

Федеральное государственное бюджетное учреждение науки Институт
математики и механики им. Н. Н. Красовского Уральского отделения
Российской академии наук

На правах рукописи

Третьяков Андрей Игоревич

**АЛГОРИТМЫ И ПРОГРАММЫ РЕШЕНИЯ
ОБРАТНЫХ ЗАДАЧ ГРАВИМЕТРИИ И
МАГНИТОМЕТРИИ НА ГРАФИЧЕСКИХ
ПРОЦЕССОРАХ**

Специальность 1.2.2 —

Математическое моделирование, численные методы и комплексы программ

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук,
профессор
Акимова Елена Николаевна

Екатеринбург — 2024

Оглавление

	Стр.
Введение	4
Глава 1. Структурные обратные задачи гравиметрии и магнитометрии	17
1.1 Постановка обратной задачи гравиметрии о восстановлении одной поверхности раздела	18
1.2 Постановка обратной задачи магнитометрии о восстановлении одной поверхности раздела	20
1.3 Постановка обратной задачи гравиметрии о нахождении нескольких поверхностей раздела	21
1.4 Постановка обратной задачи магнитометрии о нахождении нескольких поверхностей раздела	23
1.5 Постановка обратной задачи магнитометрии для случая произвольно направленной суммарной намагниченности	24
1.6 Схема аппроксимации интегрального оператора в задаче магнитометрии	26
Глава 2. Методы решения обратных задач гравиметрии и магнитометрии	30
2.1 Построение модифицированного покомпонентного метода	30
2.2 Построение модифицированного метода сопряженных градиентов	33
2.3 Построение гибридного метода сопряженных градиентов	35
2.4 Построение модифицированного метода сопряженных градиентов с весовыми множителями (ММСГВМ) для многослойной среды	36
2.5 Градиентные методы	37
2.6 Оптимизации по памяти для модифицированных методов	39
Глава 3. Комплекс программ и вычислительные эксперименты .	43
3.1 Обзор вычислительных технологий	43
3.2 Описание комплекса программ	44
3.3 Описание веб-портала	52
3.4 Нагрузочное тестирование	59

	Стр.
3.5 Численные эксперименты	64
3.5.1 Восстановление одной поверхности раздела по зашумленным гравитационным данным	65
3.5.2 Восстановление одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности	69
3.5.3 Восстановление трех поверхностей раздела по гравитационным данным с использованием реального гравитационного поля	76
3.5.4 Восстановление одной поверхности раздела по гравитационным данным	81
Заключение	85
Словарь терминов	86
Список литературы	89
Публикации автора по теме диссертации	97
Список рисунков	99
Список таблиц	102
Приложение А. Листинги программного кода	103

Введение

Актуальность и степень разработанности темы. Анализ гравитационных и магнитных полей с целью получения информации о структурных неоднородностях земной коры и верхней мантии представляет собой сложную проблему как с научной, так и с технической точек зрения. Геофизические данные применяются в задачах геологоразведки для поиска полезных ископаемых: руд, нефти, газа и многих других. Анализ магнитного поля используется, например, для изучения геологических процессов, таких как сдвиги плит и формирование горных пород. С развитием вычислительной техники появляется возможность повысить качество и степень разрешения геофизических неоднородностей. В задачах, где достигнутое разрешения является достаточным, можно существенно сократить время расчетов. Разработка программных средств для эффективного использования вычислительных систем при решении данного класса проблем является актуальной задачей.

Решение геофизических задач — вычислительно трудоемкий процесс, требующий использования сложного математического аппарата, а также передовых компьютерных технологий. При решении задач требуется хранение и обработка большого объема информации. Кроме того, алгоритмы обладают высокой вычислительной сложностью: увеличение объема входных данных существенно увеличивает время вычислений. Решение таких задач на персональном компьютере может занимать от нескольких часов до нескольких дней, поэтому целесообразно использовать параллельные вычисления как на графических ускорителях, так и на многопроцессорной системе. Использование графических ускорителей при этом является наиболее перспективным, так как позволяет сократить время решения задач на несколько порядков. Несмотря на значительное сокращение времени использование параллельных вычислений порождает ряд проблем. Во-первых, исследователю нужно быть не только специалистом в предметной области, но и в области вычислительной математики. Кроме этого, ему также необходимо владеть технологиями параллельного программирования. Во-вторых, вычислительные ресурсы, необходимые для исследований, имеют высокую стоимость и доступны далеко не каждому исследователю. Таким образом, получение результатов является сложной и трудоемкой работой, требующей не только мощных вычислительных

ресурсов, но и больших затрат человеческого труда и финансов. Вследствие этого появляется необходимость в создании специализированного веб-портала для удаленных вычислений, который позволил бы существенно облегчить и удешевить эти процессы.

Одними из важнейших задач являются обратные задачи гравиметрии и магнитометрии, впервые сформулированные в работах Б. В. Нумерова [1] и Н. Р. Малкина [2], о нахождении поверхностей раздела однородных сред с различными плотностями либо намагниченностями по известному гравитационному, либо магнитному полю. Работы различных ученых посвящены исследованию структурных обратных задач: Е. Г. Булаха [3; 4], А. И. Кобрунова [5], В. А. Кочнева [6], П. С. Мартышко [7–9], В. М. Новоселицкого, И. Л. Пруткина [10], В. Н. Страхова [11; 12], Н. В. Федоровой, А. В. Цирульского. Компании Schlumberger, Halliburton и Woodside предоставляют геологоразведочные коммерческие услуги, в том числе по интерпретации геофизических полей.

Задачи описываются нелинейными интегральными уравнениями первого рода и являются существенно некорректными задачами: имеют неединственное решение, неустойчиво зависящее от входных данных.

После дискретизации и аппроксимации задачи сводятся к системам нелинейных уравнений большой размерности. Построению численных методов для корректных нелинейных задач посвящены работы Л. В. Канторовича [13], J. Nocedal.

В 1902 году Адамар определил условия корректности математических задач. Он считал, что задачи, не удовлетворяющие этим условиям, то есть некорректные, не имеют физического смысла. Некорректные обратные задачи возникают в математической физике. В течение многих лет они решались без строгого математического обоснования, но в 1943 году академик А.Н. Тихонов опубликовал работу, в которой доказал устойчивость некоторых обратных задач при условии, что решение принадлежит компактному множеству. После публикации работы А.Н. Тихонова теория некорректных задач стала самостоятельным разделом математики. В 1950-х и 1960-х годах в СССР появились работы, в которых использовались идеи регуляризации для решения некоторых некорректных задач. Эти исследования были проведены выдающимися учеными А.Н. Тихоновым, М.М. Лаврентьевым и В.К. Ивановым. Они заложили основы современных методов регуляризации.

В. К. Иванов опубликовал работы [14; 15], в которых ввел понятие квазирешения. Метод квазирешений был разработан для решения обратных задач теории потенциала, в которых требуется найти распределение потенциала или зарядов на основании измерений потенциала или поля.

М. М. Лаврентьев в своей работе [16] предложил способы решения некорректных задач, включая обратные задачи теории потенциала.

А. Н. Тихонов [17–19] разработал наиболее общий метод решения некорректных задач - метод регуляризации. Метод регуляризации широко используется в различных областях, включая обратные задачи, обработку сигналов и машинное обучение.

Теория некорректных задач используется в различных областях науки и техники: радиофизика, геофизика, изображения, медицина. Теория некорректных задач также играет важную роль в современном машинном обучении, где она используется для обработки неполной информации и восстановления изображений с шумом. В современной математике существует множество различных методов регуляризации, каждый из которых предназначен для решения определенного типа некорректных задач.

Развитием теории некорректных задач занимались многие математики: А. Б. Бакушинский [20–23], В. В. Васин [24–28], А. Л. Агеев [24; 29], А. В. Гончарский [30], А. Г. Ягола, А. С. Леонов [30; 31], В. Г. Романов и др., С. И. Кабанихин, В. П. Танана [25; 32–36].

В работах А. Б. Бакушинского [37], М. Ю. Кокурина, В. В. Васина, В. П. Тананы, S. George, M. Hanke, V. Kaltenbacher, A. Neubauer, O. Scherzer были предложены и исследованы методы итеративной регуляризации на основе процессов градиентного и ньютоновского типов и их модифицированных аналогов. Итеративные методы регуляризации получили широкое применение в различных областях, таких как обработка изображений, медицинская визуализация, инженерные задачи, вычисления в математике и физике. Одним из главных преимуществ итеративных методов является то, что они могут эффективно решать задачи с большим количеством неизвестных, используя меньшее количество вычислений по сравнению с другими методами регуляризации.

В работах Л. В. Канторовича [13], Б. Т. Поляка [38], J. M. Ortega и W. C. Rheinboldt [39], M. J. D. Powell [40], J. E. Dennis, R. B. Schnabel, P. D. Frank, C. T. Kelley [41], R. B. Schnabel и P. D. Frank [42] были предложены методы для решения систем нелинейных уравнений в условиях регулярности.

Для решения линейных интегральных уравнений первого рода Ландвебером [43] был построен и обоснован итерационный метод. Данный метод был использован для решения нелинейных нерегулярных уравнений М. Hanke, А. Neubauer и О. Scherzer [44; 45]. Также они доказали теоремы о сходимости и исследовали скорость сходимости метода. Исследованием вопросов сходимости методов наискорейшего спуска и минимальной ошибки решения нелинейных некорректно поставленных задач занимались А. Neubauer и О. Scherzer в работах [46; 47].

А. В. Бакушинский для решения некорректных нелинейных задач предложил [21] модификацию метода Гаусса-Ньютона. Также этот метод был исследован в работах В. Kaltenbacher, А. Neubauer, О. Scherzer и др. [48; 49].

Схема построения итерационных процессов решения нелинейных нерегулярных уравнений на основе последовательной приближенной минимизации функционала Тихонова описывается и обосновывается в работах М. Ю. Кокурина [50; 51].

Модификация метода регуляризации была предложена В. П. Тананой в работе [33]. Также в данной работе доказывается сходимость регуляризованного решения двумерной обратной задачи гравиметрии.

Алгоритмы решения обратных задач математической физики на основе метода локальных поправок и его модификаций, а также методы комплексной интерпретации геофизических данных разрабатывались в ИГФ УрО РАН (П. С. Мартышко [7; 52], И. Л. Пруткин [10], И. В. Ладовский [53], Н. В. Федорова [54], А. Л. Рублев, Д. Д. Бызов, А. Г. Цидаев).

Разработкой и исследованием параллельных алгоритмов на основе итеративно регуляризованных градиентных методов и методов Ньютона и Левенберга-Марквардта занимались в ИММ УрО РАН (В. В. Васин [26], Е. Н. Акимова [55], Г. Я. Пересторонина, Л. Ю. Тимерханова, В. Е. Мислов, А. Ф. Скурыдина).

Методы типа Ньютона и Левенберга-Марквардта обладают высокой скоростью сходимости, но являются трудоемкими. В то же время методы на основе градиентного спуска имеют меньшую скорость сходимости, но являются более эффективными в вычислениях, так как требуют меньше вычислительных операций. Поэтому в зависимости от конкретной задачи и требуемой точности решения может быть целесообразным использование различных методов оптимизации.

Проблемам исследования и распараллеливания алгоритмов применительно к задачам математической физики посвящены работы В. В. Воеводина [56], Дж. Ортеги [39], Д. К. Фаддеева и В. Н. Фаддеевой [57]. Построению и исследованию параллельных алгоритмов для решения задач гравиметрии и магнитометрии посвящены работы Е. Н. Акимовой [58; 59].

Важной проблемой является доступность использования программ, реализующих численные алгоритмы на многопроцессорных вычислительных системах. Одним из способов решения задач является разработка системы удаленных вычислений, позволяющей запускать программы через веб-интерфейс. Перечислим некоторые способы улучшения доступности и удобства пользования программами:

1. Разработка интуитивно понятного интерфейса пользователя, который позволит легко настраивать параметры алгоритма и управлять его работой.
2. Предоставление документации и справочных материалов, которые помогут пользователям лучше понимать принципы работы алгоритма и эффективно использовать его для решения своих задач.
3. Предоставление возможности использовать программу в различных средах, таких как командная строка, графический интерфейс или через веб-интерфейс.
4. Обеспечение гибкости программы, чтобы она могла быть легко настроена и изменена в соответствии с потребностями пользователя.

В работе описывается веб-портал, предназначенный для решения этих проблем. В работе [60] описывается первый подход к системе удаленных вычислений. На данном этапе веб-портал позволяет пользователю запускать задание на вычисление решения обратной задачи гравиметрии и задачи Дирихле с использованием предложенных методов на МВС-1000/32. Всего на портале предустановлено 8 алгоритмов для решения данного типа задач. Пользователь может выбрать необходимый алгоритм, ввести входные параметры, выбрать количество процессоров и запустить задачу. После завершения расчетов у пользователя появляется возможность посмотреть изображение решения и скачать результаты. В работе [61] предложен ряд важных улучшений для веб-портала: система аутентификации пользователя, возможность использования графических ускорителей, выбор выходных файлов, по которым осуществляется построение изображений результатов расчета. Система аутентификации

пользователя проверяет учетную запись пользователя на суперкомпьютере МВС-ИММ. Для обеспечения безопасности добавлено SSL-шифрование. Также добавлена возможность выбора планировщика вычислительных ресурсов. В работе [62] добавлена возможность работы с суперкомпьютером УРАН.

Также имеются альтернативные наработки в данной отрасли. Во всех перечисленных далее работах описываются компоненты веб-портала по отдельности. Среди них не нашлось ни одного подходящего комплексного решения, но сами по себе компоненты достаточно полезны. В работе [63] описывается библиотека для удаленного запуска задач, без интерфейса и хранилища алгоритмов. Данная библиотека отлично подходит для автоматизации запуска задач на удаленном кластере. В работе [64] описывается веб-портал для запуска распределенных задач на вычислительном кластере. Особенностью портала является фиксированный список решаемых задач и работа с конкретным вычислительным кластером. Также существует продукт UNICORE [65] — сложная многомодульная система. Для работы требуется установка и настройка нескольких модулей, клиента для работы с порталом. Создание и загрузку задач требуется производить каждому пользователю. На базе этого продукта есть реализованные решения запуска собственных веб-порталов. В работе [66] на базе UNICORE предлагается реализация веб-портала для высокопроизводительных вычислений в области медицины. Инструмент достаточно гибкий, но при этом необходимо установить или разработать некоторое количество модулей, для того, чтобы поддержать свой процесс запуска задач. Как правило, модули необходимо устанавливать на вычислительные узлы, находящиеся в одной сети с кластером. В работе [67] описывается облачный веб-сервис для решения задач химической кинетики. Данное решение предполагает использование только предустановленных расчетных модулей с конкретным вычислительным кластером. Также можно отметить ручное заведение пользовательских учетных данных, что снижает степень автоматизации и время реакции.

Как правило, запросы на выполнение вычислений к веб-порталам осуществляются посредством различных методов API, представляющих собой комплексное решение, одно из которых описано в работе [68]. Работа представляет собой FirecREST API, обеспечивающий доступ к ресурсам высокопроизводительного компьютерного центра, что позволяет научным веб-порталам запускать задачи на удаленном кластере и впоследствии загружать на него (или скачивать с него) данные. Также имеется альтернативное решение в виде

NEWT API, описанного в работе [69]. Это API поддерживает аутентификацию пользователей и их работу с очередью задач на вычисления, с файлами, папками и другими хранимыми данными.

Объект и предмет исследования. В работе предлагаются менее ресурсоемкие по сравнению с классическими градиентными методами и процессами типа Гаусса-Ньютона итерационные методы решения нелинейных уравнений, возникающих при решении структурных обратных задач гравиметрии и магнитометрии о восстановлении поверхностей раздела сред, и их параллельная реализация на многоядерных процессорах. Применение разработанных параллельных алгоритмов и программ в значительной мере повышает эффективность решения задач.

В диссертационной работе предлагается развитие специализированного веб-портала, описанного в работах [60–62], включающее в себя идеи из альтернативных разработок и оптимизации существующих наработок. Одна из целей — сделать инструмент простым для установки, настройки и использования. Веб-портал позволяет пользователям загружать алгоритмы для решения своих собственных задач и делиться ими с другими пользователями системы. Так же появляется возможность подключения к различным вычислительным кластерам.

Целью работы является построение быстрых и экономичных по памяти итерационных алгоритмов на основе градиентных методов решения нелинейных обратных задач гравиметрии и магнитометрии о нахождении поверхностей раздела сред, реализация алгоритмов в виде комплекса программ для графических ускорителей и разработка системы удаленных вычислений для запуска разработанных программ на многопроцессорных системах.

Для достижения поставленной цели решены следующие **задачи**:

1. Разработан новый покомпонентный градиентный метод для решения обратной задачи магнитометрии в случае произвольно направленной намагниченности. Разработаны модификации метода сопряженных градиентов для задач гравиметрии и магнитометрии для двухслойной и многослойной среды с использованием блочно-теплицевой структуры матрицы.
2. Разработаны экономичные по памяти и времени счета параллельные алгоритмы (на основе методов автора и др.), реализованные в виде комплекса программ для GPU.

3. Разработана система удаленных вычислений для запуска разработанных программ на многопроцессорных системах и встроенный в нее комплекс программ.
4. Проведены эксперименты по исследованию сходимости численных методов, эффективности и ускорения параллельных алгоритмов. Для экспериментов построены модельные примеры и примеры на основе реальных гравитационных данных.

Методология и методы исследования. В диссертационной работе использован математический аппарат численных методов оптимизации, теории некорректных задач и методы математического моделирования. Для реализации алгоритмов на графических процессорах используется технология параллельного программирования NVidia CUDA.

Положения, выносимые на защиту:

1. Для решения обратной задачи магнитометрии о восстановлении поверхности раздела сред, в рамках модели двухслойной среды с произвольно направленными векторами намагниченности слоев, предложен покомпонентный градиентный метод, учитывающий поправку на отклонение от вертикали вектора разности намагниченности слоев.
2. На основе модифицированного метода сопряженных градиентов построены экономичные по памяти (с пространственной сложностью $O(n)$) алгоритмы решения обратных задач гравиметрии и магнитометрии о восстановлении поверхностей раздела сред, учитывающие блочно-теплицевую структуру матриц.
3. Разработан комплекс эффективных параллельных алгоритмов и программ решения обратных задач гравиметрии и магнитометрии с применением высокопроизводительных графических ускорителей, апробированный с помощью математического моделирования для квазиреальных данных с целью обоснования применимости предлагаемого подхода.
4. Разработан веб-портал для выполнения удаленных вычислений с возможностью запуска комплекса программ на многопроцессорных вычислительных системах.

Научная новизна. Результаты, представленные в работе, являются новыми и представляют теоретическую и практическую научную ценность.

1. Для модели двухслойной среды с произвольно направленными векторами намагниченности слоев построен новый покомпонентный градиентный метод решения структурной задачи магнитометрии о восстановлении поверхности раздела сред. Метод учитывает поправку на отклонение от вертикали вектора разности намагниченности слоев и использует экономичную схему аппроксимации интегрального оператора.
2. Построены экономичные по памяти модифицированные методы решения структурных обратных задач гравиметрии и магнитометрии, использующие блочно-теплицевую структуру матриц производных интегральных операторов, что снижает алгоритмическую и пространственную сложность методов.
3. Разработан комплекс программ для решения структурных обратных задач гравиметрии и магнитометрии на основе методов, представленных в работе. Для уменьшения времени счета и требуемого объема памяти предлагаемые алгоритмы построены на основе модифицированных градиентных методов с использованием блочно-теплицевой структуры матрицы производных.
4. Разработан веб-портал для удаленных вычислений, позволяющий запускать комплекс программ на многопроцессорных системах, добавлять новые программы и подключать новые вычислители.

Теоретическая и практическая значимость работы. Разработанные в диссертационной работе и апробированные в расчетах параллельные алгоритмы и программы могут быть эффективно использованы при численном решении на графических процессорах обратных задач теории потенциала: задач гравиметрии и магнитометрии о нахождении поверхностей раздела. Разработанные методы, алгоритмы и программы могут быть использованы в программных пакетах для решения задач гравиметрии и магнитометрии.

Степень достоверности полученных результатов. Достоверность результатов диссертации установлена проведенными численными экспериментами, надежность разработанных программ проверена расчетами при решении модельных задач. Надежность разработанного веб-портала проверена с помощью нагрузочного тестирования.

Апробация работы. Основные положения диссертационной работы докладывались и обсуждались на всероссийских и международных конференциях и семинарах:

1. Международная научная конференция «Параллельные вычислительные технологии» (Казань, 2017; Ростов-на-Дону, 2018; Калининград, 2019).
2. EAGE Geoinformatics (Киев, 2016).
3. Девятые научные чтения Ю.П. Булашевича (Екатеринбург, 2017).
4. Международный научный семинар им. Д.Г. Успенского (Пермь, 2019).
5. Национальный Суперкомпьютерный Форум (Переславль-Залесский, 2021).

Публикации. Основные результаты по теме диссертации изложены в 11 печатных изданиях, 2 из которых изданы в журналах, рекомендованных ВАК, 5 — в научных изданиях, индексируемых Web of Science и Scopus, 4 — в научных изданиях, индексируемых в РИНЦ. В Роспатент и РОСРИД зарегистрированы 3 программы для ЭВМ.

Личный вклад. Вклад автора отражен в основных положениях, выносимых на защиту и в опубликованных работах [83—93]. В работах [83; 84; 90] автору принадлежит построение параллельных алгоритмов на основе модифицированных градиентных методов с весовыми множителями, разработка программ для графических ускорителей. В работе [89] автору принадлежит разработка алгоритмов решения задач гравиметрии и магнитометрии о восстановлении поверхностей раздела сред, построение параллельных алгоритмов на основе методов градиентного типа и разработка программ для графических процессоров с использованием технологии NVidia CUDA. В работе [86] автору принадлежит построение модифицированного варианта регуляризованного метода сопряженных градиентов для решения обратных задач гравиметрии и магнитометрии. В работе [87] автору принадлежит построение модифицированного покомпонентного метода. В работах [85; 88; 92] автору принадлежит разработка программ для графических ускорителей. В работе [91] автору принадлежит построение схемы аппроксимации интегрального оператора. Автором разработан веб-портал для организации удаленных вычислений [93]. Программные компоненты комплекса программ зарегистрированы в Роспатенте [94—96].

Структура и объем работы. Диссертация состоит из введения, 3 глав, заключения и приложения. Полный объем диссертации составляет 122 стра-

ницы, включая 42 рисунка и 7 таблиц. Список литературы содержит 97 наименований.

Краткое содержание работы.

Во введении обосновывается актуальность исследований, проводимых в рамках диссертационной работы, приводится обзор научной литературы по изучаемой проблеме, формулируется цель, ставятся задачи работы, излагается научная новизна и практическая значимость представляемой работы.

Первая глава посвящена постановкам обратных структурных задач гравиметрии и магнитометрии о восстановлении поверхностей раздела сред по гравитационным либо магнитным данным. Показано, что рассматриваемые в диссертационной работе задачи сводятся к решению нелинейных уравнений. Построена схема аппроксимации интегрального оператора прямой задачи нахождения магнитного поля для поверхности раздела сред в случае произвольно направленных векторов намагниченности слоев, увеличивающая точность аппроксимации интегрального оператора по сравнению с квадратурными формулами.

В разделе 1.1 приводится постановка структурной обратной задачи гравиметрии для одной границы. В разделе 1.2 приводится постановка структурной обратной задачи магнитометрии для одной границы. В разделе 1.3 приводится постановка структурной обратной задачи гравиметрии для нескольких поверхностей раздела. В разделе 1.4 приводится постановка структурной обратной задачи магнитометрии для нескольких поверхностей раздела. В разделе 1.5 приводится постановка структурной обратной задачи магнитометрии для случая произвольно направленной суммарной намагниченности. Раздел 1.6 посвящен построению схемы аппроксимации интегрального оператора задачи нахождения поля второй производной ньютоновского потенциала для контактной поверхности в случае произвольно направленных векторов намагниченности слоев. Данная схема увеличивает точность аппроксимации интегрального оператора в уравнении по сравнению с квадратурными формулами.

Вторая глава посвящена построению оригинальных экономичных по памяти и времени счета алгоритмов решения задач, поставленных в первой главе. Построен новый покомпонентный метод для решения обратной задачи магнитометрии в случае произвольно направленного вектора намагниченности для одной границы раздела сред. Построены экономичные модифицированные алгоритмы решения обратных задач гравиметрии и магнитометрии на основе метода

сопряженных градиентов с использованием блочно-теплицевой структуры матрицы производных интегрального оператора.

Раздел 2.1 посвящен построению нового экономичного покомпонентного метода для решения обратной задачи магнитометрии для произвольно направленной намагниченности. В разделе 2.2 описывается построение модифицированного метода сопряженных градиентов. В разделе 2.3 описывается построение гибридного метода сопряженных градиентов. В разделе 2.4 предлагается модифицированный метод сопряженных градиентов с весовыми множителями для многослойной среды. В разделе 2.5 приводятся методы других авторов, реализованные автором в виде параллельных программ, встроенных в систему удаленных вычислений, используемые в диссертационной работе для решения обратных задач гравиметрии и магнитометрии: регуляризованный метод сопряженных градиентов и модифицированные альфа-процессы. В разделе 2.6 описываются оптимизации по памяти для модифицированных методов.

Третья глава посвящена разработке комплекса параллельных программ решения структурных обратных задач гравиметрии и магнитометрии на графических ускорителях и тестированию комплекса на построенных модельных примерах с использованием реальных гравитационных полей. Разработан веб-портал для удаленных вычислений, позволяющий запускать комплекс программ на многопроцессорных системах. Комплекс программ интегрирован в веб-портал. Раздел 3.1 приводится обзор технологий, используемых в комплексе программ. В том числе, OpenMP, MPI, CUDA. Раздел 3.2 посвящен построению параллельных алгоритмов для графических процессоров. На основе вышеописанных и ранее разработанных методов для решения структурных обратных задач гравиметрии и магнитометрии разработаны алгоритмы для графических ускорителей, входящие в состав комплекса программ. В разделе 3.3 приводится описание универсального веб-портала: схема БД, схема взаимодействия с пользователем, основные функциональные возможности. В разделе 3.4 приведено описание нагрузочного тестирования веб-портала. Раздел 3.5 описывает численные эксперименты для решения модельных задач с помощью разработанных методов и параллельных программ. Проведенные численные эксперименты показывают, что методы и алгоритмы, с точки зрения времени счета и требуемых вычислительных ресурсов, являются эффективными и экономичными.

В заключении приводятся основные выводы и рекомендации, перспективы дальнейшей разработки темы.

Благодарности. Автор выражает искреннюю признательность своему научному руководителю — доктору физико-математических наук, ведущему научному сотруднику ИММ УрО РАН Елене Николаевне Акимовой.

Автор выражает благодарность за полезные замечания и обсуждения члену-корреспонденту РАН, доктору физико-математических наук, главному научному сотруднику ИММ УрО РАН Владимиру Васильевичу Васину.

Автор выражает благодарность за полезные замечания члену-корреспонденту РАН, доктору физико-математических наук, зав. лабораторией математической геофизики ИГФ УрО РАН Петру Сергеевичу Мартышко.

Автор выражает благодарность за полезные замечания и обсуждения кандидату физико-математических наук, научному сотруднику ИММ УрО РАН Владимиру Евгеньевичу Мисиллову.

Глава 1. Структурные обратные задачи гравиметрии и магнитометрии

Для удобства изложения вместо громоздких формулировок «поле первой производной ньютоновского потенциала» и «обратная задача для поля первой производной ньютоновского потенциала» будем использовать «гравитационное поле» и «задача гравиметрии». Вместо формулировки «поле второй производной ньютоновского потенциала» и «обратная задача для поля второй производной ньютоновского потенциала» будем использовать «магнитное поле» и «задача магнитометрии».

Первая глава посвящена постановкам обратных структурных задач гравиметрии и магнитометрии о восстановлении поверхностей раздела сред по гравитационным либо магнитным данным. Показано, что рассматриваемые в диссертационной работе задачи сводятся к решению нелинейных уравнений. Построена схема аппроксимации интегрального оператора прямой задачи нахождения магнитного поля для поверхности раздела сред в случае произвольно направленных векторов намагниченности слоев, увеличивающая точность аппроксимации интегрального оператора по сравнению с квадратурными формулами.

Обратные задачи гравиметрии и магнитометрии являются существенно некорректными задачами, решение которых обладает сильной чувствительностью к погрешностям правых частей, полученных в результате измерений и предварительной обработки геофизических данных. Поэтому при решении задач используются методы итеративной регуляризации.

Разработанные методы, алгоритмы и программы могут быть использованы в реальных пакетах для интерпретации геофизических данных.

В разделе 1.1 приводится постановка структурной обратной задачи гравиметрии для одной границы. В разделе 1.2 приводится постановка структурной обратной задачи магнитометрии для одной границы. В разделе 1.3 приводится постановка структурной обратной задачи гравиметрии для нескольких поверхностей раздела. В разделе 1.4 приводится постановка структурной обратной задачи магнитометрии для нескольких поверхностей раздела. В разделе 1.5 приводится постановка структурной обратной задачи магнитометрии для случая произвольно направленной суммарной намагниченности.

1.1 Постановка обратной задачи гравиметрии о восстановлении одной поверхности раздела

Рассмотрим постановку обратной задачи гравиметрии о восстановлении одной поверхности раздела.

Введем трехмерную декартову систему координат, в которой плоскость xOy совпадает с земной поверхностью, а ось Oz направлена вертикально вниз. Предполагается, что нижнее полупространство состоит из двух слоев постоянной плотности, разделенных искомой поверхностью S (рис. 1.1).

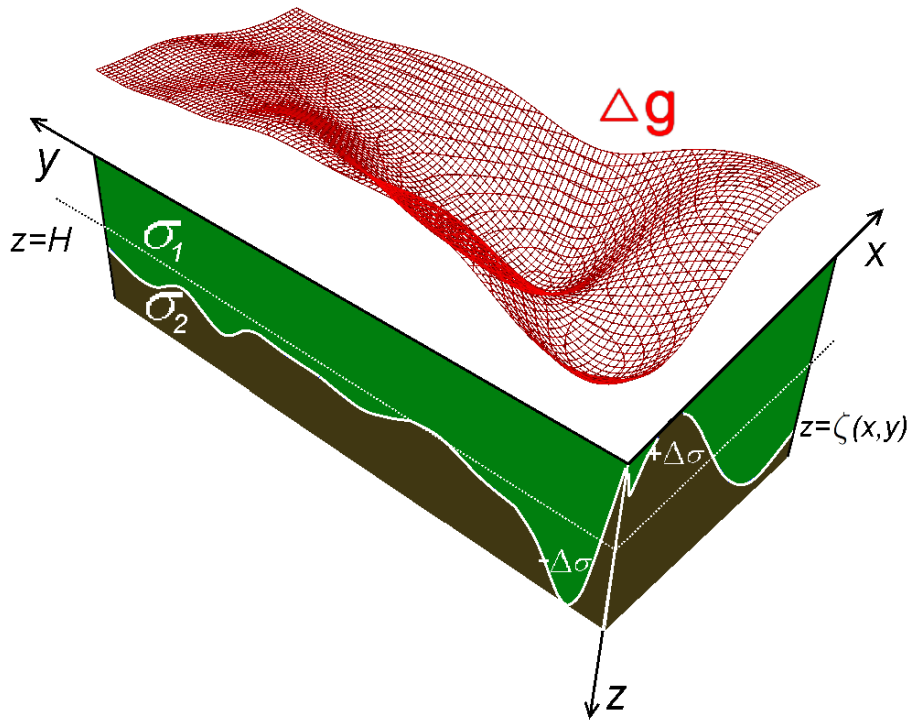


Рисунок 1.1 — Модель двухслойной среды в задаче гравиметрии

Пусть поверхность раздела задается уравнением $z = \zeta(x, y)$, скачок плотности на ней равен $\Delta\sigma$, поверхность имеют горизонтальную асимптотическую плоскость $z = h$, т.е.

$$\lim_{x \rightarrow \pm\infty, y \rightarrow \pm\infty} |\zeta(x, y) - h| = 0.$$

В предположении, что гравитационная аномалия создана отклонением исходной поверхности от горизонтальной плоскости $z = h$, в декартовой системе

координат функция, описывающая искомую поверхность, удовлетворяет двумерному нелинейному интегральному уравнению первого рода [1]

$$f\Delta\sigma \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\frac{1}{\sqrt{(x-x')^2 + (y-y')^2 + \zeta^2(x,y)}} - \frac{1}{\sqrt{(x-x')^2 + (y-y')^2 + h^2}} \right] dx dy = \Delta g(x',y',0), \quad (1.1)$$

где f — гравитационная постоянная, $\Delta\sigma$ — скачок плотности на границе, $\zeta(x,y)$ — искомая поверхность, $\Delta g(x',y',0)$ — аномальное гравитационное поле, измеренное на поверхности.

После дискретизации уравнения (1.1) на сетке $n = M \times N$ (рис. 1.2, $a \leq x \leq b, c \leq y \leq d$) с шагами $\Delta x, \Delta y$, где задана правая часть $\Delta g(x',y',0)$, имеем систему n нелинейных уравнений $A(z) = F$, где F — вектор правой части размерности n ; z — вектор решения.

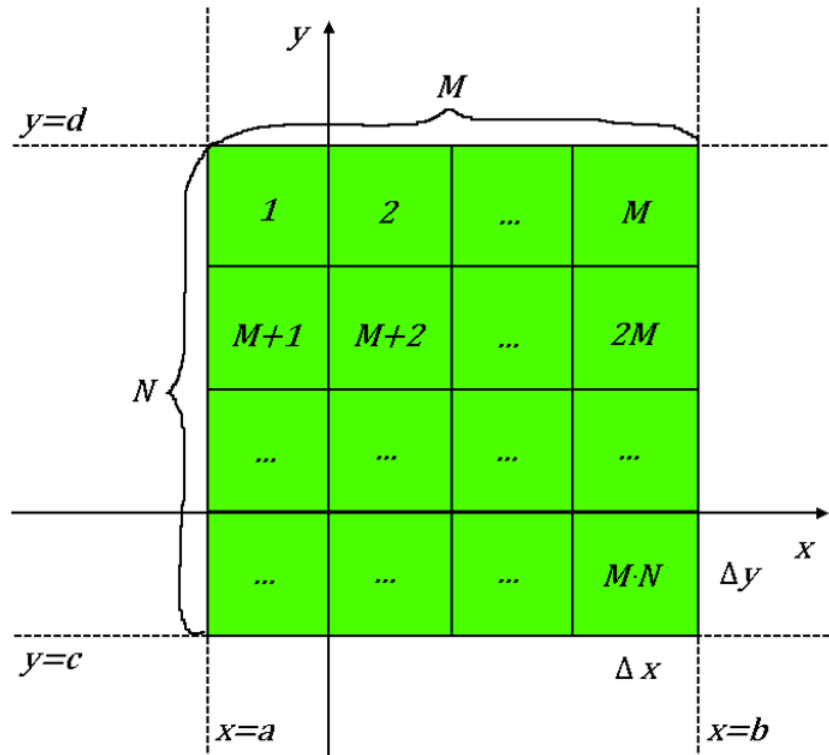


Рисунок 1.2 — Распределение индексов по сетке

1.2 Постановка обратной задачи магнитометрии о восстановлении одной поверхности раздела

Предполагается, что нижнее полупространство состоит из двух слоев постоянной вертикально направленной намагниченности J_1 и J_2 , разделенных искомой поверхностью S (рис. 1.3). Плоскость xOy совпадает с поверхностью Земли, ось Oz направлена вниз.

