

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»
(СГУ)

Кафедра дискретной математики и
информационных технологий

**МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ
ЭЛЕКТРОННОГО ПОТОКА С ЭЛЕКТРОМАГНИТНЫМ ПОЛЕМ
ВОЛНОВЕДУЩЕЙ СИСТЕМЫ В ДИСКРЕТНОМ ПРИБЛИЖЕНИИ**

АВТОРЕФЕРАТ

НАУЧНО-КВАЛИФИКАЦИОННОЙ РАБОТЫ
(ДИССЕРТАЦИИ)

аспиранта 4 курса
направления 09.06.01 – Информатика и вычислительная техника
направленности «Математическое моделирование, численные методы и
комплексы программ»

ВЫСОЦКОГО АЛЕКСАНДРА ВИТАЛЬЕВИЧА

Научный руководитель
д.ф.-м.н., профессор

Д. К. Андрейченко

Зав. кафедрой

Д. К. Андрейченко

Саратов 2019

СОДЕРЖАНИЕ

Введение	3
1. Разработка программного комплекса	12
1.1 Реализация алгоритма поиска собственных значений методом Хаусхолдера в среде MS Visual Studio на языке C++	12
1.2 Реализация алгоритма поиска собственных значений методом Хаусхолдера в среде графического программирования LabVIEW	18
1.3 Реализация алгоритма поиска глобального экстремума ЦФ методом отжига на языке C++ и проведение численного эксперимента	20
1.4 Реализация метода крупных частиц на языке C++ и в пакете прикладных программ для решения задач технических вычислений MATLAB и проведение численного эксперимента	25
Заключение	32
Список использованных источников	33

ВВЕДЕНИЕ

Актуальность. При проектировании ламп бегущей волны (ЛБВ) необходимо решать множество задач, связанных как с расчетом различных параметров непосредственно самой системы, так и с моделированием работы разработанной ЛБВ.

При проведении вычислений на первом применяется математическое моделирование динамических систем, описываемых дифференциальными уравнениями. При этом приходится часто решать задачу связанную с поиском собственных значений матриц (например, при нахождении собственных частот резонансных структур). Одной из важных проблем, возникающих при этом, является большие вычислительные затраты, а следовательно, длительное время вычислений. Существует большое количество методов, позволяющих решать проблему поиска собственных значений матриц. Нахождение собственных значений матриц, имеющих размерность порядка 10^3 - 10^6 и более, требует больших вычислительных затрат, а следовательно, длительное время вычислений. Но с другой стороны задаче нахождения собственных значений матриц присущ внутренний параллелизм, что дает основания использовать параллельные вычислительные системы для их решения.

Одним из возможных подходов к нахождению собственных значений матрицы является использование метода Хаусхолдера, приводящего исходную симметричную матрицу к трехдиагональной симметричной форме, вместе с QL-алгоритмом. Метод Хаусхолдера заключается в умножении матрицы на каждом шаге справа и слева на ортогональные матрицы отражения. Каждое преобразование обнуляет часть строки и столбца в преобразовываемой матрице. В QL-алгоритме используются двусторонние умножения на элементарные матрицы вращения.

Предполагаемое численное или физическое моделирование процессов в разрабатываемых приборах СВЧ, которые могут быть описаны набором дифференциальных уравнений, требует последующую оптимизацию прибора с целью достижения требуемых значений выходных параметров. В настоящее время решение большого количества технических задач требует поиска экстремума некоторой целевой функции (ЦФ) на заданной области определения. Наличие всевозможных ограничений на оптимизируемые параметры и многоэкстремальность ЦФ, как правило, приводит к большим временным затратам. Применение параллельных ВС позволяет повысить производительность вычислений благодаря большому числу одновременно работающих ядер, а программные-платформы предоставляют широкий набор методов и инструментов для эффективной реализации распараллеленных алгоритмов.

Существует широкий набор методов оптимизации, такие как: градиентный метод поиска, генетический алгоритм, метод Гаусса-Зейделя, статистические методы поиска, метод отжига и другие.

В данной работе рассматривается применение метода отжига к задаче поиска глобального экстремума ЦФ.

Метод отжига — это техника оптимизации, использующая упорядоченный случайный поиск на основе аналогии с процессом образования в веществе кристаллической структуры с минимальной энергией при охлаждении.

Большим преимуществом метода отжига является возможность избежать так называемых «ловушек» в локальных минимумах оптимизируемой функции за счет принятия изменений, временно ухудшающих результат, что отражает нагрев расплава для предотвращения его быстрого остывания при протекании соответствующего физического процесса. Еще одним преимуществом является то, что даже в условиях нехватки вычислительных ресурсов для нахождения

глобального минимума метод отжига, как правило, выдает неплохое решение — один из локальных минимумов

При численном моделировании процессов в ЛБВ широкое применение находит метод крупных частиц. Этот метод находит широкое применение при моделировании физических процессов, происходящих в потоке заряженных частиц, в частности для симуляции взаимодействия электронного потока с внешним электромагнитным полем, учитывая граничные условия и прочие факторы.

Метод крупных частиц применяется к задачам, в которых рассматривается эволюция во времени некоторой среды, либо результат такой эволюции. Это может быть несжимаемая жидкость, плотный или разреженный газ, плазма из ионов и электронов и многое другое. Характерной особенностью данного метода является разбиение среды на множество дискретных объектов, модельных частиц, рассматриваемых как некоторая сетка подвижных узлов.

Моделирование движения заряженных частиц является сложным и затратным с точки зрения вычислительных ресурсов процессом. Необходимо учитывать внешние и внутренние электромагнитные поля, взаимодействие частиц между собой и с окружающими предметами. Метод крупных частиц позволяет решить данную задачу за приемлемое количество времени с заданной точностью.

В качестве практического примера применимости данного метода можно привести задачу моделирования тонких ленточных пучков. Данная задача представляет все больший интерес в связи с созданием и развитием терагерцовых ЛБВ. Провести анализ траектории движения заряженных частиц и оценить изменение формы пучка с течением времени в зависимости от различных внешних условий и позволяет данный метод.

Проектирование СВЧ приборов и решаемые при этом задачи требуют большого количества вычислительных и временных ресурсов. Для ускорения

данных ресурсоемких вычислений применяются различные параллельные вычислительные системы. GPU наиболее эффективны при решении задач, обладающих параллелизмом по данным, число арифметических операций в которых велико по сравнению с операциями над памятью.

Одной из платформ, ориентированной для осуществления различных матричных преобразований, является программно-аппаратная платформа NVIDIA CUDA. Современные графические ускорители (GPU) NVIDIA имеют сотни упрощенных вычислительных ядер и большие объемы встроенной памяти, позволяющие максимально эффективно решать задачи, в которых необходимо одинаковые операции применить к множеству независимых данных.

Аппаратные возможности графических процессоров NVIDIA в совокупности с поставляемой с ними библиотекой CUBLAS – набором подпрограмм, предназначенных для вычислений задач линейной алгебры и использующие прямой доступ к ресурсам GPU, можно использовать для поиска собственных значений у матриц высоких порядков, моделирования физических процессов и решения задач оптимизации.

При реализации решений на параллельных вычислительных системах возникает ряд характерных проблем, связанный с доступом к данным, которые еще не были рассчитаны отдельным потоком.

Так, например, при распараллеливании метода Хаусхолдера возникает необходимость взаимодействия процессов, осуществляющих ортогонализацию, что достаточно сложно реализовать на CUDA, где фактически отсутствует возможность какого-либо взаимодействия между исполняемыми процессами, кроме как через глобальную память, что оказывает существенное влияние на производительность.

Реализуемые решения могут выступать как самостоятельный программный комплекс, так и в качестве интеграционного модуля для других

программных продуктов (в виде подключаемых программных dll библиотек). Среди таковых можно выделить среду графического программирования LabVIEW компании National Instruments (NI), обеспечивающая быстрое создание эффективных программ с наглядным интерфейсом, а так же прикладной пакет MATLAB, встроенные операции для работы с матрицами, численного дифференцирования, а также мощный графический пакет которого позволяет существенно сократить время разработки.

Цель диссертационной работы. Целью является разработка и отладка программного комплекса для расчета физических процессов в СВЧ приборах путем численного моделирования с применением распараллеливания алгоритмов на графических ускорителях NVIDIA с технологией CUDA.

Для достижения поставленной цели в работе решаются следующие основные задачи:

- Поиск эффективных способов распараллеливания метода отжига, с точки зрения минимизации времени и повышения надежности вычислений. Программная реализация поиска глобального экстремума методом имитации отжига, распараллеленного различными способами, проведение численного эксперимента для оценки эффективности способов распараллеливания. Анализ результатов численного эксперимента.
- Изучение методов поиска собственных значений симметричных матриц, адаптация алгоритмов поиска собственных значений матрицы для вычислений на GPU. Программная реализация метода поиска собственных значений, оценка его эффективности.
- Разработка параллельного алгоритма метода крупных частиц для моделирования движения заряженного потока частиц и его реализация в виде программного комплекса.

Методы исследования. Методами исследования в работе являются математическое моделирование на основе интегральных уравнений

электродинамики, проекционные и итерационные методы их решения, численные методы для реализации алгоритмов в программном комплексе.

Научная новизна работы.

