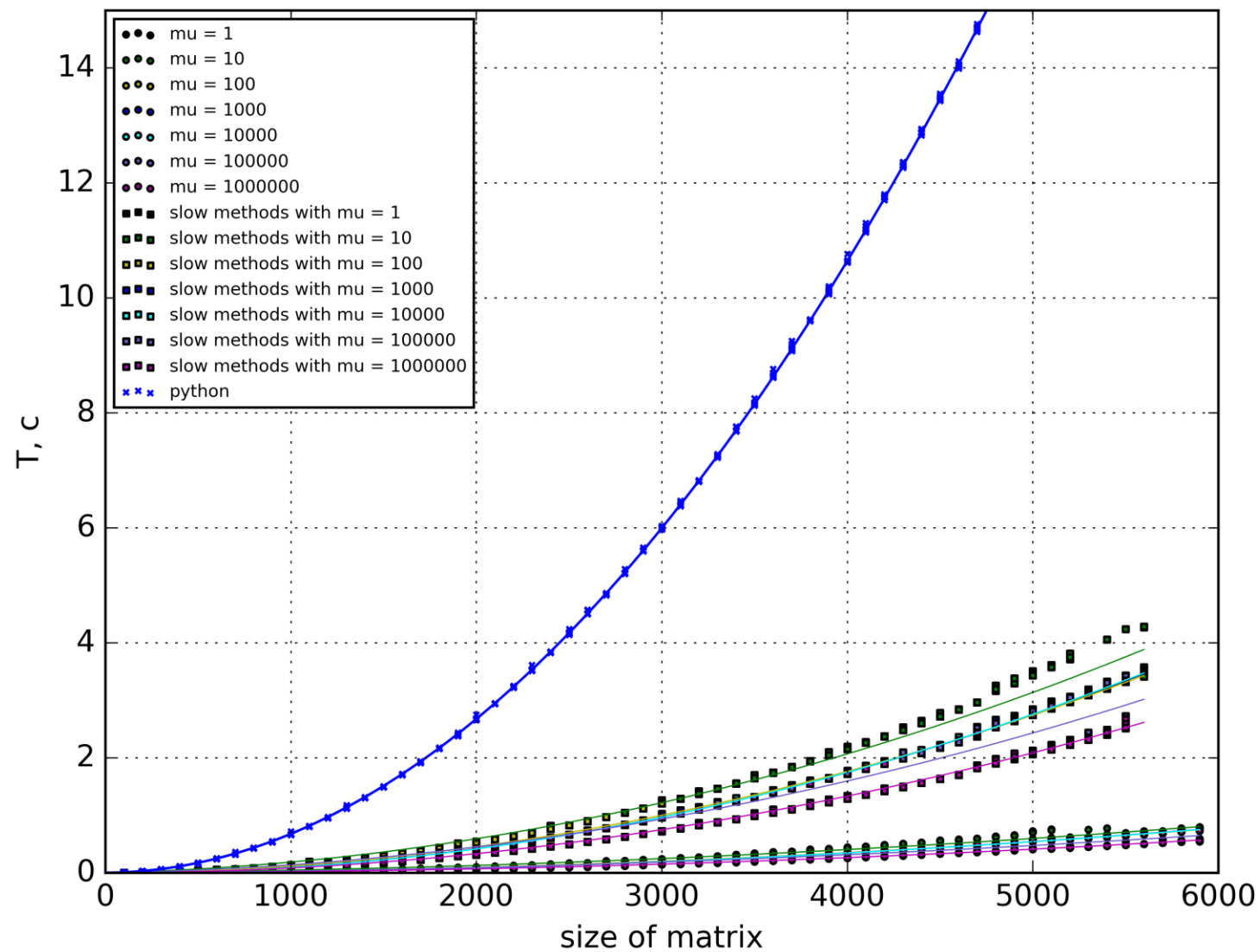
Результаты

Скорость работы метода Гаусса