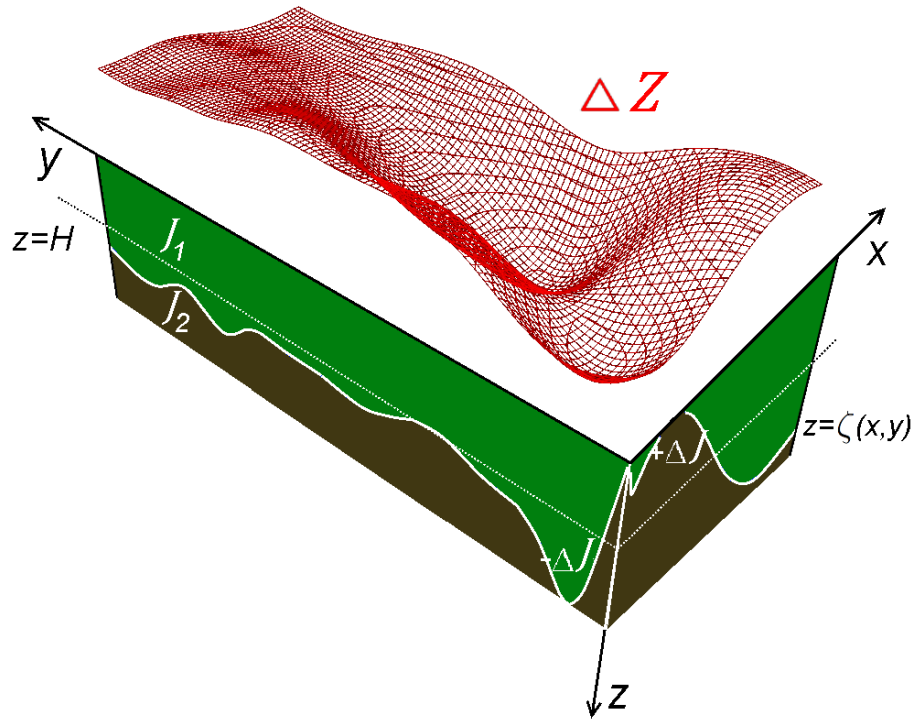


Рисунок 1.3 — Модель двухслойной среды в задаче магнитометрии

Пусть поверхность раздела задается уравнением $z = \zeta(x, y)$, поверхность имеет горизонтальную асимптотическую плоскость $z = h$, т.е.

$$\lim_{x \rightarrow \pm\infty, y \rightarrow \pm\infty} |\zeta(x, y) - h| = 0.$$

Функция, описывающая искомую поверхность раздела, удовлетворяет двумерному интегральному уравнению первого рода [2]

$$\frac{\Delta J}{4\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\frac{\zeta(x, y)}{\left((x - x')^2 + (y - y')^2 + \zeta^2(x, y) \right)^{3/2}} - \frac{h}{\left((x - x')^2 + (y - y')^2 + h^2 \right)^{3/2}} \right] dx dy = \Delta Z(x', y', 0), \quad (1.2)$$

где ΔJ — скачок вертикальной компоненты вектора намагниченности, $\zeta(x,y)$ — искомая поверхность, $\Delta Z(x',y',0)$ — аномальное магнитное поле (вертикальная составляющая напряженности магнитного поля).

После дискретизации уравнения (1.2) на сетке $n = M \times N$ (рис. 1.2) с шагами $\Delta x, \Delta y$, где задана правая часть $\Delta Z(x',y',0)$, имеем систему n нелинейных уравнений $B(z) = F$, где F — вектор правой части размерности n ; z — вектор решения.

1.3 Постановка обратной задачи гравиметрии о нахождении нескольких поверхностей раздела

Рассмотрим постановку обратной задачи гравиметрии о нахождении нескольких поверхностей раздела. Данная модель является обобщением модели двухслойной среды для задачи гравиметрии.

Введем трехмерную декартову систему координат, в которой плоскость xOy совпадает с земной поверхностью, а ось Oz направлена вертикально вниз. Предполагается, что нижнее полупространство состоит из нескольких слоев постоянной плотности, разделенных искомыми поверхностями $S_l, l = 1, \dots, L$, где L — известное число границ раздела (рис. 1.4).

Пусть поверхности раздела задаются уравнениями $z_l = \zeta_l(x,y)$, скачки плотности на них равны $\Delta\sigma_l$, поверхности имеют горизонтальные асимптотические плоскости $z_l = h_l$, т.е.

$$\lim_{x \rightarrow \pm\infty, y \rightarrow \pm\infty} |\zeta_l(x,y) - h_l| = 0.$$

Поле от суперпозиции границ равно [7]

$$f \sum_{l=1}^L \Delta\sigma_l \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\frac{1}{\sqrt{(x-x')^2 + (y-y')^2 + \zeta_l^2(x,y)}} - \frac{1}{\sqrt{(x-x')^2 + (y-y')^2 + h_l^2}} \right] dx dy = \Delta g(x',y',0), \quad (1.3)$$

где f — гравитационная постоянная, $\Delta\sigma_l$ — скачки плотности на границах, $\zeta_l(x,y)$ — искомые поверхности, $\Delta g(x',y',0)$ — аномальное гравитационное поле, заданное на поверхности.

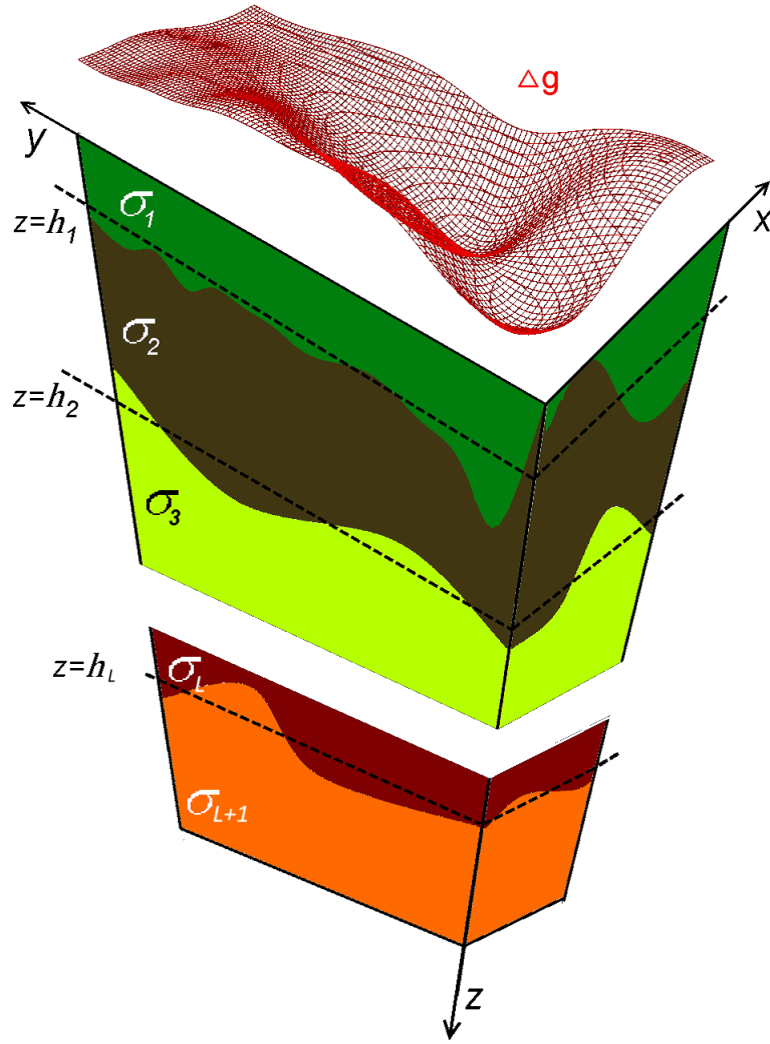


Рисунок 1.4 — Модель многослойной среды в задаче гравиметрии

Предполагается, что решение существует. Для исключения неединственности используется априорная информация: начальное приближение (достаточно близкое к решению); известные плотности слоев (разности плотностей); глубины асимптотических плоскостей. Весовые множители выбираются из разделенных полей. Нелинейное интегральное уравнение первого рода неустойчиво, поэтому используется регуляризация.

После дискретизации уравнения (1.3) на сетке $n = M \times N$ с шагами Δx , Δy , где задана правая часть $\Delta g(x', y', 0)$, имеем систему n нелинейных уравнений $A(z) = F$, где $F = (F_1, F_2, \dots, F_L)^\top$ — вектор правой части размерности n ; $z = (z_{1,1}, z_{1,2}, \dots, z_{1,n}, z_{2,1}, \dots, z_{2,n}, \dots, z_{L,1}, \dots, z_{L,n})^\top$ — вектор решения, составленный из векторов для каждой границы размерности $L \cdot n = L \cdot M \cdot N$.

1.4 Постановка обратной задачи магнитометрии о нахождении нескольких поверхностей раздела

Предполагается, что нижнее полупространство состоит из нескольких слоев постоянной вертикально направленной намагниченности $J_l = J_l^z$, $l = 1, \dots, L + 1$, разделенных искомыми поверхностями $S_l, l = 1, \dots, L$, где L — априорно известное число границ раздела. Плоскость xOy совпадает с поверхностью Земли, ось Oz направлена вниз.

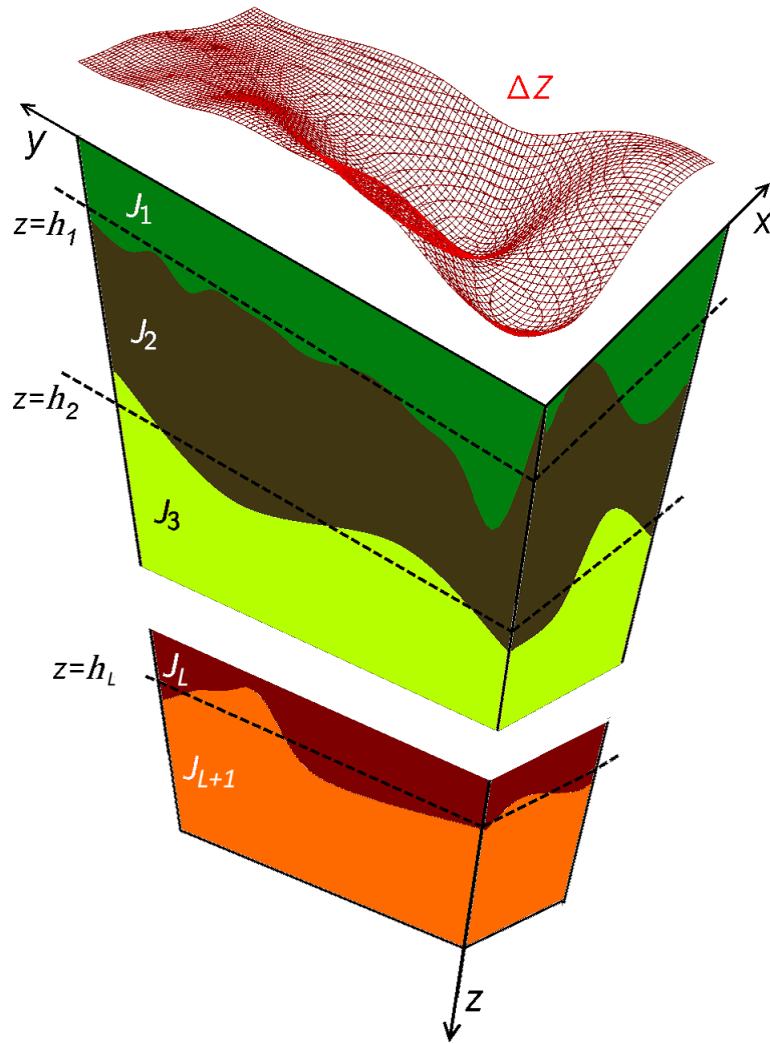


Рисунок 1.5 — Модель многослойной среды в задаче магнитометрии

Пусть поверхности раздела задаются уравнениями $z_l = \zeta_l(x, y)$, поверхности имеют горизонтальные асимптотические плоскости $z_l = h_l$, т.е.

$$\lim_{x \rightarrow \pm\infty, y \rightarrow \pm\infty} |\zeta_l(x, y) - h_l| = 0.$$

Поле от суперпозиции границ равно [52]

$$\sum_{l=1}^L \frac{\Delta J_l}{4\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\frac{\zeta_l(x,y)}{\left((x-x')^2 + (y-y')^2 + \zeta_l^2(x,y) \right)^{3/2}} - \frac{h}{\left((x-x')^2 + (y-y')^2 + h_l^2 \right)^{3/2}} \right] dx dy = \Delta Z(x',y',0), \quad (1.4)$$

где $\Delta J_l = J_{l+1} - J_l$ — скачки вертикальных компонент векторов намагниченности слоев, $\zeta_l(x,y)$ — искомые поверхности, $\Delta Z(x',y',0)$ — аномальное магнитное поле, измеренное на поверхности.

Предполагается, что решение существует. Для исключения неединственности используется априорная информация: начальное приближение (достаточно близкое к решению); известные плотности слоев (разности плотностей); глубины асимптотических плоскостей. Весовые множители выбираются из разделенных полей. Нелинейное интегральное уравнение первого рода неустойчиво, поэтому используется регуляризация.

После дискретизации уравнения (1.4) на сетке $n = M \times N$ (рис. 1.2) с шагами Δx , Δy , где задана правая часть $\Delta Z(x',y',0)$, имеем систему n нелинейных уравнений $B(z) = F$, где $F = (F_1, F_2, \dots, F_L)^T$ — вектор правой части размерности n ; $z = (z_{1,1}, z_{1,2}, \dots, z_{1,n}, z_{2,1}, \dots, z_{2,n}, \dots, z_{L,1}, \dots, z_{L,n})^T$ — вектор решения, составленный из векторов для каждой границы размерности $L \cdot n = L \cdot M \cdot N$.

1.5 Постановка обратной задачи магнитометрии для случая произвольно направленной суммарной намагниченности

В данном разделе приводится постановка структурной обратной задачи магнитометрии для случая произвольно направленной намагниченности. Разное направление остаточной намагниченности говорит о разном возрасте пород в слоях.

Предполагается, что нижнее полупространство состоит из двух слоев произвольно направленной намагниченности J_1 и J_2 (различными по модулю и по направлению), разделенных искомой поверхностью (рис. 1.6).

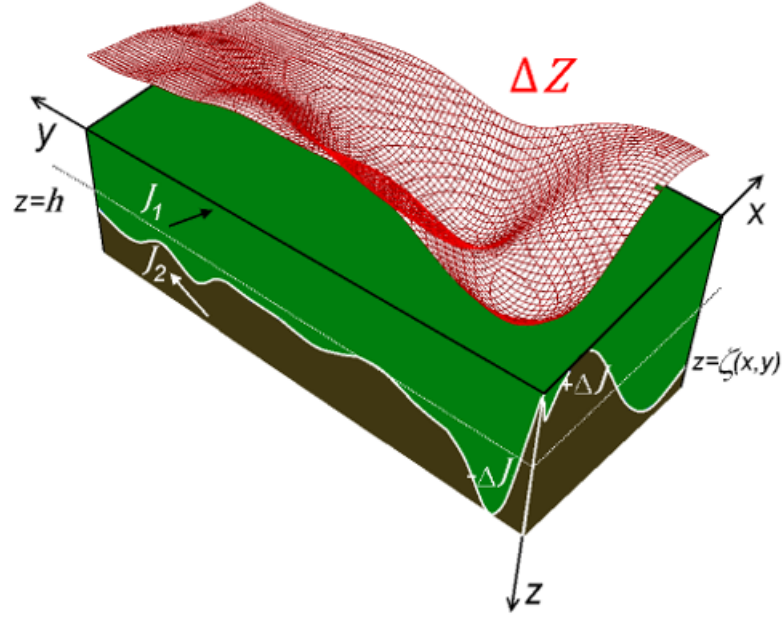


Рисунок 1.6 — Модель двухслойной среды в задаче магнитометрии для случая произвольно направленной намагниченности

Пусть поверхность раздела задается уравнением $z = \zeta(x, y)$, $\Delta J = J_2 - J_1 = (\Delta J_x, \Delta J_y, \Delta J_z)^\top$ — разность векторов намагниченности слоев. Поверхность имеет горизонтальную асимптотическую плоскость $z = h$, т.е.

$$\lim_{x \rightarrow \pm\infty, y \rightarrow \pm\infty} |\zeta(x, y) - h| = 0.$$

Поле от поверхности раздела равно [70]

$$\frac{1}{4\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\frac{\Delta J_x (x - x') + \Delta J_y (y - y') - \Delta J_z h}{\left((x - x')^2 + (y - y')^2 + h^2 \right)^{3/2}} - \frac{\Delta J_x (x - x') + \Delta J_y (y - y') - \Delta J_z \zeta(x, y)}{\left((x - x')^2 + (y - y')^2 + \zeta^2(x, y) \right)^{3/2}} \right] dx dy = \Delta Z(x', y', 0), \quad (1.5)$$

где $\Delta J = J_2 - J_1 = (\Delta J_x, \Delta J_y, \Delta J_z)^\top$ — разность векторов намагниченности слоев, $\Delta Z(x', y', 0)$ — аномальное магнитное поле, измеренное на поверхности.

После дискретизации уравнения (1.5) получаем систему нелинейных уравнений $B(z) = F$, где F — вектор правой части размерности $M \cdot N$.

1.6 Схема аппроксимации интегрального оператора в задаче магнитометрии

Построим схему аппроксимации интегрального оператора задачи нахождения поля второй производной ньютоновского потенциала для контактной поверхности в случае произвольно направленных векторов намагниченности слоев. Эксперименты (см. главу 3) показали, что предложенная схема увеличивает точность аппроксимации интегрального оператора в уравнении (1.5) по сравнению с квадратурными формулами.

В отличие от подхода получения интегрального уравнения с последующей дискретизацией, предлагается подход на основе идеи конечных объемов. Такой подход для задачи гравиметрии был предложен в работе [7]. Дискретизация проводится в процессе получения уравнения.

Для каждой ячейки сетки с индексами (i, j) строится элементарный параллелепипед высотой $|\Delta h|$, $\Delta h = \zeta(x_i, y_j) - h$, имеющий намагниченность $J = (J_x, J_y, J_z) = \text{sign } \Delta h \cdot \Delta J$. Поле от криволинейной поверхности будет равно сумме полей от всех элементов разбиения:

$$\Delta Z(x', y', z') \approx \sum_{ij} \Delta Z_{ij}(x', y', z').$$

Найдем выражение для поля прямоугольного параллелепипеда со сторонами, параллельными координатным осям. Для этого воспользуемся теоремой Пуассона [71].

$$U(x', y', z') = -\frac{1}{4\pi} \left\langle J, \text{grad } V(x', y', z') \right\rangle,$$

где $V(x', y', z')$ — гравитационный потенциал, создаваемый однородным параллелепипедом в точке (x', y', z') . Вертикальная компонента напряженности магнитного поля (производная потенциала U по z) имеет вид:

$$Z(x', y', z') = \frac{\partial U(x', y', z')}{\partial z} = -\frac{1}{4\pi} (J_x V_{xz} + J_y V_{yz} + J_z V_{zz}).$$

Выражения для вторых производных V_{xz} и V_{yz} потенциала приводятся в [72]

$$V_{xz} = \left| \left| \left| (\eta - y') + \sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2} \right| \right|_{\xi_1}^{\xi_2} \right|_{\eta_1}^{\eta_2} \Big|_{\zeta_1}^{\zeta_2},$$

$$V_{yz} = \left| \left| \left| (\xi - x') + \sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2} \right| \right|_{\xi_1}^{\xi_2} \right|_{\eta_1}^{\eta_2} \Big|_{\zeta_1}^{\zeta_2}.$$

Получим выражение для V_{zz} . Производная потенциала однородного тела произвольной формы:

$$V_{zz} = 3 \int_V \frac{(\zeta - z')^2}{\left((\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2 \right)^{5/2}} d\xi d\eta d\zeta.$$

Перейдем от интеграла по объему к тройному интегралу с пределами, соответствующими параллелепипеду $V = (x', y', z') : \xi_1 \leq x' \leq \xi_2, \eta_1 \leq y' \leq \eta_2, \zeta_1 \leq z' \leq \zeta_2$

$$V_{zz} = 3 \int_{\xi_1}^{\xi_2} \int_{\eta_1}^{\eta_2} \int_{\zeta_1}^{\zeta_2} \frac{(\zeta - z')^2}{\left((\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2 \right)^{5/2}} d\xi d\eta d\zeta.$$

Возьмем первообразную по ζ и применим формулу Ньютона-Лейбница

$$\begin{aligned}
V_{zz} = & \left| \int_{\xi_1}^{\xi_2} \int_{\eta_1}^{\eta_2} \left[-3 \left((\xi - x')^2 + (\eta - y')^2 \right) \times \right. \right. \\
& \times \left. \left\{ -(\zeta - z') \left(\sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2} - \zeta + z' \right) + \right. \right. \\
& \left. \left. + (\xi - x')^2 + (\eta - y')^2 \right\} + \right. \\
& \left. + 3 \left\{ -(\zeta - z') \left(\sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2} - \zeta + z' \right) + \right. \right. \\
& \left. \left. + (\xi - x')^2 + (\eta - y')^2 \right\}^2 + \right. \\
& \left. + \left((\xi - x')^2 + (\eta - y')^2 \right)^2 \right] / \\
& / \left[-(\zeta - z') \left(\sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2} - \zeta + z' \right) + \right. \\
& \left. + (\xi - x')^2 + (\eta - y')^2 \right]^3 d\xi d\eta \Big|_{\zeta_1}^{\zeta_2}.
\end{aligned}$$

Аналогично по η

$$\begin{aligned}
V_{zz} = & \left\| \int_{\xi_1}^{\xi_2} \frac{2 \operatorname{arctg} \left(\frac{\sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2} - \eta - \zeta + y' + z'}{\xi - x'} \right)}{x' - \xi} \right. \\
& \left. - \frac{(\eta - y')(\zeta - z')}{\left((\xi - x')^2 + (\zeta - z')^2 \right) \sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2}} d\xi \right\|_{\zeta_1}^{\zeta_2} \Big|_{\eta_1}^{\eta_2}.
\end{aligned}$$

Аналогично по ξ

$$V_{zz} = \left\| \left\| \operatorname{arctg} \left(\frac{(\eta - y')(\zeta - z')}{(\zeta - z')^2 - (\xi - x') \left(\sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2} - \xi + x' \right)} \right) + \int \frac{2 \operatorname{arctg} \left(\frac{\sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2} - \eta - \zeta + y' + z'}{\xi - x'}}{x' - \xi} d\xi \right) \right\|_{\zeta_1}^{\zeta_2} \right\|_{\eta_1}^{\eta_2}.$$

В конечном итоге получаем уравнение:

$$V_{zz} = \left\| \left\| \operatorname{arctg} \left(\frac{(\xi - x')(\eta - y')}{(\zeta - z') \sqrt{(\xi - x')^2 + (\eta - y')^2 + (\zeta - z')^2}} \right) \right\|_{\zeta_1}^{\zeta_2} \right\|_{\eta_1}^{\eta_2} \Big|_{\xi_1}^{\xi_2}.$$

Для элементарного параллелепипеда в ячейке сетки (i, j) пределы будут следующими: $\xi_1 = x_i$, $\xi_2 = x_{i+1}$; $\eta_1 = y_j$, $\eta_2 = y_{j+1}$; $\zeta_1 = \zeta(x_i, y_j)$, $\zeta_2 = h$.

Таким образом, выражение для поля от поверхности раздела (аналог формулы (1.5)) принимает вид:

$$-\frac{1}{4\pi} \sum_{j=1}^N \sum_{i=1}^M (J_x V_{xz_{ij}} + J_y V_{yz_{ij}} + J_z V_{zz_{ij}}) = \Delta Z(x', y', 0),$$

$$V_{xz_{ij}} = \left\| \left\| \left(\eta - y' + \sqrt{(\xi - x')^2 + (\eta - y')^2 + \zeta^2} \right) \right\|_{\xi=x_i}^{\xi=x_{i+1}} \right\|_{\eta=y_i}^{\eta=y_{i+1}} \Big|_{\zeta=\zeta(x_i, y_j)}^{\zeta=h},$$

$$V_{yz_{ij}} = \left\| \left\| \left(\xi - x' + \sqrt{(\xi - x')^2 + (\eta - y')^2 + \zeta^2} \right) \right\|_{\xi=x_i}^{\xi=x_{i+1}} \right\|_{\eta=y_i}^{\eta=y_{i+1}} \Big|_{\zeta=\zeta(x_i, y_j)}^{\zeta=h},$$

$$V_{zz_{ij}} = \left\| \left\| \operatorname{arctg} \left(\frac{(\xi - x')(\eta - y')}{\zeta \sqrt{(\xi - x')^2 + (\eta - y')^2 + \zeta^2}} \right) \right\|_{\zeta=\zeta(x_i, y_j)}^{\zeta=h} \right\|_{\eta=y_i}^{\eta=y_{i+1}} \Big|_{\xi=x_i}^{\xi=x_{i+1}}.$$

Примечание. Оценка погрешности по правилу Рунге показала, что предложенная схема аппроксимации дает меньшую погрешность (в 2 раза) по сравнению с аппроксимацией интеграла по квадратурным формулам.

Глава 2. Методы решения обратных задач гравиметрии и магнитометрии

Вторая глава посвящена построению экономичных по памяти и времени счета алгоритмов решения задач, поставленных в первой главе. Разработан новый покомпонентный метод для решения обратной задачи магнитометрии в случае произвольно направленного вектора намагниченности для одной границы раздела сред. Разработаны экономичные модифицированные алгоритмы решения обратных задач гравиметрии и магнитометрии на основе метода сопряженных градиентов с использованием блочно-теплицевой структуры матрицы производных интегрального оператора.

Раздел 2.1 посвящен построению нового экономичного покомпонентного метода для решения обратной задачи магнитометрии для произвольно направленной намагниченности. В разделе 2.2 описывается построение модифицированного метода сопряженных градиентов. В разделе 2.3 описывается построение гибридного метода сопряженных градиентов. В разделе 2.4 предлагается модифицированной методѣ сопряженных градиентов с весовыми множителями для многослойной среды. В разделе 2.5 приводятся методы других авторов, реализованные автором в виде параллельных программ, встроенных в систему удаленных вычислений, используемые в диссертационной работе для решения обратных задач гравиметрии и магнитометрии: регуляризованный метод сопряженных градиентов и модифицированные альфа-процессы. В разделе 2.6 описываются оптимизации по памяти для модифицированных методов.

2.1 Построение модифицированного покомпонентного метода

Данный раздел посвящен построению нового экономичного покомпонентного метода для решения обратной задачи магнитометрии для произвольно направленной намагниченности. В работе [73] для задачи гравиметрии предложен метод локальных поправок. Идея основана на том, что компонента решения оказывает наибольшее влияние на компоненту правой части (поля),

соответствующую точку на поверхности над собой. В работе [74] получен покомпонентный метод в случае вертикально направленной намагниченности (рис. 2.1, а).

Покомпонентный метод имеет вид:

$$z_i^{k+1} = z_i^k - \psi \frac{A_i(z^k) - F_i}{\|\nabla A_i(z^k)\|^2} \left(\frac{\partial A_i(z^k)}{\partial z_i} \right), \quad (2.1)$$

где z_i — i -й компонент приближенного решения, $i = 1, \dots, n$; $k = 0, 1, 2, \dots$; $\psi > 0$ — демпфирующий множитель.

Примечание. Во всех методах, описанных в данной главе, используется условие останова:

$$\frac{\|A(z) - F\|}{\|F\|} < \varepsilon, \quad \varepsilon > 0. \quad (2.2)$$

Построим модифицированный вариант покомпонентного метода. Модификация покомпонентного метода для случая произвольного направленной намагниченности основана на смещении индексов компоненты решения и компоненты правой части в зависимости от угла отклонения от вертикали вектора намагниченности слоев (рис. 2.1, б). Предполагается, что искомая поверхность расположена достаточно глубоко и влияние глубины залегания границы сравнимо с влиянием наклона намагниченности.

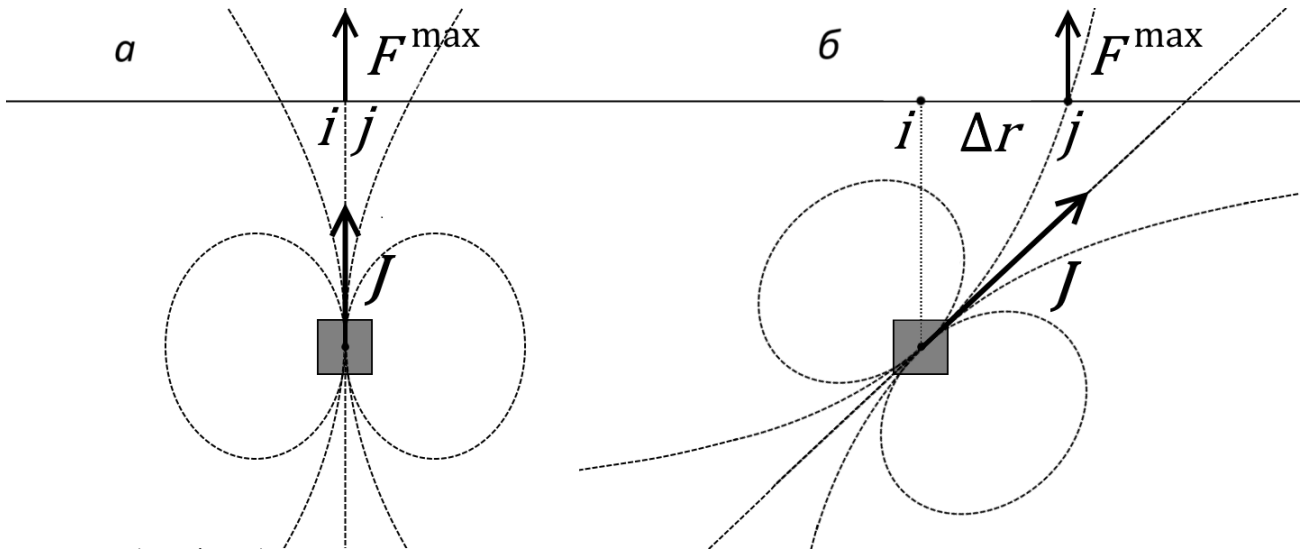


Рисунок 2.1 — Расположение точки j , соответствующей компоненте правой части, на которую оказывает максимальное влияние компонента решения, соответствующая точке i

Найдем приближение новой точки j , в которой наибольшее влияние на значение поля F_j оказывает значение глубины залегания z_i для случая произвольно направленной намагниченности. Эта точка смещена относительно точки i на \bar{x} и \bar{y} . Для нахождения \bar{x} , решим уравнение:

$$\bar{x} = \arg \max_x \left[- \frac{\Delta J_x \cdot x - \Delta J_z \cdot h}{(x^2 + h^2)^{3/2}} \right].$$

Необходимое условие максимума:

$$\frac{d}{dx} \left[- \frac{\Delta J_x \cdot x - \Delta J_z \cdot h}{(x^2 + h^2)^{3/2}} \right] = 0.$$

Посчитаем производную:

$$- \frac{\Delta J_x \cdot (2x^2 - h^2) + 3\Delta J_z \cdot h \cdot x}{(x^2 + h^2)^{5/2}} = 0.$$

Очевидно, $x \neq 0$ для случая невертикальной намагниченности и поверхности, находящейся ниже уровня земной поверхности, то есть $h > 0$, соответственно получаем

$$\Delta J_x(2x^2 - h^2) + 3\Delta J_z h x = 0.$$

Вычислим корни уравнения:

$$\bar{x}_{1,2} = \frac{(-3\Delta J_z \pm \sqrt{8\Delta J_x^2 + 9\Delta J_z^2})h}{4\Delta J_x}.$$

Предположим, что $\Delta J_z > 0$. Тогда равенство $\text{sign}(\Delta J_x) = \text{sign}(\bar{x})$ должно выполняться. Только один корень (со знаком плюс) удовлетворяет условию. Для $\Delta J_z < 0$ условию удовлетворяет корень со знаком минус.

Для получения смещения индексов нужно разделить полученное значение на Δx и округлить до ближайшего целого числа.

Смещение Δy можно найти аналогичным образом.

Модифицированный покомпонентный метод имеет вид:

$$\begin{aligned} z_i^{k+1} &= z_i^k - \psi \frac{A_j(z^k) - F_j}{\|\nabla A_j(z^k)\|^2} \left(\frac{\partial A_j(z^k)}{\partial z_i} \right), \\ j &= i + M \left[\frac{\left(-3\Delta J_z + \text{sign}(\Delta J_z) \sqrt{8\Delta J_y^2 + 9\Delta J_z^2} \right) \cdot h}{4\Delta J_y \Delta y} \right] + \\ &+ \left[\frac{\left(-3\Delta J_z + \text{sign}(\Delta J_z) \sqrt{8\Delta J_x^2 + 9\Delta J_z^2} \right) \cdot h}{4\Delta J_x \Delta x} \right], \end{aligned} \quad (2.3)$$

где Δx и Δy — шаги сетки, h — глубина залегания асимптотической плоскости, z_i — i -й компонент приближенного решения, $i = 1, \dots, n$; $k = 0, 1, 2, \dots$; $\psi > 0$ — демпфирующий множитель, $[\cdot]$ — операция округления до ближайшего целого вниз.

В случае, когда смещенные индексы выходят за границы сетки, используется продолжение путем копирования краевых значений.

Замечание. Покомпонентный метод является более экономичным по числу действий по сравнению с градиентными методами, но более чувствительным к возмущенным данным.

2.2 Построение модифицированного метода сопряженных градиентов

В данном разделе описывается построение модифицированного метода сопряженных градиентов. В работе [75] исследуется модифицированный метод наискорейшего спуска. Матрица производных интегрального оператора вычисляется один раз в начальной точке и не пересчитывается в дальнейшем процессе.

Метод сопряженных градиентов был предложен в работе [76] для решения корректных задач нелинейной оптимизации.

Рассмотрим систему нелинейных уравнений вида

$$A(z) = F,$$

где $A : R^n \Rightarrow R^m$ — конечномерная аппроксимация интегрального оператора первого рода, дифференцируемая по Фреше. Если решение существует, то задача его нахождения эквивалентна задаче минимизации

$$\min \left\{ \frac{1}{2} \|A(z) - F\|^2 : z \in R^n \right\}.$$

Обозначим градиент квадратичной невязки

$$\nabla \left[\frac{1}{2} \|A(z) - F\|^2 \right] = A'(z)^\top (A(z) - F) \equiv S(z),$$

где $A'(z)$ — матрица производных размерности $n \times m$.