- Изучены и предложены способы распараллеливания алгоритмов поиска собственных значений матриц методом Хаусхолдера, метода отжига для решения задач оптимизации ЦФ, и метода крупных частиц на графических процессорах.
- На основе предложенных параллельных алгоритмов для методов Хаусхолдера, метода отжига, и метода крупных частиц разработаны их реализации в виде программных комплексов для выполнения расчетов на параллельных вычислительных системах с поддержкой технологии NVIDIA CUDA.
- Проведен сравнительный анализ результатов запуска программного комплекса для оптимизации ЦФ. Показано, что разбиение пространства решений на области с увеличением числа блоков GPU, на которых производится расчет, сокращает число итераций, и соответствующая зависимость имеет экспоненциальный характер. Значительное влияние на количество вычислений ЦФ и соответственно на эффективность метода оптимизации оказывает значение температурного коэффициента τ . В ходе проведения численного эксперимента был сделан вывод, что изменение температурного коэффициента τ с 0.95 до 0.9999 приводит к значительному увеличению времени вычислений ЦФ, однако при этом точность нахождения решения повышается.
- Исследованы результаты численных экспериментов по поиску собственных значений матриц методом Хаусхолдера. Продемонстрировано преимущество параллельной реализации данного метода перед его последовательной версией. Показано, что с ростом размерности матриц ($n > 4000$) эффективность метода снижается из-за

большого количества ветвлений и последовательных действий в QL-алгоритме.

- Исследованы результаты численных экспериментов по моделированию движения заряженных частиц в одномерном приближении. Показано, что при сведении к минимуму копирования больших объемов данных из RAM в GPU, запуске расчетов на всех доступных блоках графического процессора, и применении функций из программной библиотеки CUBLAS для эффективной работы с матрицами получен заметный прирост производительности с ростом числа крупных частиц, ускорение времени расчета одного шага составляет от 1.8 до 2.8.
- Проведена интеграция разработанных программных комплексов с пакетом прикладных программ для решения задач технических вычислений MATLAB и LabVIEW, показана возможность их универсального использования в виде консольных программ с графическим выводом результатов работы (вывод траекторий частиц и различных параметров электронного пучка; вывод вектора собственных значений), и проведен сравнительный анализ производительности реализованных модулей и штатных средств.

Научная и прикладная значимость. Научной значимостью обладают предложенные способы распараллеливания рассмотренных алгоритмов. В прикладном аспекте результаты работы представляют интерес для проектирования и расчета СВЧ устройств.

Достоверность результатов. Достоверность результатов работы основана на использовании строгих электродинамических моделей анализа. Достоверность части численных результатов подтверждена их совпадением и сравнением с аналогичными как теоретическими, так и экспериментальными результатами других авторов.

Положения и результаты, выносимые на защиту:

1. Разработанные программные комплексы с модульной структурой, обладающие широкими возможностями внедрения в различные программные продукты, основанный на предложенных параллельных версиях рассматриваемых алгоритмов и поддерживающие технологии высокопроизводительных параллельных вычислений CUDA на аппаратной платформе графических процессоров NVIDIA, позволяют получать решения для собственных значений матриц, поиска глобального экстремума функций многих переменных, моделирования движения заряженных частиц, и обладают возможностями расширения функционала.
2. Перенос расчетов основных шагов метода крупных частиц с CPU на параллельную вычислительную систему на основе GPU позволяет добиться кратного сокращения времени вычислений в целом, и одной итерации алгоритма в частности.
3. При увеличении значения температурного коэффициента метода отжига существует порог, при котором происходит кратное увеличение времени работы алгоритма при незначительных изменениях точности вычислений.
4. Частые обмены данными между CPU и GPU не позволяют добиться линейного улучшения производительности с ростом размерности матриц при поиске собственных значений методом Хаусхолдера.

Апробация результатов работы. Основные результаты по теме диссертационного исследования докладывались на следующих международных конференциях, школах и семинарах: международная научная конференция "Компьютерные науки и информационные технологии" Саратов 2014 и 2016, 16-я Международная суперкомпьютерная конференция "Научный сервис в сети Интернет: многообразие суперкомпьютерных миров." Новороссийск 2014, 5-я Международная научно-практической конференции "Современные концепции научных исследований" Москва 2014, Научной конференция "Presenting

Academic Achievements to the World. Natural Science” Саратов 2015, XIV Международная научно-техническая конференция «Физика и технические приложения волновых процессов» Самара 2016, Saratov Fall Meeting SFM'16 – International Symposium Optics and Biophotonics IV Саратов 2016. Принял участие в работе international scientific school “Recent Advances in Algorithms” Санкт-Петербург 2017.

Публикации. По теме диссертации было опубликовано 7 печатных работ, в том числе 1 статья в рецензируемых российских журналах, рекомендованных ВАК, 6 статей в сборниках международных и российских конференций.

Личный вклад автора. Лично автором разработаны все программы, произведена основная часть расчетов и интерпретирована значительная часть полученных в работе результатов. Постановка задач и разработка алгоритмов проводилась совместно с научными руководителями.

1. РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА

1.1 Реализация алгоритма поиска собственных значений методом Хаусхолдера в среде MS Visual Studio на языке C++

На основе представленной выше схемы в п. 1.3, на языке C++, с использованием технологии параллельных вычислений NVIDIA CUDA, было реализовано и отлажено приложение, предназначенное для поиска собственных значений симметричной матрицы. Программный код реализованной программы приведен в приложении А.

При распараллеливании метода Хаусхолдера возникает необходимость взаимодействия процессов, осуществляющих ортогонализацию, что достаточно сложно реализовать на CUDA, где фактически отсутствует возможность какого-либо взаимодействия между исполняемыми процессами. Повышение эффективности метода было достигнуто за счет использования подпрограмм CUBLAS, реализующих матрично-векторные преобразования на GPU, и параллельной обработки массивов данных.

Приложение представляет собой консольную программу. Входными параметрами являются размерность входной матрицы, расположение конфигурационного файла с матрицей и способ отображения результатов поиска собственных значений – вывод на экран (ключ “-s”) или запись в файл (ключ “-f”, с обязательным указанием пути сохранения результата).

Конфигурационный файл с матрицей представляет собой текстовый файл, в котором виде квадратной таблицы записаны элементы матрицы.

На рисунках 5 и 6 представлен интерфейс работы консольного приложения.

```
Администратор: C:\Windows\system32\cmd.exe
C:\>
C:\>
C:\>C:\Householder.exe 100 C:/matrix.txt -s

Найденные собственные значения:
-349186.219      0.958      202.471      404.976      606.985
  808.987      1010.986      1212.988      1414.989      1616.989
 1818.990      2020.991      2222.994      2424.991      2626.992
 2828.993      3030.993      3232.993      3434.994      3636.993
 3838.996      4040.995      4242.997      4444.999      4647.000
 4849.000      5050.997      5252.999      5454.999      5656.999
 5859.000      6061.000      6263.001      6465.003      6666.999
 6869.009      7071.034      7273.097      7475.272      7677.692
 7880.588      8084.311      8289.276      8495.903      8704.469
 8915.103      9127.705      9342.069      9557.875      9774.735
 9992.228     10209.932     10427.422     10644.271     10860.064
11074.417     11287.011     11497.614     11706.166     11912.768
12117.722     12321.431     12524.323     12726.743     12928.910
13130.985     13333.003     13535.007     13737.005     13939.004
14141.006     14343.006     14545.012     14747.006     14949.008
15151.006     15353.015     15555.012     15757.011     15959.013
16161.010     16363.010     16565.012     16767.008     16969.010
17171.016     17373.014     17575.010     17777.023     17979.018
18181.018     18383.012     18585.018     18787.016     18989.014
19191.018     19393.027     19595.018     19797.025     19999.023

Время поиска собственных значений составило 0.325 с.
C:\>
```

Рисунок 5 – Интерфейс работы приложения с выводом найденных собственных значений на экран

```
Администратор: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\User>C:\Householder.exe 100 C:/matrix.txt -f C:/eigenvalues.txt

Найденные собственные сохранены в C:/eigenvalues.txt.
Время поиска собственных значений составило 0.375 с.
C:\Users\User>
```

Рисунок 6 – Интерфейс работы приложения с сохранением найденных собственных значений в файл

На рисунке 7 приведена блок-схема алгоритма поиска собственных значений с параллельной обработкой данных и вызовом процедур CUBLAS.