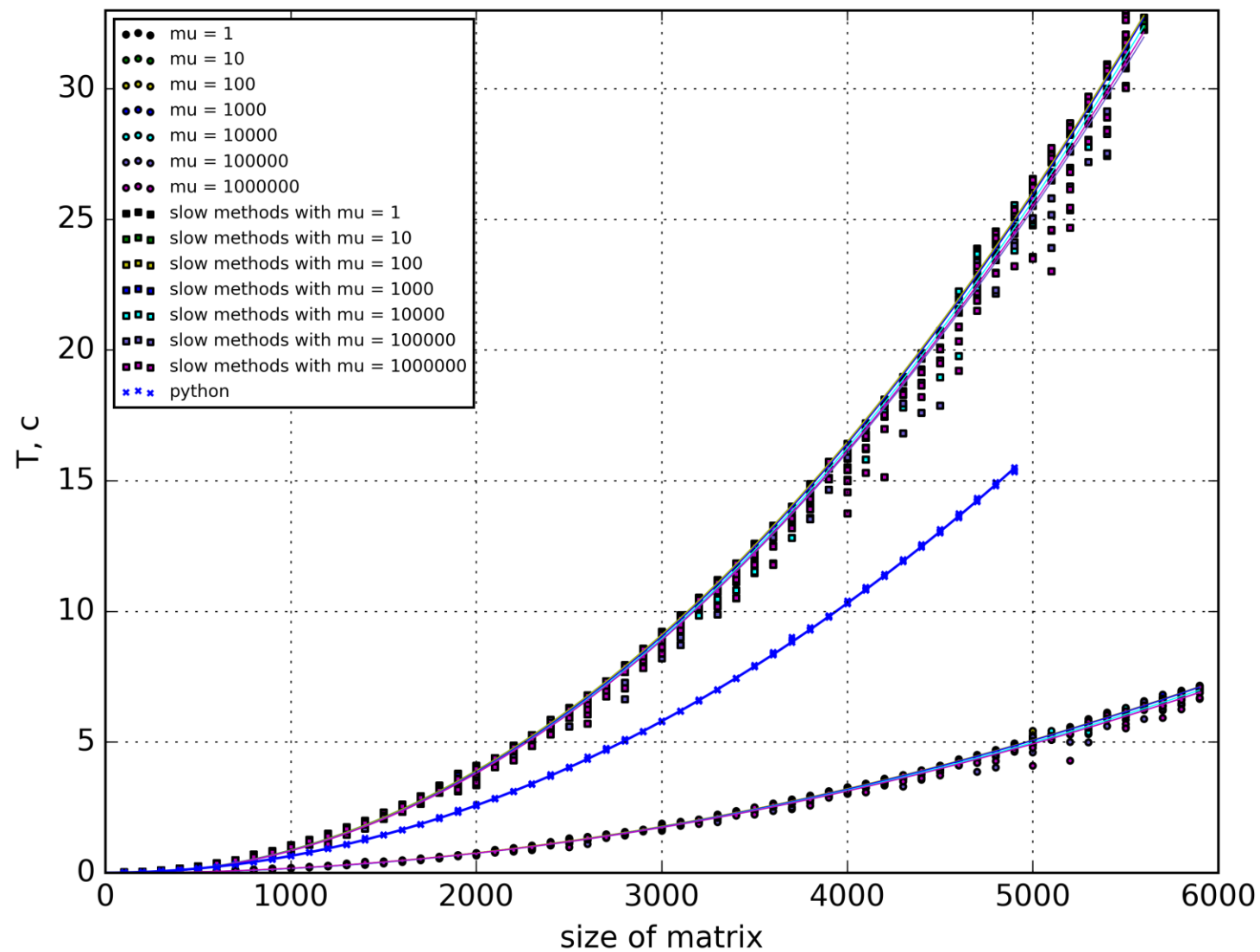
Результаты

Скорость работы метода Якоби



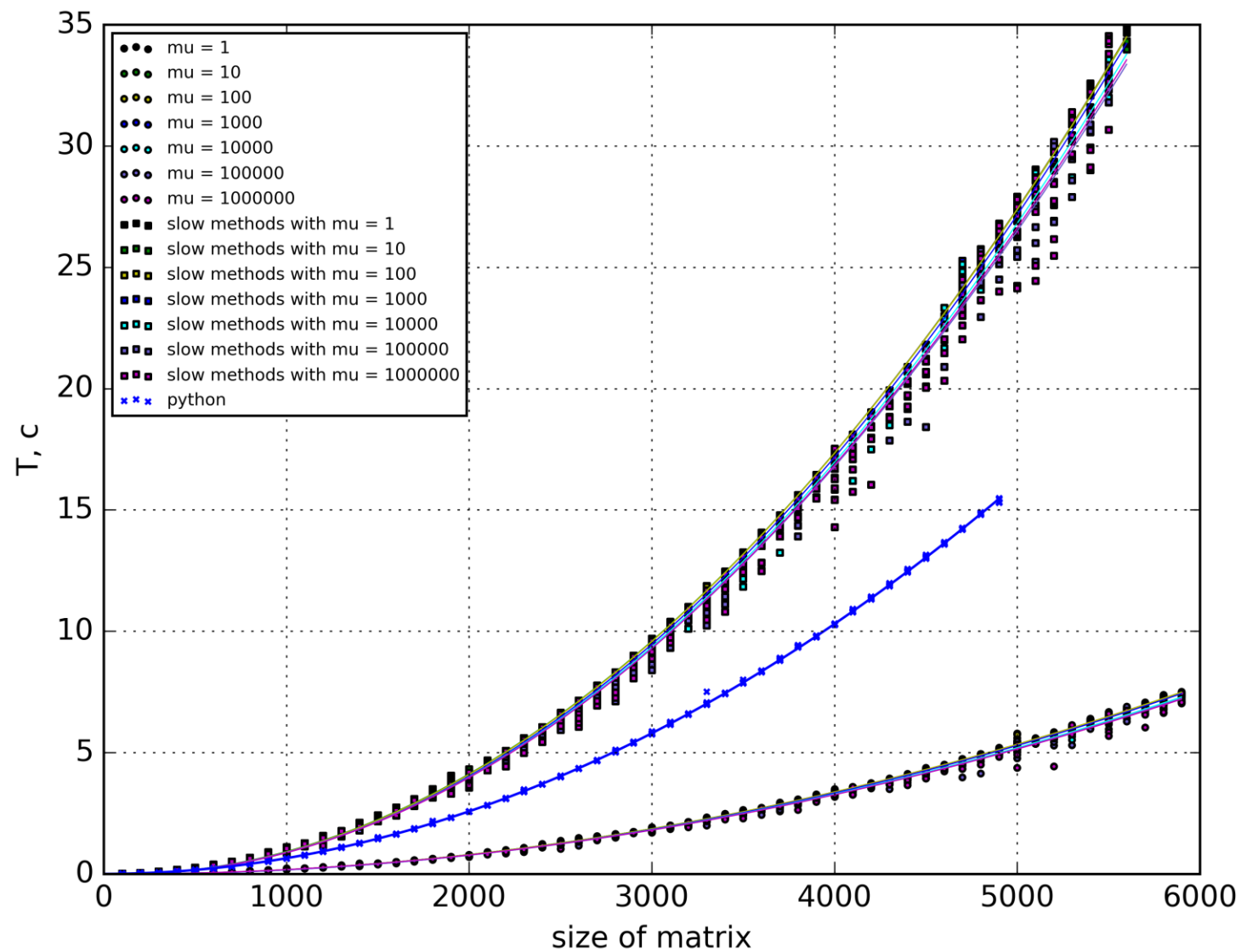