В соответствии с работой [75], обозначим градиент регуляризованной невязки

$$S_\alpha^0(z) = A'(z^0)^\top(A(z) - F) + \alpha(z - z^0) = 0, \quad (2.4)$$

где z^0 — начальное приближение, $\alpha \geq 0$ — параметр регуляризации.

Построим процесс вида

$$z^{k+1} = z^k + \psi \chi^k p^k,$$

где

$$\begin{aligned} p^k &= S_\alpha^0(z^k) + \beta^k p^{k-1}, \quad k > 0, \\ p^0 &= S_\alpha^0(z^0), \\ \beta^k &= \max \left\{ \frac{\langle S_\alpha^0(z^k), (S_\alpha^0(z^k) - S_\alpha^0(z^{k-1})) \rangle}{\|S_\alpha^0(z^{k-1})\|^2}, 0 \right\}. \end{aligned}$$

Найдем выражение для приближенного размера шага χ^k , используя линейную аппроксимацию A .

$$\begin{aligned} \chi^k &\approx \operatorname{argmin}_\chi \left\{ \|A(z^k) + A'(z^0)(z^k + \chi p^k - z^k) - F\|^2 + \right. \\ &\quad \left. + \alpha \|z^k + \chi p^k - z^0\|^2 \right\} = \\ &= \operatorname{argmin}_\chi \left\{ \|(A(z^k) - F)\|^2 + 2\chi \langle A'(z^0)^\top(A(z^k) - F), p^k \rangle + \right. \\ &\quad \left. + \chi^2 \|A'(z^0)p^k\|^2 + \alpha \|z^k - z^0\|^2 + \right. \\ &\quad \left. + 2\alpha\chi \langle (z^k - z^0), p^k \rangle + \alpha\chi^2 \|p^k\|^2 \right\} = \\ &= \operatorname{argmin}_\chi \left\{ \|(A(z^k) - F)\|^2 + \alpha \|z^k - z^0\|^2 + \right. \\ &\quad \left. + \chi^2 \|A'(z^0)p^k\|^2 + \alpha\chi^2 \|p^k\|^2 + 2\chi \langle S_\alpha^0, p^k \rangle \right\}. \end{aligned}$$

Используя необходимое условие минимума,

$$\begin{aligned} \frac{\partial}{\partial \chi} \left(\| (A(z^k) - F) \|^2 + \alpha \| z^k - z^0 \|^2 + \chi^2 \| A'(z^0) p^k \|^2 + \right. \\ \left. + \alpha \chi^2 \| p^k \|^2 + 2\chi \langle S_\alpha^0, p^k \rangle \right) = 0; \\ 2\chi \| A'(z^0) p^k \|^2 + 2\alpha \chi \| p^k \|^2 + 2 \langle S_\alpha^0, p^k \rangle = 0, \end{aligned}$$

получаем

$$\chi^k \approx \tilde{\chi}^k = - \frac{\langle p^k, S_\alpha^0(z^k) \rangle}{\| A'(z^0) p^k \|^2 + \alpha \| p^k \|^2}.$$

Таким образом, модифицированный регуляризованный метод сопряженных градиентов принимает вид:

$$\begin{aligned} z^{k+1} &= z^k - \psi \frac{\langle p^k, S_\alpha^0(z^k) \rangle}{\| A'(z^0) p^k \|^2 + \alpha \| p^k \|^2} p^k, \\ p^k &= S_\alpha^0(z^k) + \beta^k p^{k-1}, \quad k > 0, \\ p^0 &= S_\alpha^0(z^0), \\ \beta^k &= \max \left\{ \frac{\langle S_\alpha^0(z^k), (S_\alpha^0(z^k) - S_\alpha^0(z^{k-1})) \rangle}{\| S_\alpha^0(z^{k-1}) \|^2}, 0 \right\}, \\ S_\alpha^0(z) &= A'(z^0)^\top (A(z) - F) + \alpha (z - z^0), \end{aligned} \tag{2.5}$$

где z^0 — начальное приближение, z^k — приближенное решение на k -ой итерации, $k = 0, 1, 2, \dots$; $\alpha \geq 0$ — параметр регуляризации, $\psi > 0$ — демпфирующий множитель.

2.3 Построение гибридного метода сопряженных градиентов

В разделе описывается построение гибридного метода сопряженных градиентов. Идея гибридного метода заключается в том, чтобы пересчитывать матрицу производных один раз за несколько итераций.

Построим гибридный регуляризованный вариант линейризованного метода сопряженных градиентов, используя регуляризацию по методу Тихонова, и зафиксируем матрицу производных на несколько итераций.

Гибридный регуляризованный метод сопряженных градиентов принимает вид:

$$\begin{aligned}
z^{k+1} &= z^k - \psi \frac{\langle p^k, S_\alpha(z^k) \rangle}{\|A'(z^k)p^k\|^2 + \alpha \|p^k\|^2} p^k, \\
p^k &= S_\alpha(z^k) + \beta^k p^{k-1}, \quad k > 0, \\
p^0 &= S_\alpha(z^0), \\
\beta^k &= \max \left\{ \frac{\langle S_\alpha(z^k), (S_\alpha(z^k) - S_\alpha(z^{k-1})) \rangle}{\|S_\alpha(z^{k-1})\|^2}, 0 \right\}, \\
S_\alpha(z) &= A'(z^k)^\top (A(z) - F) + \alpha(z - z^0) = 0, \\
A'(z^k) &= \begin{cases} A'(z^0), & 0 \leq k < j, \\ A'(z^j), & j \leq k < 2j, \\ A'(z^{2j}), & 2j \leq k < 3j, \\ \dots & \end{cases},
\end{aligned} \tag{2.6}$$

где z^0 — начальное приближение, z^k — приближенное решение на k -ой итерации, $k = 0, 1, 2, \dots$; $\alpha \geq 0$ — параметр регуляризации, $\psi > 0$ — демпфирующий множитель.

Данный алгоритм является компромиссным по вычислительным затратам между методом сопряженных градиентов (с пересчетом матрицы на каждой итерации) и модифицированным методом сопряженных градиентов (с вычислением матрицы один раз).

2.4 Построение модифицированного метода сопряженных градиентов с весовыми множителями (ММСГВМ) для многослойной среды

В разделе предлагается модифицированной методой сопряженных градиентов с весовыми множителями для многослойной среды. В работе [77] построен метод сопряженных градиентов с весовыми множителями для многослойной среды.

Предлагаемая автором модификация заключается в том, что производная интегрального оператора вычисляется один раз в фиксированной точке.

Построенный модифицированный метод сопряженных градиентов с весовыми множителями (ММСГВМ) для многослойной среды имеет вид:

$$\begin{aligned}
 z^{k+1} &= z^k - \psi \frac{\langle p^k, S_0(z^k) \rangle}{\|A'(z^0)p^k\|^2} p^k, \\
 p^k &= v^k + \beta^k p^{k-1}, \quad k > 0, \\
 p^0 &= v^0, \\
 v^k &= \Lambda S_0(z^k), \\
 \beta^k &= \max \left\{ \frac{\langle v^k, (v^k - v^{k-1}) \rangle}{\|v^{k-1}\|^2}, 0 \right\}, \\
 S_0(z) &= A'(z^0)^\top (A(z) - F),
 \end{aligned} \tag{2.7}$$

где z^0 — начальное приближение, z^k — приближенное решение на k -ой итерации, $k = 0, 1, 2, \dots$; $\psi > 0$ — демпфирующий множитель, Λ — диагональная матрица, состоящая из весовых множителей γ .

Весовые множители γ_i выбираются для каждой компоненты z_i из соответствующей компоненты f_i :

$$(f_1, f_2, \dots, f_n, f_{n+1}, \dots, f_{L \cdot n})^\top \rightarrow (\gamma_1, \gamma_2, \dots, \gamma_n, \gamma_{n+1}, \dots, \gamma_{L \cdot n})^\top.$$

Выделенные поля $F_l = (f_{(l-1)n+1}, f_{(l-1)n+2}, \dots, f_{(l-1)n+n})$ от поверхностей $l = 1, \dots, L$ находятся по методике повысотных трансформаций П. С. Мартышко и И. Л. Пруткина [73]).

В качестве начального приближения можно взять асимптотические плоскости.

2.5 Градиентные методы

В комплексе программ для решения обратных задач гравиметрии и магнитометрии для случая двухслойной среды также реализованы методы других авторов:

• регуляризованный линеаризованный метод сопряженных градиентов (РЛМСГ) [86; 87]

$$\begin{aligned}
 z^{k+1} &= z^k - \psi \frac{\langle p^k, S_\alpha(z^k) \rangle}{\|A'(z^k)p^k\|^2 + \alpha \|p^k\|^2} p^k, \\
 p^k &= S_\alpha(z^k) + \beta^k p^{k-1}, \quad k > 0, \\
 p^0 &= S_\alpha(z^0), \\
 \beta^k &= \max \left\{ 0, \frac{\langle S_\alpha(z^k), S_\alpha(z^k) - S_\alpha(z^{k-1}) \rangle}{\|S_\alpha(z^{k-1})\|^2} \right\}, \quad (2.8)
 \end{aligned}$$

$$S_\alpha(z) = A'(z)^T (A(z) - F) + \alpha (z - z^0),$$

где z^0 — начальное приближение, $k = 0, 1, 2, \dots$; z^k — приближенное решение на k -ой итерации, $\alpha \geq 0$ — параметр регуляризации, $\psi > 0$ — демпфирующий множитель.

• Модифицированные α -процессы [88]

— метод минимальной невязки (ММН)

$$z^{k+1} = z^k - \psi \frac{\langle (A'(z^0) + \bar{\alpha}I) \bar{S}_\alpha(z^k), \bar{S}_\alpha(z^k) \rangle}{\|(A'(z^0) + \bar{\alpha}I) \bar{S}_\alpha(z^k)\|^2} \bar{S}_\alpha(z^k); \quad (2.9)$$

— метод наискорейшего спуска (МНС)

$$z^{k+1} = z^k - \psi \frac{\|\bar{S}_\alpha(z^k)\|^2}{\langle (A'(z^0) + \bar{\alpha}I) \bar{S}_\alpha(z^k), \bar{S}_\alpha(z^k) \rangle} \bar{S}_\alpha(z^k); \quad (2.10)$$

— метод минимальной ошибки (ММО)

$$z^{k+1} = z^k - \psi \frac{\langle (A'(z^0) + \bar{\alpha}I)^{-1} \bar{S}_\alpha(z^k), \bar{S}_\alpha(z^k) \rangle}{\|\bar{S}_\alpha(z^k)\|^2} \bar{S}_\alpha(z^k), \quad (2.11)$$

где z^0 — начальное приближение, z^k — приближенное решение на k -ой итерации, $k = 0, 1, 2, \dots$; $\alpha \geq 0$ — параметр регуляризации, $\psi > 0$ — демпфирующий множитель. В качестве условия останова используется (2.2) при достаточно малом $\varepsilon > 0$.

Примечание. При реализации данных методов можно использовать ниже описанные оптимизации.

Для решения обратных задач гравиметрии и магнитометрии для случая многослойной среды используется:

• линеаризованный метод сопряженных градиентов с весовыми множителями (ЛМСГВМ) [86]

$$z^{k+1} = z^k - \psi \frac{\langle p^k, S_\alpha(z^k) \rangle}{\|A'(z^k)p^k\|^2 + \alpha\|p^k\|^2} p^k,$$

$$p^k = v^k + \beta^k p^{k-1}, \quad k > 0,$$

$$p^0 = v^0,$$

$$v^k = \Lambda S_\alpha(z^k),$$

$$\beta^k = \max \left\{ 0, \frac{\langle v^k, v^k - v^{k-1} \rangle}{\|v^{k-1}\|^2} \right\},$$

$$S_\alpha(z) = A'(z)^\top (A(z) - F),$$

где z^0 — начальное приближение, z^k — приближенное решение на k -ой итерации, $k = 0, 1, 2, \dots$; $\alpha \geq 0$ — параметр регуляризации, $\psi > 0$ — демпфирующий множитель, Λ — диагональная матрица, состоящая из весовых множителей. В качестве условия останова используется (2.2) при достаточно малом $\varepsilon > 0$.

2.6 Оптимизации по памяти для модифицированных методов

Заметим, что при использовании модифицированных градиентных методов, основанных на вычислении матрицы производных в фиксированной точке, при выборе начального приближения $z^0 = \text{const}$ матрица является теплицево-блочной-теплицевой (ТБТ).

Для задачи гравиметрии элементы матрицы принимают вид

$$a_{i,j,u,v} = a_{(j-1)M+i, (v-1)M+u} = f \Delta \sigma \Delta x \Delta y \frac{-z^0}{((x_u - x_i)^2 + (y_v - y_j)^2 + (z^0)^2)^{3/2}}.$$

Для задачи магнитометрии элементы матрицы принимают вид

$$\begin{aligned} a_{i,j,u,v} &= a_{(j-1)M+i, (v-1)M+u} = \\ &= -\frac{1}{4\pi} \Delta x \Delta y \frac{J_z \left((x_u - x_i)^2 + (y_v - y_j)^2 - 2(z^0)^2 \right) + 3z^0 \left(J_x((x_u - x_i) + J_y(y_v - y_j)) \right)}{\left((x_u - x_i)^2 + (y_v - y_j)^2 + (z^0)^2 \right)^{5/2}}. \end{aligned}$$

Справедливо следующее утверждение.

Утверждение 1. При смещении пары индексов блока (i, u) или пары индексов элементов внутри блока (j, v) на одну и ту же величину значения элементов не меняются

$$\forall r : ((u + r) \in [1, M]) \wedge ((i + r) \in [1, M]) \Rightarrow a_{i, j, u, v} = a_{i+r, j, u+r, v},$$

$$\forall t : ((v + t) \in [1, N]) \wedge ((j + t) \in [1, N]) \Rightarrow a_{i, j, u, v} = a_{i, j+t, u, v+t}.$$

Другими словами, значения элементов, для которых $(i - u) = \text{const}$ или $(j - v) = \text{const}$, совпадают. На диагоналях, параллельных главной, стоят одинаковые блоки, а внутри каждого блока каждая диагональ состоит из одинаковых элементов (рис. 2.3), т.е. матрица является теплицево-блочной-теплицевой.

Ранее ТБТ-структура матрицы использовалась в работе [78] при решении линейной задачи гравиметрии о нахождении переменной плотности в слое.

На рис. 2.2 изображена структура матрицы производных интегрального оператора для случая двухслойной среды, где $i = 1, \dots, M; j = 1, \dots, N$ — индексы блока, $u = 1, \dots, M; v = 1, \dots, N$ — индексы элемента внутри блока. Блоки с одинаковыми индексами совпадают, элементы внутри блока с одинаковыми индексами имеют одно и то же значение. На рис. 2.3 изображена структура матрицы производных интегрального оператора для случая многослойной среды. Она состоит из L больших блоков, где L — число границ раздела. Каждый такой блок имеет блочно-теплицевую структуру. Для хранения такой матрицы достаточно хранить одну строку (рис. 2.4), для вычисления любого элемента в матрице достаточно применить соответствующий сдвиг относительно сохраненной строки.

Утверждение 2. Пространственная сложность алгоритмов, использующих ТБТ-матрицу, составляет $O(n)$, где $n = M \times N$.

Алгоритмы с полной матрицей имеют сложность $O(n^2)$.

Доказательство (по построению): пусть нам нужно получить элемент матрицы производных с индексами $(j + i \cdot M \cdot N, v + u \cdot M \cdot N)$, соответствующий точке интегрирования (x_u, y_v) и точке наблюдения (x'_i, y'_j) . Индекс элемента в хранимой строке определяется путем сложения нескольких компонент.

1. В v -й строке внутри блока нужно взять элемент с индексом j .
2. v -я строка внутри блока получается смещением малого окна ширины $2M - 1$ элементов (число уникальных элементов в блоке) вправо на $M - v - 1$ элементов.
3. i -ый блок в u -й блочной строке начинается с $(2M - 1) \cdot i$ -го элемента.

4. u -я блочная строка получается смещением большого окна ширины $(2M - 1)N$ вправо на $(2M - 1)(N - u - 1)$ элементов.

Такой подход позволяет сократить время счета и затраты на оперативную память. Вместо вычисления и хранения полной матрицы размерности $MN \times MN$ достаточно хранить только $(2M - 1)(2N - 1)$ уникальных элементов.

Для расчетной сетки 512×512 и формата с двойной точностью матрица коэффициентов займёт 512 ГБ. Отметим, что элементы матрицы производной оператора вычисляются «на лету» в момент выполнения векторно-матричных операций. При использовании модифицированных методов хранения теплицево-блочной матрицы для той же сетки требует менее 8 МБ. Использование требуемой памяти уменьшается в 65536 раз.

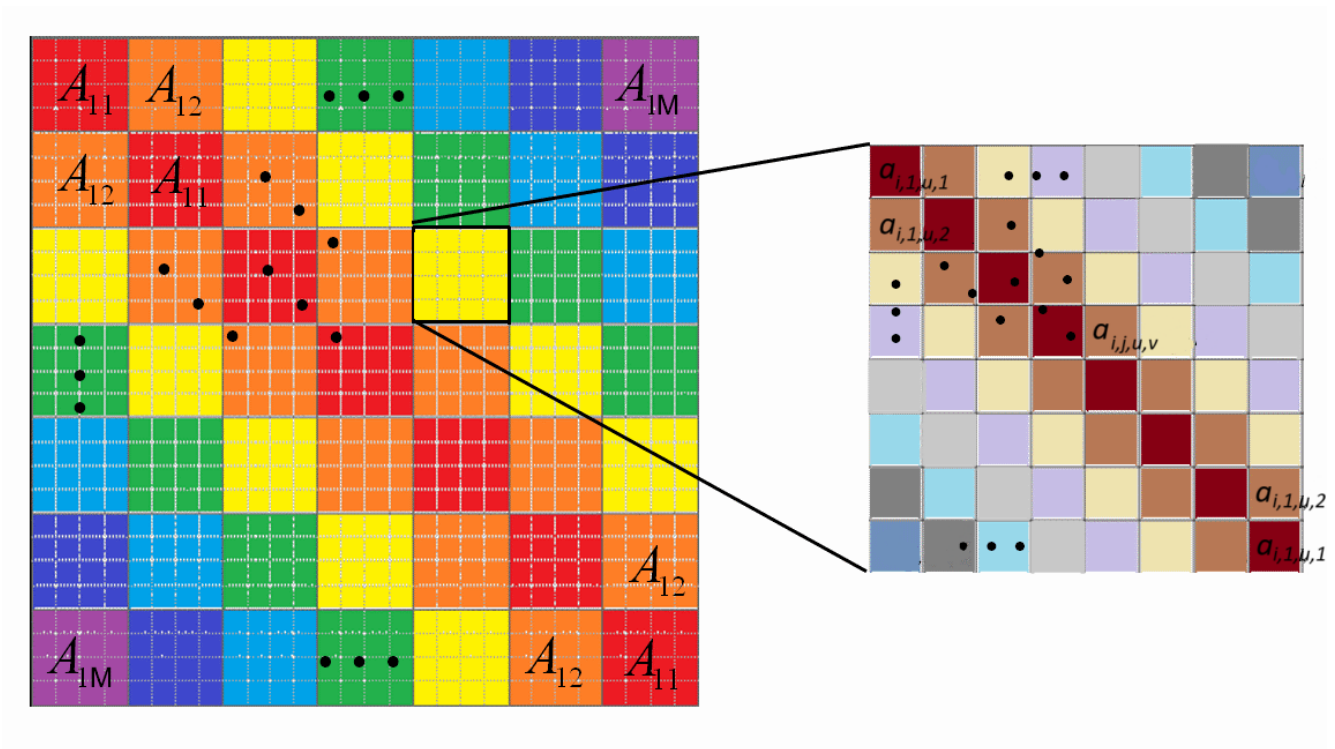


Рисунок 2.2 — Структура матрицы производных интегрального оператора для случая двухслойной среды

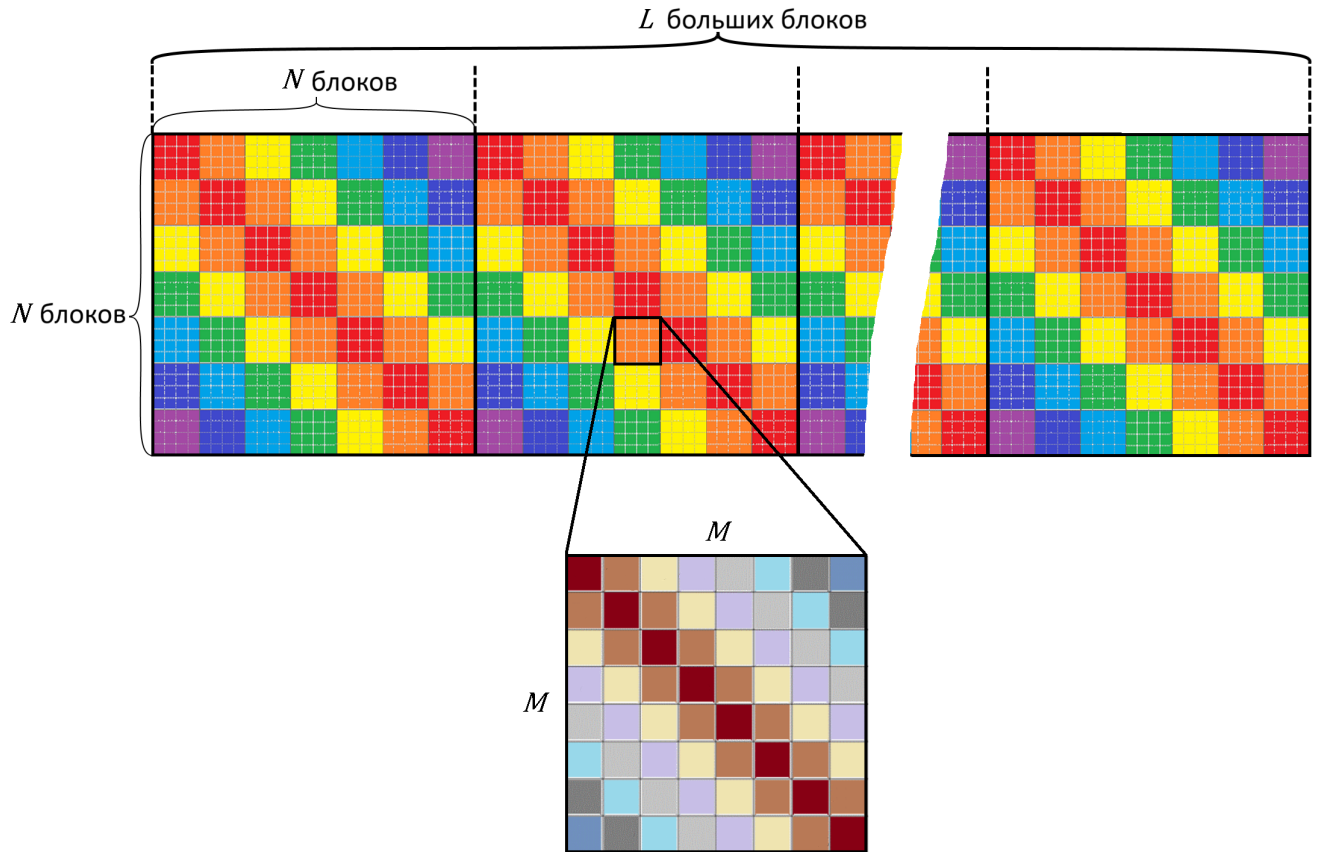


Рисунок 2.3 — Структура матрицы производных интегрального оператора для случая многослойной среды

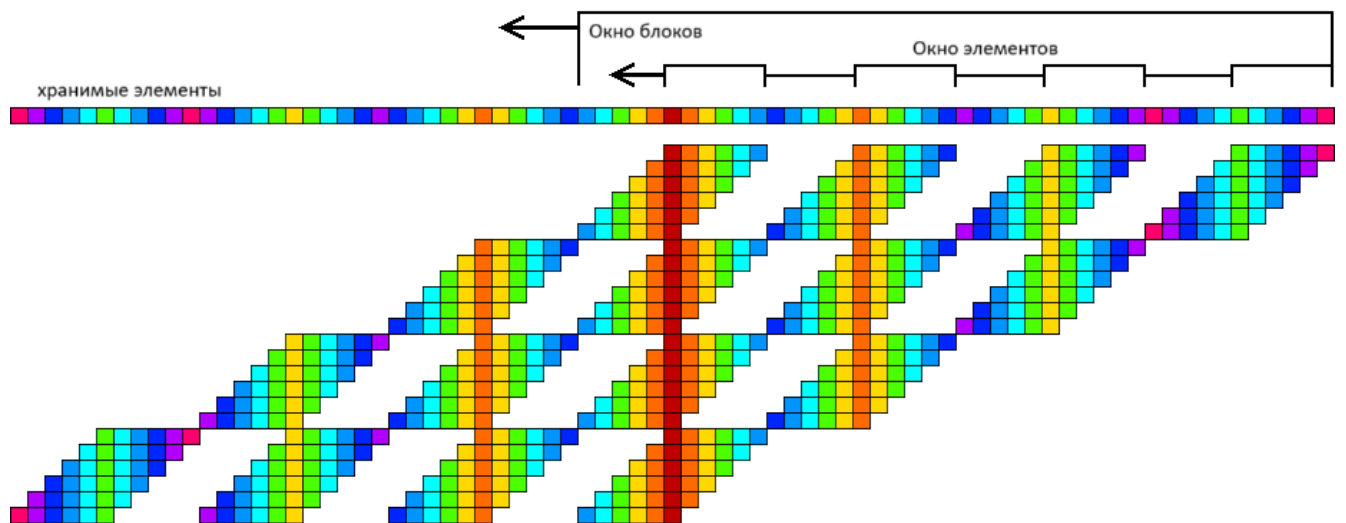


Рисунок 2.4 — Способ хранения блочно-теплицевой матрицы

Глава 3. Комплекс программ и вычислительные эксперименты

Третья глава посвящена разработке комплекса параллельных программ решения структурных обратных задач гравиметрии и магнитометрии на графических ускорителях и тестированию комплекса на построенных модельных примерах с использованием реальных гравитационных полей. Разработан веб-портал для удаленных вычислений, позволяющий запускать комплекс программ на многопроцессорных системах. Комплекс программ интегрирован в веб-портал.

В разделе 3.1 приводится обзор технологий, используемых в комплексе программ. В том числе, OpenMP, MPI, CUDA. Раздел 3.2 посвящен построению параллельных алгоритмов для графических процессоров. В разделе 3.3 приводится описание универсального веб-портала. В разделе 3.4 приведено описание нагрузочного тестирования веб-портала. Раздел 3.5 описывает численные эксперименты для решения модельных задач с помощью разработанных методов и параллельных программ.

3.1 Обзор вычислительных технологий

Для эффективного решения задач, алгоритмы которых хорошо распараллеливаются на несколько подзадач, существует множество технологий и библиотек.

В случае вычислений на центральном многоядерном процессоре одним из стандартов является OpenMP Open Multi-Processing. Для стандарта существуют реализации в виде библиотек на языках C, C++ и фортран. Разработка спецификаций стандарта ведётся некоммерческой организацией OpenMP Architecture Review Board (ARB).

Для многопроцессорных систем используется технология MPI (Message Passing Interface). В данной технологии для синхронизации данных используется сеть. Это позволяет производить вычисления на процессорах, находящихся на физически различных вычислительных узлах. Но также это оказывает негативное влияние на эффективность вычислений, так как сетевые операции

занимают достаточно много времени, относительно вычислительных операций и при необходимости частой синхронизации вычислительных потоков происходит значительное снижение производительности алгоритмов.

Также для вычислений можно использовать графические ускорители. Техника, при которой обобщенные вычисления (запускаемые на центральном процессоре) запускаются на видеоускорителях, называется GPGPU (General-purpose computing on graphics processing units). Для вычислений на графических ускорителях также существует несколько общепринятых технологий.

Еще одной технологией для вычислений на графических ускорителях является CUDA (Compute Unified Device Architecture). В отличие от предыдущей технологии OpenCL, CUDA — программно-аппаратная архитектура, которая может использоваться исключительно на графических процессорах Nvidia. Существует реализация в виде библиотек на языках C, C++ и Фортран. Для программирования с использованием CUDA используется синтаксис, похожий на язык программирования C. Также важной особенностью является то, что для данной технологии разработаны высокоэффективные библиотеки, в том числе для векторно-матричных операций (Cublas). Данное обстоятельство позволяет экономить на разработке программного обеспечения и при этом писать высокоэффективный код. При использовании CUDA можно достаточно удобно писать код, который будет исполняться на нескольких графических ускорителях, установленных в одном вычислительном узле. Для этого не требуются дополнительные инструменты и библиотеки.

3.2 Описание комплекса программ

На основе вышеописанных методов для решения структурных обратных задач гравиметрии и магнитометрии разработаны алгоритмы для графических ускорителей, входящие в состав комплекса программ. Алгоритмы встроены в систему удаленных вычислений. Для параллельной реализации алгоритмов выбрана технология CUDA по нескольким причинам:

- на суперкомпьютере УРАН, на котором проводились эксперименты установлены графические ускорители Nvidia Tesla

- для технологии CUDA реализована библиотека для эффективных векторно-матричных операций Cublas
- задачи хорошо ложатся на архитектуру графических ускорителей

В задачах используются сетки больших размерностей — это позволяет загрузить все вычислительные ядра видеоускорителя. Также важен тот факт, что вычисления в узлах сетки независимы друг от друга и синхронизация потоков требуется только при переходе на следующую итерацию. Это позволяет эффективно использовать как ядра графических ускорителей, так и несколько видеоускорителей одновременно. Использование нескольких видеоускорителей позволяет решить не только проблему производительности, но и снизить нагрузку на оперативную память одного ускорителя. Таким образом использование нескольких графических ускорителей позволяет увеличить размерность сетки и точность решения.

Описание входных и выходных данных, а также рекомендации по использованию также загружены на веб-портал. Для запуска задачи необходимо загрузить файл с аномальным полем, указать глубину и количество графических ускорителей. После окончания работы программы на выходе будет файл с восстановленной поверхностью и ее изображение.

Комплекс программ включает реализации следующих методов на графических ускорителях (рис. 3.1):

- Задача гравиметрии для одной границы:
 - регуляризованный линейризованный МСГ (2.8)
 - модифицированный регуляризованный линейризованный МСГ (2.5)
 - гибридный МСГ (2.6) [87]
 - линейризованный МНС (2.6) [87]
 - покомпонентный градиентный метод (2.1)
 - модифицированные альфа процессы (2.9)(2.10)(2.11)
- Задача гравиметрии для нескольких границ:
 - линейризованный МСГ с весовыми множителями (2.12)
 - модифицированный МСГ с весовыми множителями (2.7)
- Задача магнитометрии для одной границы:
 - регуляризованный линейризованный МСГ (2.8)
 - покомпонентный градиентный метод (2.1)

- модифицированный покомпонентный градиентный метод (2.3) [87]
- линеаризованный МНС [88]
- модифицированный МНС [88]
- Задача магнитометрии для нескольких границ:
 - линеаризованный МСГ с весовыми множителями (2.12)

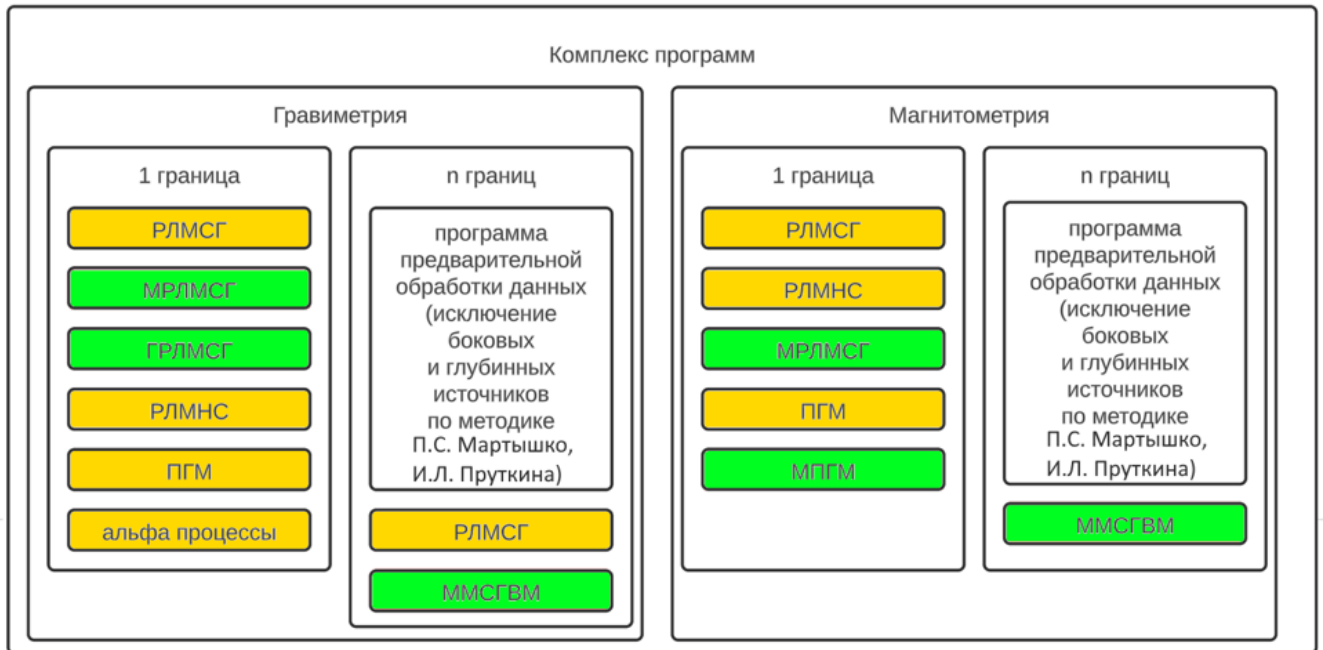


Рисунок 3.1 — Структура комплекса программ

Рекомендации по использованию методов:

- для зашумленных данных в задачах гравиметрии и магнитометрии лучше подходят модифицированные α -процессы
- для модельных задач и слабозашумленных данных рекомендуется покомпонентный метод
- для задач с несколькими границами необходимо использовать МСГ с весовыми множителями.

Алгоритмы решения обратной задачи гравиметрии на основе градиентных методов численно реализованы на суперкомпьютере кластерного типа «Уран». Алгоритмы реализованы на видеоускорителях NVIDIA Tesla с помощью технологии CUDA и библиотеки CUBLAS.

Комплекс программ предусматривает запуск алгоритмов на вычислительных узлах с несколькими установленными графическими ускорителями NVIDIA, поддерживающими технологию CUDA. Рассмотрим основную структуру алгоритмов.

- Первый шаг является подготовительным: на всех графических ускорителях выделяется память под вектора и матрицы, копируется вектор правой части уравнения.
- На следующем шаге на каждом устройстве запускается набор ядерных функций. Данные функции реализованы таким образом, чтобы минимизировать необходимость синхронизации данных между различными ускорителями. Этот шаг повторяется до тех пор, пока не будет достигнута необходимая точность, либо не закончится время, отведенное на расчет задачи.
- На последнем шаге производится выгрузка результатов с графического ускорителя на центральный процессор. Данные сохраняются на диск, оперативная память видеокарты и хоста освобождается. Работа алгоритма завершается.