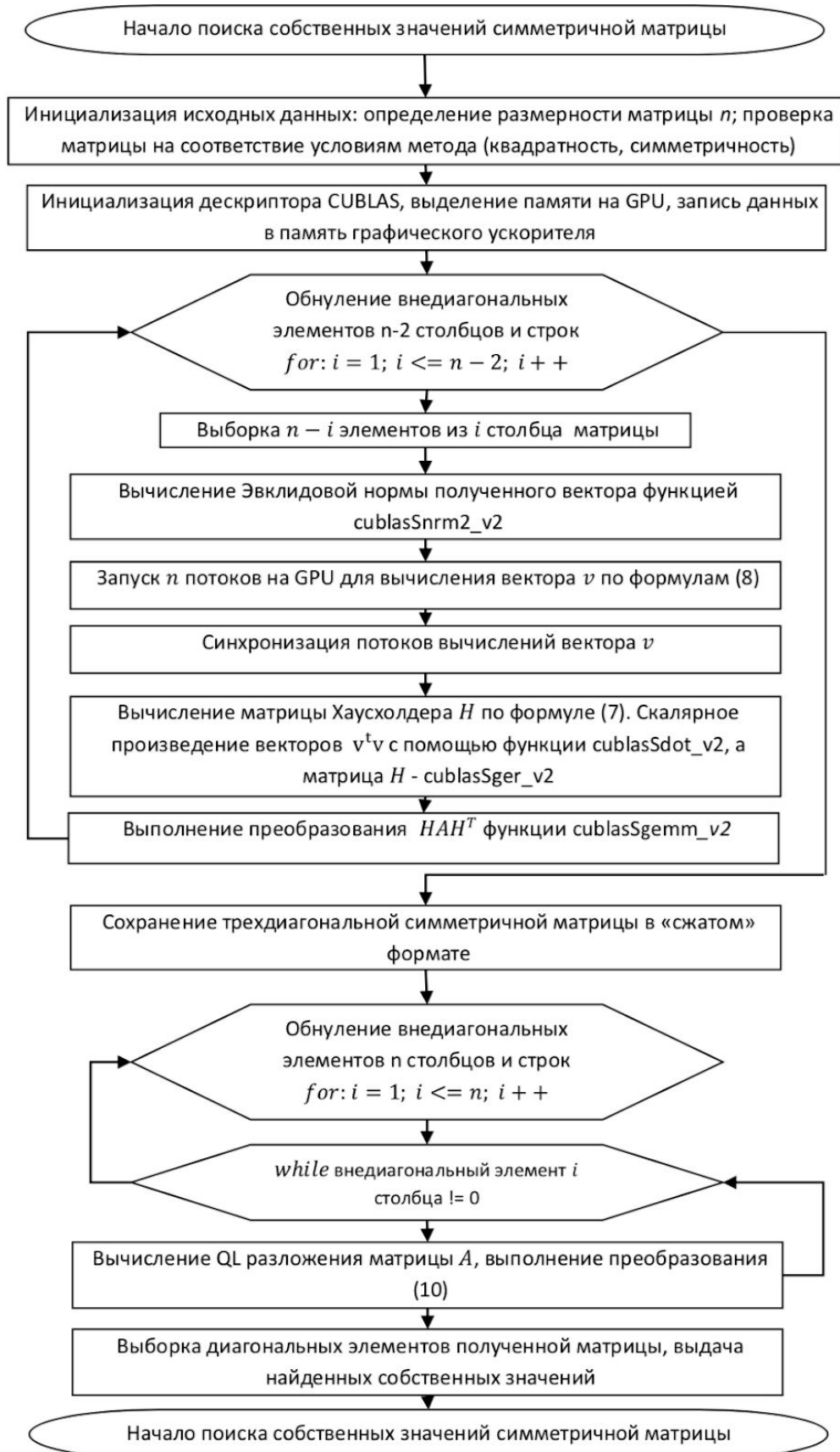


Рисунок 7 – Блок-схема алгоритма поиска собственных значений с параллельной обработкой данных и вызовом процедур CUBLAS

Тестирование и отладка приложения проводилась в системе со следующими техническими характеристиками: CPU Intel Core2 Quad 2.33 GHz, 4GB RAM, NVIDIA GeForce GTX 650 2GB.

На рисунке 8 приведен график зависимости времени поиска собственных значений, усредненного по итогам 100 запусков, от размера матрицы.

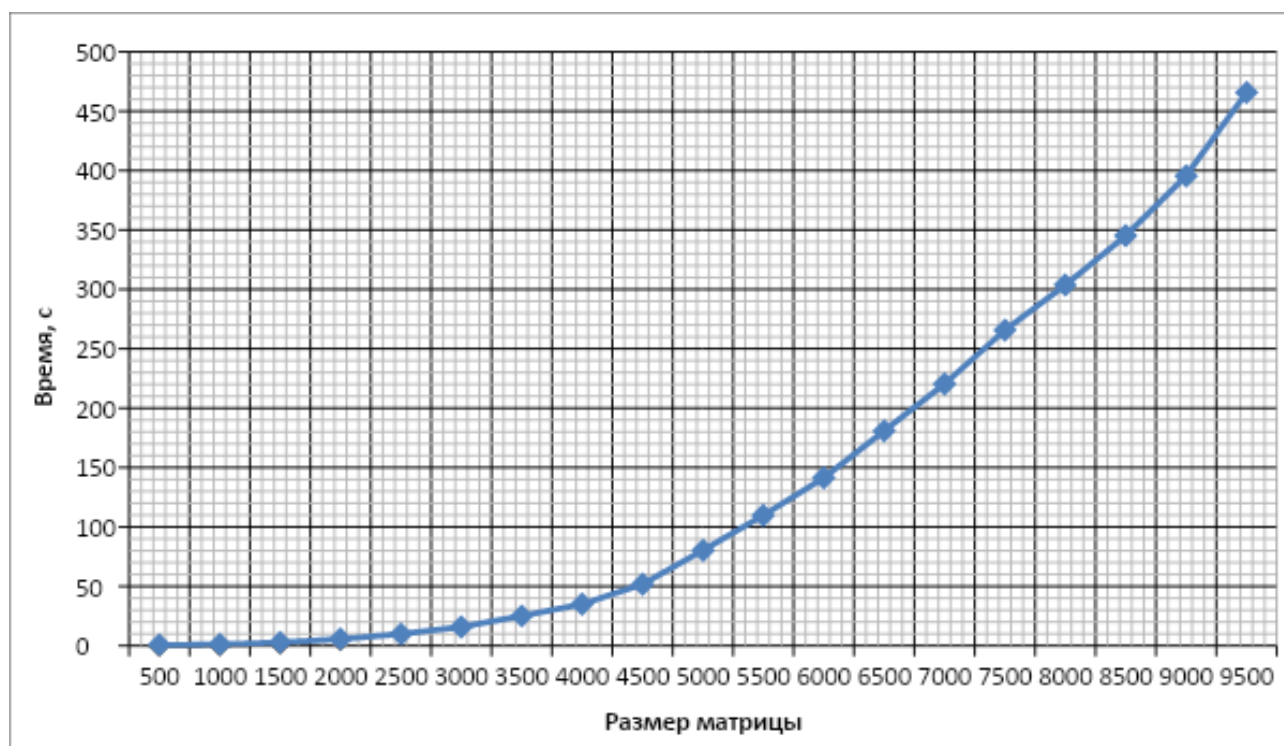


Рисунок 8 – Зависимость времени (в секундах) поиска собственных значений от размеров матрицы

График наглядно показывает увеличение времени поиска с увеличением размерности матрицы: с 0.325 с. для матрицы размерностью 500×500 до 465.571 с. для матрицы размерностью 9500×9500 .

На рисунке 9 представлен сравнительный график зависимости времени поиска собственных значений, усредненного по итогам 100 запусков, от размера матрицы для последовательной и распараллеленной версии приложения.

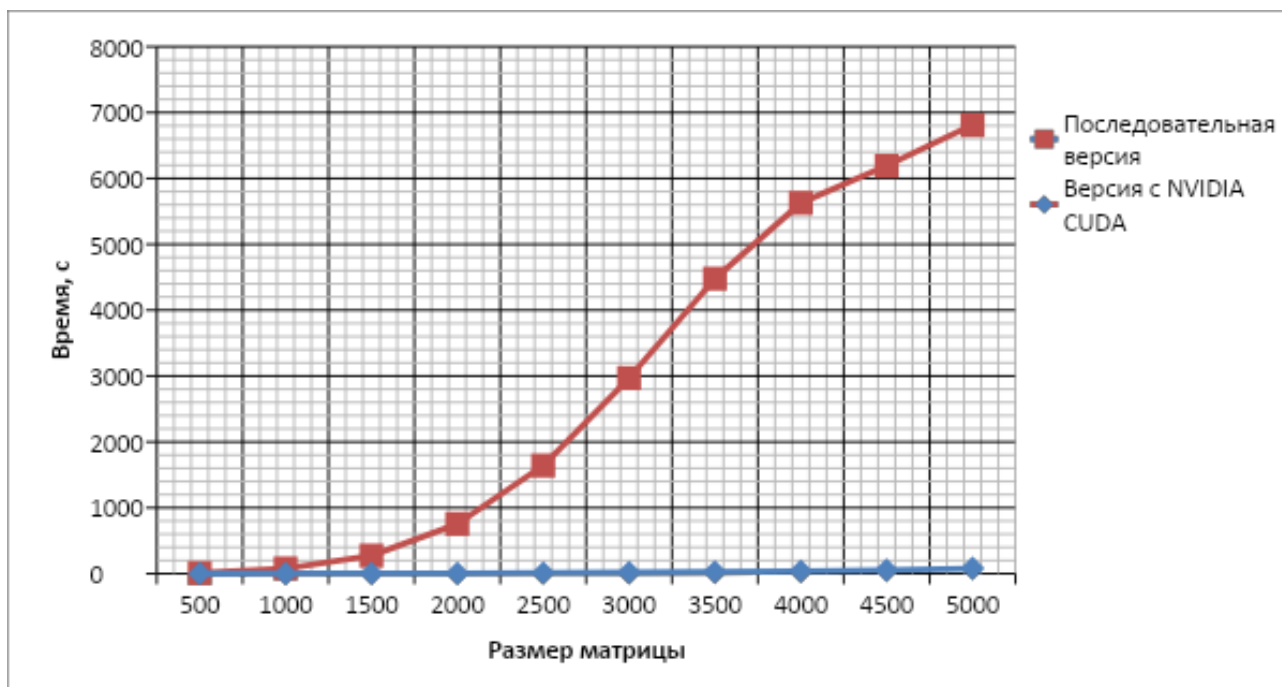


Рисунок 9 – Зависимость времени (в секундах) выполнения поиска собственных значений симметричной матрицы от ее размера параллельным и последовательным методом

Из представленного графика видно, что использование GPU в вычислениях дает значительный прирост производительности по сравнению с последовательной реализацией.

Для упрощения интеграции с различными приложениями, реализованная программа была скомпилирована в виде динамически подключаемой библиотеки (DLL). Входными параметрами полученного модуля являются указатель сохраненную в памяти матрицу, указатель на выделенный для сохранения результатов поиска массив, размерность матрицы.

1.2 Реализация алгоритма поиска собственных значений методом Хаусхолдера в среде графического программирования LabVIEW

На основе представленной выше схемы в п. 1.3, в среде графического программирования LabVIEW разработан и отлажен программный модуль, предназначенный для поиска собственных значений симметричной матрицы.