Результаты

Скорость работы метода нижней релаксации



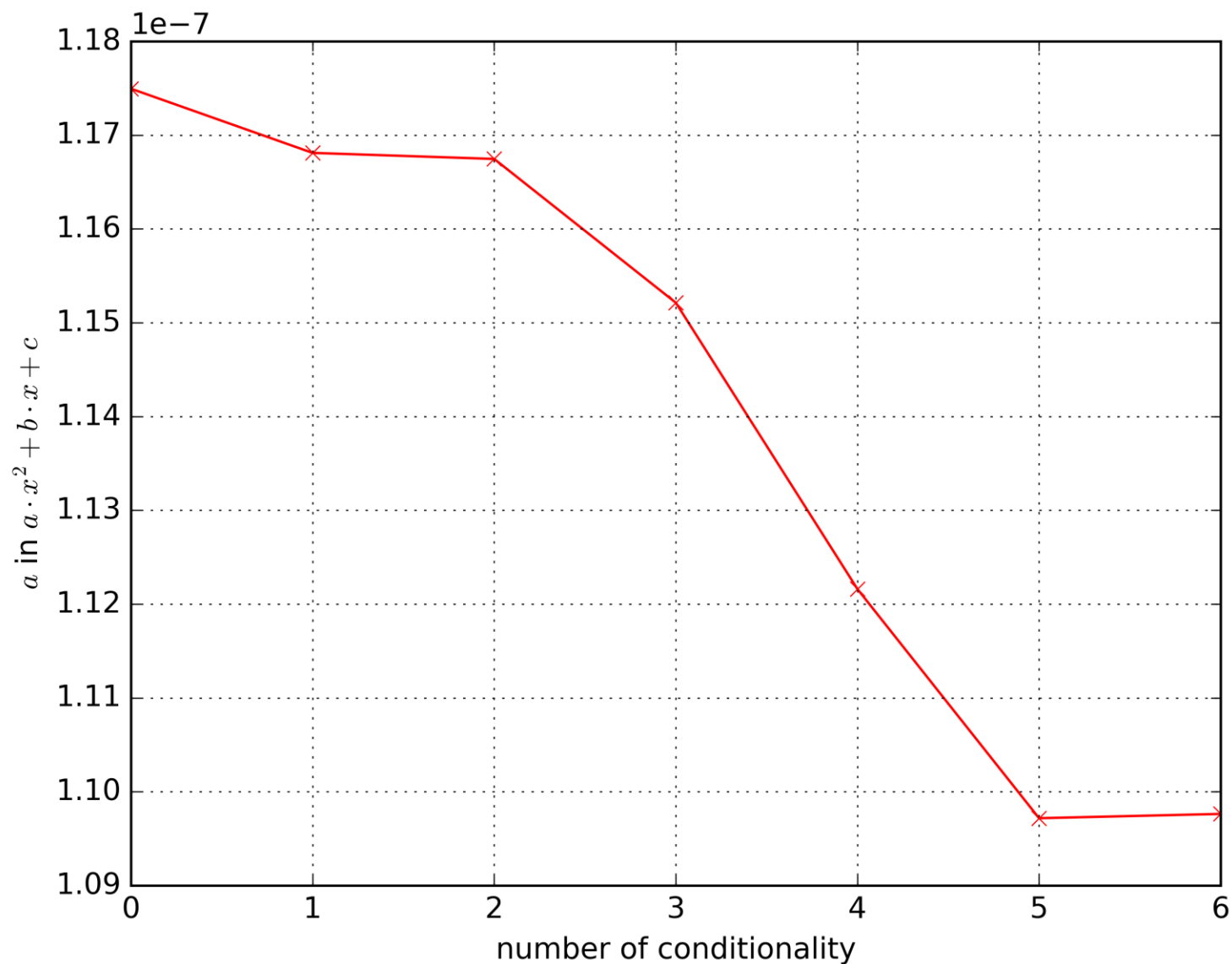
Результаты