Например, в алгоритме, реализующем РЛМСГ, имеется 3 ядерных функции, которые выполняются в рамках каждой итерации:

1. На данном этапе вычисляются элементы матрицы A' , компоненты векторов S_α , p^k , β^k . После завершения функции CalculateStep1 на центральном процессоре собирается коэффициент β^k и передается на все графические ускорители.

2. Затем вычисляется числитель и знаменатель коэффициента шага χ^k . После завершения функции CalculateStep2 на центральном процессоре производится деление числителя на знаменатель.

3. Полученные вычисления используются для окончательного расчета вектора z^{k+1} в функции CalculateZ.

Предполагается, что ускорители установлены на 1 вычислительном узле — это позволяет минимизировать сетевые расходы. Центральный процессор в такой схеме занимается диспетчеризацией видеокарт. Для сетки 512×512 хранение матрицы производных требует 512 ГБ. Поэтому элементы матрицы производной оператора вычисляются «на лету» во время выполнения векторно-матричных операций. Используется способ автоматической настройки параметров выполнения подпрограмм на GPU: найден оптимальный размер блока для эталонного размера сетки 256×256 , для другого размера сетки параметры пропорционально пересчитываются.

Для запуска комплекса программ необходимо выполнить команду и передать необходимые аргументы:

./CudaGradient gpu-count method-name method-params input-data,

где

- CudaGradient — название исполняемого файла (бинарный код программы),
- gpu-count — количество используемых графических ускорителей (не должно превышать количество установленных ускорителей на узле),
- method-name — название запускаемого метода,
- method-params — параметры метода (для разных методов набор параметров может отличаться)
- input-data — пути до файлов с входными данными

В качестве входных данных принимаются файлы формата *.dat. Данный формат представляет собой описание значений сеточной функции поля в виде:

X Y <значение в данной координате>

Выходные файлы также формируются в формате *.dat.

Во время работы комплекса в консоль будет выводиться отладочная информация. По результатам работы будет выведена сводная информация по методу и сформированы выходные файлы.

Описание методов и их параметров. В текущей реализации в качестве аргументов комплекса программ могут использоваться следующие комбинации методов и параметров:

1. Регуляризованный линеаризованный МСГ для решения задачи гравиметрии.
 method-name: gravity_msg.
 method-params:
 - h — глубина залегания;
 - alpha — параметр регуляризации;
 - psi — демпфирующий множитель;
 - eps — условие останова.
2. Модифицированный регуляризованный линеаризованный МСГ для решения задачи гравиметрии.
 method-name: gravity_msg_modified.
 method-params:
 - h — глубина залегания;
 - alpha — параметр регуляризации;
 - psi — демпфирующий множитель;

- eps — условие останова.
3. Гибридный МСГ для решения задачи гравиметрии.
method-name: gravity_msg_hybrid.
method-params:
 - h — глубина залегания;
 - alpha — параметр регуляризации;
 - psi — демпфирующий множитель;
 - eps — условие останова.
 4. Линеаризованный МНС для решения задачи гравиметрии.
method-name: gravity_mns.
method-params:
 - h — глубина залегания;
 - alpha — параметр регуляризации;
 - psi — демпфирующий множитель;
 - eps — условие останова.
 5. Покомпонентный градиентный метод для решения задачи гравиметрии.
method-name: gravity_pgm.
method-params:
 - h — глубина залегания;
 - alpha — параметр регуляризации;
 - psi — демпфирующий множитель;
 - eps — условие останова.
 6. Модифицированные альфа процессы для решения задачи гравиметрии.
method-name: gravity_alpha_process.
method-params:
 - process — {mns, msg, mmo}, где mns — метод наискорейшего спуска, msg — метод сопряженных градиентов, mmo — метод минимальной ошибки;
 - h — глубина залегания;
 - alpha — параметр регуляризации;
 - psi — демпфирующий множитель;
 - eps — условие останова.
 7. линеаризованный МСГ с весовыми множителями для решения задачи гравиметрии с несколькими границами.

method-name: gravity_msg_multilayer.

method-params:

- layers — количество слоев;
- h — глубина залегания;
- alpha — параметр регуляризации;
- psi — демпфирующий множитель;
- weights — весовые множители;
- eps — условие останова.

8. Модифицированный МСГ с весовыми множителями для решения задачи гравиметрии с несколькими границами.

method-name: gravity_msg_modified_multilayer.

method-params:

- layers — количество слоев;
- h — глубина залегания;
- alpha — параметр регуляризации;
- psi — демпфирующий множитель;
- weights — весовые множители;
- eps — условие останова.

9. Регуляризованный линейризованный МСГ с весовыми множителями для решения задачи магнитометрии.

method-name: gravity_msg.

method-params:

- h — глубина залегания;
- alpha — параметр регуляризации;
- psi — демпфирующий множитель;
- eps — условие останова.

10. Покомпонентный градиентный метод с весовыми множителями для решения задачи магнитометрии.

method-name: magn_pgm.

method-params:

- h — глубина залегания;
- alpha — параметр регуляризации;
- psi — демпфирующий множитель;
- eps — условие останова.

11. Модифицированный покомпонентный градиентный метод с весовыми множителями для решения задачи магнитометрии.

method-name: magn_pgm_modified.

method-params:

- h — глубина залегания;
- alpha — параметр регуляризации;
- psi — демпфирующий множитель;
- dJx — значение скачка намагниченности по x на искомой поверхности;
- dJy — значение скачка намагниченности по y на искомой поверхности;
- dJz — значение скачка намагниченности по z на искомой поверхности;
- eps — условие останова.

12. Линеаризованный МНС с весовыми множителями для решения задачи магнитометрии.

method-name: magn_mns.

method-params:

- h — глубина залегания;
- alpha — параметр регуляризации;
- psi — демпфирующий множитель;
- eps — условие останова.

13. Модифицированный МНС с весовыми множителями для решения задачи магнитометрии.

method-name: magn_mns_modified.

method-params:

- h — глубина залегания;
- alpha — параметр регуляризации;
- psi — демпфирующий множитель;
- eps — условие останова.

14. Линеаризованный МСГ с весовыми множителями с весовыми множителями для решения задачи магнитометрии с несколькими границами.

method-name: magn_msg_multilayer.

method-params:

- layers — количество слоев;

- h — глубина залегания;
- α — параметр регуляризации;
- ψ — демпфирующий множитель;
- weights — весовые множители;
- eps — условие останова.

3.3 Описание веб-портала

В данной работе предлагается веб-портал, в котором учтены недостатки работ [60–62] и используются удачные идеи из работ [63; 65]. Из работы [63] используется идея общения с кластером с использованием личных учетных данных пользователя, с учетом этого появляется возможность работы с произвольным кластером. Идея о возможности самостоятельной загрузки алгоритмов пользователями взята из [65].

Предлагаемый веб-портал является универсальным решением для запуска задач на удаленных кластерах. При подключении удаленного кластера не требуется производить установку дополнительного ПО на самом кластере, также нет необходимости в привилегированной учетной записи. Общение с кластером ведется через ssh-туннель. Пользователю, для работы с сервисом, требуется действующая учетная запись. Взаимодействие с центром обработки данных (ЦОД) берет на себя портал, пользователям нет необходимости в прямом подключении к кластеру.

Преимуществами веб-портала являются:

- Нет привязки к конкретному кластеру и его ПО.
- Нет ограничений на список алгоритмов/задач. Пользователи сами могут загружать алгоритмы, делать их публичными на всех/определенную группу людей
- Гибкая настройка пред- и постобработки задач.

Важной особенностью является то, что работа по планированию (учет времени и приоритизация) пользовательских задач остается на стороне кластера, что позволяет сильно упростить логику взаимодействия с удаленными узлами и не конфликтовать с пользователями, использующими кластер напрямую.

Входные данные и результаты работы хранятся непосредственно на кластере — это позволяет уменьшить требования к вычислительным ресурсам и объему хранилища. В базе данных хранятся учетные данные пользователей, авторизационная информация и список задач. Система управления базой данных (СУБД) может быть установлена вместе с основным веб-приложением на одних и тех же серверах, т.к. объем хранимых данных невелик. При необходимости СУБД может быть вынесена на отдельные серверы.

Описание входных и выходных данных, а также рекомендации по использованию также загружены на веб-портал. Для запуска задачи необходимо загрузить файл с аномальным полем, указать глубину и количество графических ускорителей. После окончания работы программы на выходе будет файл с восстановленной поверхностью и ее изображение.

Портал может масштабироваться при росте количества пользователей и задач за счет реплицирования основного приложения и шардирования БД по пользователям. Так же за счет реплицирования повышается надежность и отказоустойчивость.

Веб-портал имеет следующую структуру (рис. 3.2):

- Веб-интерфейс — написан с использованием технологии blazor [79] (реализация web assembler под .net core). Blazor — перспективная технология, позволяющая писать код для браузера на языке C#. Впоследствии код компилируется веб-ассемблером в двоичный код для последующего исполнения браузером. Данный подход позволяет писать достаточно эффективный код, т.к. не происходит преобразований в javascript-код и последующей интерпретации. Общение с серверной частью происходит по протоколу http. Аутентификация происходит по протоколу OpenID Connect. С порталом поставляется реализованный модуль аутентификации. Также имеется возможность в конфигурации указать внешний сервис аутентификации, совместимый с протоколом OAuth2 или OpenID Connect. Это позволит не дублировать учетные записи пользователей при наличии собственного сервиса аутентификации.
- Серверная часть — использует ASP.NET core [80] фреймворк и язык C#. В пользу выбора данной платформы были следующие аргументы: простота синтаксиса, большое количество реализованных функций в стандартной поставке и возможность написания кода, способного запус-

каться на различных платформах. Для взаимодействия с базой данных используется EF core [81]. Данный фреймворк позволяет абстрагироваться от конкретной СУБД и взаимодействовать с СУБД посредством LINQ запросов. Таким образом можно использовать любую СУБД, для которой реализован драйвер под EF core.

- СУБД — может использоваться любая реляционная СУБД, для которой реализован драйвер под EF core. В рамках данной работы в качестве СУБД используется PostgreSQL. Данная СУБД имеет широкое распространение, хорошую документация, а также открытый исходный код. В рамках проведенных экспериментов показала достаточную производительность для базовых сценариев, используемых в работе.



Рисунок 3.2 — Структура веб-портала

На рис. 3.3 представлена структура БД. Таблицы с префиксом `AspNet` созданы фреймворком `ASP.NET` для корректной работы авторизации. В таблице `ssh_user_settings` хранится информация об учетных данных пользователей, необходимых для подключения к кластеру. В таблице `programs` содержится информация о загруженных алгоритмах и скриптах для предобработки входных данных и постобработки результатов вычислений. Таблица `tasks` хранит информацию о запущенных и завершенных заданиях на расчеты.

Логические модули:

- Раздел администрирования. Здесь ведется учет пользователей и групп. В данном разделе можно пригласить новых пользователей и настроить уровень доступа к необходимым программам.
- Программы (рис. 3.4). В данном разделе находится список доступных программ, производится загрузка новых программ, редактирования существующих и запуск. Так же здесь настраиваются параметры запуска задачи по умолчанию. По кнопке создания новой программы (`Create`

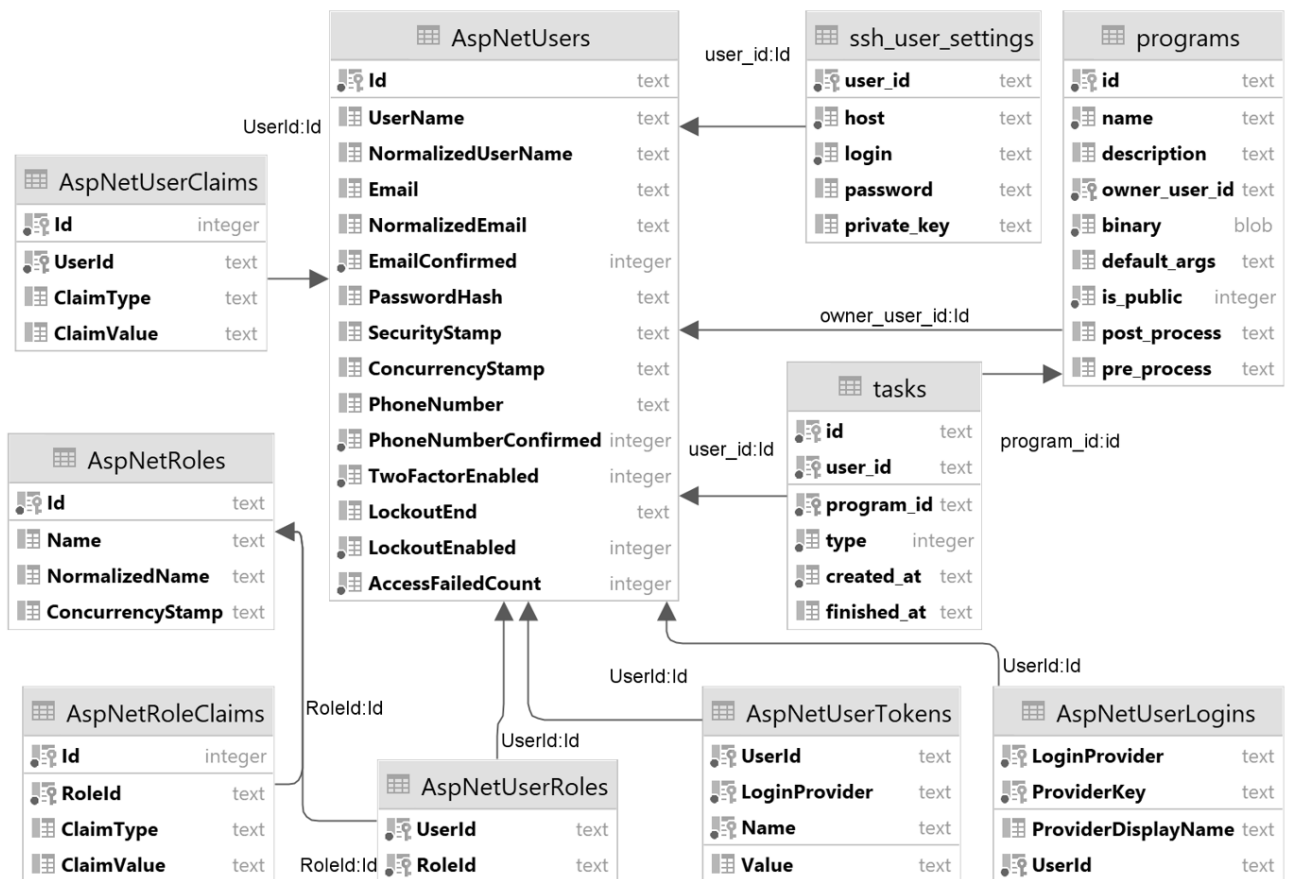
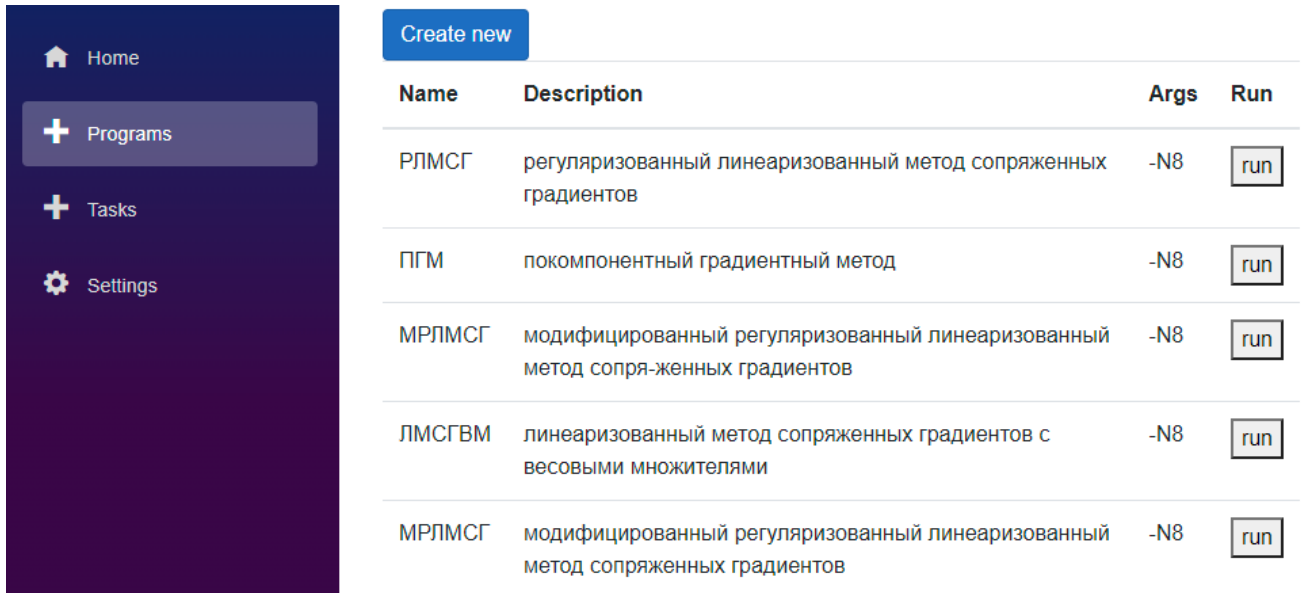


Рисунок 3.3 — Структура базы данных

new) открывается форма создания новой программы (рис. 3.5). На данной форме заполняется краткое название программы, подробное описание, значения аргументов по умолчанию и сам файл. По кнопке запустить откроется форма-подтверждение, где можно изменить параметры запуска программы. После чего программа помещается в очередь задач на выполнение.

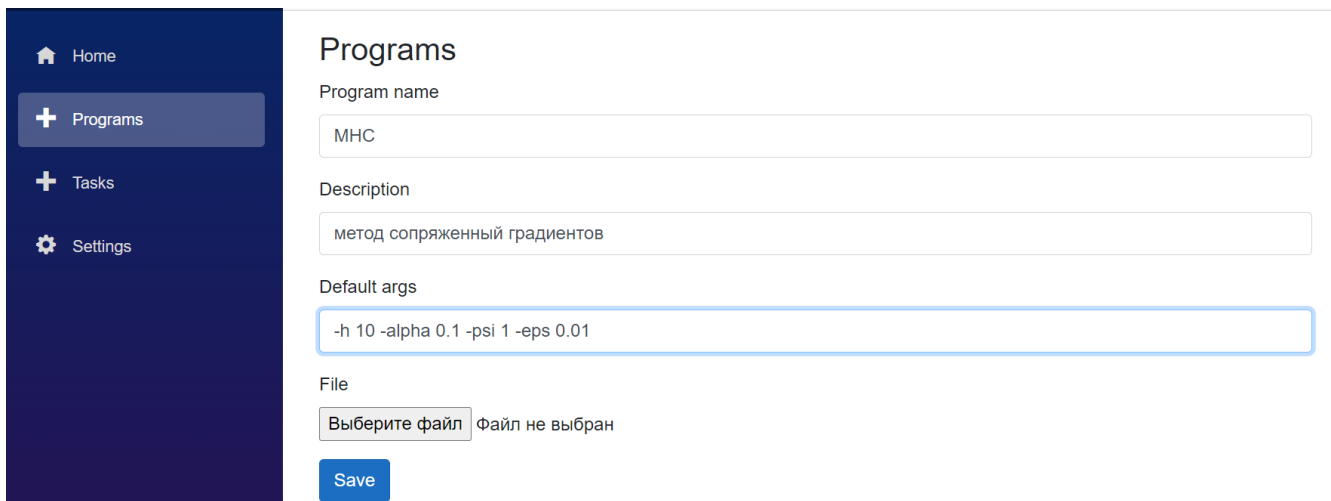
- Задания (рис. 3.6). В данном разделе находится список запущенных задач. Для исполняющихся задач можно посмотреть время запуска и статус. Для готовых есть возможность скачать архив с выходными данными.
- Настройки (рис. 3.7). Здесь описываются настройки для подключения к вычислительному кластеру: dns-имя или ip-адрес кластера, ssh-ключ для подключения (также есть возможность воспользоваться логином и паролем). Использование ssh-ключей позволяют сделать взаимодействие с кластером более безопасным, т.к. не требуется передачи и хранения логина и пароля, а также появляется возможность отзыва

скомпрометированных ключей. Для проверки корректности сохраненных настроек, имеется возможность отправить на кластер тестовую команду. В случае успеха, на странице отобразится результат выполнения команды. В случае неудачи отобразится ошибка, описывающая место и параметры возникновения.



Name	Description	Args	Run
РЛМСГ	регуляризованный линейаризованный метод сопряженных градиентов	-N8	<input type="button" value="run"/>
ПГМ	покомпонентный градиентный метод	-N8	<input type="button" value="run"/>
МРЛМСГ	модифицированный регуляризованный линейаризованный метод сопряженных градиентов	-N8	<input type="button" value="run"/>
ЛМСГВМ	линейаризованный метод сопряженных градиентов с весовыми множителями	-N8	<input type="button" value="run"/>
МРЛМСГ	модифицированный регуляризованный линейаризованный метод сопряженных градиентов	-N8	<input type="button" value="run"/>

Рисунок 3.4 — Список загруженных программ



Programs

Program name

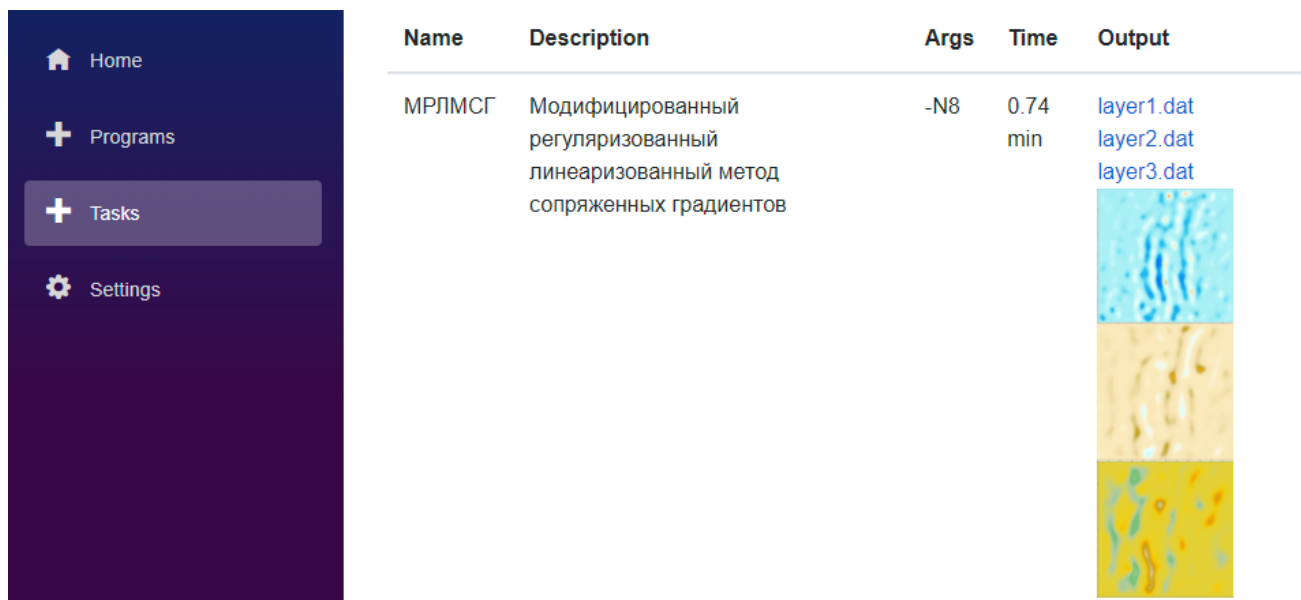
Description

Default args

File
 Файл не выбран

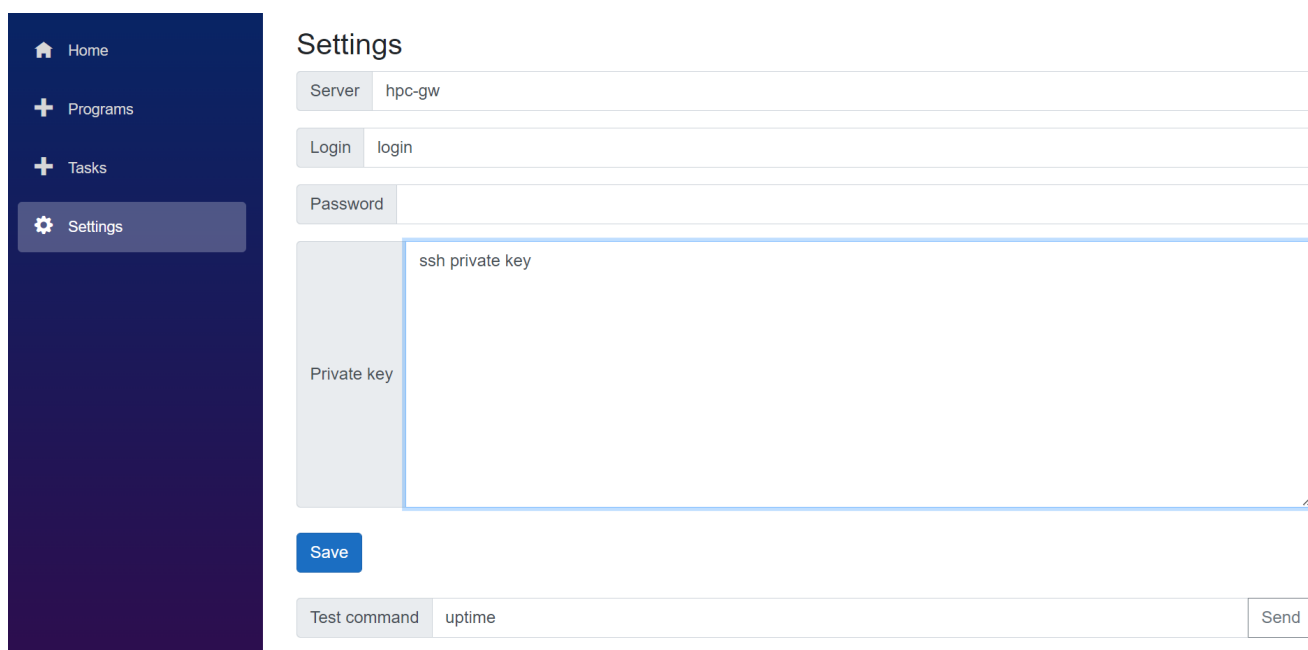
Рисунок 3.5 — Форма создания новой программы

На рис. 3.8 представлена схема взаимодействия пользователя с веб-порталом. При первом входе на веб-портал пользователю необходимо пройти регистрацию и указать учетные данные для подключения к вычислительному кластеру. Далее он может создать и настроить алгоритмы решения необходимых ему задач (загруженные программы могут использоваться в личных целях, либо могут быть доступны всем пользователям). Загружать можно как



Name	Description	Args	Time	Output
МРЛМСГ	Модифицированный регуляризованный линеаризованный метод сопряженных градиентов	-N8	0.74 min	layer1.dat layer2.dat layer3.dat

Рисунок 3.6 — Список запущенных заданий



Settings

Server hpc-gw

Login login

Password

Private key ssh private key

Save

Test command uptime Send

Рисунок 3.7 — Страница настроек веб-портала

готовый исполняемый файл, так и исходный код (при наличии соответствующего компилятора или интерпретатора). После загрузки формируется шаблон для запуска задачи, в котором описываются параметры, необходимы для запуска и требуемые шаги для предобработки входных данных и постобработки выходных файлов. После этого пользователь выбирает алгоритм, загружает входные данные и ставит задание в очередь на выполнение. После завершения выполнения задания происходит постобработка выходных данных и появляется возможность скачать результаты работы.

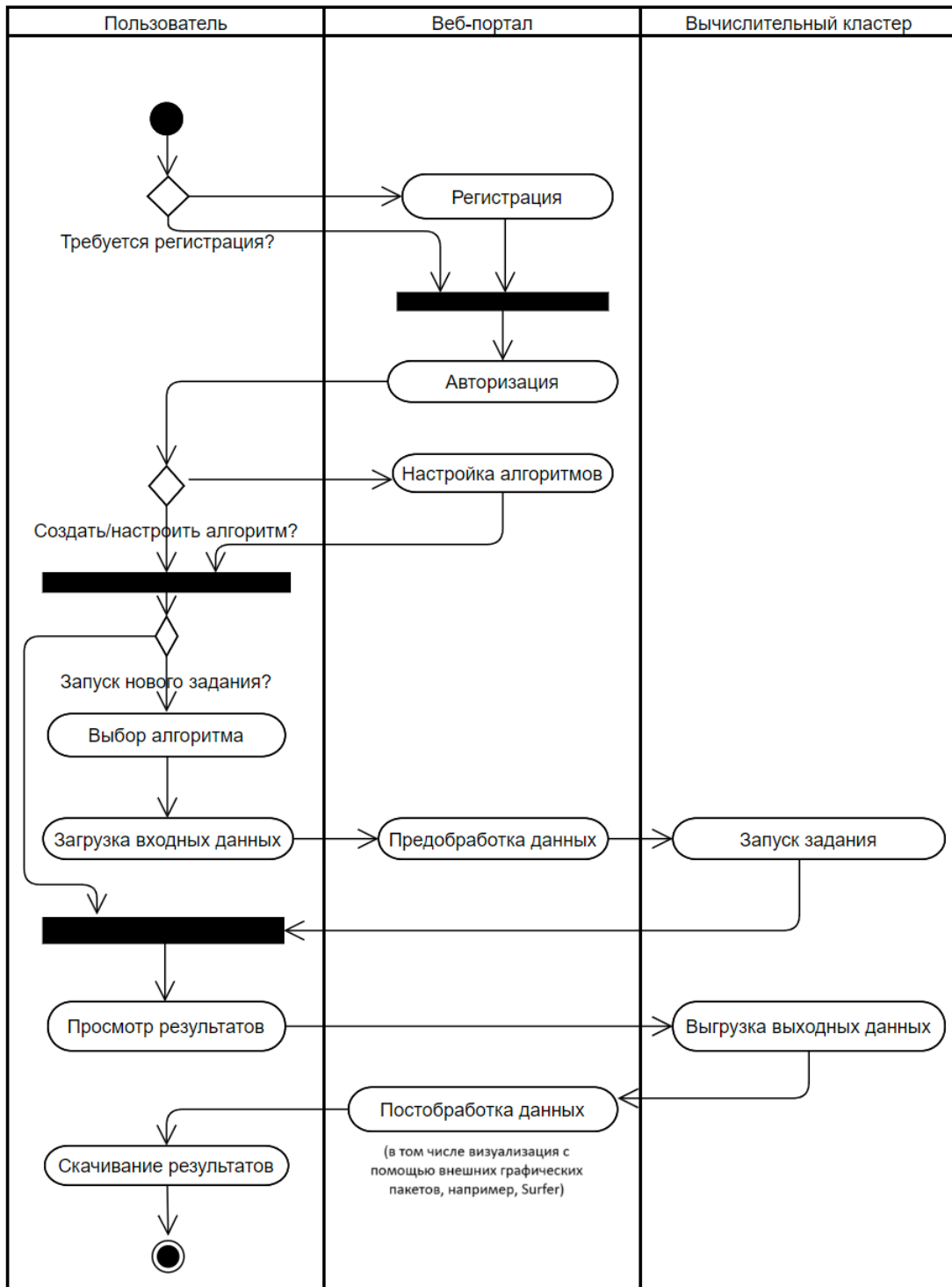


Рисунок 3.8 — Схема взаимодействия пользователя с веб-порталом

Предварительно на портал загружен набор алгоритмов для решения обратных задач гравиметрии и магнитометрии. Запуск этих задач возможен на многопроцессорных вычислительных узлах, либо на узлах, оборудованных графическими ускорителями.

При запуске задачи создается соответствующая запись в БД. После чего производится копирование входных данных на выбранный кластер и запуск задачи с переданными параметрами.

Далее, до тех пор, пока задача не завершится, производится периодический опрос статуса. После завершения работы в БД появляется отметка об окончании и ссылка на архив с выходными данными. На этом этапе пользователь может ознакомиться с результатами работы задачи в виде предпросмотра или скачать полный набор выходных данных для последующего анализа на локальной машине.

Пользователю не обязательно находится на странице для ожидания результата выполнения задачи. Опрос статуса задачи автоматизирован и происходит в фоновом режиме. Также есть возможность настроить уведомление о завершении задачи. Для этого пользователю необходимо указать свою почту при регистрации.

3.4 Нагрузочное тестирование

В рамках разработки веб-портала было произведено нагрузочное тестирование для оценки производительности сервера.

Существует множество инструментов для нагрузочного тестирования, ниже перечислены некоторые из них:

- Яндекс.Танк
- Apache JMeter
- Studio Team System
- OpenSTA
- IBM Rational Performance Tester
- HP LoadRunner
- Siege
- HTTPerf
- QALoad

В рамках диссертационной работы использовался расширяемый инструмент нагрузочного тестирования Яндекс.Танк. Данный инструмент был выбран по ряду причин:

- Хорошая документация.
- Открытый исходный код.
- Простота в использовании.

В основе инструмента лежит высокопроизводительный асинхронный генератор нагрузки phantom. Данный инструмент позволяет оценить количество запросов в секунду и посчитать статистику по временам ответа по заданному профилю нагрузки. Профиль нагрузки задается в виде набора http вызовов и подбирается таким образом, чтобы наиболее полно соответствовать предполагаемому набору вызовов реальных пользователей. В Яндекс.Танк профиль нагрузки задается в виде «патронов», которыми впоследствии «обстреливается» целевой сервер. На рис. 3.9 представлена архитектура Яндекс.Танк. Можно отметить, что инструмент имеет гибкую систему плагинов, множество точек встраивания и гибкую систему настроек. Например, на схеме используется плагин для слежения за ресурсами тестового агента, подключенного по ssh-протоколу, а также указан критерий для автоостановки теста.

На рис. 3.10 представлена схема жизненного цикла нагрузочного теста. Сначала происходит конфигурирование и подготовка тестовых данных. Затем происходит запуск теста и ожидание его завершения. После чего собираются артефакты, подготавливаются отчеты и удаляются временные файлы.

На рис. 3.11 представлена схема взаимодействия модулей системы.

В качестве тестового стенда использовалась виртуальная машина со следующей конфигурацией:

- 2 виртуальных ядра центрального процессора.
- 2 гигабайта оперативной памяти.
- 50 гигабайт дискового пространства.
- 1 гигабит сетевой пропускной способности.
- Операционная система Ubuntu 22.04.

Яндекс.Танк запускался на отдельной независимой виртуальной машине с аналогичными характеристиками.

В качестве сценария для нагрузочного тестирования был выбран следующий профиль:

- Регистрация пользователя на веб-портале.
- Настройка параметров подключения к вычислительному кластеру, сохранение приватного ssh-ключа.
- Создание и загрузка программы, которая ничего не вычисляет.