Входными параметрами модуля являются симметричная матрица и метод поиска собственных значений. Выходными параметрами являются массив найденных собственных значений и время поиска. Лицевая панель программного модуля представлена на рисунке 10.

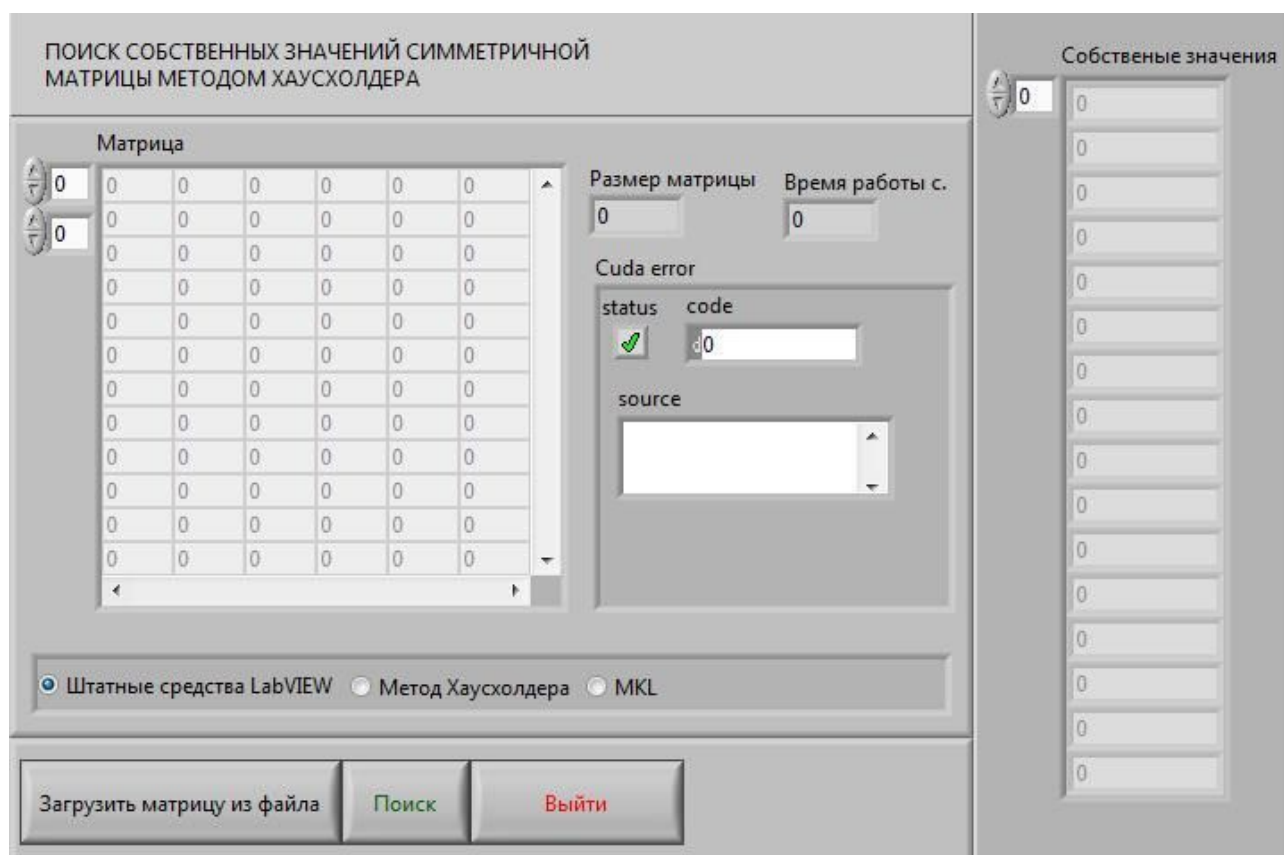


Рисунок 10 – Лицевая панель программного модуля, реализующего поиск собственных значений методом Хаусхолдера

Блок-диаграмма программного модуля, реализующего поиск собственных значений симметричной матрицы методом Хаусхолдера приведена в приложении Б.

По итогам сравнительных тестов было выявлено, что применение метода Хаусхолдера для поиска собственных значений симметричных матриц дает значительный выигрыш по времени по сравнению со штатными средствами LabVIEW, не использующими технологии распараллеливания. Так же модуль показал преимущество перед реализацией на Intel MKL.

1.3 Реализация алгоритма поиска глобального экстремума ЦФ методом отжига на языке C++ и проведение численного эксперимента

Оценка надёжности и эффективности распараллеленного алгоритма имитации отжига по схеме Больцмановского тушения осуществлялась на многоэкстремальных ЦФ, с количеством параметров оптимизации n , равным 2:

– гладкая многоэкстремальная функция:

$$f(x_1, x_2) = - \frac{1}{\frac{x_1^2 + x_2^2}{200} - \cos \cos(x_1) \cos \cos\left(\frac{x_2}{\sqrt{2}}\right) + 2}, \quad (1)$$

где $-0.5 < x_j < 1$, $j = 1, 2$; $= -1$.

– многоэкстремальная функция Растригина:

$$f(x_1, x_2) = 20 + x_1^2 - 10 \cos(2\pi x_1) + x_2^2 - 10 \cos(2\pi x_2), \quad (2)$$

где $-0.5 < x_j < 1$, $j = 1, 2$; $= 0$.

– негладкая 4-х экстремальная функция:

$$\begin{aligned} f_1 &= -5 * e^{-3*(|x_1+1|^{0.6} + |x_2+1|^{0.6})}; \\ f_2 &= -10 * e^{-2*(|x_1| + |x_2|)}; \\ f_3 &= -7 * e^{-2.5*(|x_1-1|^{0.8} + |x_2-1|^{0.8})}; \\ f_4 &= -3 * e^{-(|x_1-2|^{0.9} + |x_2-2|^{0.9})}; \\ f(x_1, x_2) &= f_1 + f_2 + f_3 + f_4, \end{aligned} \quad (3)$$

где $-0.5 < x_j < 1$, $j = 1, 2$; $= -10.1$.

– негладкая 4-х экстремальная функция, имеющая три равных

по значению глобальных минимума:

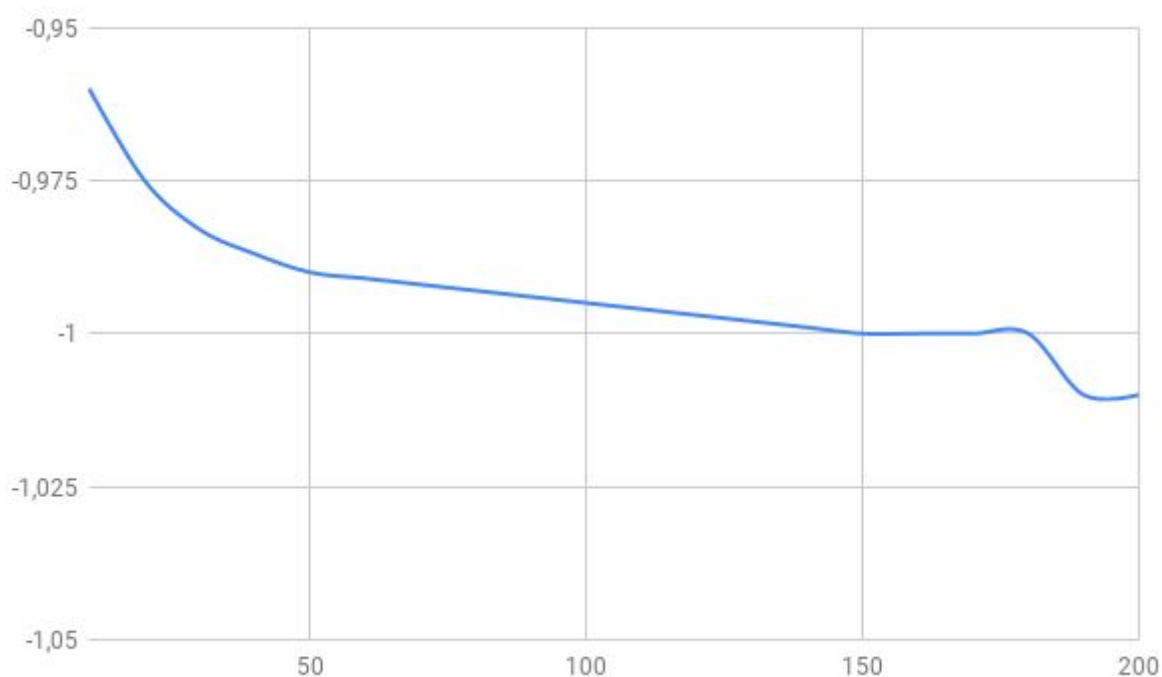
$$\begin{aligned} f_1 &= -7 * e^{-3*(|x_1+1|^{0.6} + |x_2+1|^{0.6})}; \\ f_2 &= -7 * e^{-2*(|x_1| + |x_2|)}; \\ f_3 &= -7 * e^{-2.5*(|x_1-1|^{0.8} + |x_2-1|^{0.8})}; \\ f_4 &= -3 * e^{-(|x_1-2|^{0.9} + |x_2-2|^{0.9})}; \\ f(x_1, x_2) &= g_1 + g_2 + g_3 + f_4, \end{aligned} \quad (4)$$

где $-0.5 < x_j < 1$, $j = 1, 2$; $= -7.5$.