Скорость работы метода верхней релаксации, $\tau = 1.5$



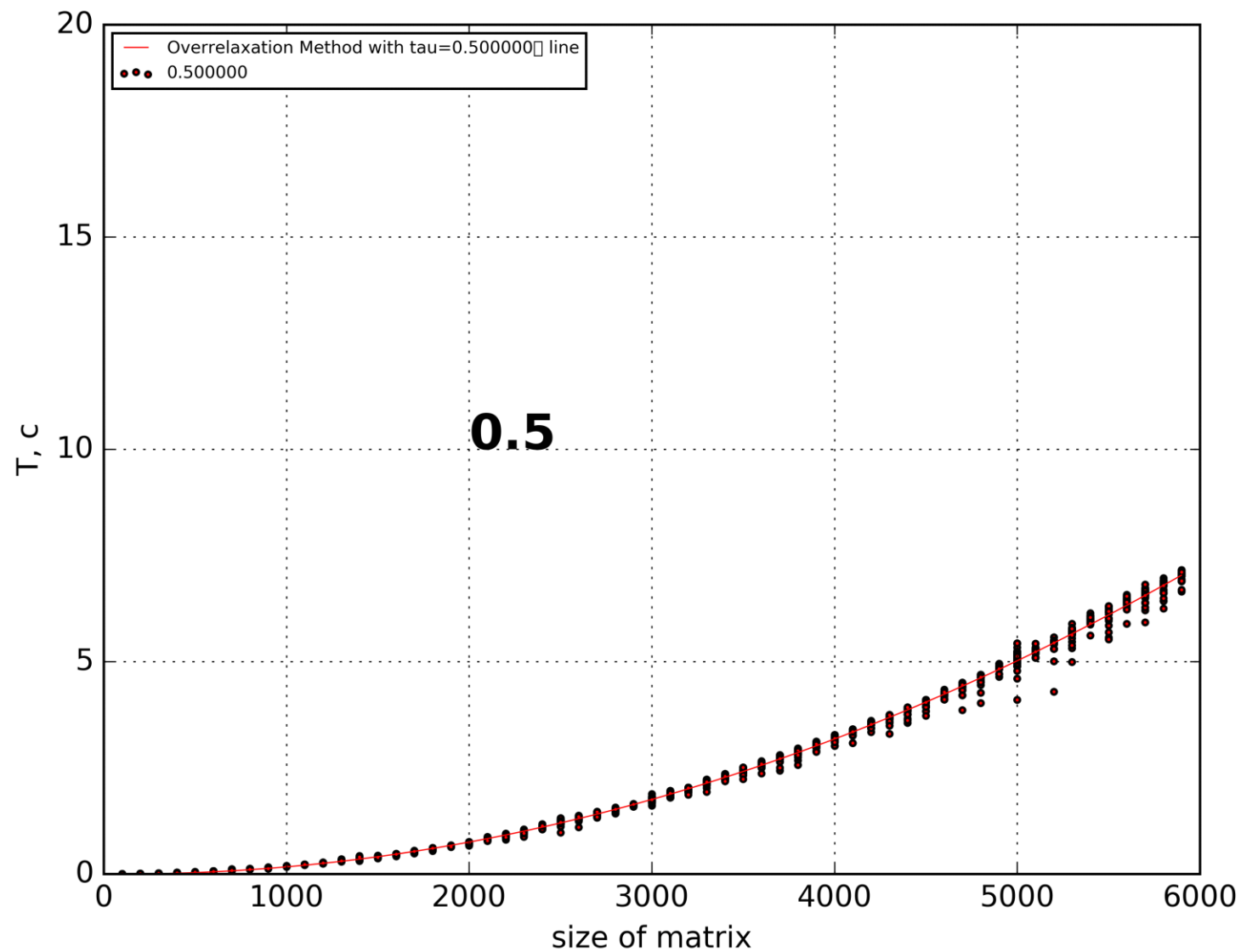
Результаты

Зависимость скорости работы метода верхней релаксации от числа обусловленности, $\tau = 1.25$