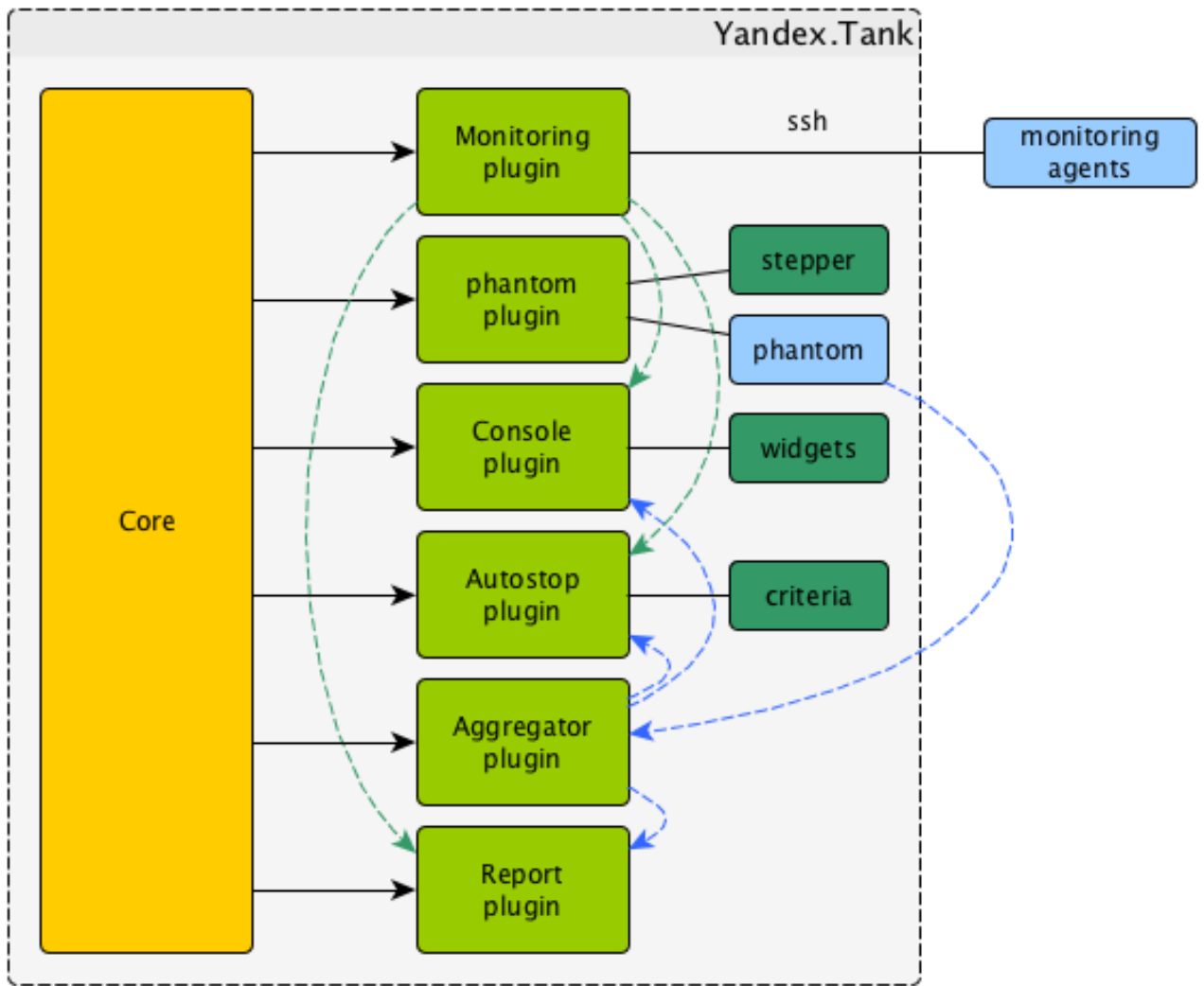


Рисунок 3.9 — Архитектура сервиса Яндекс.Танк

- Запуск 10 экземпляров задач.
- По каждой задаче 3 раза запрашивается статус выполнения.
- По каждой задаче скачивается результат выполнения.

Важно отметить, что на этапе создания программы происходит загрузка исполняемого файла, который не производит никаких вычислений. В рамках нагрузочного тестирования требуется оценить производительность только веб-портала и таким образом мы снижаем влияние вычислительного кластера на результаты. Так же снижается нагрузка на вычислительный кластер и, соответственно, влияние на других пользователей вычислительного кластера.

Шаблон файла с патронами представлен в листинге 3.1. Переменные в шаблоне, взятые в фигурные скобки, заполняются на этапе предварительной подготовки нагрузочного теста специальным скриптом.

Профиль количества запросов выглядит следующим образом:

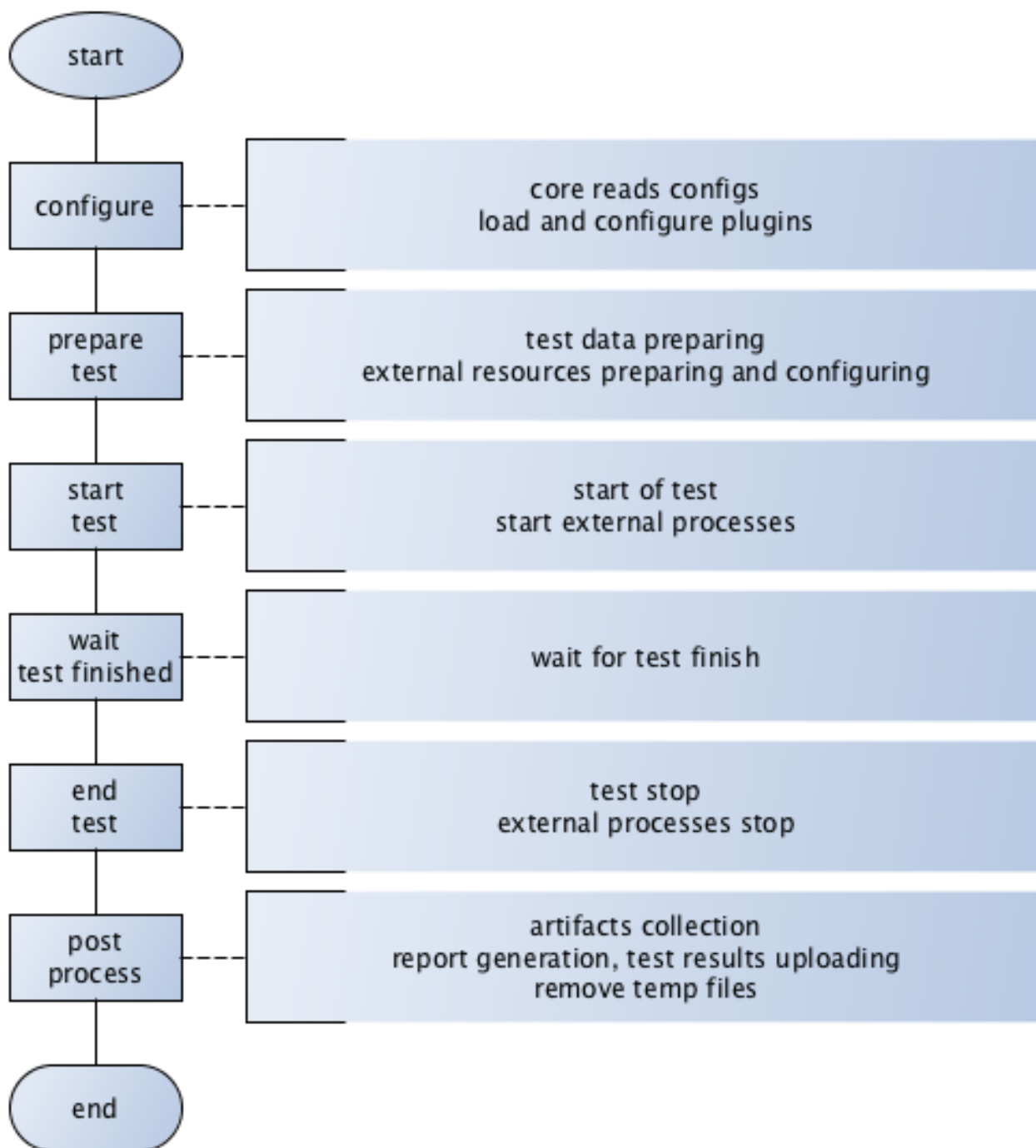


Рисунок 3.10 — Жизненный цикл нагрузочного теста

- Линейный рост от 0 запросов в секунду до 300 запросов в секунду в течение 1 минуты. Эта стадия необходима для того, чтобы сервер смог плавно выйти на полную мощность.
- Постоянное количество запросов в количестве 300 запросов в секунду в течение 9 минут.

Критерии экстренного останова:

- Появление кодов ответа, отличных от 2xx.

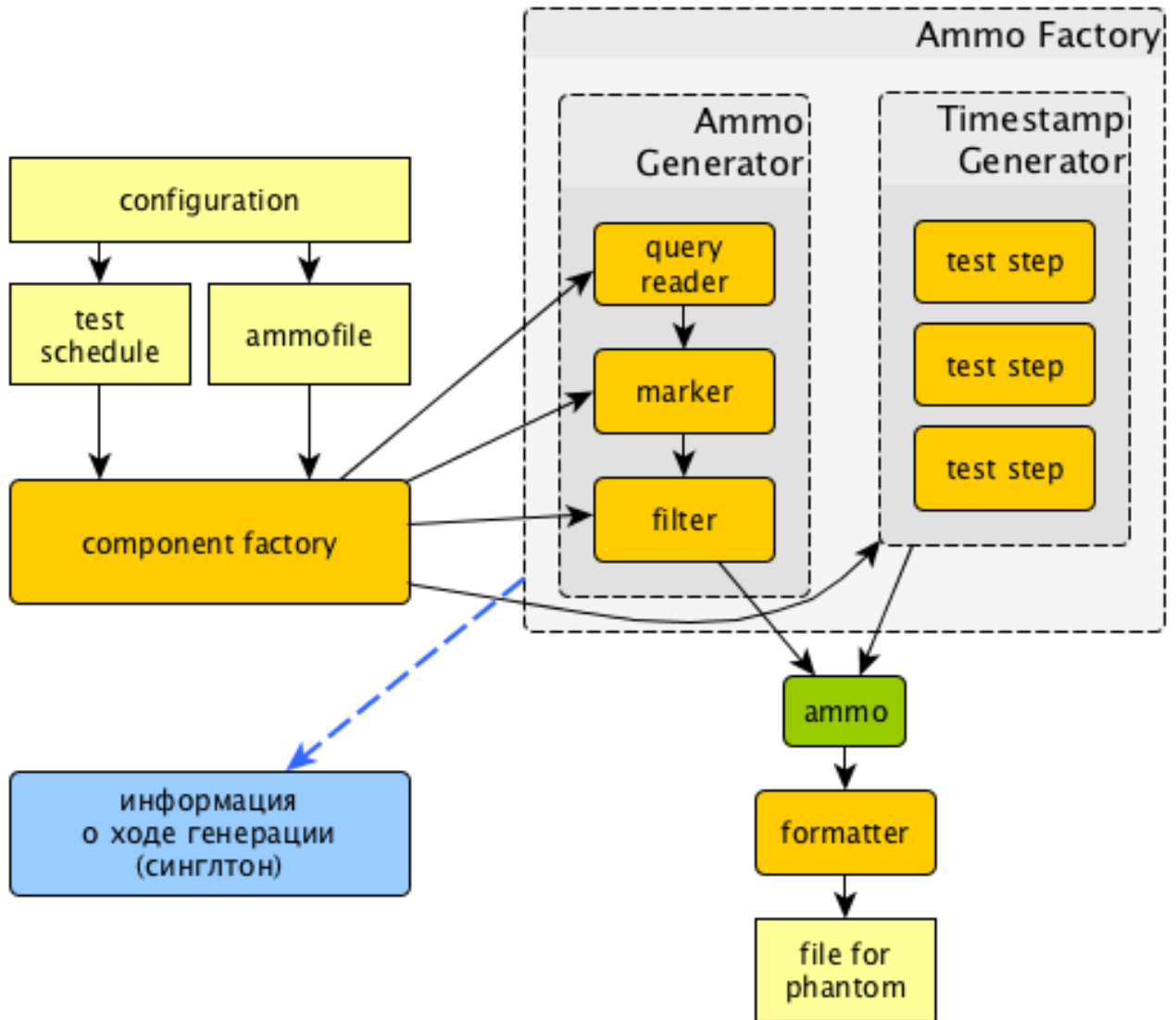


Рисунок 3.11 — Схема взаимодействия модулей системы Яндекс.Танк

Листинг 3.1 Шаблон файла с патронами

```

5 | [Connection: close]
   | [Host: web-portal]
   | [Cookie: {cookie}]
   |
   | {registration-ammo}
   | {config-ammo}
   | {run-ammo}
   | {download-ammo}
  
```

– Превышение времени ответа сервера 5 секунд.

На рис. 3.12 представлены графики с результатами нагрузочного теста. На графиках видно, что при нагрузке в 300 запросов в секунду время ответа сервера не превышало 1 секунду, при этом было использовано менее 200 потоков. Данные результаты показывают, что сервер, на котором запущен веб-портал, хорошо справляется с нагрузкой и способен обслужить более 300 одновременных сессий пользователей на тестовой конфигурации. Также имеется возможность горизонтального и вертикального масштабирования путем увеличения мощности сервера и количества запущенных экземпляров сервиса.

Запуск нагрузочного тестирования после каждого изменения позволяет следить за производительностью сервиса, вовремя реагировать на отклонения результатов теста и сопоставлять их с реальными изменениями в коде.

3.5 Численные эксперименты

Численные эксперименты проводились на суперкомпьютере УРАН с использованием графических ускорителей M2090 с помощью разработанного веб-портала.

Суперкомпьютер «Уран» собран на базе Blade серверов фирмы Hewlett-Packard. Он состоит из 81 вычислительного узла, которые установлены в модулях с высокой плотностью упаковки.

Вычислительные узлы оснащены процессорами Intel Xeon, работающими на частотах 2.2-3.1 ГГц, 48-384 гигабайтами оперативной памяти и графическими ускорителями NVIDIA Tesla.

В общей сложности пользователям доступно 1542 вычислительных ядра CPU, 167 платы GPU и 13 Тбайт оперативной памяти. Система хранения суперкомпьютера «Уран» позволяет разместить до 140 Тбайт данных. Для передачи данных между вычислительными узлами используются высокоскоростные сети Infiniband с пропускной способностью 20 Гбит/с и 100 Гбит/с. Доступ к суперкомпьютеру «Уран» осуществляется через городскую сеть УрО РАН в Екатеринбурге по технологии 10Gi Ethernet со скоростью 10 Гбит/с.

Проведенные численные эксперименты показывают, что методы и алгоритмы, с точки зрения времени счета и требуемых вычислительных ресурсов, являются эффективными и экономичными.

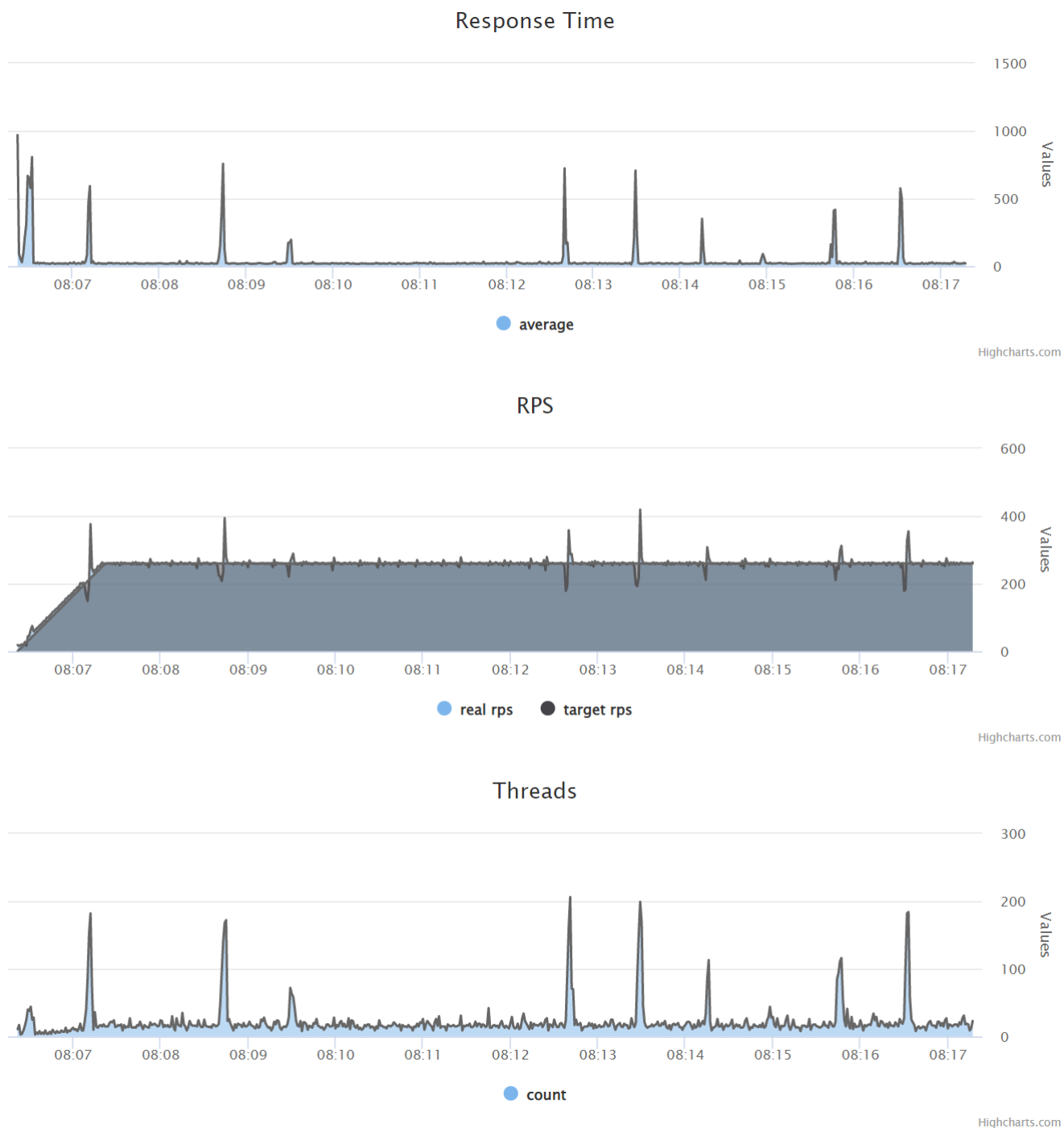


Рисунок 3.12 — Графики с результатами нагрузочного теста

3.5.1 Восстановление одной поверхности раздела по зашумленным гравитационным данным

Целью эксперимента является сравнение итерационных методов РЛМНС, РЛМСГ, МРЛМНС, МРЛМСГ, ГРЛМСГ решения обратной задачи гравимет-

рии для одной границы по времени счета с помощью разработанного комплекса программ, встроенных в универсальный веб-портал.

В экспериментах использовалась квазиреальная модель 2-слойной среды размерностью $2^9 \times 2^9$. На рис. 3.13 представлено гравитационное поле, полученное путем решения прямой задачи гравиметрии, на которое равномерный шум с максимальным отклонением от исходного значения равным 10%. Асимптотическая плоскость $H = 6$ км, плотность $\Delta\sigma = 0.1$ г/см³.

При решении задачи использовались следующие параметры: $\alpha = 0.1$. Значения были подобраны экспериментальным путем. Условием останова было выбрано $\frac{\|A(z)-F\|}{\|F\|} < \varepsilon$, $\varepsilon = 0.05$. Относительная погрешность при этих условиях составляет $\delta = \frac{\|z-z^*\|}{\|z^*\|} = 0.06$, где z — приближенное решение, z^* — точное решение. Начальная погрешность составляла 0.206.

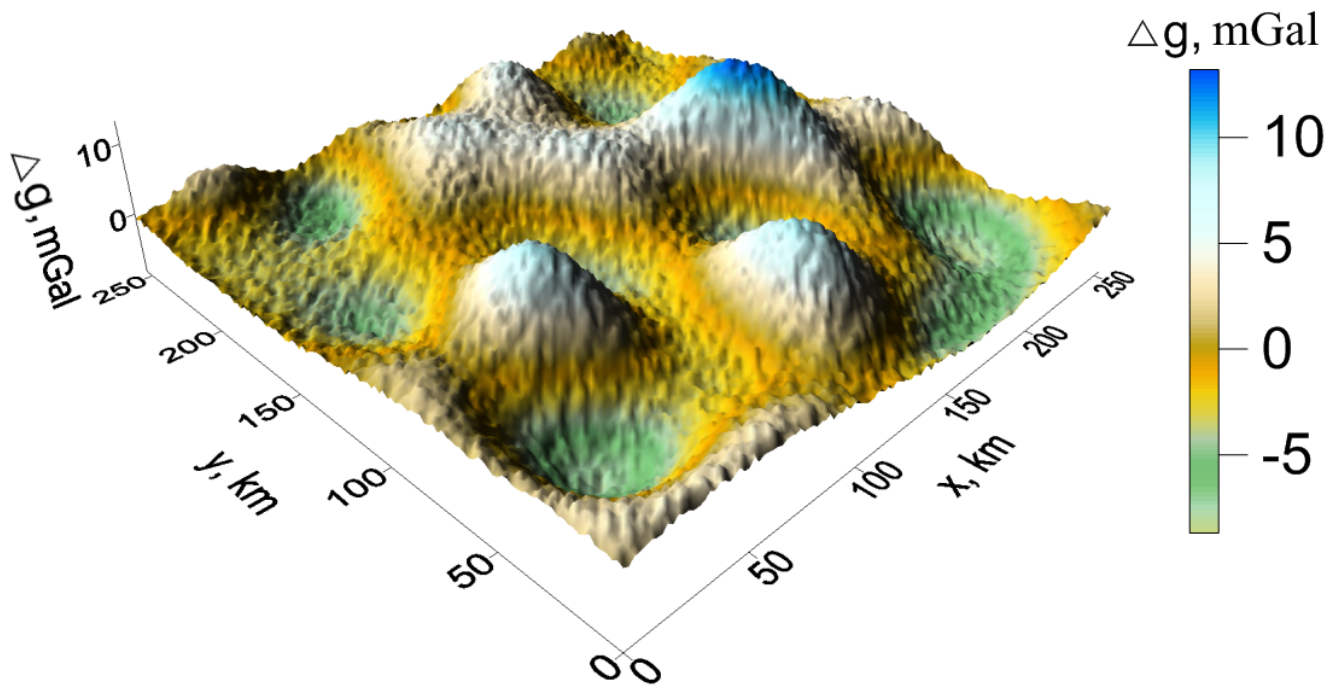


Рисунок 3.13 — Модельное гравитационное поле для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным

На рис. 3.14 представлено точное решение задачи, на рис. 3.15 - приближенное решение.

В таблице 1 приводится сравнение методов по количеству итераций и среднему времени счета для 10 запусков на узле с использованием одного графического ускорителя для сетки 512×512 . Применение модифицированных вариантов сокращает общее время вычислений в 1.4 раза по сравнению с немодифицированными вариантами. Гибридный вариант оказался эффективнее

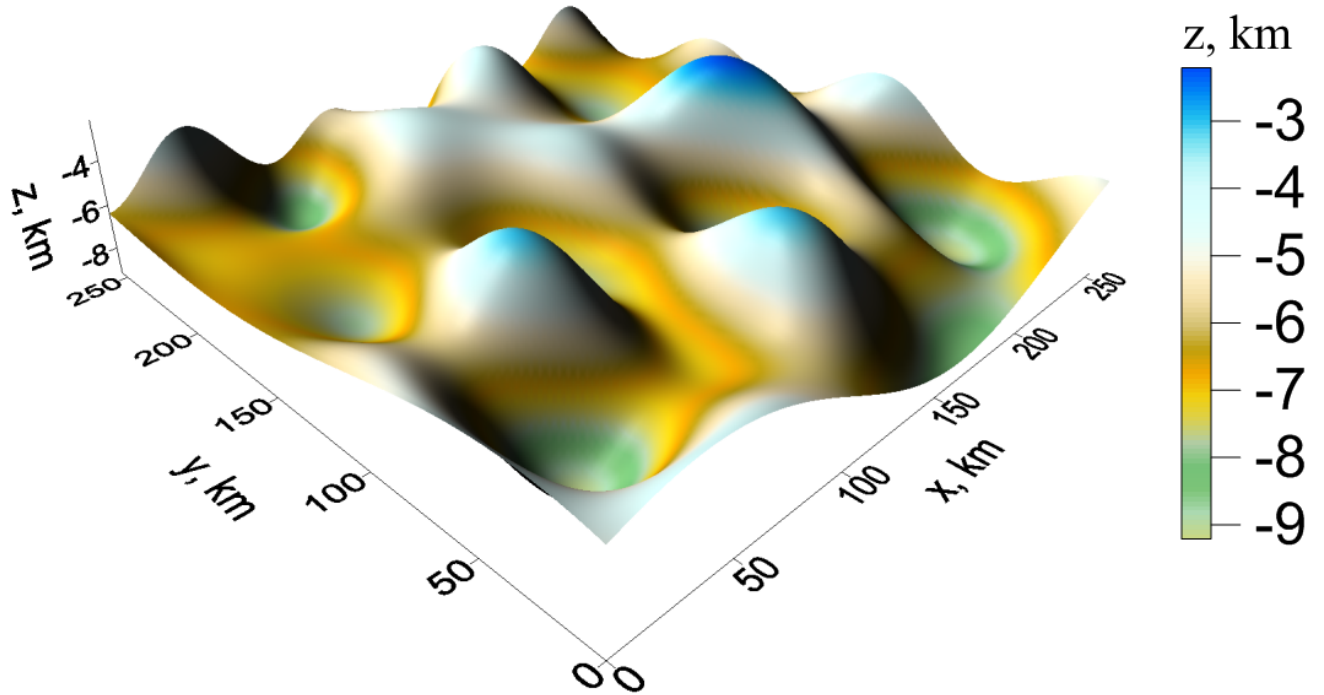


Рисунок 3.14 — Точное решение для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным

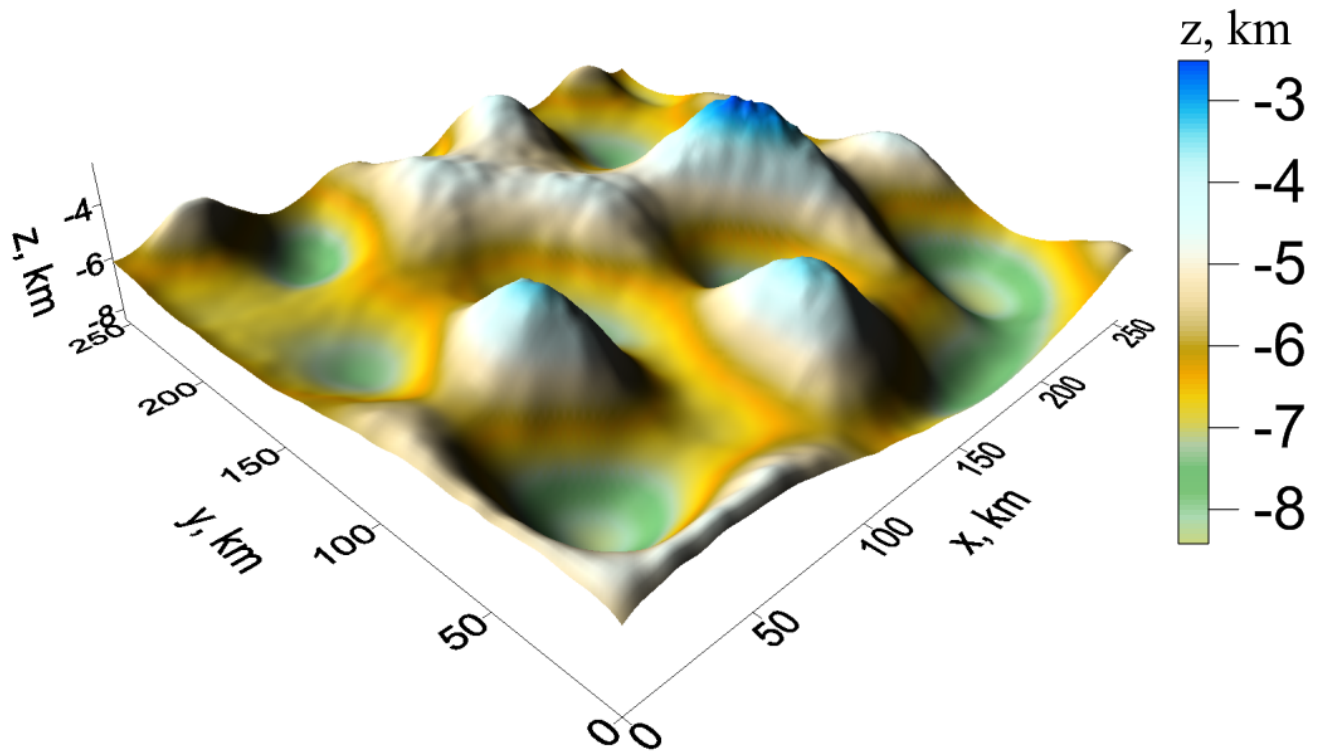


Рисунок 3.15 — Приближенное решение для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным

немодифицированного метода, но его время счета больше модифицированного варианта при таком же количестве итераций и точности.

Таблица 1 — Результаты численных экспериментов для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным

Метод	N, число итер.	T	T на 1 итер.
МНС (без модификации)	5	120 сек	24 сек
МСГ (без модификации)	4	115 сек	28 сек
ММНС (модифицированный, ТБТ)	7	70 сек	10 сек
ММСГ (модифицированный, ТБТ)	6	72 сек	10 сек
ГМСГ (гибридный, ТБТ)	6	114 сек	19 сек

В таблице 2 приведено сравнение эффективности и ускорения для метода ММСГ при использовании различного количества графических ускорителей. Ускорение (рис. 3.16) для 8 GPU составляет 7.24, эффективность (рис. 3.17) составляет 90%.

Таблица 2 — Результаты численных экспериментов для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным

Число ускорителей	Время, сек	Ускорение, S	Эффективность, E
1	72	1	1
2	38	1.9	0.94
4	19	3.8	0.93
8	10	7.2	0.9

Вывод. Применение модифицированных методов незначительно увеличивает число итераций для достижения сравнимой точности, но за счет экономии вычислений на каждой итерации сокращает общее время счета до 2 раз. Экономия памяти при использовании ТБТ матрицы составила 65 тысяч раз.

Ускорение

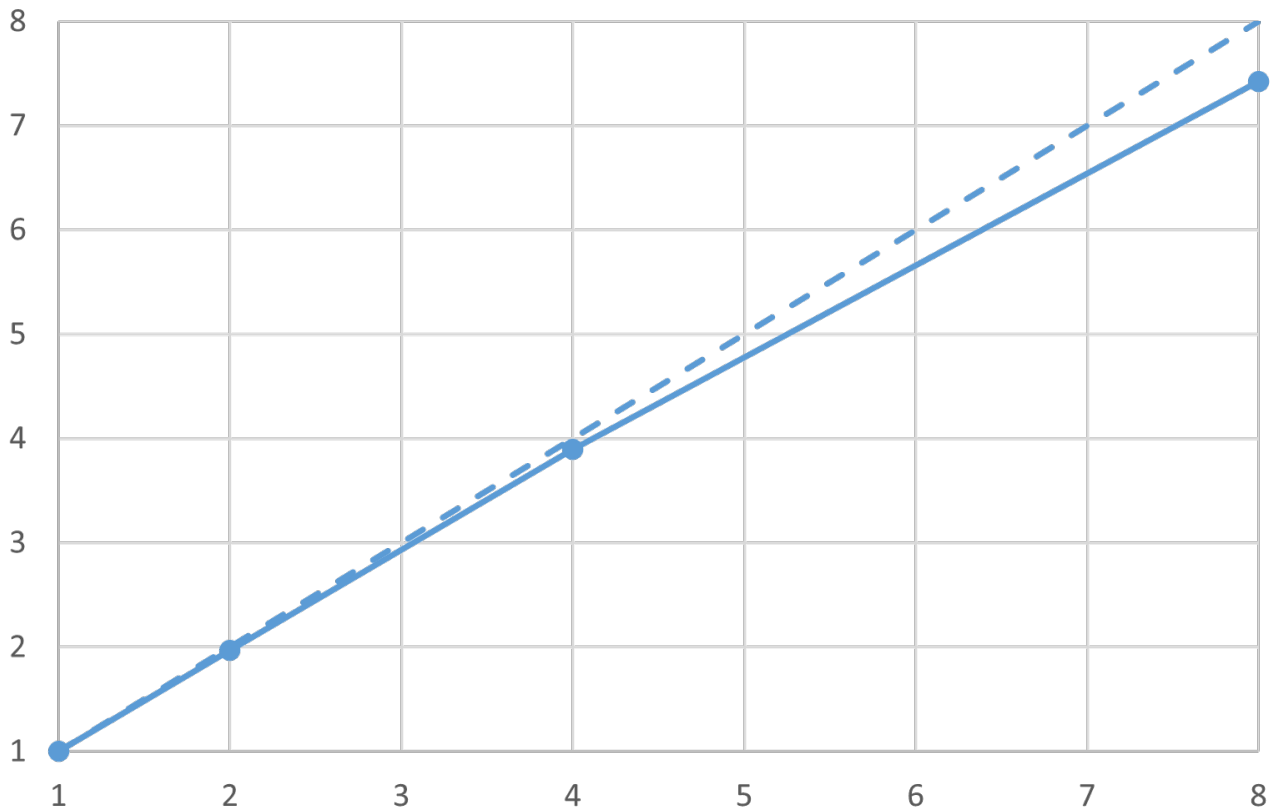


Рисунок 3.16 — Ускорение для метода ММСГ для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным

3.5.2 Восстановление одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

Целью эксперимента является сравнение методов РЛМСГ, ПГМ, МПГМ решения обратной задачи магнитометрии для случая произвольно направленного вектора намагниченности по времени счета.

В экспериментах использовалась квазиреальная модель 2-слойной среды размерностью $2^9 \times 2^9$.