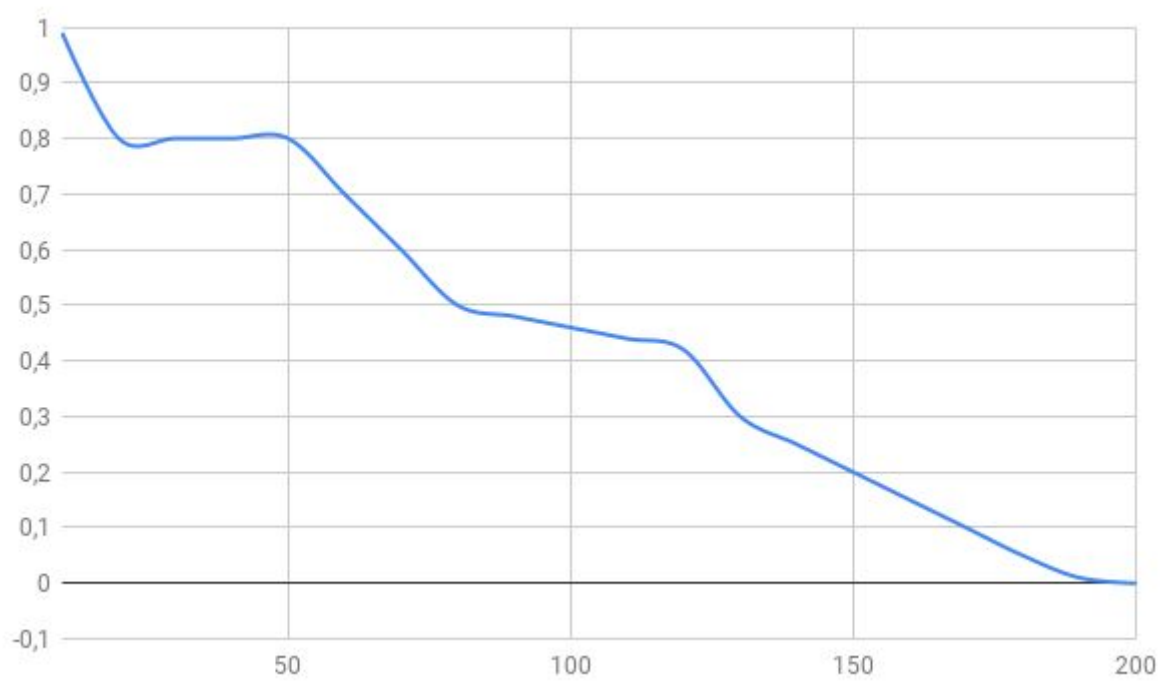
Тестирование проводилось при следующих условиях: начальная температура $T_0 = 10^9$, конечная $T_{end} = 0.1$, коэффициент изменения температуры $r = 0.97$. При таких значениях параметров обеспечивалось

оптимальное соотношение точности результата и времени вычисления алгоритма.

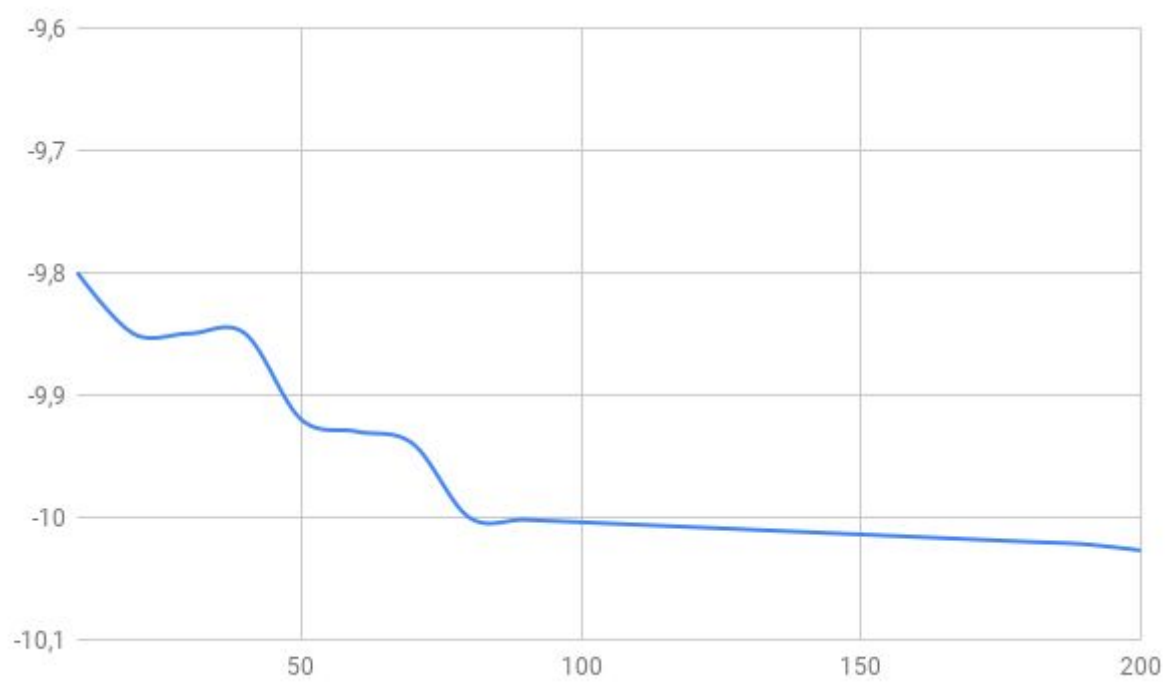
Результаты поиска минимума ЦФ (1), (2), (3) и (4) тремя распараллеленными методами имитации отжига в зависимости от числа одновременно работающих блоков и потоков, усредненные по итогам 100 запусков вычислительной системы, приведены на рисунках ниже для каждой из функций.



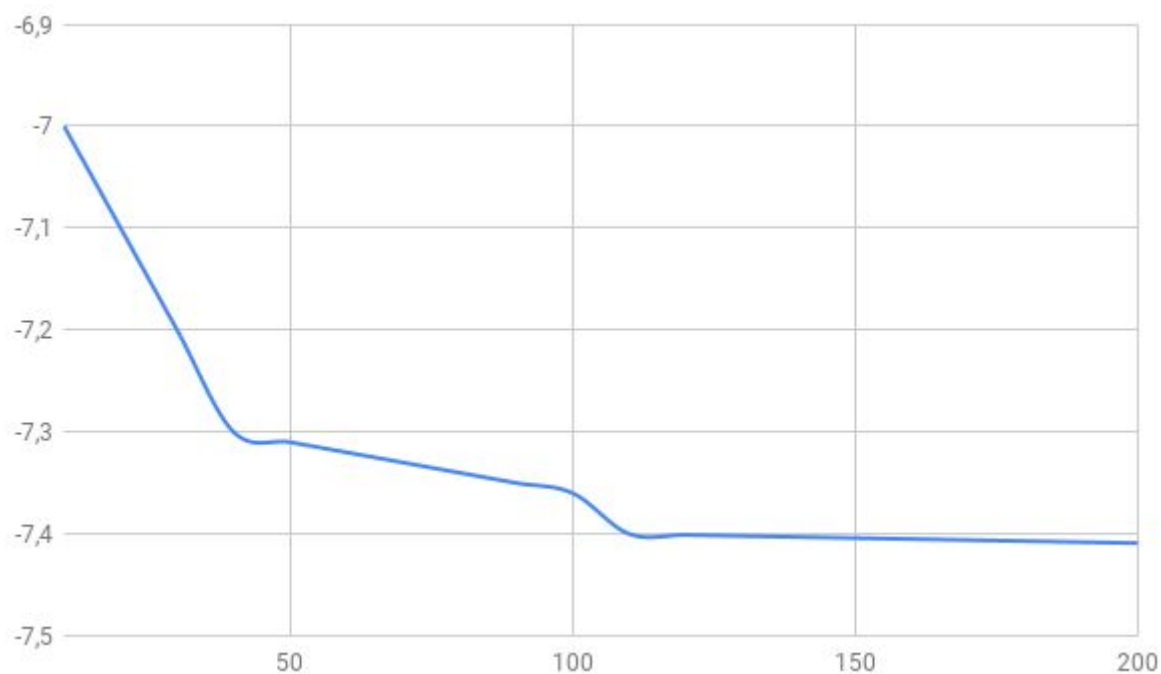
а



б



в



Г

Рис. Зависимость минимального значения целевой функции от количества одновременно работающих блоков GPU: а — для гладкой многоэкстремальной функции, б — многоэкстремальной функции Растригина, в — для негладкой 4-экстремальной функции, г — для негладкой 4-экстремальной функции с тремя равными минимумами

Число итераций сокращается с увеличением вычислительных узлов в случае разбиения пространства решений на области; соответствующая зависимость имеет экспоненциальный характер:

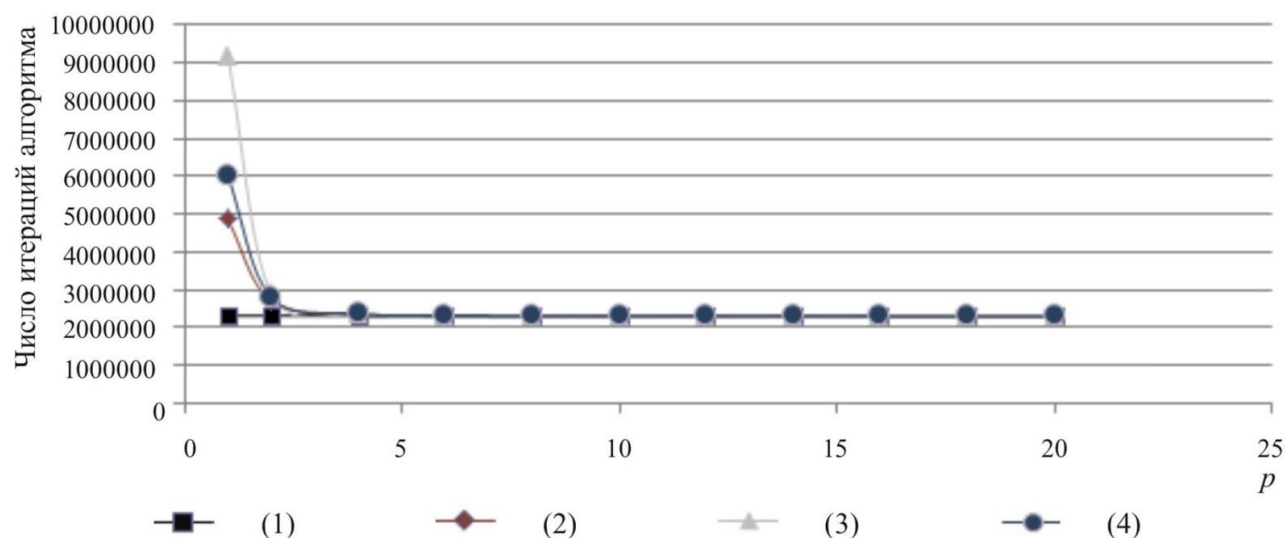


Рис. Число итераций алгоритма поиска минимума ЦФ 1–4 при увеличении числа блоков GPU