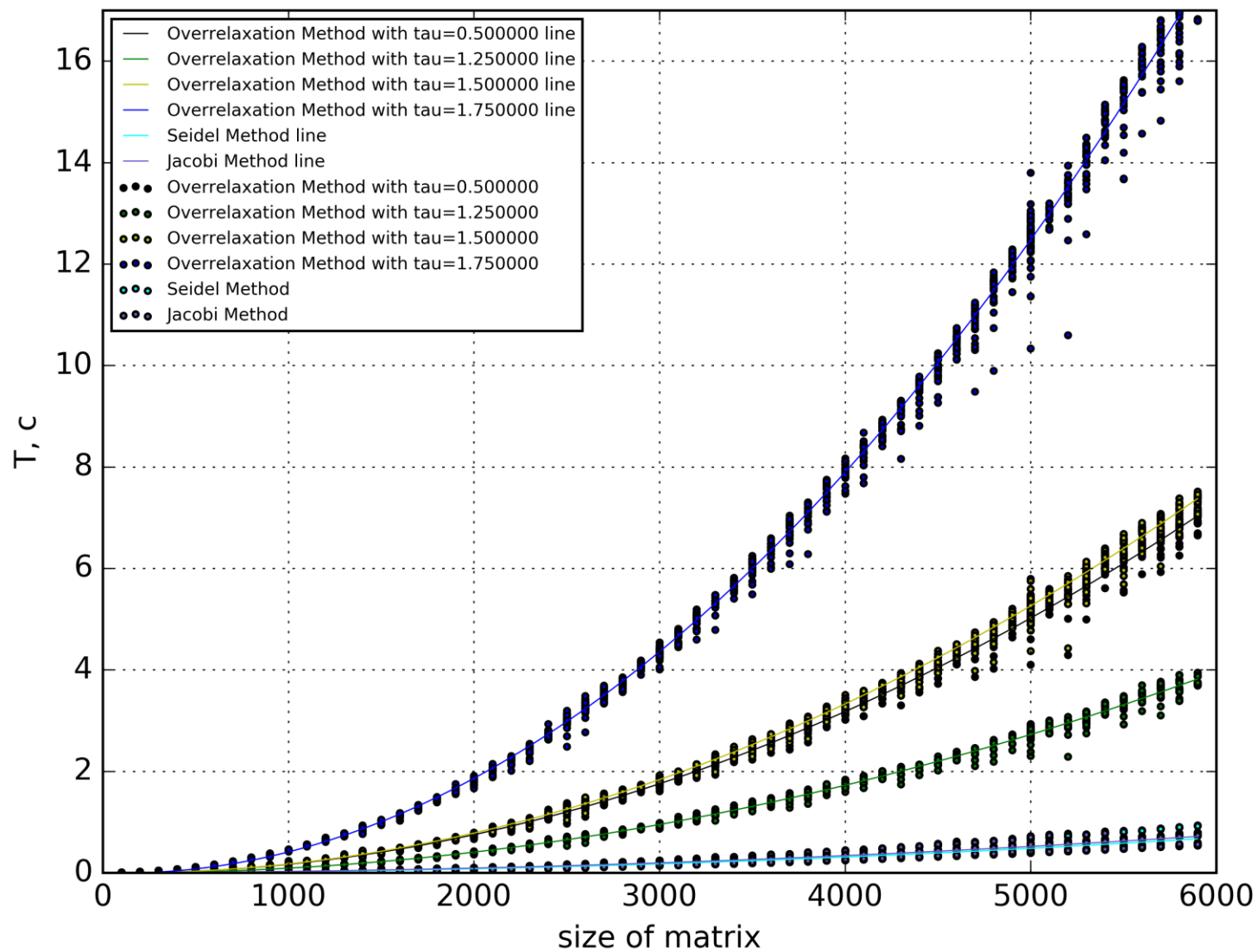
Результаты

Зависимость скорости работы метода релаксации от τ



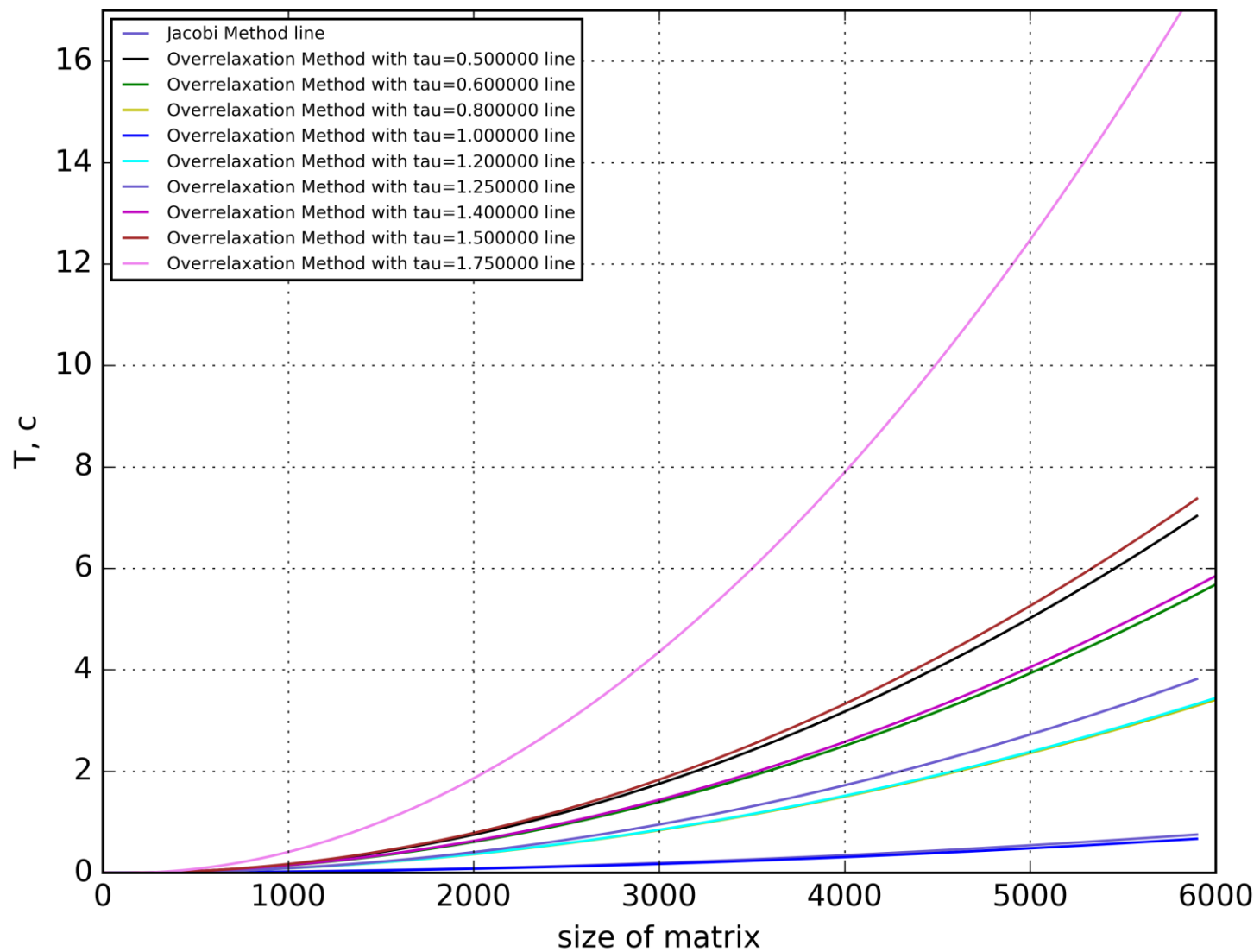
Результаты

Сравнительный график для всех методов



Результаты

Сравнительный график для всех методов без экспериментальных точек



Результаты и выводы

- Была реализована программа, осуществляющая сравнение методов решения СЛАУ
- Теоретическая сложность методов совпала с экспериментальной
 - Для прямого метода Гаусса – $O(n^3)$
 - Для численных методов – $O(n^2)$
- Наибольшей производительностью обладают
 1. Метод Гаусса-Зейделя ($\tau = 1$)
 2. Метод Якоби
- Быстродействие падает с ростом числа обусловленности
- Оптимальное значение параметра релаксации $\tau_{opt} \in (1.0; 2.0)$

Дальнейшее развитие

- Увеличение количества сравниваемых алгоритмов
- Увеличение собираемой статистики
- Нахождение τ_{opt} для верхней релаксации
- Демонстрация к курсу «Вычислительная математика»

[github.com/Ivan-Sergeyev/slae alg comp](https://github.com/Ivan-Sergeyev/slae_alg_comp)

Дополнительная часть

Результаты (Python)