На рис. 3.18 представлено точное решение задачи.

$$z = 20 - 5.21e^{-(x/6.13)^4 - (y/9.59)^4} + 6.11e^{-(x/4.11 + 8.12)^4 - (y/7.5 - 3.65)^4} + 8.27e^{-(x/6.13 - 4.9)^4 - (y/6.72 - 3.65)^4}.$$

Эффективность

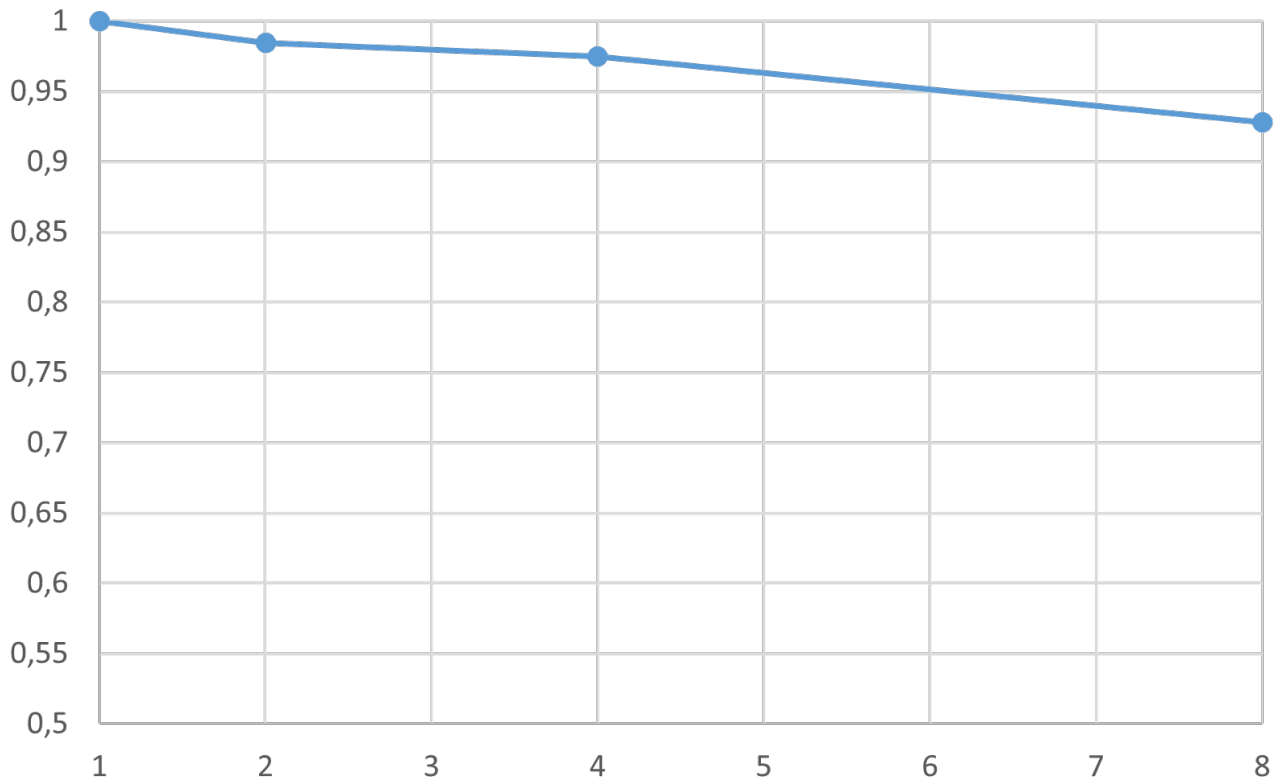


Рисунок 3.17 — Эффективность для метода ММСГ для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным

На рис. 3.19 — 3.23 - представлены модельные поля $\Delta Z_i(x,y,0)$ для различных углов отклонения от вертикали векторов намагниченности ($0^\circ - 80^\circ$). Эти поля получены путем решения прямой задачи магнитометрии с асимптотической плоскостью $H = 20$ км и следующими намагниченностями слоев:

$$\begin{aligned}
 \Delta J_1 &= (0,0,1)\text{A/m}, \\
 \Delta J_2 &= (0.19,0.19,1)\text{A/m}, \\
 \Delta J_3 &= (0.41,0.41,1)\text{A/m}, \\
 \Delta J_4 &= (0.71,0.71,1)\text{A/m}, \\
 \Delta J_5 &= (1.23,1.23,1)\text{A/m}, \\
 \Delta J_6 &= (1.94,1.94,1)\text{A/m}, \\
 \Delta J_7 &= (4.01,4.01,1)\text{A/m}.
 \end{aligned}
 \tag{3.1}$$

Эти намагниченности соответствуют углам отклонения намагниченности от вертикали $0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 70^\circ, 80^\circ$.

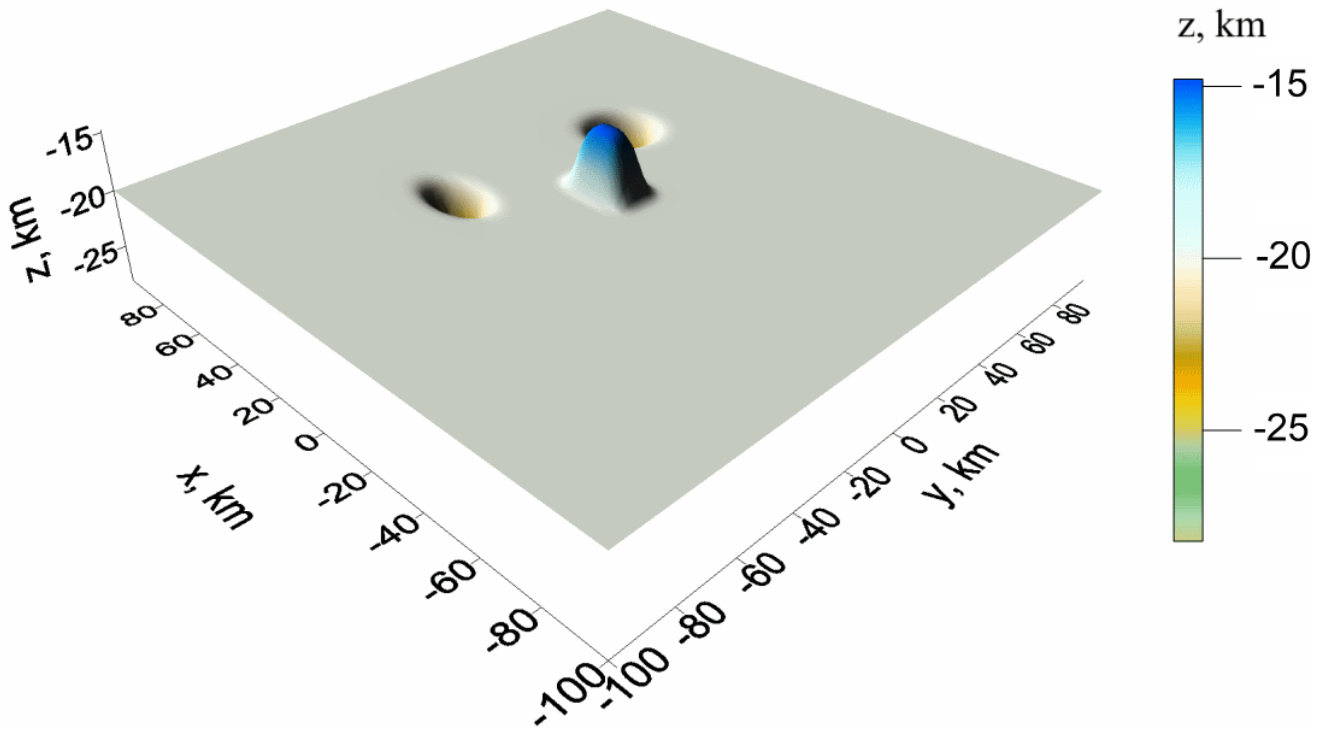


Рисунок 3.18 — Точное решение для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

На рис. 3.24 изображено приближенное решение.

Эксперименты проводились на суперкомпьютере Уран на узлах с 8 видеоускорителями NVIDIA Tesla M2090.

В качестве условия останова использовалось $\|A(z) - F\|/\|F\| < \varepsilon$, $\varepsilon = 0.01$ для всех методов. Параметр $\psi = 0.85$ был использован в РЛМСГ для 60° , в ПГМ и МПГМ для 45° .

Параметр $\psi = 0.75$ был использован в ПГМ и МПГМ для 60° . Значение параметра $\psi = 0.35$ использовалось в ПГМ и МПГМ для углов 70° и 80° . В остальных случаях был использован $\psi = 1$. РЛМСГ выполнялся с параметром регуляризации $\alpha = 0$.

В таблице 3 приводится сравнение методов по количеству итераций и среднему времени счета для 10 запусков на узле с использованием одного графического ускорителя для сетки 512×512 . Применение модифицированного варианта сокращает общее время вычислений в 3 раза для небольших углов и в 1.5 раза для больших по сравнению с LCGM. Обозначение «расх.» означает, что не удалось подобрать параметр ψ , при котором невязка уменьшается до требуемого значения.

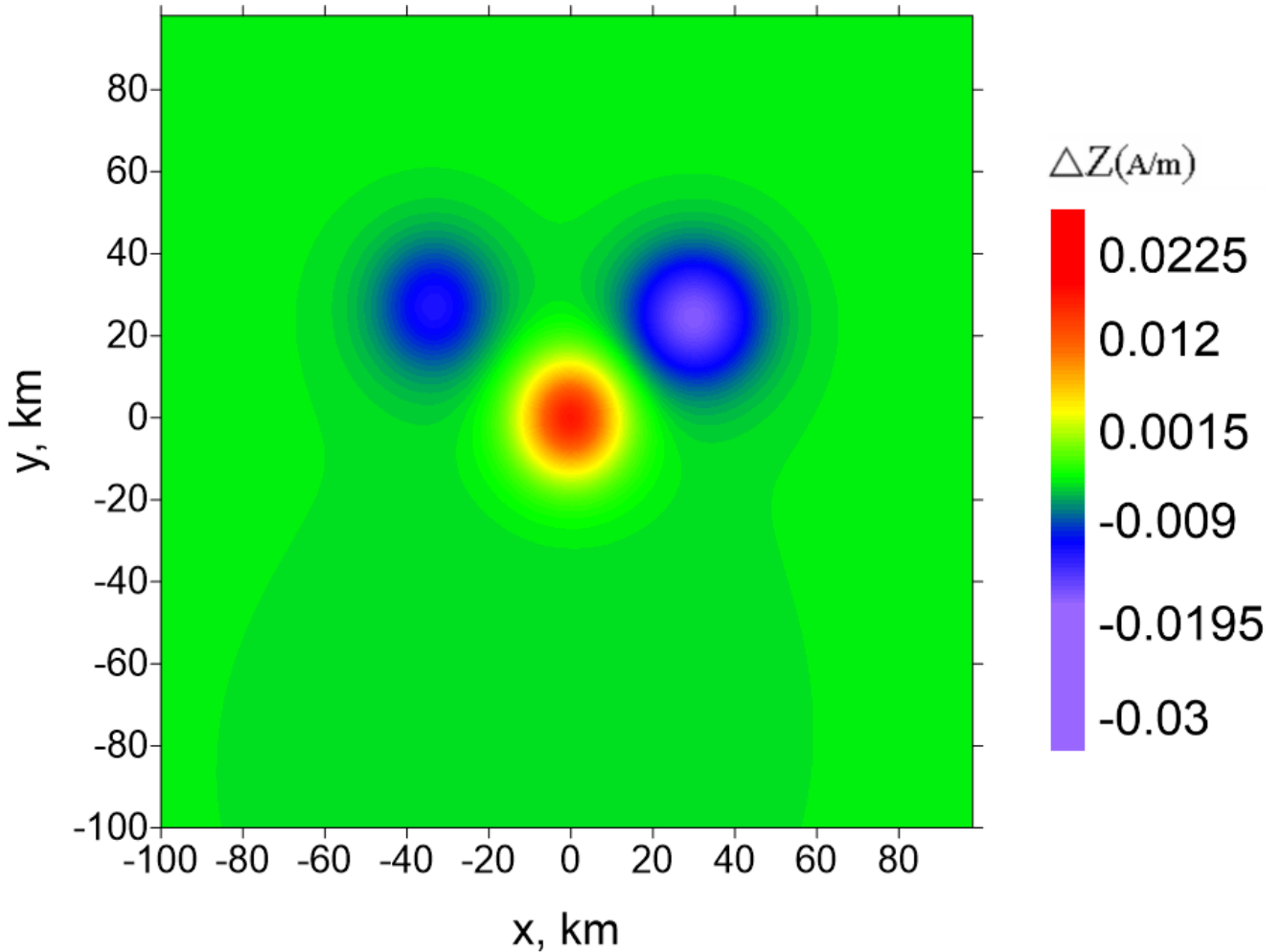


Рисунок 3.19 — Модельное гравитационное поле для угла 0 для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

Таблица 3 — Сравнение методов для задачи 3.5.2 о восстановлении поверхности раздела для случая произвольно направленного вектора намагниченности

Угол вектора	РЛМСГ (2.8)		ПГМ (2.1)		МПГМ (2.3)	
	N	T , мин	N	T , мин	N	T , мин
0°	20	22	6	7.6	6	7.6
15°	20	22	6	7.6	6	7.6
30°	20	22	8	8.3	7	8
45°	25	28	10	10.8	9	10.3
60°	26	29	16	17.7	14	17
70°	64	70	расх.	расх.	19	22
80°	расх.	расх.	расх.	расх.	100	110

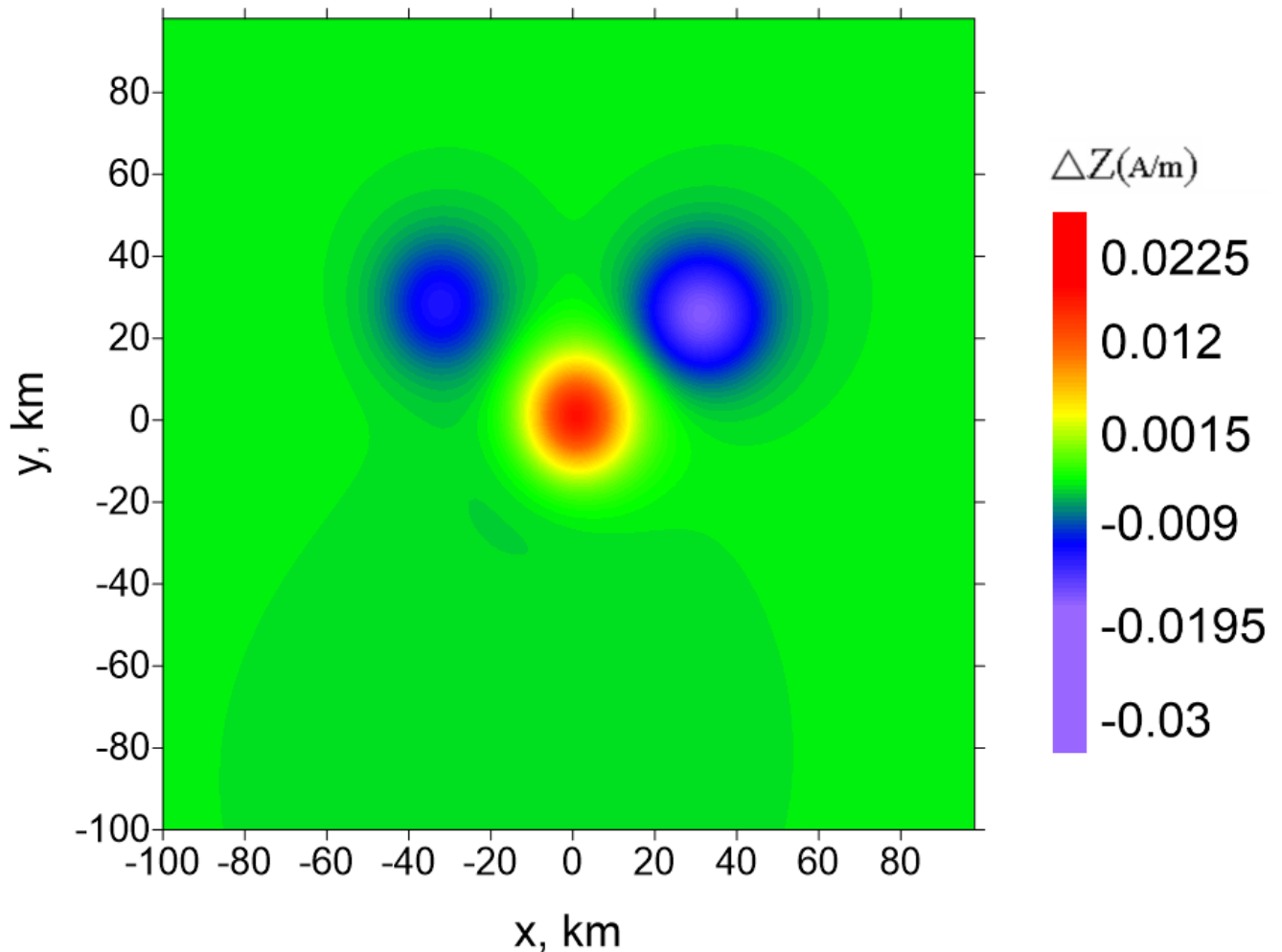


Рисунок 3.20 — Модельное гравитационное поле для угла 15 для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

На рис. 3.25–3.26 изображены графики изменение невязки по правой части и погрешности решения в зависимости от числа итераций для углов 70° и 80° .

При значениях углов от 0° до 60° относительная погрешность решения для трех рассмотренных методов составила $\delta = \|z - z^*\|/\|z^*\| < 0.01$. При значении угла 70° ПГМ расходится (невязка по правой части увеличивается с ростом числа итераций, см. рис. 3.25а). При использовании методов РЛМСГ и МПГМ относительная норма невязки уменьшается до $\varepsilon = 0.01$. При этом время счета МПГМ в 3 раза меньше, чем РЛМСГ. Погрешности решений, полученных методами РЛМСГ и МПГМ, составили 0.015.

При значении угла 80° РЛМС и ПГМ расходятся (см. рис. 3.25б). При использовании метода МПГМ относительная норма невязки уменьшается до $\varepsilon = 0.012$ за 100 итераций, при этом относительная погрешность полученного решения составляет 0.015.

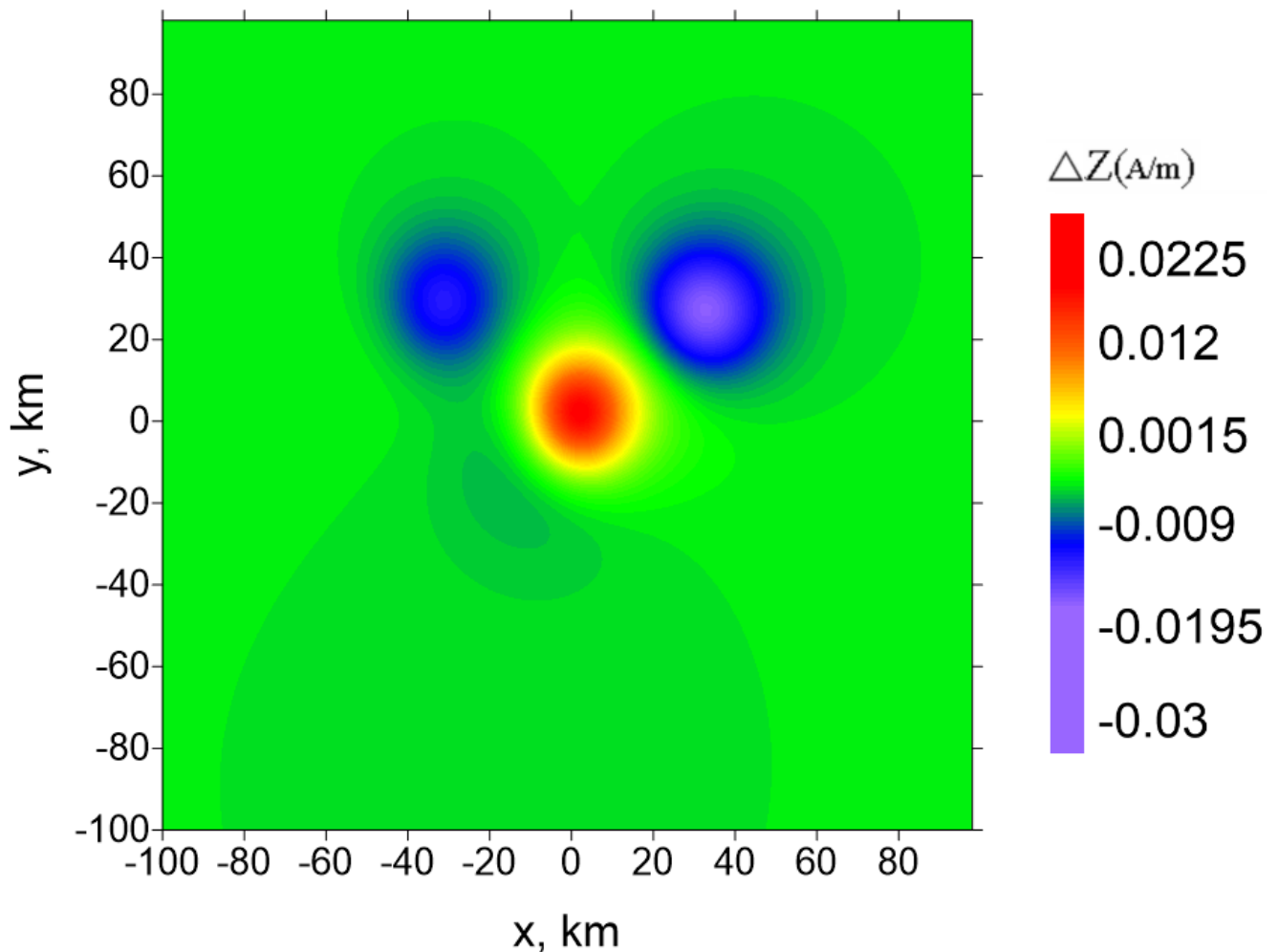


Рисунок 3.21 — Модельное гравитационное поле для угла 30° для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

В таблице 4 приводится сравнение по времени счета для различного числа ускорителей. А также приводится эффективность и ускорение. Ускорение для 8 GPU составляет 7.83, эффективность составляет 97%.

Вывод. Модифицированный покомпонентный метод обладает преимуществом по сравнению с ПГМ и РЛМСГ. Метод позволяет решать задачу магнитометрии при углах отклонения вектора намагниченности до 80° . Применение модифицированного покомпонентного метода сокращает общее время вычислений в 2-3 раза по сравнению с РЛМСГ.

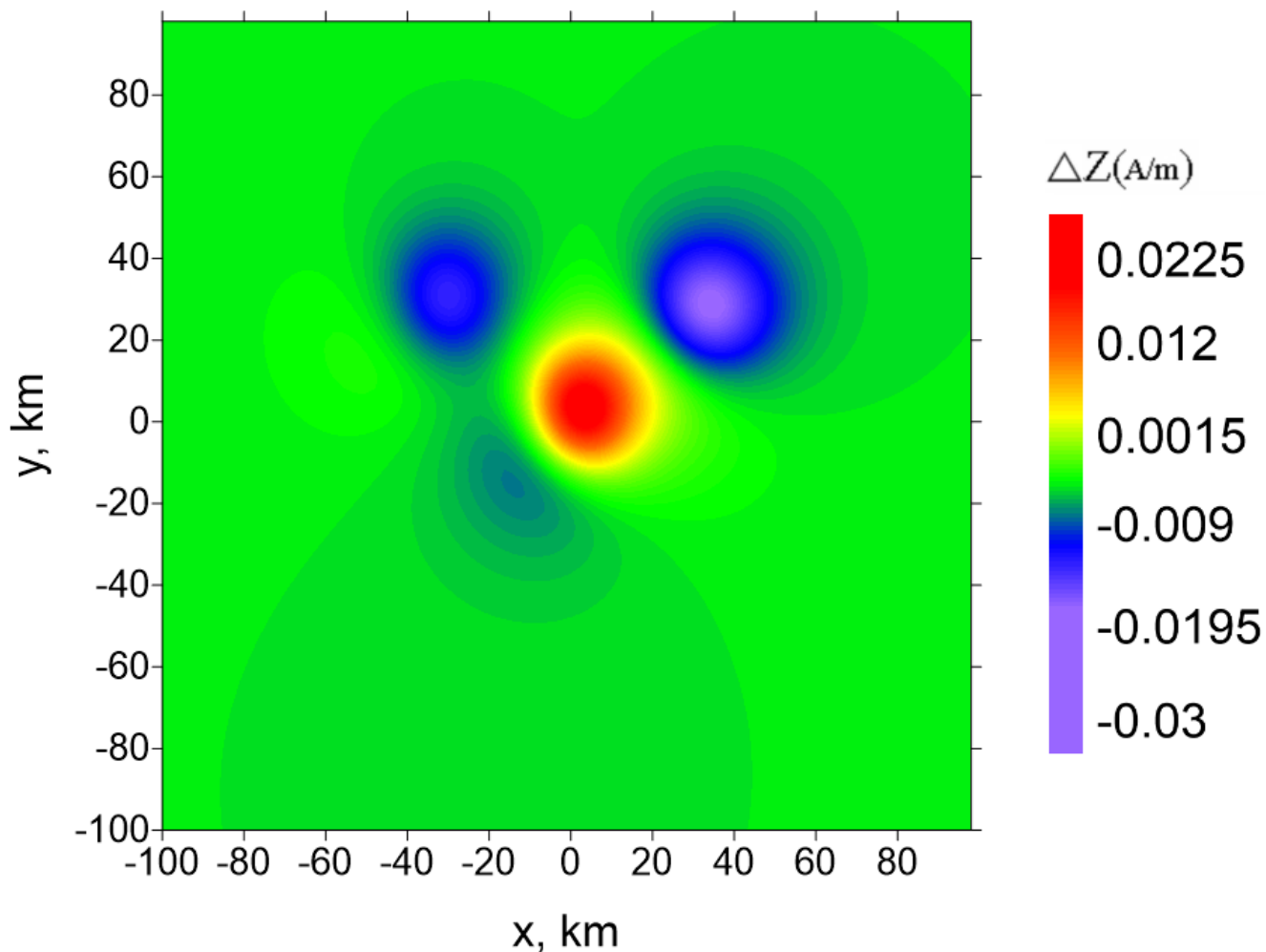


Рисунок 3.22 — Модельное гравитационное поле для угла 45° для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

Таблица 4 — Время счета МПГМ для различного числа GPU для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

Угол вектора	Количество GPU			
	1	2	4	8
	Время T , мин			
0°	4.3	2.2	1.1	0.55
15°	4.3	2.2	1.1	0.55
30°	5	2.5	1.3	0.65
45°	6.3	3.2	1.6	0.8
60°	10	5	2.5	1.25

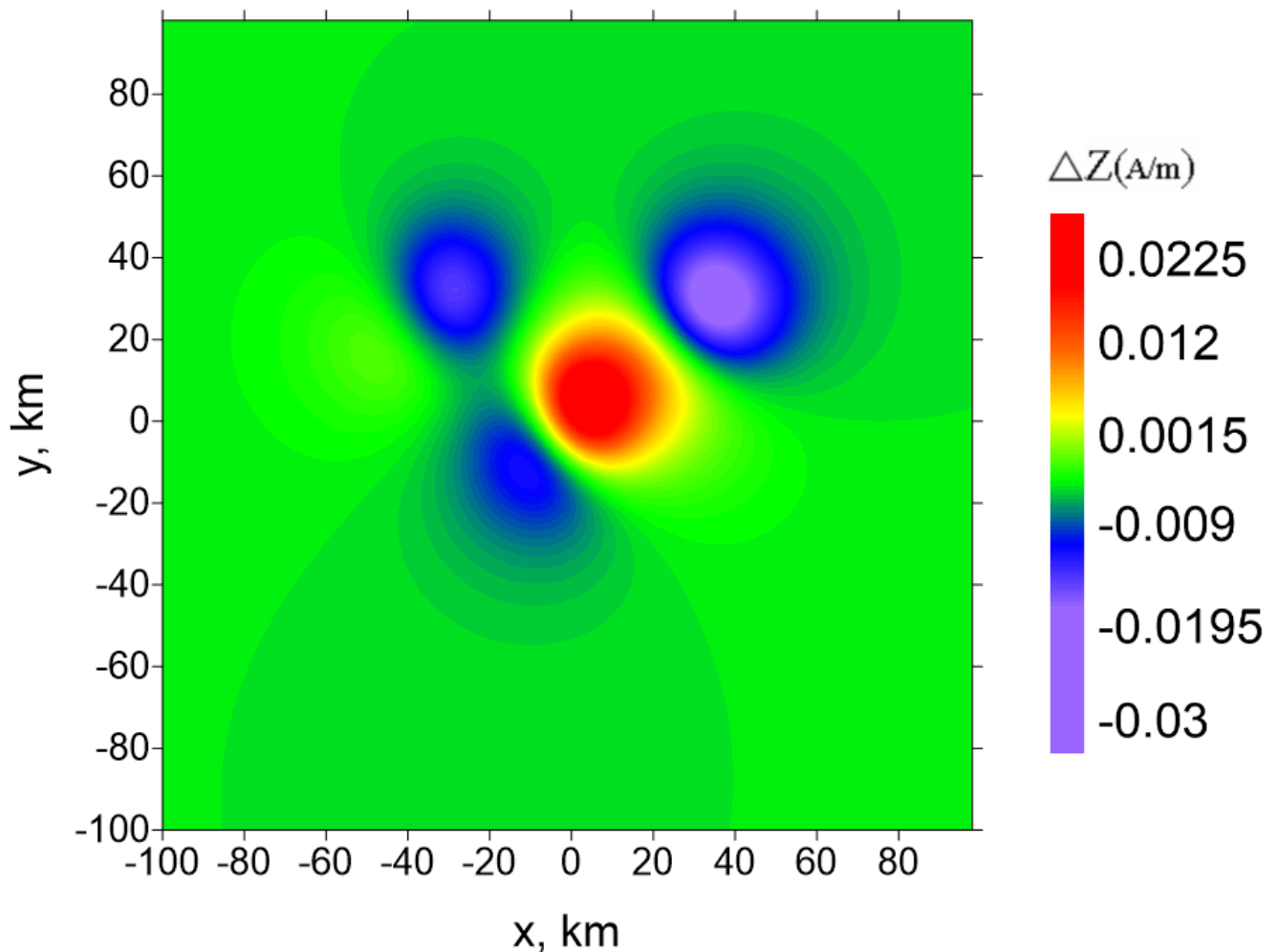


Рисунок 3.23 — Модельное гравитационное поле для угла 60° для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

3.5.3 Восстановление трех поверхностей раздела по гравитационным данным с использованием реального гравитационного поля

Целью эксперимента является сравнение методов ЛМСГВМ и ММСГВМ решения обратной задачи гравиметрии для нескольких границ по времени счета с помощью разработанного веб-портала.

Гравитационное поле от трех поверхностей построено на основе карт гравитационных полей [82]. Кадрирован район Среднего Урала, произведен пересчет градусов в км. Выделено поле, соответствующее области $600 \times 600 \times 45$ км между глубинами 5 и 50 км.

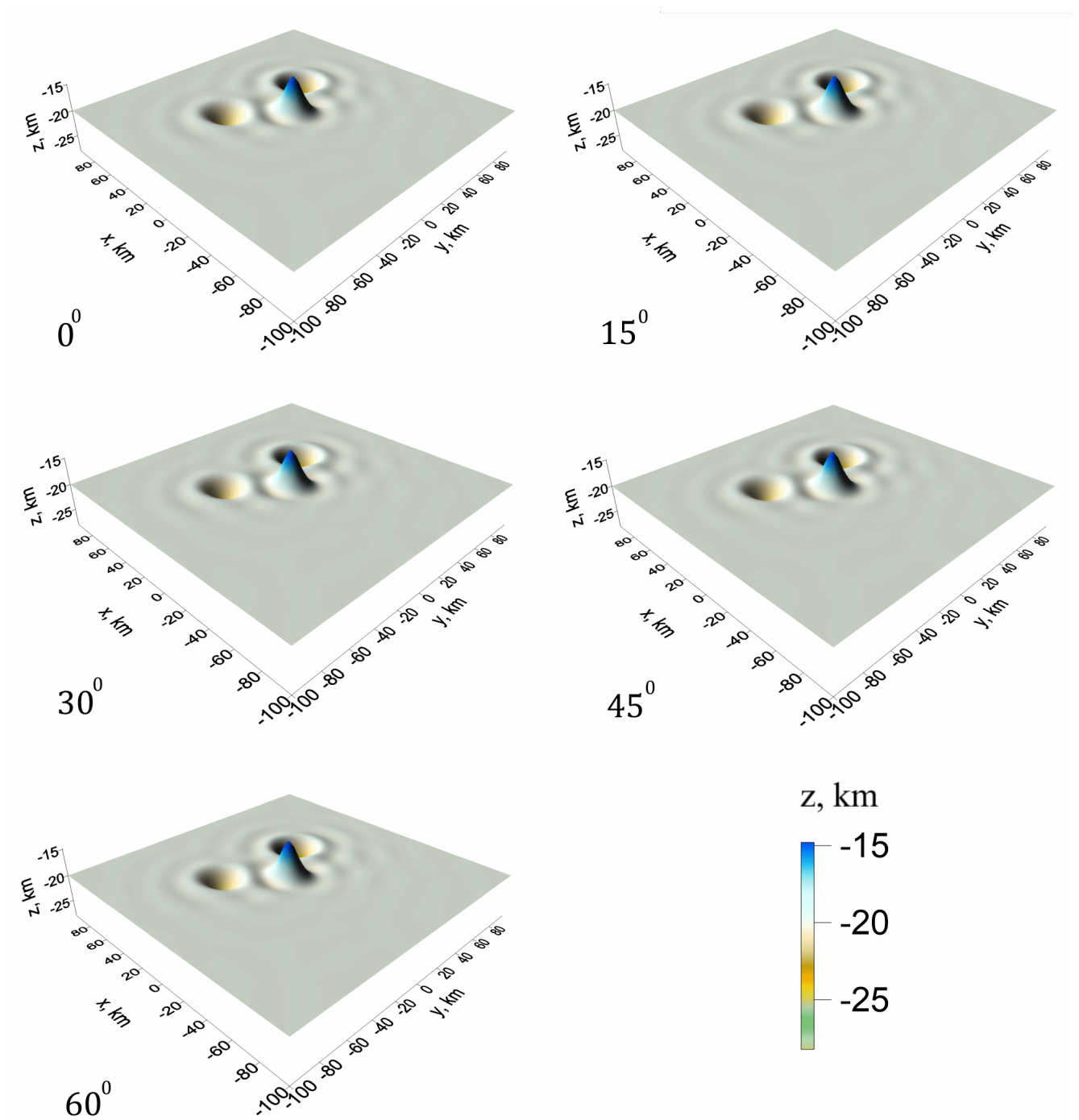


Рисунок 3.24 — Приближенные решения для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности

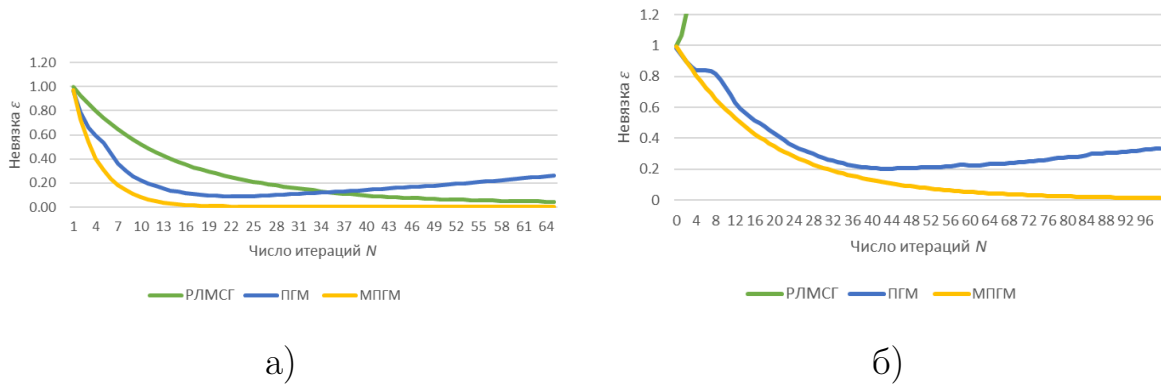


Рисунок 3.25 — Изменение невязки по правой части по итерациям а) угол 70° ; б) угол 80°

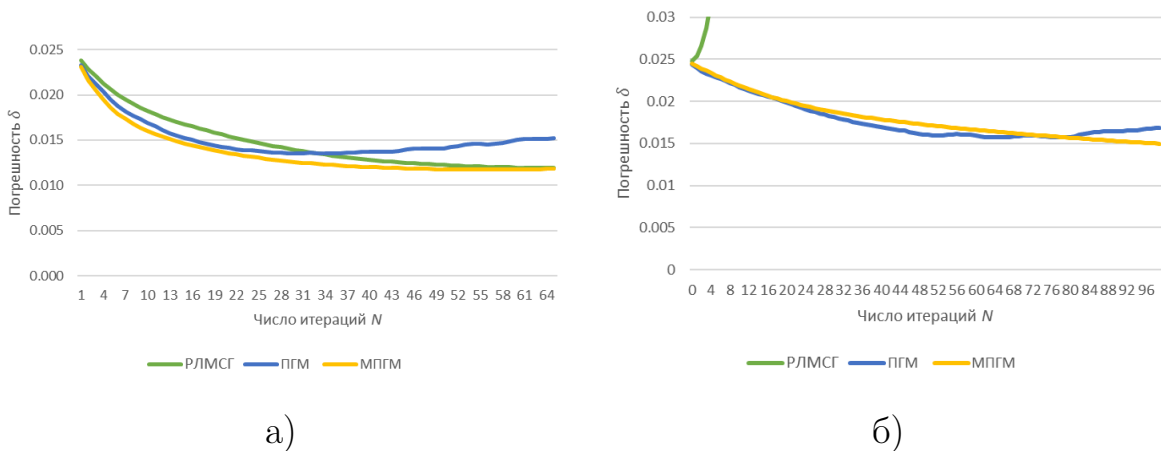


Рисунок 3.26 — Изменение погрешности решения по итерациям а) угол 70° ; б) угол 80°

Одним из способов получения реальных данных по плотности используется зависимость между скоростью распространения сейсмических волн и плотностью, рассмотренная в работе [52].