Значительное влияние на количество вычислений ЦФ и соответственно на эффективность тестируемого метода оптимизации оказывает значение температурного коэффициента r . В ходе проведения численного эксперимента был сделан вывод, что изменение температурного коэффициента r с 0.95 до 0.9999 приводит к увеличению времени вычислений ЦФ практически в 6 раз, однако при этом точность нахождения решения повышается.

1.4 Реализация метода крупных частиц на языке C++ и в пакете прикладных программ для решения задач технических вычислений MATLAB и проведение численного эксперимента

В рамках данной работы было реализовано консольное приложение, осуществляющее расчет траекторий заряженных частиц в одномерном приближении. Приложение было разработано на языке C++ в двух версиях, последовательной и оптимизированной, с использованием технологии NVIDIA CUDA.

Последовательная реализация приложения выполнялась на CPU Intel Core2 Quad, оптимизированная — на GPU NVIDIA GeForce GT 740M. На рис. 1 представлены графики времени вычислений в зависимости от числа моделируемых частиц для последовательной и параллельной реализации.

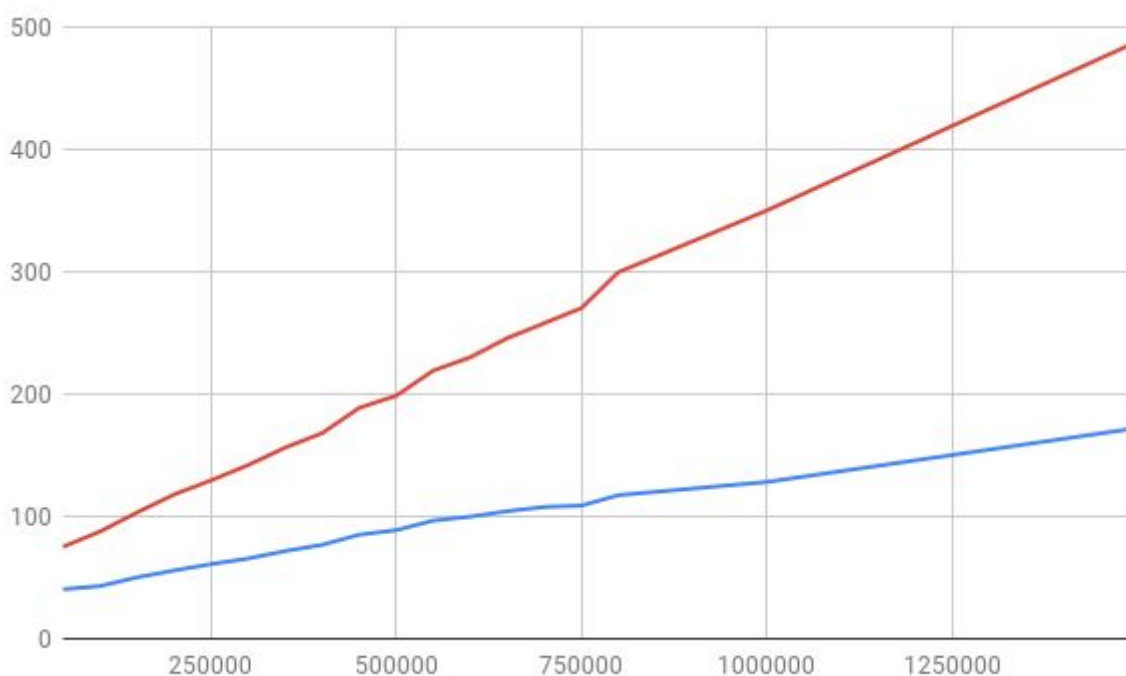


Рис. Графики времени работы последовательной и параллельной версии программы

Параметрами GPU, оказывающими существенное влияние на производительность, являются количество блоков (blocks) в сетке (grid),

количество вычислительных потоков (threads) в рамках блока, количество разделяемой памяти, доступной блоку. В каждом блоке выполняется равное число потоков. Исполнение потоков может быть синхронизировано в рамках текущего блока процедурой `__syncthreads()`.

Рассматривая параллельную вычислительную систему с NVIDIA CUDA, можно выделить четыре вида памяти. Оперативная память (RAM) является доступной только для процессов, запускаемых на CPU. GPU (устройство, device) имеет три вида памяти: постоянная память, глобальная память и разделяемая память. Постоянная и глобальная память доступны всем потокам. Глобальная память доступна для чтения и записи всем потокам. Постоянная память предоставляет разрешение только на чтение. Host может как передавать данные из RAM в глобальную или постоянную область памяти GPU так и наоборот. Общая память - это память, используемая всеми потоками блока. Все потоки в одном блоке имеют разрешение на чтение и запись для использования общей памяти блока.

Для повышения производительности реализации алгоритма необходимо свести к минимуму копирование больших объемов данных из RAM в GPU и обратно, и использование глобальной памяти GPU. Таким образом, физические характеристики графического процессора, объем встроенной памяти в частности, накладывают ограничения на точность симуляции (количество узлов в сетке разбиения рассматриваемого пространства и частиц, участвующем в моделировании). Также для увеличения производительности можно отказаться от использования метода Гаусса-Зейделя, из-за его медленной сходимости в ряде случаев, в пользу других, более эффективных итерационных методов.

Большое количество запускаемых потоков на GPU и эффективная работа с матрицами и массивами уменьшить время вычисления. Из представленного графика на рис. 2 можно сделать вывод, что использование GPU дает заметный

прирост производительности с ростом числа крупных частиц, ускорение времени расчета одного шага составляет от 1.8 до 2.8.

