Лабораторная работа №2

Максимов Артем ИВТ-21М

Решения уравнений методом Гаусса

Метод Гаусса предназначен для решения уравнений.

Уравнение выглядит следующим образом

Также эту систему уравнений можно записать следующим способом

Где

A – матрица коэффициентов

b – вектор свободных членов

x – вектор неизвестных

Теперь после такого описания уравнение метод Гаусса предполагает с помощью элементарных преобразований над строками свести основную матрицу коэффициентов к ступенчатому виду. После этого выполняется обратная подстановка для поиска вектора x с следующей формулой

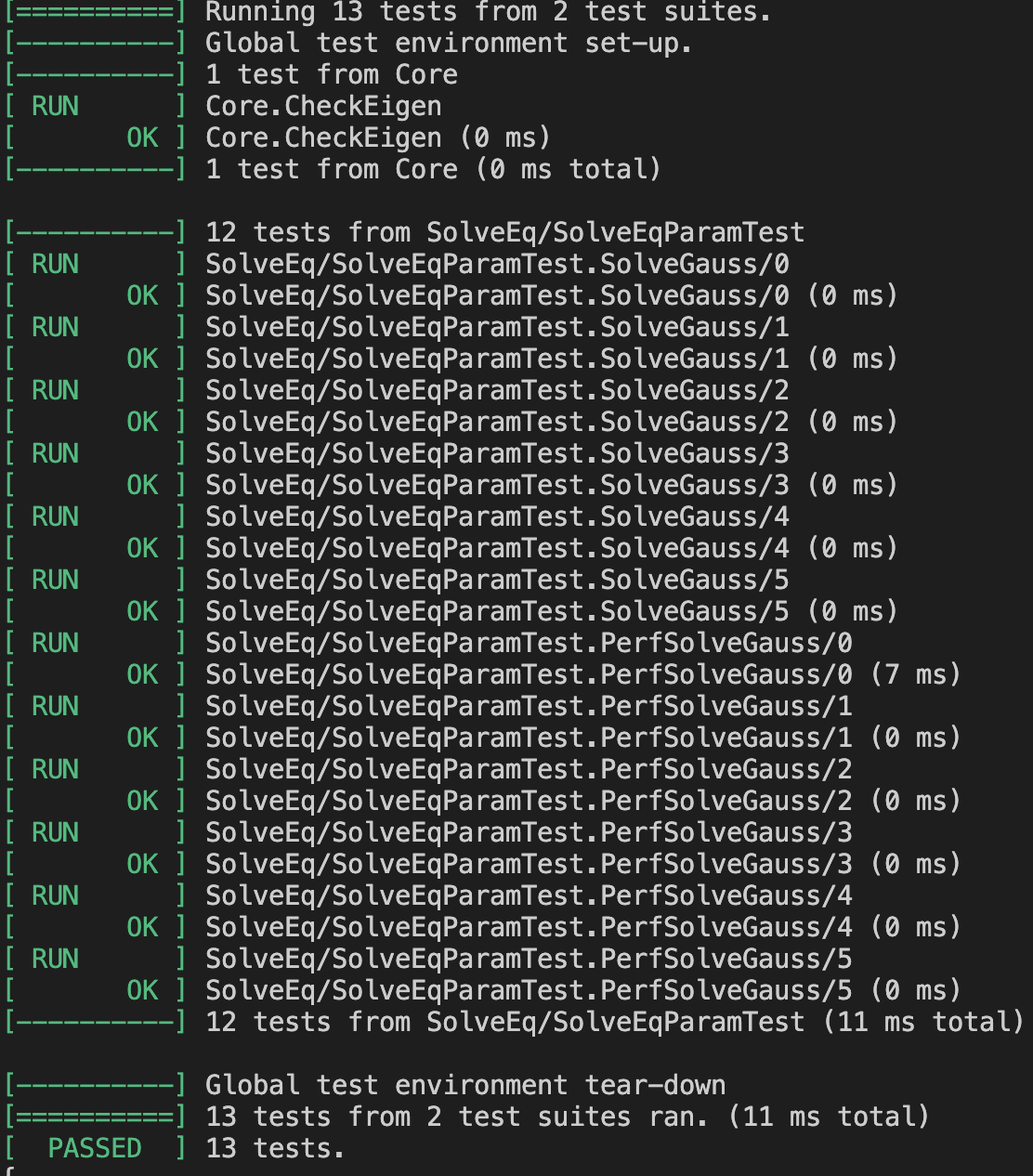
Где — это итоговая полученная ступенчатая матрица из матрицы коэффициентов и также модифицированный вектор свободных членов после преобзования в ступенчатый вид матрица коэффициентов

Основной код находится в *labs/lab2/include/solve\_eq.hpp*

В методе Гаусса основная нагрузка происходит на этапе элиминации, когда в процессе происходит обнуление элементов по диагонали. На этом этапе происходят операции над разными строками, которые является независимыми друг от друга