В данной задаче взяты модельные данные глубин и плотностей для трех поверхностей с асимптотическими плоскостями $H_1 = 10$ км, $H_2 = 20$ км, $H_3 = 30$ км и скачками плотности $\sigma_1 = 0.3$ г/см³, $\sigma_2 = 0.2$ г/см³, $\sigma_3 = 0.1$ г/см³.

Общее поле методом повысотных трансформаций [52] разделено на три компоненты, соответствующие глубинам 5–15, 15–25 и 25–50 км. Модельные поверхности подготовлены решением независимых задач для каждого поля.

Далее решена задача об одновременном восстановлении трех модельных поверхностей по гравитационному полю. Относительные погрешности для каждой восстановленной поверхности не превышают 3%.

В экспериментах использовалась квазиреальная модель 4-слойной среды размерностью $2^9 \times 2^9$. На рис. 3.27 представлено суммарное гравитационное

поле, полученное путем решения прямой задачи с тремя модельными поверхностями S_1, S_2, S_3 (рис. 3.28). На рис. 3.29 представлены восстановленные поверхности.

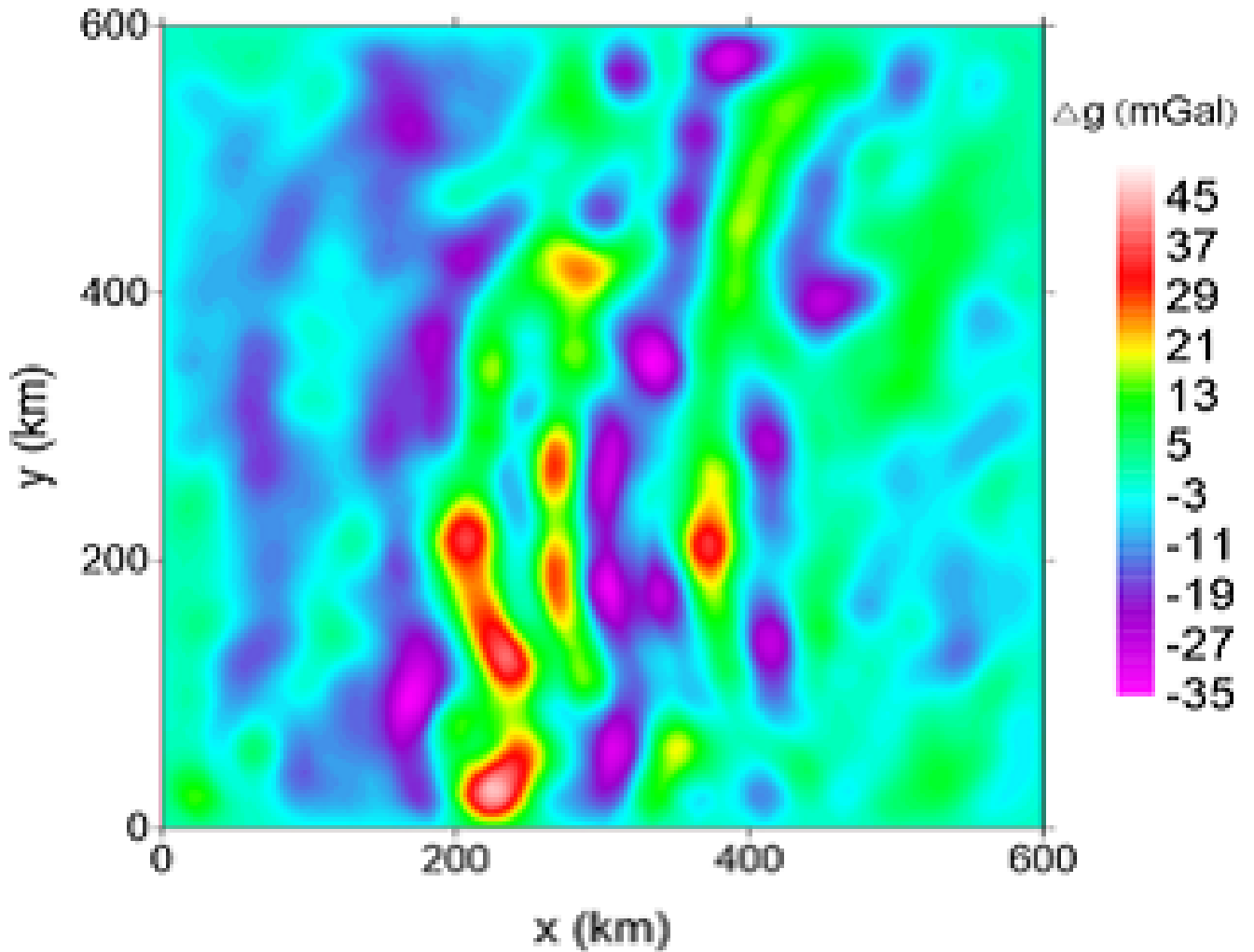


Рисунок 3.27 — Суммарное гравитационное поле для задачи 3.5.3 о восстановлении трех поверхностей раздела по гравитационным данным

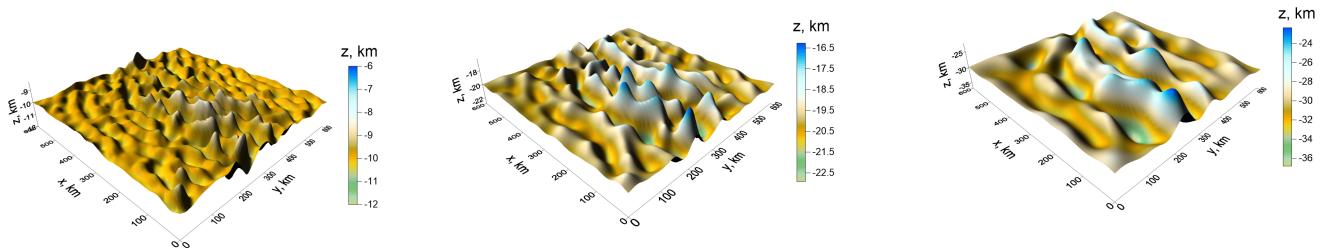


Рисунок 3.28 — Модельные поверхности S_1, S_2, S_3 для задачи 3.5.3 о восстановлении трех поверхностей раздела по гравитационным данным

При решении задачи использовались следующие параметры: $\psi = 1$, $\alpha = 0.1$. Значения были подобраны экспериментальным путем. Условием останова

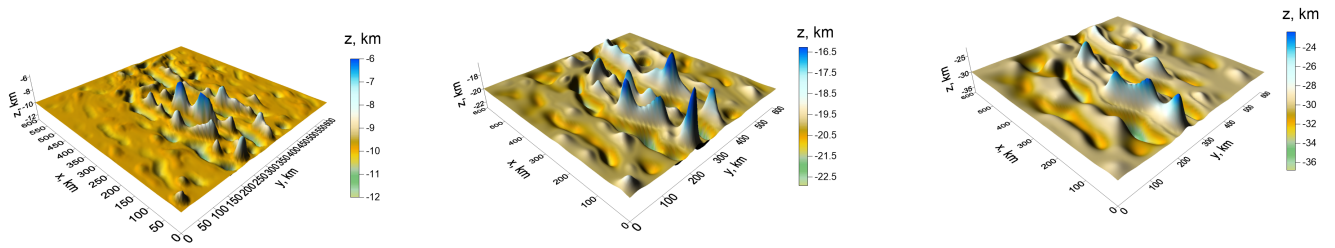


Рисунок 3.29 — Восстановленные поверхности S_1, S_2, S_3 для задачи 3.5.3 о восстановлении трех поверхностей раздела по гравитационным данным

было выбрано $\frac{\|A(z)-F\|}{\|F\|} < \varepsilon$, $\varepsilon = 0.25$. Относительная погрешность при этих условиях составляет $\delta = \frac{\|z-z^*\|}{\|z^*\|} < 0.03$.

Запуск расчетов производится на странице списка загруженных программ. Результаты вычислений можно посмотреть на странице списка задач. Для приведенных алгоритмов для постобработки результатов используется модуль взаимодействия с Surfer. Это позволяет вывести изображения поверхностей на странице результатов.

Для сравнения времени счета решения задачи используются формулы ускорения и эффективности параллельных алгоритмов:

$$S_m = \frac{T_1}{T_m}, E_m = \frac{S_m}{m},$$

где S_m — ускорение для m графических ускорителей, T_1 — время счета на 1 графическом ускорителе, T_m — время счета на m графических ускорителях, E_m — эффективность для m графических ускорителей.

В таблице представлены результаты эксперимента для ММСГ, МСГ с весовыми множителями:

Вывод. Параллельные программы показали высокую эффективность до 90%. Время счета уменьшилось с 15 часов на одном процессоре до 45 секунд на 8 графических ускорителях. Модифицированный метод с ТБТ матрицей работает в 2 раза быстрее, чем метод без модификации.

Таблица 5 — Результаты численных экспериментов для задачи 3.5.3 о восстановлении трех поверхностей раздела по гравитационным данным

Алгоритм	Вычислитель		Время T	Ускорение, S	Эффек- тивность, E
МСГ с весовыми множителями	Intel Xeon E5-2650	1 поток	14.7 ч.	-	-
	NVIDIA Tesla M2090	1 x GPU	12 мин.	68	-
		8 x GPU	1.73 мин.	510 (6.93)	(0.86)
мод. МСГ с весовыми множителями	Intel Xeon E5-2650	1 поток	7.5 ч.	-	-
	NVIDIA Tesla M2090	1 x GPU	5.3 мин.	85	-
		8 x GPU	0.74 мин.	610 (7.16)	(0.89)

3.5.4 Восстановление одной поверхности раздела по гравитационным данным

В экспериментах использовалась квазиреальная модель 2-слойной среды размерностью $2^9 \times 2^9$. Модель построена на основе гравитационного поля для поверхности 3 из предыдущей задачи. На рис. 3.30 представлено модельное гравитационное поле, полученное путем решения прямой задачи гравиметрии, используя известную поверхность z с асимптотической плоскостью $H = 30$ км и плотностью $\Delta\sigma = 0.1$ г/см³.

На рис. 3.31 представлено точное решение задачи, на рис. 3.32 - приближенное решение.

Задача решалась на суперкомпьютере УРАН на узле с установленными 8 графическими ускорителями NVIDIA Tesla M2090. Для решения использовались 3 метода:

- метод сопряженных градиентов (РЛМСГ)
- модифицированный метод сопряженных градиентов (МРЛМСГ)
- покомпонентный метод (ПГМ)

В качестве условия останова использовалось $\|A(z) - F\|/\|F\| < \varepsilon$, $\varepsilon = 0.01$ для всех методов.

Для всех методов в качестве параметров использовались $\psi = 1$ и $\alpha = 0.1$.

Относительная погрешность для всех методов составила $\delta = \|z - z^*\|/\|z^*\| < 0.01$.

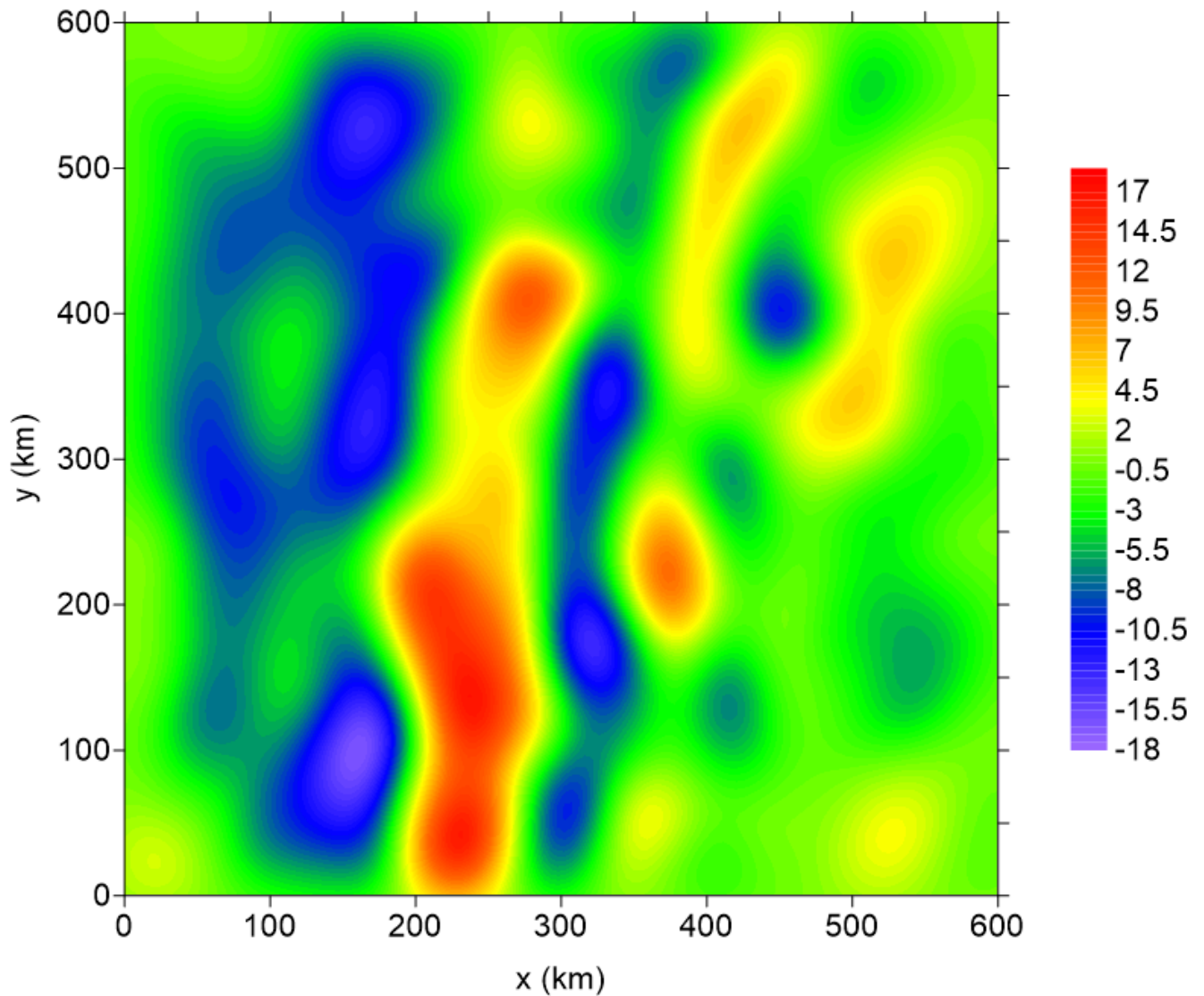


Рисунок 3.30 — Модельное гравитационное поле для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным

В таблице 6 приводится сравнение методов по количеству итераций и среднему времени счета для 10 запусков на узле с использованием одного графического ускорителя для сетки 512×512 .

Таблица 6 — Результаты численных экспериментов для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным

Метод	Количество итераций	Время, мин
РЛМСГ	30	36
МРЛМСГ	40	33
ПГМ	21	18

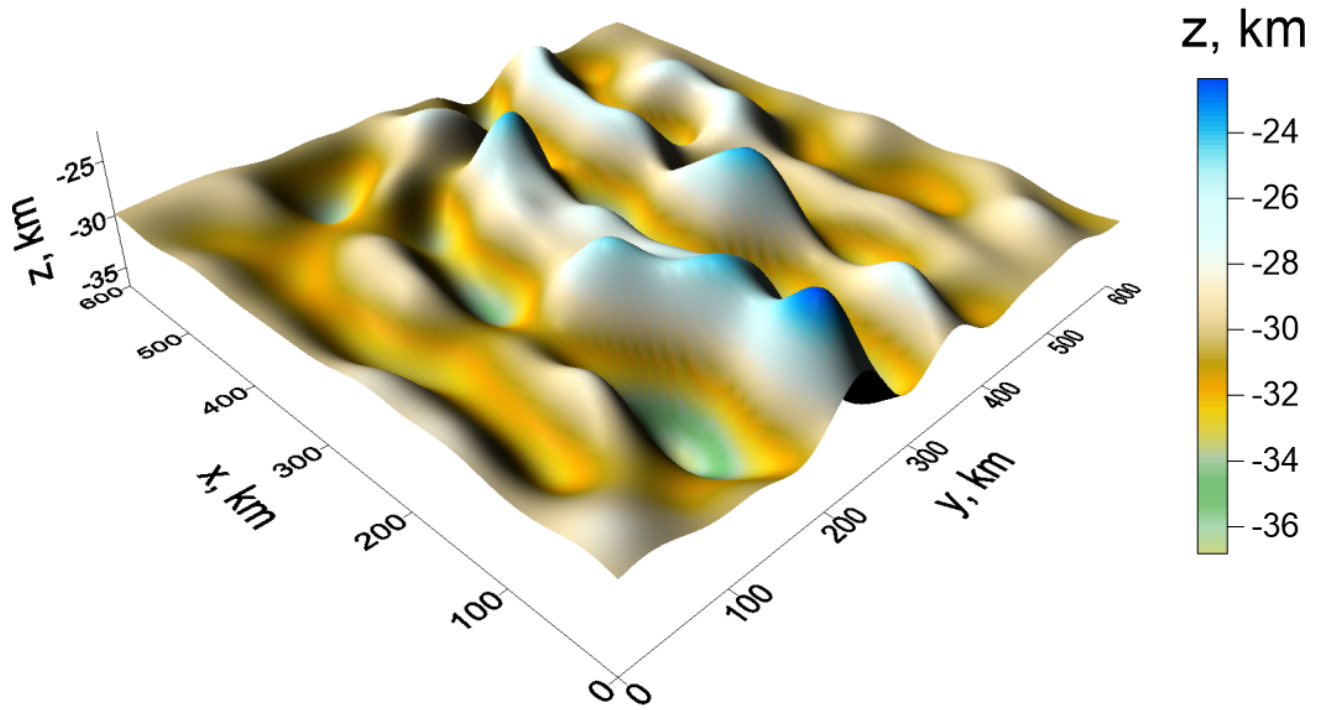


Рисунок 3.31 — Точное решение для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным

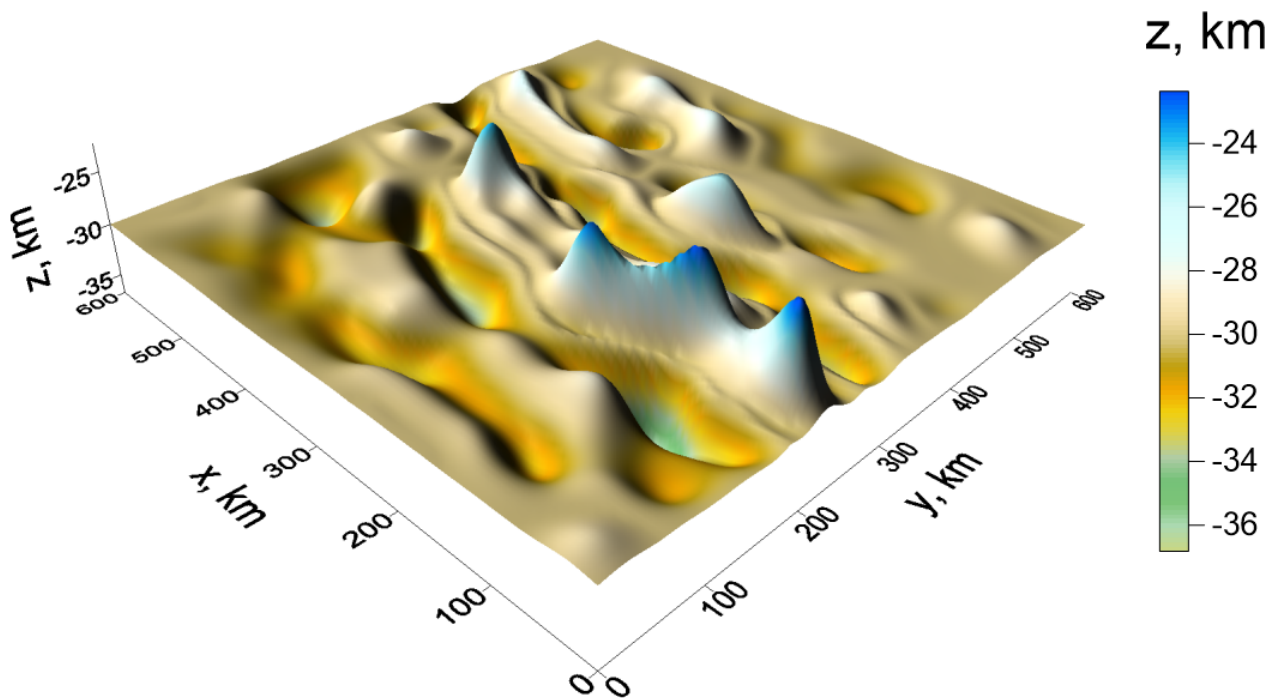


Рисунок 3.32 — Приближенное решение для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным

В таблице 7 приводится сравнение по времени счета для различного числа ускорителей для модифицированного метода сопряженных градиентов. А также приводится эффективность и ускорение.

Таблица 7 — Результаты численных экспериментов для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным

Число ускорителей	Время, мин	Ускорение, S	Эффективность, E
1	5.3	-	-
2	2.8	1.89	0.94
4	1.5	3.6	0.9
8	0.7	7.16	0.89

Вывод. Модифицированный метод сопряженных градиентов показал высокую эффективность распараллеливания. Метод является более экономичным по количеству итераций, количеству вычислительных операций на итерации и времени счета каждой итерации по сравнению с методом сопряженных градиентов. Эффективность модифицированного алгоритма составила 90% при использовании вычислительного узла с установленными 8 графическими ускорителями.

Заключение

Для решения структурной обратной задачи магнитометрии о нахождении поверхности раздела сред для случая произвольно направленных векторов намагниченности слоев разработан новый покомпонентный градиентный численный метод, учитывающий свойства математической модели — поправка на отклонение от вертикали вектора разности намагниченности слоев и использованием схемы аппроксимации интегрального оператора, более экономичной по сравнению с квадратурными формулами.

Для решения структурных обратных задач гравиметрии и магнитометрии о восстановлении поверхностей раздела сред разработаны экономичные по памяти модифицированные методы градиентного типа, использующие блочно-теплицевую структуру матриц производных конечномерных аппроксимаций интегральных операторов. В результате получено снижение пространственной сложности с $O(n^2)$ до $O(n)$.

Разработан комплекс параллельных алгоритмов и программ для решения структурных обратных задач гравиметрии и магнитометрии о восстановлении поверхностей раздела сред на графических ускорителях на основе описанных в работе методов. Комплекс программ протестирован на построенных модельных задачах, с использованием гравитационного поля.

Разработан веб-портал для удаленных вычислений, позволяющий запускать комплекс программ на многопроцессорных системах, добавлять новые программы и подключать новые вычислители. Нагрузочное тестирование показало возможность портала быстро обслуживать большое количество пользователей и заданий одновременно.

Рекомендации и дальнейшие перспективы разработки темы. В дальнейшем предполагается численное решение обратных задач гравиметрии и магнитометрии о нахождении поверхностей раздела с реальными данными на графических процессорах с помощью разработанных параллельных алгоритмов и программ. Веб-портал предполагается использовать для решения задач математической физики на многопроцессорных системах.

Словарь терминов

CPU : Central processing unit, центральный процессор — электронный блок либо интегральная схема, исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера

OpenMP : Open Multi-Processing, открытый стандарт для распараллеливания программ на языках Си, Си++ и Фортран

MPI : Message Passing Interface, программный интерфейс (API) для передачи информации, который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу

GPU : Graphics processing unit, графический ускоритель — отдельное устройство персонального компьютера, выполняющее графический рендеринг

GPGPU : General-purpose computing on graphics processing units, неспециализированные вычисления на графических процессорах

OpenCL : Open Computing Language, неспфреймворк для написания компьютерных программ, связанных с параллельными вычислениями на различных графических и центральных процессорах, а также FPGA

CUDA : Compute Unified Device Architecture, программно-аппаратная архитектура параллельных вычислений, которая позволяет существенно увеличить вычислительную производительность благодаря использованию графических процессоров фирмы Nvidia

СУБД : Система управления базами данных, совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных

SQL : Structured Query Language, декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных

LINQ : Language-Integrated Query представляет простой и удобный язык запросов к источнику данных. Аббревиатура LINQ обозначает целый набор технологий, создающих и использующих возможности интеграции запросов непосредственно в язык C#. Традиционно запросы к данным выражаются в виде простых строк без проверки типов при компиляции или поддержки

IntelliSense. Кроме того, разработчику приходится изучать различные языки запросов для каждого типа источников данных: баз данных SQL, XML-документов, различных веб-служб и т. д. Технологии LINQ превращают запросы в удобную языковую конструкцию, которая применяется аналогично классам, методам и событиям

IntelliSense : технология автодополнения Microsoft

XML : eXtensible Markup Language - «расширяемый язык разметки». Рекомендован Консорциумом Всемирной паутины (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов как программами, так и человеком, с акцентом на использование в Интернете

PostgreSQL : свободная объектно-реляционная система управления базами данных (СУБД). Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows

OAuth2 : протокол авторизации, предназначенный для организации доступа клиентских приложений к ресурсам, или данным учетных записей, пользователя на другом сервисе. В качестве клиентских приложений выступают веб-сервисы, мобильные и десктопные приложения

OpenID Connect : открытый стандарт децентрализованной системы аутентификации, предоставляющей пользователю возможность создать единую учётную запись для аутентификации на множестве не связанных друг с другом интернет-ресурсов, используя услуги третьих лиц. Базовой функцией OpenID является предоставление портативного, клиент-ориентированного, цифрового идентификатора для свободного и децентрализованного использования

Веб-портал : сайт в компьютерной сети, который предоставляет пользователю различные интерактивные интернет-сервисы, которые работают в рамках этого сайта

API : Application Programming Interface, описание способов, взаимодействия одной компьютерной программы на другую

WebAssembly : язык программирования низкого уровня для стековой виртуальной машины, спроектированный как портативная цель компиляции для высокоуровневых языков, таких как Си, C++, C#, Rust, Go

C# : объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270

Список литературы

1. *Нумеров Б. В.* Интерпретация гравитационных наблюдений в случае одной контактной поверхности // Докл. АН СССР. — 1930. — № 21. — С. 569—574.
2. *Малкин Н. Р.* О решении обратной магнитометрической задачи для случая одной контактной поверхности (случай пластообразно залегающих масс) // Докл. АН СССР. Сер. А. — 1931. — № 9. — С. 232—235.
3. *Булах Е. Г.* Обзор работ по методам минимизации в обратных задачах гравиметрии и магнитометрии // Геофизический журнал. — 1999. — Т. 1, № 4. — С. 5—19.
4. *Булах Е. Г., Марков М. Н., Бойко П. Д.* Математическое обеспечение автоматизированной системы интерпретации гравитационных аномалий. — Киев : Наук. думка, 1984. — 112 с.
5. *Кобрунов А. И.* К теории интерпретации данных гравиметрии для слоистых сред (равномерная оптимизация) // Изв. АН СССР. Сер. Физика Земли. — 1988. — № 8. — С. 33—44.
6. Технологии решения прямых и обратных задач 3D гравиметрии и магнитометрии / В. А. Кочнев [и др.] // Мат.32-й сессии Межд.научн.сем. им.Д.Г.Успенского «Вопросы теории и практики геологической интерпретации гравитационных, магнитных и электрических полей». Пермь. — 2005. — С. 134—137.
7. *Мартышко П. С., Ладовский И. В., Цидаев А. Г.* Построение региональных геофизических моделей на основе комплексной интерпретации гравитационных и сейсмических данных // Физика земли. — 2010. — № 11. — С. 23—35.
8. *Мартышко П. С., Пруткин И. Л.* О решении прямой и обратной задач магниторазведки // Геофизический журнал. — 1982. — Т. 4, № 6. — С. 39—49.
9. *Мартышко П. С., Цидаев А. Г.* Построение региональных геофизических моделей на основе комплексной интерпретации гравитационных и сейсмических данных // Материалы Международной конференции, посвященной 50летию Института геофизики УрО РАН. Екатеринбург: ИГФ УрО РАН. — 2008. — С. 167—170.

10. *Пруткин И. Л.* О решении трехмерной обратной задачи гравиметрии в классе контактных поверхностей методом локальных поправок // Известия АН СССР. Физика Земли. — 1986. — Т. 1. — С. 67—77.
11. *Страхов В. Н.* О решении некорректных задач магнито- и гравиметрии, представляемых интегральными уравнениями типа свертки // Известия АН СССР. Физика Земли. — 1967. — № 4. — С. 36—54.
12. *Страхов В. Н., Лапина М. И.* Монтажный метод решения обратной задачи гравиметрии // Доклады АН СССР. — 1976. — Т. 227, № 2. — С. 344—347.
13. *Канторович Л. В.* О методе наискорейшего спуска // Доклады Академии наук СССР. — 1947. — Т. 56. — С. 233—236.
14. *Иванов В. К.* О линейных некорректных задачах // Доклады АН СССР. — 1962. — Т. 145, № 2. — С. 270—272.
15. *Иванов В. К.* О некорректно поставленных задачах // Математический сборник. — 1963. — Т. 61(103), № 2. — С. 211—223.
16. *Лаврентьев М. М.* О некоторых некорректных задачах математической физики. — Новосибирск : Изд-во Сибирского отделения АН СССР, 1962. — 92 с.
17. *Тихонов А. Н.* О регуляризации некорректно поставленных задач // Доклады АН СССР. — 1963. — Т. 153, № 1. — С. 49—52.
18. *Тихонов А. Н.* О решении некорректно поставленных задач и методе регуляризации // Доклады АН СССР. — 1963. — Т. 151, № 3. — С. 501—504.
19. *Тихонов А. Н., Гласко В. Б.* Применение метода регуляризации в нелинейных задачах // Журнал вычислительной математики и математической физики. — 1965. — Т. 5, № 3. — С. 463—473.
20. *Бакушинский А. Б.* Регуляризирующий алгоритм на основе метода Ньютона — Канторовича для решения вариационных неравенств // Журнал вычислительной математики и математической физики. — 1976. — Т. 16, № 6. — С. 1397—1404.
21. *Бакушинский А. Б.* К проблеме сходимости итеративно-регуляризованного метода Гаусса-Ньютона // Журнал вычислительной математики и математической физики. — 1992. — Т. 32, № 9. — С. 1503—1509.
22. *Бакушинский А. Б., Гончарский А. В.* Итеративные методы решения некорректных задач. — Москва : Наука, 1989. — 128 с.
23. *Бакушинский А. Б., Поляк Б. Т.* О решении вариационных неравенств // Доклады Академии наук СССР. — 1974. — Т. 219, № 5. — С. 1038—1041.

24. *Васин В. В., Агеев А. Л.* Некорректные задачи с априорной информацией. — Уральская изд. фирма «Наука», 1993. — 263 с.
25. *Иванов В. К., Васин В. В., Танана В. П.* Приближенное решение операторных уравнений. — Москва : Наука, 1978. — 205 с.
26. Решение трехмерных обратных задач гравиметрии и магнитометрии для трехслойной среды / В. В. Васин [и др.] // Математическое моделирование. — 2003. — Т. 15, № 2. — С. 69—76.
27. *Васин В. В.* Проксимальный алгоритм с проектированием в задачах выпуклого программирования. — Уральск. научн. центр, Ин-т матем. и механ., 1982. — 47 с.
28. *Васин В. В.* Итерационные методы решения некорректных задач с априорной информацией в гильбертовых пространствах // Журнал вычислительной математики и математической физики. — 1988. — Т. 28, № 7. — С. 971—980.
29. *Агеев А. Л.* Регуляризация нелинейных операторных уравнений на классе функций ограниченной вариации // Журнал вычислительной математики и математической физики. — 1980. — Т. 20, № 4. — С. 819—826.
30. *Гончарский А. В., Леонов А. С., Ягола А. Г.* Обобщенный принцип невязки // Журнал вычислительной математики и математической физики. — 1973. — Т. 13, № 2. — С. 294—302.
31. *Тихонов А. Н., Леонов А. С., Ягола А. Г.* Нелинейные некорректные задачи. — Москва : Курс, 2017. — 400 с.
32. *Танана В. П.* О приближенном решении нелинейных операторных уравнений // Известия Челябинского научного центра. — 2003. — Т. 21, № 4. — С. 6—8.
33. *Танана В. П.* О сходимости регуляризованных решений нелинейных операторных уравнений // Сибирский журнал промышленной математики. — 2003. — Т. 6, № 3. — С. 119—133.
34. *Танана В. П.* О решении операторных уравнений первого рода с многозначными операторами и их приложение // Известия вузов. Математика. — 1977. — Т. 7. — С. 87—93.
35. *Танана В. П.* Методы решения операторных уравнений. — Москва : Наука, 1987.