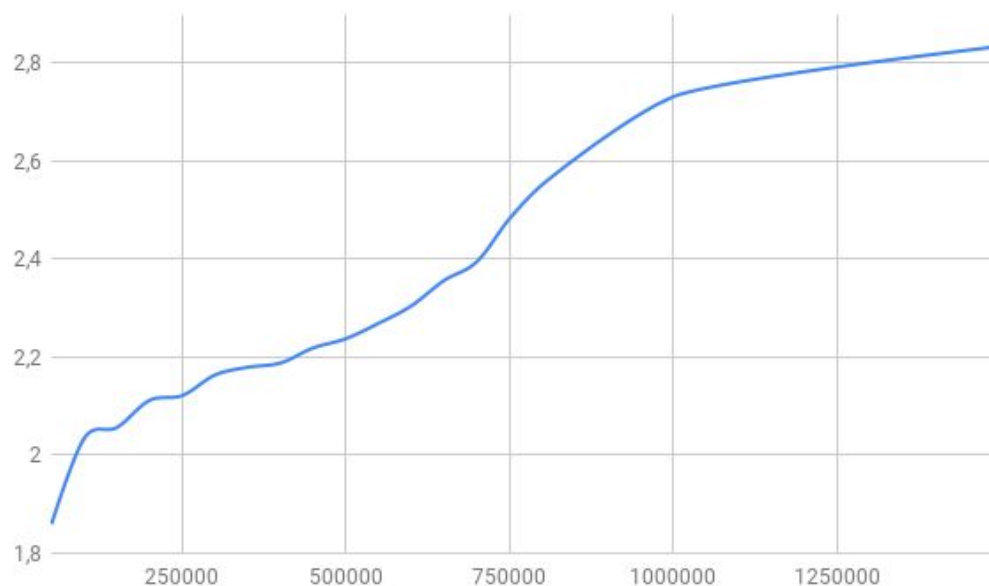


Рис. График ускорения вычислений в зависимости от числа моделируемых частиц

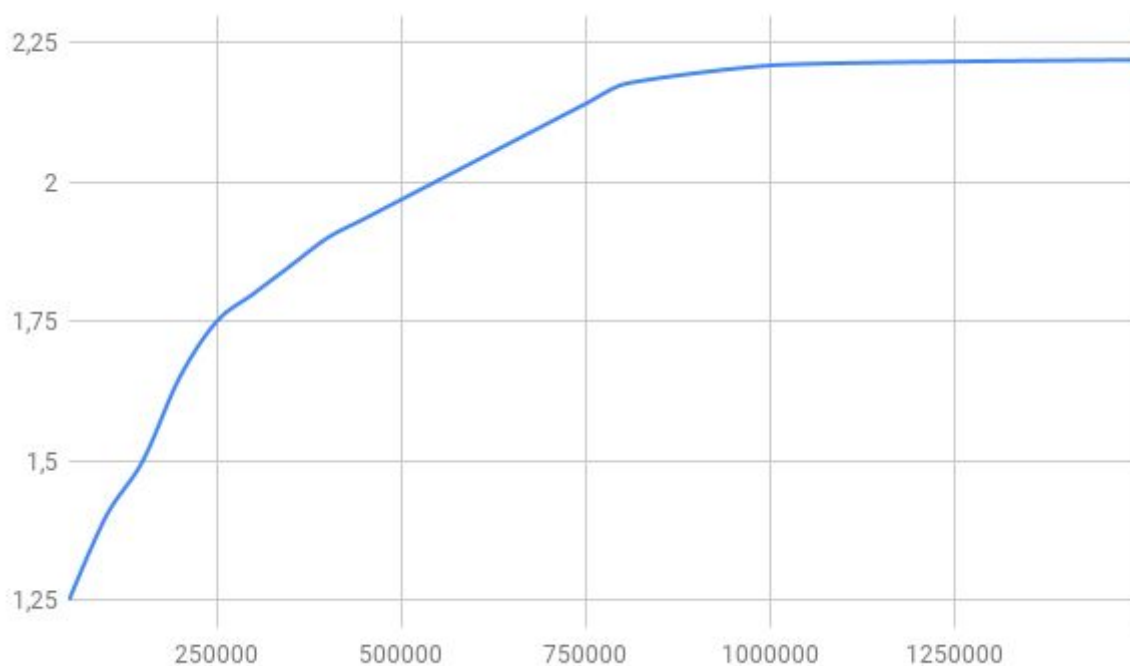


Рис. График ускорения вычислений в зависимости от числа моделируемых частиц с расчетом положений частиц на GPU

Сравнив полученные данные с реализацией, где основная часть вычислений проводилась на CPU, а на GPU были вынесены лишь расчеты шагов (1) и (4), можно отметить также уменьшение времени расчетов, где максимальное ускорение вычислений составило 2.17 раз.

Удобным инструментом в реализации решения данной задачи является прикладной пакет MATLAB. Встроенные операции для работы с матрицами, численного дифференцирования, а также мощный графический пакет позволяет существенно сократить время разработки.

В среде MATLAB была реализованная программа для анализа траекторий заряженных частиц, которая представляет собой консольное приложение с графическим выводом траекторий частиц и различных параметров электронного пучка. Работы приложения разделена на три этапа. На первом этапе, этапе инициализации, происходит разбиение среды на отдельные ячейки, добавление заряженных частиц в заданную область, координаты и скорость которых определяются случайным образом (с учетом общих требований). На втором этапе проводятся основные вычисления в соответствии с описанным выше алгоритмом. Основные шаги вычислительного цикла для i -го узла приведены ниже.

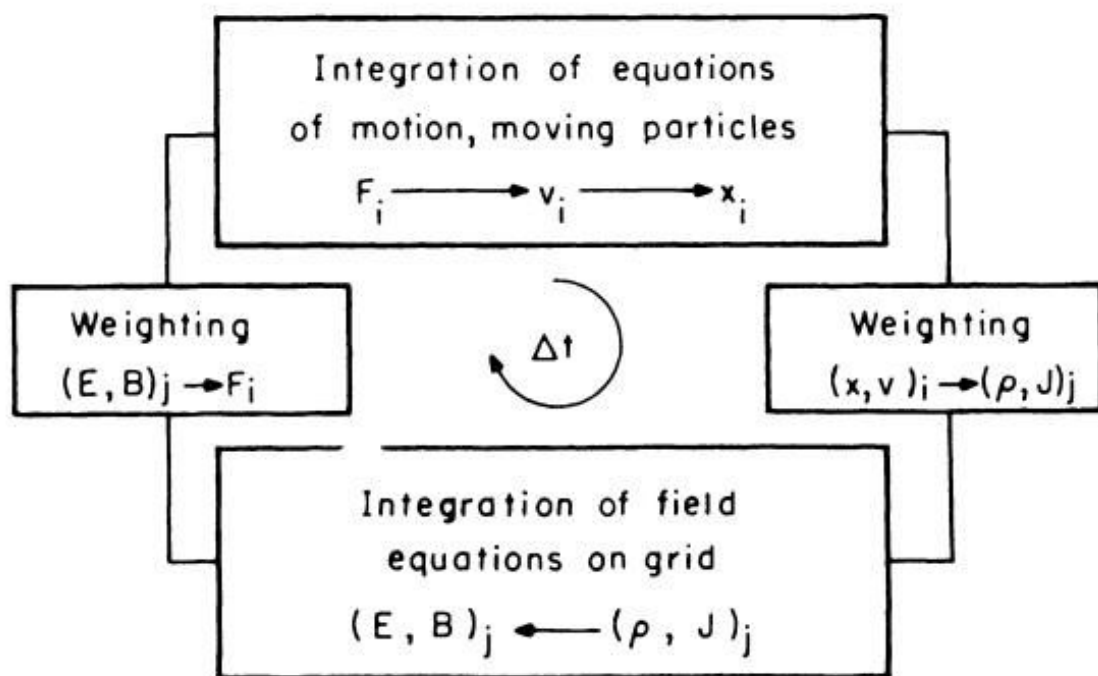


Рис. Блок-схема алгоритма метода крупных частиц

Интерполяция заряда на узлы сетки основана на cloud-in-cell алгоритме. Заряд частицы будет распределен пропорционально между узлами сетки в зависимости от положения частицы. На рисунке ниже приведен пример распределения заряда.

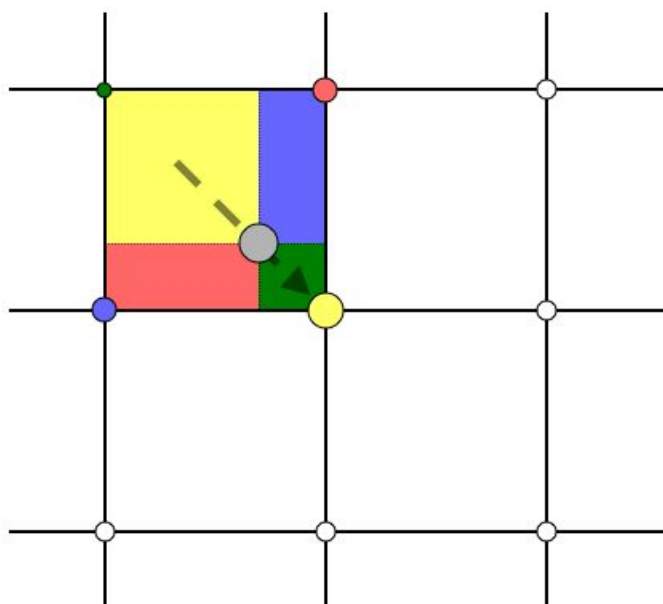


Рис. Пример распределения заряда по узлам сетки

После вычисления значения электромагнитного поля происходит обновление скоростей и положения частиц. Решение уравнения движения осуществляется методом leapfrog, который позволяет сократить время вычисления с сохранением заданной точности.

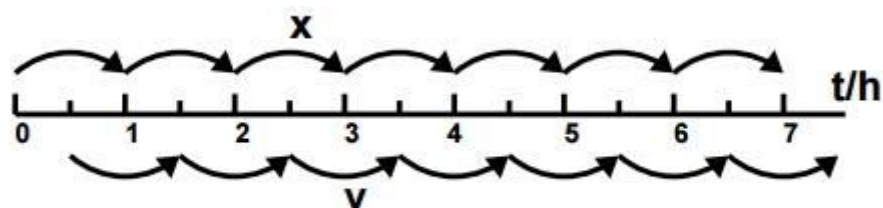


Рис. Принцип метода leapfrog

На рисунке ниже приведен результат работы данного алгоритма, анализировался пролет заряженных частиц через замедляющие системы.