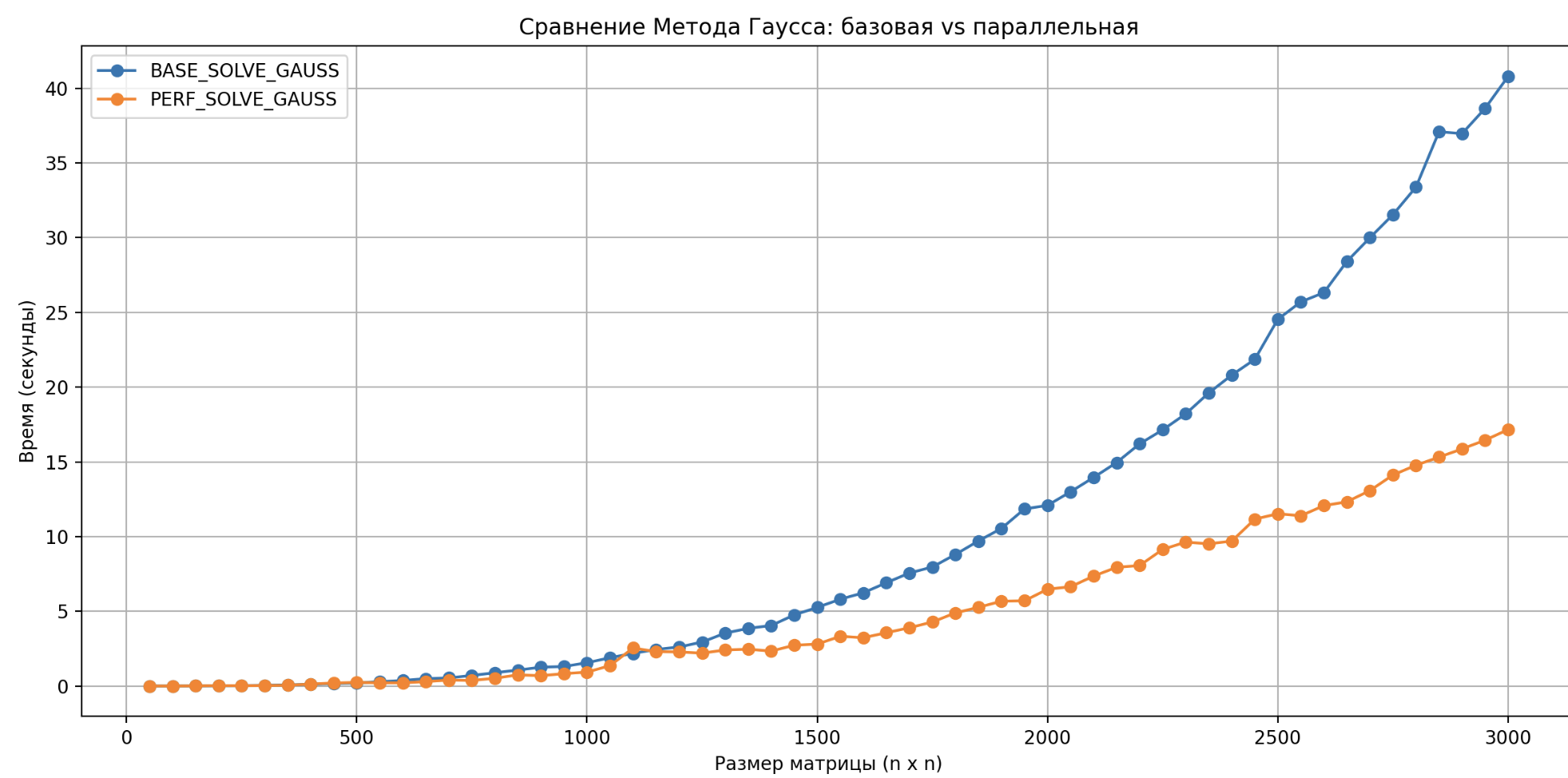
В итоге как видно каждая строка – k обрабатывается независимо, поэтому этап можно распараллелить

Для проверки результатов, полученных из обоих вариантов метода Гаусса, была взята библиотека *Eigen*, где есть метод для вычисления результаты методом Гаусса. В тесте *Core.CheckEigen* была проведена проверка самого вычисления библиотеки и дальше в других тестах полученные результаты сравнивались с результатами полученные из готовой библиотеки. Также для тестов использовалась библиотека gtest. Тесты находятся по пути *labs/lab2/include/test.hpp*



Для снятия изменение производительности использовалась библиотека google benchmark. Тест для перфа находятся в *labs/lab2/include/bench.hpp*

Полученные результаты



В итоге, смотря на график можно сказать, что распараллеливание предложенного участка кода положительно повлияло на производительность. Видно, что при размере меньше, чем 1000 разница не сильно видна, однако стоит также учитывать, что результаты схожи при таком маленьком количество размера и параллельность и ухудшает результат в этот период. После размера 1000 по графику видно, что с увлечением результат, показанный улучшенной версии показывает все больше и больший отрыв от изначальной версии.