



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.Ломоносова



Кафедра ВТМ

Компьютерный практикум по учебному курсу
«Практикум на ЭВМ»

ЗАДАНИЕ № 1

QR-разложение матриц отражениями Хаусхолдера

гор. Москва
2022 год

0.0.1 Постановка задачи

В моем варианте работы мне нужно было реализовать QR-разложение матриц отражениями Хаусхолдера, протестировать на разных входных размерах матриц и рассчитать асимптотическую сложность алгоритма в данной реализации.

0.0.2 Описание программы

Все программы доступны в репозитории

0.0.3 Тесты

Таблица 1: Таблица результатов времени работы (в секундах) для начальной версии алгоритма QR разложения отражениями Хаусхолдера(qr reflect v1.cpp), и этой же версии перенесенной с vector на static arrays(qr reflect static.cpp)

N	Basic method C++	Basic method C++ with optimization (-ofast, -march)	Basic method C++ with static arrays
128	0.045	0.005	0.019
256	0.359	0.063	0.165
512	3.003	0.692	1.445
1024	34.074	7.709	18.503
2048	365.258	89.486	202.263
4096	3123.711	896.840	1753.305

Таблица 2: Таблица результатов времени работы оптимизированного алгоритма(qr reflect final.cpp), включая с++ оптимизацию с ключами -Ofast, -march=native, и начального

N	Basic method + static arrays	Improved method C++ (static arrays)	Improved method C++ with optimization (-ofast, -march)
128	0.019	0.018	0.001
256	0.165	0.137	0.015
512	1.445	1.087	0.119
1024	18.503	8.733	1.350
2048	202.263	69.302	12.197
4096	1753.305	552.585	100.519

Таблица 3: Таблица результатов времени работы оптимизированного алгоритма(qr reflect final.cpp) и алгоритма на fortran(qr reflect fortran.f90), с учетом оптимизаций и без

N	Improved method C++ (static arrays)	Improved method C++ with optimization (-ofast, -march)	Fortran	Fortran with -ofast, -march optimization
128	0.018	0.001	0.0186	0.018
256	0.137	0.015	0.138	0.015
512	1.087	0.119	1.090	0.113
1024	8.733	1.350	8.772	1.333
2048	69.302	12.197	69.734	12.181
4096	552.585	100.519	554.922	100.394

Таблица 4: Таблица результатов времени работы опт. алгоритма на C++ с использованием blas и алгоритма на fortran с использованием blas

N	C++ with blas	Fortran with blas
128	0.006	0.00649
256	0.046	0.0459
512	0.378	0.370
1024	3.282	4.109
2048	26.886	37.111
4096	218.260	307.751

Таблица 5: Таблица результатов времени работы алгоритма на C++ с использованием библиотеки lapack и алгоритма на Fortran с использованием lapack

N	C++ lapack	Fortran lapack
128	0.001	0.0012
256	0.008	0.0086
512	0.063	0.0628
1024	0.495	0.496
2048	4.000	4.002
4096	33.235	33.249