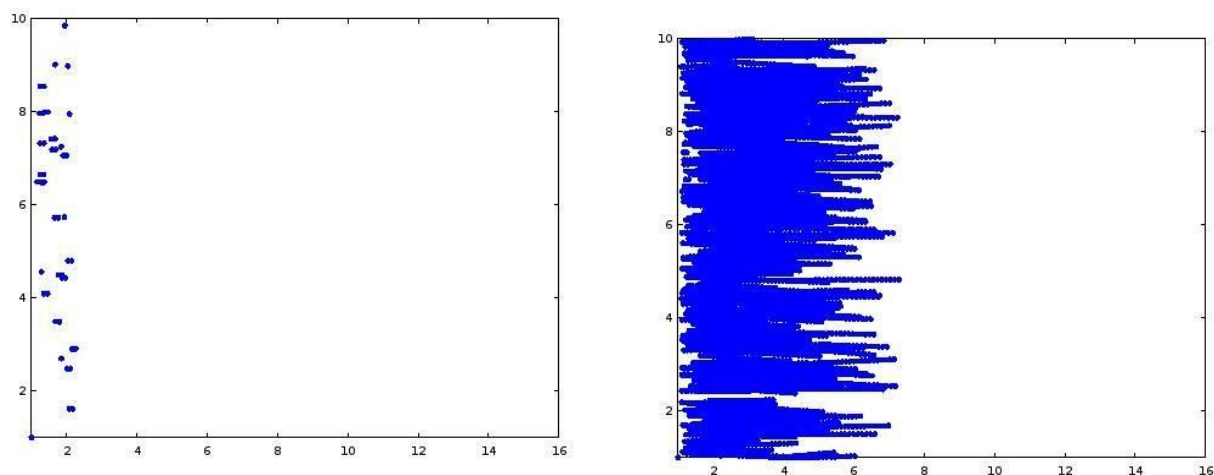


Рис. Пример вывода программы анализа траекторий электронов

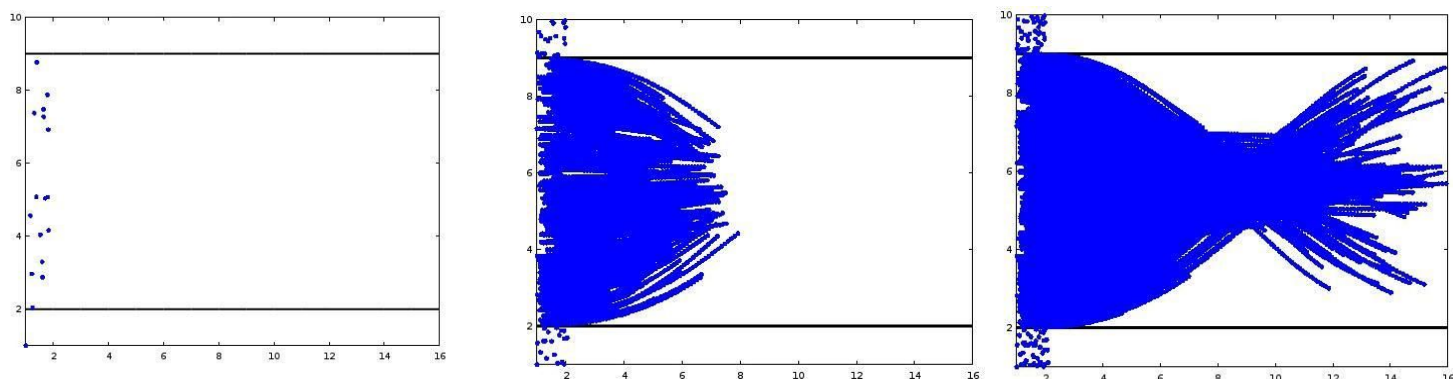


Рис. Пример вывода программы анализа траекторий электронов при
пролете между положительно заряженными пластинами

Из данных скриншотов можно сделать вывод, что данные, полученные во время моделирования, соотносятся с теоретическими ожиданиями.

ЗАКЛЮЧЕНИЕ

В данной был проведен анализ методов, используемых при численном моделировании процессов, протекающих в СВЧ приборах. Предложены параллельные версии алгоритмов с учетом особенностей их реализации на GPU NVIDIA.

Был предложен способ распараллеливания метода отжига, с точки зрения минимизации времени и повышения надежности вычислений. Программная реализация поиска глобального экстремума методом имитации отжига представлена в виде консольного приложения C++ и проведен численный эксперимент для оценки эффективности предложенного способа распараллеливания.

Предложена реализация поиска собственных значений симметричных матриц методом Хаусхолдера с адаптацией для вычислений на GPU. Программная реализация метода выполнена в двух частях: консольное приложение на языке C++ и динамически подключаемая библиотека, которая была подключена к реализации в программном пакете LabVIEW.

Предложен способ реализации параллельного алгоритма метода крупных частиц для моделирования движения заряженного потока частиц и его реализация в виде программного комплекса на языке C++.

Полученные комплексы обладают возможностями расширения функционала. Наибольшая эффективность достигается при использовании параллельных вычислительных систем на основе современных графических ускорителей NVIDIA.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Ильин, В.А. Линейная алгебра / В.А. Ильин, Э.Г. Позняк. М.: 1974, «Наука». 292 с.
- 2 Sanders, J. CUDA by Example: An Introduction to General-Purpose GPU Programming / Sanders J., Kandrot E. // Addison-Wesley, 2010. p. 6.
- 3 National Instruments Россия [Электронный ресурс]. URL: <http://www.labview.ru> (дата обращения: 10.06.2014). Загл. с экрана. Яз. рус.
- 4 Калиткин, Н. Н. Численные методы / под ред. А. А. Самарского. М. : Из-во «Наука», 1978. 157 с.
- 5 Крылов, В. И. Вычислительные методы высшей математики / В. И. Крылов, В. В. Бобков, П. И. Монастырный. Минск: Издательство «Высшая школа», 1972. 229 с.
- 6 Численные методы решения задач на собственные значения и собственные векторы матриц [Электронный ресурс]: Учитель.ру. URL: http://www.uchites.ru/files/nummethod_book_chapter1-2.pdf (дата обращения: 10.06.2014). Загл. с экрана. Яз. рус.
- 7 Парлетт, Б. Симметричная проблема собственных значений. Численные методы. Москва «Мир», 1983 г. С. 168.
- 8 Уилкинсон, Дж. Х. Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра. Перевод с английского под ред. д-ра техн. наук проф. Ю. И. Топчиева. М., «Машиностроение», 1976 г. 216 с.
- 9 About CUDA. NVIDIA Developer [Электронный ресурс]. URL: <https://developer.nvidia.com/about-cuda> (дата обращения: 10.06.2014). Загл. с экрана. Яз. англ.
- 10 История развития GPU. Параллельное программирование с использованием технологии CUDA [Электронный ресурс]. URL:

- <http://edu.chpc.ru/cuda/mainse3.html> (дата обращения: 10.06.2014). Загл. с экрана. Яз. рус.
- 11 Что такое вычисления с гри-ускорением? GPU и процессоры NVIDIA [Электронный ресурс]. URL: <http://www.nvidia.ru/object/gpu-computing-ru.html> (дата обращения: 10.06.2014). Загл. с экрана. Яз. рус.
- 12 CUDA: Как работает GPU / Хабрахабр [Электронный ресурс]. URL: <http://habrahabr.ru/post/54707/> (дата обращения: 10.06.2014). Загл. с экрана. Яз. рус.
- 13 Боресков, А. В. Параллельные вычисления на GPU. Архитектура и программная модель CUDA / А. В Боресков, А. А. Харламов, Н. Д. Марковский. М. : Изд-во Московского университета, 2012. 30 с.
- 14 GNU Scientific Library – Reference Manual: BLAS Support [Электронный ресурс]. URL: https://www.gnu.org/software/gsl/manual/html_node/BLAS-Support.html (дата обращения: 10.06.2014). Загл. с экрана. Яз. англ.
- 15 cuBLAS :: CUDA Toolkit Documentation [Электронный ресурс]. URL: <http://docs.nvidia.com/cuda/cublas/> (дата обращения: 10.06.2014). Загл. с экрана. Яз. англ.
- 16 Лопатин А. С. Метод отжига // Стохастическая оптимизация в информатике : межвуз. сб. СПб. : Изд-во СПбГУ, 2005. Вып. 1. С. 133–149.
- 17 Савин А. Н, Тимофеева Н. Е. Применение алгоритма оптимизации методом имитации отжига на системах параллельных и распределённых вычислений // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2012. Т. 12, вып. 1. С. 110–116.
- 18 Kirkpatrick S. A., Gelatt C. D., Vecchi M. P. Optimization by simulated annealing // Science. N.S. 1983. Vol. 220, No 4598. P. 671–680.

- 19 Kirkpatrick S. A., Gelatt C. D., Vecchi M. P. Optimization by simulated annealing // Science. N.S. 1983. Vol. 220, No 4598. P. 671–680.
- 20 Высоцкий А.В., Тараканов А.С., Шоломов К.И., Тимофеева Н.Е., Ерофтиев А.А. Исследование эффективности поиска глобального экстремума методом имитации отжига, распараллеленного различными способами // Изв. Саратов. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2013. Т. 13, вып. 3. С. 87–94.
- 21 Morey I. J., Birdsall C. K. Travelling-wave-tube simulation; The IBC code. — Berkeley, 2010. — С. 4–12.
- 22 Langdon A. B., Birdsall C. K. Plasma Physics Via Computer Simulation. — Boca Raton : CRC Press, 2004. — С. 29–51.
- 23 Григорьев Ю. Н. Численные методы «частицы-в-ячейках» — М. : Наука, 2000. — С. 77–106.
- 24 Hockney R. W., Eastwood J. W. Computer simulations using particles. — New-York : McGraw-Hill, 1981. — С. 184–205.
- 25 Высоцкий А. В., Савин А. Н. Применение технологии NVIDIA CUDA для реализации метода крупных частиц, Компьютерные науки и информационные технологии : Материалы Междунар. науч. конф. — Саратов : Издат. центр «Наука», 2016. – 496 с. : ил. ISBN 978-5-9999-2651-7.
- 26 Высоцкий А. В., “Применение среды MATLAB для анализа траекторий в плоских тонких ленточных пучках”, Тезисы докладов XIV Международной научно-технической конференции: Приложение к журналу «Физика волновых процессов и радиотехнические системы» / Под ред. В.А. Неганова. – Казань: ООО «16ПРИНТ», 2016. - 272 с.
- 27 Тимофеева Н.Е., Высоцкий А.В., Савин А.Н. Применение технология NVIDIA CUDA для поиска собственных значений симметричных матриц методом Хаусхолдера, Компьютерные науки и информационные технологии, Материалы Международной научной конференции, 30

июня-3 июля 2014 г., г. Саратов, Россия. – Саратов: Издательский центр «Наука», 2014. – С 95-97. – ISBN 978- 5-9999-2100-0.

- 28 Тимофеева Н.Е., Высоцкий А.В., Шоломов К.И., Савин А.Н. Применение технологий параллельных вычислений NVIDIA CUDA для поиска собственных значений матриц, Современные концепции научных исследований, Материалы 5-й Международной научно- практической конференции, 29-30 августа 2014 г., г. Москва. – М.: Изд-во ЕСУ, 2014. – С 52-56
- 29 Тимофеева Н.Е., Высоцкий А.В., Шоломов К.И., Савин А.Н. Применение технологий параллельных вычислений NVIDIA CUDA для поиска собственных значений матриц, Научный сервис в сети Интернет: многообразие суперкомпьютерных миров. Материалы 16- й Международной суперкомпьютерной конференции, 22-27 сентября 2014 г., г. Новороссийск. – М.: Изд-во МГУ, 2014. – С 289-295