36. *Танана В. П.* Об аппроксимации регуляризованного решения нелинейного уравнения // Сибирский математический журнал. — 1997. — Т. 2. — С. 416—423.
37. *Bakushinskiy A., Goncharsky A.* Ill-posed problems: theory and applications. — Springer Science & Business Media, 1994. — 258 p.
38. *Поляк Б. Т.* Метод сопряженных градиентов в задачах на экстремум // Журнал вычислительной математики и математической физики. — 1969. — Т. 9, № 4. — С. 807—821.
39. *Ортега Д.* Введение в параллельные и векторные методы решения линейных систем. — Москва : Мир, 1991. — 367 с.
40. *Powell M.* A hybrid method for nonlinear equations // Numerical methods for nonlinear algebraic equations. — 1970. — Vol. 7. — P. 87—114.
41. *Kelley C.* Iterative Methods for Linear and Nonlinear Equations. — Philadelphia: Siam, 1995.
42. *Schnabel R. B., Frank P. D.* Tensor Methods for Nonlinear Equations // SIAM Journal on Numerical Analysis. — 1983. — Vol. 21, no. 5. — P. 815—843.
43. *Landweber L.* An Iteration Formula for Fredholm Integral Equations of the First Kind // American Journal of Mathematics. — 1951. — Vol. 73, no. 3. — P. 615—624.
44. *Hanke M., Neubauer A., Scherzer O.* A convergence analysis of the Landweber iteration for nonlinear ill-posed problems // Numerische Mathematik. — 1995. — Vol. 72, no. 1. — P. 21—37.
45. *Neubauer A.* On Landweber iteration for nonlinear ill-posed problems in Hilbert scales // Numerische Mathematik. — 2000. — Vol. 85, no. 2. — P. 309—328.
46. *Neubauer A., Scherzer O.* A convergence rate result for a steepest descent method and a minimal error method for the solution of nonlinear ill-posed problems // Zeitschrift für Analysis und ihre Anwendungen. — 1995. — Vol. 14, no. 2. — P. 369—377.
47. *Scherzer O.* A convergence analysis of a method of steepest descent and a two-step algorithm for nonlinear ill-posed problems // Numerical functional analysis and optimization. — 1996. — Vol. 17. — P. 197—214.

48. *Blaschke B., Neubauer A., Scherzer O.* On convergence rates for the iteratively regularized Gauss-Newton method // IMA Journal of Numerical Analysis. — 1997. — Vol. 17, no. 3. — P. 421—436.
49. *Kaltenbacher B., Neubauer A., Ramm A. G.* Convergence rates of the continuous regularized Gauss—Newton method // Journal of Inverse and Ill-Posed Problems. — 1995. — Vol. 10, no. 3. — P. 261—280.
50. *Кокурин М. Ю.* О выпуклости функционала Тихонова и итеративного регуляризованных методах решения нерегулярных нелинейных операторных уравнений // Журнал вычислительной математики и математической физики. — 2010. — Т. 50, № 4. — С. 651—664.
51. *Кокурин М. Ю.* Об организации глобального поиска при реализации схемы Тихонова // Изв. вузов. Матем. — 2010. — № 12. — С. 30—31.
52. Теория и методы комплексной интерпретации геофизических данных / П. С. Мартышко [и др.]. — Екатеринбург : УрО РАН, 2016. — 94 с.
53. *Мартышко П. С., Ладовский И. В., Гемайдинов Д. В.* О выборе параметра регуляризации в задаче аналитического продолжения гравитационных полей (разделение аномалий от разноглубинных источников) // Геология и геофизика. — 2023. — Т. 64, № 1. — С. 112—122.
54. *Martyshko P. S., Fedorova N. V., Rublev A. L.* Numerical algorithms for structural magnetometry inverse problem solving // Russian Journal of Earth Sciences. — 2021.
55. *Akimova E. N., Vasin V. V.* Stable parallel algorithms for solving the inverse gravimetry and magnetometry problems // Intern. J. Engineering Modelling. — 2004. — Vol. 17, no. 1/2. — P. 13—19.
56. *Воеводин В. В., Воеводин В. В.* Параллельные вычисления. — Санкт-Петербург : БХВ-Петербург, 2002. — 608 с.
57. *Фаддеева В. Н., Фаддеев Д. К.* Параллельные вычисления в линейной алгебре // Кибернетика. — 1977. — № 3. — С. 28—40.
58. *Акимова Е. Н.* Параллельные алгоритмы решения обратных задач гравиметрии и магнитометрии на МВС-1000 // Вестник ННГУ. — 2009. — № 4. — С. 181—189.
59. *Акимова Е. Н.* Параллельные алгоритмы решения задач грави-магнитометрии и упругости на многопроцессорных системах с распределенной

- памятью : Дисс. д-ра физ.-мат. наук / Акимова Елена Николаевна. — ИММ УрО РАН, 2009.
60. *Акимова Е. Н., Гемайдинов Д. В.* Параллельные алгоритмы решения обратной задачи гравиметрии и организация удаленного взаимодействия между МВС-1000 и пользователем // Вычислительные методы и программирование. — 2008. — Т. 9. — С. 129—140.
 61. *Акимова Е. Н., Белоусов Д. В., Мислов В. Е.* Алгоритмы решения обратных геофизических задач на многопроцессорных вычислительных системах // Сибирский журнал вычислительной математики. — 2013. — Т. 16, № 2. — С. 107—121.
 62. Specialized web portal for solving problems on multiprocessor computing systems / E. N. Akimova [et al.] // CEUR Workshop Proceedings. — 2015. — Vol. 1513. — P. 123—129.
 63. *Tsidaev A.* .NET Library for Seamless Remote Execution of Supercomputing Software // CEUR-WS. — 2017. — Vol. 1990. — P. 79—83.
 64. *Kuklin E., Pravdin S.* A Web-based System for Launching Large Experiment Series on Supercomputers // CEUR-WS. — 2018. — Vol. 2281. — P. 136—145.
 65. *Erwin, Dietmar.* UNICORE—a grid computing environment // Concurrency and Computation: Practice and Experience. — 2002. — Vol. 14. — P. 1395—1410.
 66. A Web-Based Service Portal to Steer Numerical Simulations on High-Performance Computers / A. Grosch [et al.] // 8th European Medical and Biological Engineering Conference. — 2020. — Vol. 80. — P. 57—65.
 67. Облачный сервис решения задач химической кинетики с использованием параллельных вычислений / М. В. Сахибгареева [и др.] // Системы и средства информатики. — 2017. — Т. 27, № 1. — С. 155—166.
 68. FirecREST: a RESTful API to HPC systems / F. A. Cruz [et al.] // 2020 IEEE/ACM International Workshop on Interoperability of Supercomputing and Cloud Technologies (SuperCompCloud). — 2020. — P. 21—26.
 69. *Cholia S., Skinner D., Boverhof J.* NEWT: A RESTful service for building High Performance Computing web applications // Gateway Computing Environments Workshop (GCE). — 2010. — P. 1—11.

70. *Misilov V. E.* On solving the structural inverse magnetic problem of finding a contact surface in the case of arbitrary directed magnetization // EAGE Geoinformatics 2016: 15th International Conference on Geoinformatics – Theoretical and Applied Aspects. — 2016.
71. *Яновский Б. М.* *Земной магнетизм. Л: Изд-во ЛГУ 1.* Земной магнетизм. — Л : Изд-во ЛГУ, 1978. — 592 с.
72. *Шванк О. А., Люстих Е. Н.* Интерпретация гравитационных наблюдений. — Л. : Гостоптехиздат, 1947. — 400 с.
73. *Мартышко П. С., Пруткин И. Л.* Технология разделения источников гравитационного поля по глубине // Геофизический журнал. — 2003. — Т. 25, № 1. — С. 159—168.
74. *Akimova E. N., Misilov V. E.* A fast componentwise gradient method for solving structural inverse gravity problem // International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management, SGEM. — 2015. — Vol. 3, no. 1. — P. 775—781.
75. *Васин В. В.* Модифицированный метод наискорейшего спуска для нелинейных нерегулярных операторных уравнений // ДАН. — 2015. — Т. 462, № 3. — С. 264—267.
76. *Fletcher R., Reeves C. M.* Function minimization by conjugate gradients // The computer journal. — 1964. — Vol. 7, no. 2. — P. 149—154.
77. *Мартышко П. С., Акимова Е. Н., Мисилов В. Е.* О решении структурной обратной задачи гравиметрии модифицированными методами градиентного типа // Физика земли. — 2016. — № 5. — С. 82—86.
78. An Efficient Numerical Technique for Solving the Inverse Gravity Problem of Finding a Lateral Density / E. N. Akimova [et al.] // Applied Mathematics and Information Sciences. — 2016. — Vol. 10, no. 5. — P. 1681—1688.
79. Introduction to ASP.NET Core Blazor. Microsoft. 2021. [Электронный ресурс]. — URL: <https://docs.microsoft.com/en-us/aspnet/core/blazor> (дата обр. 29.10.2021).
80. Документация по ASP.NET. Microsoft. 2021. [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/aspnet/core> (дата обр. 29.10.2021).
81. Entity Framework Core. Microsoft. 2021. [Электронный ресурс]. — URL: <https://docs.microsoft.com/ru-ru/ef/core> (дата обр. 29.10.2021).

82. World Gravity Map. Commission for the Geological Map of the World / S. Bonvalot [et al.]. — Paris : Eds. BGI-CGMW-CNES-IRD, 2012.

Публикации автора по теме диссертации

83. Градиентные методы решения структурных обратных задач гравиметрии и магнитометрии на суперкомпьютере «Уран» / Е. Н. Акимова [и др.] // Вычислительные методы и программирование: Новые вычислительные технологии (электронный научный журнал). — 2015. — Т. 16, № 1. — С. 155—164.
84. *Akimova E. N., Misilov V. E., Tretyakov A. I.* Regularized methods for solving nonlinear inverse gravity problem // 15th EAGE International Conference on Geoinformatics – Theoretical and Applied Aspects. — 2016. — P. 1—5.
85. On solving the inverse structural magnetic problem for large grids on GPUs / E. N. Akimova [et al.] // AIP Conference Proceedings. — 2017. — Vol. 1863.
86. *Akimova E. N., Misilov V. E., Tretyakov A. I.* Optimized algorithms for solving structural inverse gravimetry and magnetometry problems on GPUs // Communications in Computer and Information Science. — 2017. — Vol. 753. — P. 144—155.
87. *Akimova E. N., Misilov V. E., Tretyakov A. I.* Modified Componentwise Gradient Method for Solving Structural Magnetic Inverse Problem // Communications in Computer and Information Science. — 2018. — Vol. 910. — P. 162—173.
88. *Akimova E. N., Misilov V. E., Tretyakov A. I.* Using Multicore and Graphics Processors to Solve the Structural Inverse Gravimetry Problem in a Two-Layer Medium by Means of α -Processes // Communications in Computer and Information Science. — 2019. — Vol. 1063. — P. 285—296.
89. *Акимова Е. Н., Мисилов В. Е., Третьяков А. И.* Экономичные алгоритмы решения обратных задач гравиметрии и магнитометрии на графических процессорах // Вычислительные методы и программирование: Новые вычислительные технологии (электронный научный журнал). — 2023. — Т. 24. — С. 368—385.
90. Градиентные методы решения структурных обратных задач гравиметрии и магнитометрии на суперкомпьютере «Уран» / Е. Н. Акимова [и др.] // Труды международной конференции «Параллельные вычислительные технологии (ПАВТ'2015)». — 2015. — С. 8—18.

91. *Акимова Е. Н., Мисилов В. Е., Третьяков А. И.* О решении прямой задачи нахождения второй производной ньютоновского потенциала // Девятые научные чтения Ю.П. Булашевича. Глубинное строение, геодинамика, тепловое поле Земли, интерпретация геофизических полей. — 2017. — С. 18—22.
92. *Акимова Е. Н., Третьяков А. И.* Решение нелинейных обратных задач гравиметрии и магнитометрии с использованием графических ускорителей // Сборник научных трудов 46-й сессии Международного научного семинара им. Д.Г. Успенского «Вопросы теории и практики геологической интерпретации гравитационных, магнитных и электрических полей». — 2019. — С. 16—21.
93. *Третьяков А. И.* Разработка комплекса параллельных программ решения обратных задач гравиметрии и магнитометрии для сеток большой размерности // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. — 2022. — Т. 11, № 1. — С. 57—78.
94. *Свидетельство о государственной регистрации программы для ЭВМ. Grav_1Surface_regCG_CUDA* / Е. Н. Акимова, В. Е. Мисилов, А. И. Третьяков ; Ф. государственное бюджетное учреждение науки Институт математики и механики им. Н. Н. Красовского Уральского отделения Российской академии наук. — № 2020663100 ; заявл. 05.10.2020 ; опубл. 22.10.2020, 2020661482 (Рос. Федерация).
95. *Свидетельство о государственной регистрации программы для ЭВМ. MagJz_1Surface_regCG_CUDA* / Е. Н. Акимова, В. Е. Мисилов, А. И. Третьяков ; Ф. государственное бюджетное учреждение науки Институт математики и механики им. Н. Н. Красовского Уральского отделения Российской академии наук. — № 2020662619 ; заявл. 05.10.2020 ; опубл. 16.10.2020, 2020661563 (Рос. Федерация).
96. *Свидетельство о государственной регистрации программы для ЭВМ. Web-Portal_Paral_Prog* / Е. Н. Акимова, Д. Гемайдинов, А. И. Третьяков ; Ф. государственное бюджетное учреждение науки Институт математики и механики им. Н. Н. Красовского Уральского отделения Российской академии наук. — № 2023683778 ; заявл. 10.11.2023 ; опубл. 14.11.2023, 2023684234 (Рос. Федерация).

Список рисунков

1.1	Модель двухслойной среды в задаче гравиметрии	18
1.2	Распределение индексов по сетке	19
1.3	Модель двухслойной среды в задаче магнитометрии	20
1.4	Модель многослойной среды в задаче гравиметрии	22
1.5	Модель многослойной среды в задаче магнитометрии	23
1.6	Модель двухслойной среды в задаче магнитометрии для случая произвольно направленной намагниченности	25
2.1	Расположение точки j , соответствующей компоненте правой части, на которую оказывает максимальное влияние компонента решения, соответствующая точке i	31
2.2	Структура матрицы производных интегрального оператора для случая двухслойной среды	41
2.3	Структура матрицы производных интегрального оператора для случая многослойной среды	42
2.4	Способ хранения блочно-теплицевой матрицы	42
3.1	Структура комплекса программ	46
3.2	Структура веб-портала	54
3.3	Структура базы данных	55
3.4	Список загруженных программ	56
3.5	Форма создания новой программы	56
3.6	Список запущенных заданий	57
3.7	Страница настроек веб-портала	57
3.8	Схема взаимодействия пользователя с веб-порталом	58
3.9	Архитектура сервиса Яндекс.Танк	61
3.10	Жизненный цикл нагрузочного теста	62
3.11	Схема взаимодействия модулей системы Яндекс.Танк	63
3.12	Графики с результатами нагрузочного теста	65
3.13	Модельное гравитационное поле для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным	66
3.14	Точное решение для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным	67

3.15	Приближенное решение для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным . . .	67
3.16	Ускорение для метода ММСГ для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным	69
3.17	Эффективность для метода ММСГ для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным	70
3.18	Точное решение для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности	71
3.19	Модельное гравитационное поле для угла 0 для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности . .	72
3.20	Модельное гравитационное поле для угла 15 для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности . .	73
3.21	Модельное гравитационное поле для угла 30 для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности . .	74
3.22	Модельное гравитационное поле для угла 45 для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности . .	75
3.23	Модельное гравитационное поле для угла 60 для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности . .	76
3.24	Приближенные решения для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности	77
3.25	Изменение невязки по правой части по итерациям а) угол 70°; б) угол 80°	78
3.26	Изменение погрешности решения по итерациям а) угол 70°; б) угол 80°	78
3.27	Суммарное гравитационное поле для задачи 3.5.3 о восстановлении трех поверхностей раздела по гравитационным данным	79

3.28	Модельные поверхности S_1, S_2, S_3 для задачи 3.5.3 о восстановлении трех поверхностей раздела по гравитационным данным	79
3.29	Восстановленные поверхности S_1, S_2, S_3 для задачи 3.5.3 о восстановлении трех поверхностей раздела по гравитационным данным	80
3.30	Модельное гравитационное поле для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным	82
3.31	Точное решение для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным	83
3.32	Приближенное решение для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным	83

Список таблиц

1	Результаты численных экспериментов для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным	68
2	Результаты численных экспериментов для задачи 3.5.1 о восстановлении одной поверхности раздела по зашумленным гравитационным данным	68
3	Сравнение методов для задачи 3.5.2 о восстановлении поверхности раздела для случая произвольно направленного вектора намагниченности	72
4	Время счета МПГМ для различного числа GPU для задачи 3.5.2 о восстановлении одной поверхности раздела по магнитным данным для случая произвольно направленного вектора намагниченности	75
5	Результаты численных экспериментов для задачи 3.5.3 о восстановлении трех поверхностей раздела по гравитационным данным	81
6	Результаты численных экспериментов для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным	82
7	Результаты численных экспериментов для задачи 3.5.4 о восстановлении одной поверхности раздела по гравитационным данным	84

Приложение А

Листинги программного кода

Листинг А.1 программы решения обратных задач гравиметрии

Листинг А.1 Листинг из внешнего файла

```

#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <fstream>
5 #include <string>
#include <iomanip>
#include <cmath>
#include <time.h>

10 using namespace std;

#define Z0 6
#define eps 0.005
#define grav 6.673
15 #define gamma 0.15

#define dx 2
#define dy 2
20 #define Nx 128
#define Ny 128

#define H 6
#define dG 0.1
25 #define alpha 0.01

#define MAX_ITER 10000

#define coef grav*dG*dx*dy
30 double f[Nx*Ny], f1[Nx*Ny], x[Nx], y[Ny], z[Nx*Ny], z_exact[Nx*Ny]; //
    , Ank3[Nx*Ny][Nx*Ny], B[Nx*Ny][Nx*Ny];
double norm;
int size, rank2, h, hx;

```

```

35 int ndevices;
   struct config_t
   {
   double *Adev,*fdev,*zdev,*Xdev,*Ydev,*Sdev,*Sprevdev,*pdev;
   float *b1dev,*b2dev,*p1dev,*p21dev,*p22dev;
40 };
   config_t *configs; //[ndevices];

   __global__ void gradient_step1_new_kernel(double *x, double *y,
   double *f, double *S, double *S_prev, double *An, float *b1,
   float *b2, double *z, int  idevice, int ndevicesDev)
   {
45   int l=threadIdx.x + blockIdx.x*blockDim.x+(Nx/ndevicesDev)*
   idevice;
   int k=threadIdx.y + (blockIdx.y)*blockDim.y;
   double sum,sum1,p1,p2,p3;

   sum=0;
50   sum1=0;
   for(int i=0;i<Ny;i++)
   {
   for(int j=0;j<Nx;j++)
   {
55     p3=(x[l]-x[j])*(x[l]-x[j])+(y[k]-y[i])*(y[k]-y[i]);
     p1=1/sqrt(p3+z[j*Ny+i]*z[j*Ny+i]);
     p2=1/sqrt(p3+H*H);
     sum+=((p1-p2));
     p3=(x[l]-x[j])*(x[l]-x[j])+(y[k]-y[i])*(y[k]-y[i])+z[j*Ny+
60     i]*z[j*Ny+i];
     p1=z[j*Ny+i]/sqrt(p3*p3*p3);
     //sum1 -= (grav*dG*p1*dx*dy);
     sum1 -= (p1*coef);
     //Ank[l*Ny+k][j*Ny+i]=sum1;
   }
65 }

   int i=l*Ny+k;
   sum=sum*coef-f[i];
   sum1=sum1*coef;
70 An[i]=sum1;

   S[i]=sum1*(sum)+alpha*(z[i]-Z0);

```

```

75  //bk up
    double b1_ = S[i]*(S[i]-S_prev[i]);
    atomicAdd(&b1[0],b1_);
    double b2_ = S_prev[i]*S_prev[i];
    atomicAdd(&b2[0],b2_);
80  }

__global__ void gradient_step2_kernel(double *S, double *An,
    double *p, float *p1, float *p21, float *p22, double bk,int
    idevice,int ndevicesDev)
{
85  //int i=threadIdx.x + (blockIdx.x+idevice*300)*blockDim.x;
    int i=threadIdx.x + (blockIdx.x)*blockDim.x+idevice*(Nx*Ny/
        ndevicesDev);

    p[i]=S[i]+bk*p[i];

90  double p1_ = p[i]*S[i];
    atomicAdd(&p1[0],p1_);
    double p21_ = An[i]*An[i]*p[i]*p[i];
    atomicAdd(&p21[0],p21_);
    double p22_ = p[i]*p[i];
95  atomicAdd(&p22[0],p22_);

    //z[i]-=gamma*s1[0]*S[i]/s2[0];
}

100 __global__ void gradient_step3_kernel(double *p, double *z,
    double coefdev, int idevice, int ndevicesDev)
{
    int i=threadIdx.x + (blockIdx.x)*blockDim.x+idevice*(Nx*Ny/
        ndevicesDev);

    z[i]-=coefdev*p[i];
105 }

void WriteOutputFile()
{
    ofstream of1("output_multi.dat");
110  for(int i=0;i<Nx;i++)

```

```

    {
        for(int j=0;j<Ny;j++)
        {
            of1<<(x[i])<<" "<<(y[j])<<" "<<(-z[i*(Ny)+j])<<endl;
115        }
    }
}

void err(cudaError_t i)
120 {
    if(i!=0){
        printf("Error: %i\t%s\n",i,cudaGetErrorString(i));
        exit(-1);
    }
125 }

double calcNorm()
{
    double res=0,res1=0;
130     for(int i=0;i<Nx*Ny;i++)
    {
        res+=(-z_exact[i]-z[i])*(-z_exact[i]-z[i]);
        res1+=z_exact[i]*z_exact[i];
    }
135     return sqrt(res/res1);
}

void gradient()
{
140 //cout<<"OK\n";
    dim3 threads=dim3(16/ndevices,32,1);//2
    //cout<<"OK\n";
    dim3 blocks=dim3(8,4,1);
    dim3 threads2=dim3(32,1,1);
145 dim3 blocks2=dim3((512)/ndevices,1,1);//2
    float zero[1];
    zero[0]=0;
    float tmp;
    float b1,b2,p1,p21,p22;
150 int nb=Nx*Ny*sizeof(double);
    int index=(Nx*Ny/ndevices);//*idevice;
    //cout<<"OK\n";
    for(int i=0;i<50;i++)

```

```

{
155 b1=0;b2=0;
    p1=0;p21=0;p22=0;
    for(int idevice=0;idevice<ndevices;idevice++)
    {
        config_t* config=configs+idevice;
160 // cout<<"OK1\n";
        err(cudaSetDevice(idevice));
        //cout<<"OK2\n";

        double* tmp = config->Sprevdev;
165 config->Sprevdev = config->Sdev;
        config->Sdev = tmp;

        err(cudaMemcpy(config->zdev,z,nb,cudaMemcpyHostToDevice));
        err(cudaMemcpy(config->b1dev,zero,sizeof(float),
cudaMemcpyHostToDevice));
170 err(cudaMemcpy(config->b2dev,zero,sizeof(float),
cudaMemcpyHostToDevice));
        //calcAkernel<<<blocks,threads>>>(config->Adev,config->
Adev,config->Xdev,config->Ydev,config->zdev,idevice,ndevices
);
        //cout<<"OK3\n";
        gradient_step1_new_kernel<<<blocks,threads>>>
        (config->Xdev,config->Ydev,config->fdev,config->Sdev,
config->Sprevdev,config->Adev,config->b1dev,config->b2dev,
config->zdev,idevice,ndevices);
175 //cout<<"OK4\n";
    }

    for(int idevice=0;idevice<ndevices;idevice++)
    {
180 //cout<<"OK5\n";
        config_t* config=configs+idevice;
        err(cudaSetDevice(idevice));

        err(cudaMemcpy(&tmp,config->b1dev,sizeof(float),
185 cudaMemcpyDeviceToHost));
        b1+=tmp;
        err(cudaMemcpy(&tmp,config->b2dev,sizeof(float),
cudaMemcpyDeviceToHost));
        b2+=tmp;

```

```

190     }
        double bk = max(b1/b2,0.0f);
        if(i==0)
        {
            bk=0;
195     }

        for(int idevice=0;idevice<ndevices;idevice++)
        {
            //cout<<"OK5\n";
200         config_t* config=configs+idevice;
            err(cudaSetDevice(idevice));

            err(cudaMemcpy(config->p1dev,zero,sizeof(float),
cudaMemcpyHostToDevice));
            err(cudaMemcpy(config->p21dev,zero,sizeof(float),
cudaMemcpyHostToDevice));
205         err(cudaMemcpy(config->p22dev,zero,sizeof(float),
cudaMemcpyHostToDevice));
            gradient_step2_kernel<<<blocks2, threads2>>> (config->Sdev
,config->Adev,config->pdev,config->p1dev,config->p21dev,
config->p22dev,bk,idevice,ndevices);
        }

210     for(int idevice=0;idevice<ndevices;idevice++)
        {
            //cout<<"OK5\n";
            config_t* config=configs+idevice;
            err(cudaSetDevice(idevice));
215

            err(cudaMemcpy(&tmp,config->p1dev,sizeof(float),
cudaMemcpyDeviceToHost));
            p1+=tmp;
220         err(cudaMemcpy(&tmp,config->p21dev,sizeof(float),
cudaMemcpyDeviceToHost));
            p21+=tmp;
            err(cudaMemcpy(&tmp,config->p22dev,sizeof(float),
cudaMemcpyDeviceToHost));
            p22+=tmp;

```



```

225     }
        double coefdev = gamma*p1/(p21+alpha*p22);

230

        for(int  idevice=0;idevice<ndevices;idevice++)
        {
            config_t* config=configs+idevice;
235            err(cudaSetDevice(idevice));

            gradient_step3_kernel<<<blocks2, threads2>>> (config->pdev
,config->zdev,coefdev,idevice,ndevices);
            }
        for(int  idevice=0;idevice<ndevices;idevice++)
240        {
            config_t* config=configs+idevice;
            err(cudaSetDevice(idevice));
            //cout<<"OK8\n";
            err(cudaDeviceSynchronize());
245            //cout<<"OK\n";
            err(cudaMemcpy(&z[index*idevice],&config->zdev[index*
idevice],(int)(nb/ndevices),cudaMemcpyDeviceToHost));
            //cout<<"OK9\n";
            //if(i%50 == 0)cout<<"Index: "<<index*idevice<<"\t Size:
" <<(int)(nb/ndevices)<<"\t All: " <<nb<<endl;
250        }

        if(i%5 == 0)
        {
            cout<<"Iteration: " <<i<<" - " <<calcNorm()<<endl;
255            if(calcNorm()<eps)break;
        }
    }

    /*for(int  idevice=0;idevice<ndevices;idevice++)
260        {
            config_t* config=configs+idevice;
            err(cudaSetDevice(idevice));

```

```

    err(cudaMemcpy(z, config->zdev, nb, cudaMemcpyDeviceToHost));
265 }
    /*
    WriteOutputFile();
}

270 void cudaInit()
{
    int nb=Nx*Ny*sizeof(double);
    int nx=Nx*sizeof(double);
275 int ny=Ny*sizeof(double);
    //int ndevices=1;

    for(int  idevice=0;idevice<ndevices;idevice++)
280 {
        config_t* config=configs+idevice;
        err(cudaSetDevice(idevice));

        //err(cudaMemcpyToSymbol(&ndevicesDev,&ndevices,sizeof(int),0,
        //cudaMemcpyHostToDevice));
285

        //err(cudaMalloc((void**)&config->Adev,nb));
        //err(cudaMalloc((void**)&config->Adev,nb));
        err(cudaMalloc((void*)&config->Sdev,nb));
        err(cudaMalloc((void*)&config->Sprevdev,nb));
290 err(cudaMalloc((void*)&config->Adev,nb));
        err(cudaMalloc((void*)&config->pdev,nb));
        err(cudaMalloc((void*)&config->fdev,nb));
        err(cudaMalloc((void*)&config->zdev,nb));
        err(cudaMalloc((void*)&config->Xdev,nx));
295 err(cudaMalloc((void*)&config->Ydev,ny));
        err(cudaMalloc((void*)&config->b1dev,sizeof(float)));
        err(cudaMalloc((void*)&config->b2dev,sizeof(float)));
        err(cudaMalloc((void*)&config->p1dev,sizeof(float)));
        err(cudaMalloc((void*)&config->p21dev,sizeof(float)));
300 err(cudaMalloc((void*)&config->p22dev,sizeof(float)));

        err(cudaMemcpy(config->fdev,f,nb,cudaMemcpyHostToDevice));
        err(cudaMemcpy(config->zdev,z,nb,cudaMemcpyHostToDevice));
        err(cudaMemcpy(config->Xdev,x,nx,cudaMemcpyHostToDevice));
305 err(cudaMemcpy(config->Ydev,y,ny,cudaMemcpyHostToDevice));

```

```
    }  
}  
  
void init(int argc, char* argv[])  
310 {  
    if(argc==2)  
    {  
        ndevices=atoi(argv[1]);  
    }  
315 else  
        ndevices=1;  
  
    configs=new config_t[ndevices];  
320  
  
    string fname="128x128_H6_ds0.1_field_clean.dat";  
    string fname2="128x128_H6_surface_original.dat";  
  
325    ifstream if1(fname.c_str());  
    ifstream if2(fname2.c_str());  
  
    double a;  
    for(int i=0;i<Ny;i++)  
330 {  
        for(int j=0;j<Nx;j++)  
        {  
            z[i*(Nx)+j]=Z0;  
            if1>>x[j]>>y[i]>>f[j*(Ny)+i];  
335            if2>>a>>a>>z_exact[j*(Ny)+i];  
        }  
    }  
  
340    cudaInit();  
}  
  
void finalize()  
345 {  
    for(int ideoice=0;ideoice<ndevices;ideoice++)  
    {  
        config_t* config=configs+ideoice;
```

```

    err(cudaSetDevice(idevice));
350 //err(cudaFree(config->Adev));
    //err(cudaFree(config->Adev));
    err(cudaFree(config->Sdev));
    err(cudaFree(config->fdev));
    err(cudaFree(config->zdev));
355 err(cudaFree(config->Xdev));
    err(cudaFree(config->Ydev));
    err(cudaFree(config->b1dev));
    err(cudaFree(config->b2dev));
    err(cudaFree(config->p1dev));
360 err(cudaFree(config->p21dev));
    err(cudaFree(config->p22dev));
    }
    delete [] configs;
}
365
int main(int argc, char* argv[])
{
    init(argc, argv);
    cout<<"GPU: " <<ndevices<<endl;
370 clock_t t0=clock();
    gradient();
    double t1=double(clock()-t0)/CLOCKS_PER_SEC;
    cout<<"time = " <<t1<<endl;
    finalize();
375 return 0;
}

```

Листинг А.2 программы решения обратных задач магнитометрии

Листинг А.2 Листинг из внешнего файла

```

#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <fstream>
5 #include <string>
#include <iomanip>
#include <cmath>
#include <time.h>

10 using namespace std;

#define Z0 10

```

```

#define eps 0.005
#define grav 0.0795774715459477
15 #define gamma 1

#define dx 1.559055118
#define dy 1.559055118
20 #define Nx 128
#define Ny 128

#define H 10
#define dG -1
25 #define alpha 0

#define MAX_ITER 100

#define coef grav*dG*dx*dy
30
double f[Nx*Ny],f1[Nx*Ny],x[Nx],y[Ny],z[Nx*Ny],z_exact[Nx*Ny];//
    ,Ank3[Nx*Ny][Nx*Ny],B[Nx*Ny][Nx*Ny];
double norm2;
int size,rank2,h,hx;

35 int ndevices;
struct config_t
{
double *Adev,*fdev,*zdev,*Xdev,*Ydev,*Sdev,*Sprevdev,*pdev;
float *b1dev,*b2dev,*p1dev,*p21dev,*p22dev;
40 };
config_t *configs;//[ndevices];

__global__ void gradient_step1_new_kernel(double *x, double *y,
double *f, double *S, double *S_prev, double *An, float *b1,
float *b2,double *z,int idevice,int ndevicesDev)
{
45 int l=threadIdx.x + blockIdx.x*blockDim.x+(Nx/ndevicesDev)*
idevice;
int k=threadIdx.y + (blockIdx.y)*blockDim.y;
double sum,sum1,p1,p2,p3;

sum=0;
50 sum1=0;
for(int i=0;i<Ny;i++)

```

```

{
  for(int j=0;j<Nx;j++)
  {
55     p3=(x[l]-x[j])*(x[l]-x[j])+(y[k]-y[i])*(y[k]-y[i]);
        p1=1/pow(p3+z[j*Ny+i]*z[j*Ny+i],1.5);
        p2=1/sqrt(p3+H*H);
        sum+=((p1*z[j*Ny+i]-p2*p2*p2*H));
        p1=(p1-p1*p1*p1*p1*p1*3*pow(z[j*Ny+i],2));
60     sum1-=(p1*coef);
        }
    }

    int i=l*Ny+k;
65     sum=sum*coef-f[i];
        //sum1=sum1*coef;
        An[i]=sum1;

    S[i]=sum1*(sum)+alpha*(z[i]-Z0);
70

    //bk up
    double b1_ = S[i]*(S[i]-S_prev[i]);
    //double b1_ = sum;
75     atomicAdd(&b1[0],b1_);
    double b2_ = S_prev[i]*S_prev[i];
    atomicAdd(&b2[0],b2_);
}
80

__global__ void gradient_step2_kernel(double *S, double *An,
    double *p, float *p1, float *p21, float *p22, double bk,int
    idevice,int ndevicesDev)
{
    //int i=threadIdx.x + (blockIdx.x+idevice*300)*blockDim.x;
    int i=threadIdx.x + (blockIdx.x)*blockDim.x+idevice*(Nx*Ny/
    ndevicesDev);
85

    p[i]=S[i]+bk*p[i];

    double p1_ = p[i]*S[i];
    atomicAdd(&p1[0],p1_);
90     double p21_ = An[i]*An[i]*p[i]*p[i];
    atomicAdd(&p21[0],p21_);

```

```

double p22_ = p[i]*p[i];
atomicAdd(&p22[0],p22_);

95 //z[i]-=gamma*s1[0]*S[i]/s2[0];
}

__global__ void gradient_step3_kernel(double *p, double *z,
double coefdev, int idevice, int ndevicesDev)
{
100 int i=threadIdx.x + (blockIdx.x)*blockDim.x+idevice*(Nx*Ny/
ndevicesDev);

z[i]-=coefdev*p[i];
}

105 void WriteOutputFile()
{
ofstream of1("output_multi.dat");
for(int i=0;i<Nx;i++)
{
110 for(int j=0;j<Ny;j++)
{
of1<<(x[i])<<" "<<(y[j])<<" "<<(-z[i*(Ny)+j])<<endl;
}
}
115 }

void err(cudaError_t i)
{
if(i!=0){
120 printf("Error: %i\t%s\n",i,cudaGetErrorString(i));
exit(-1);
}
}

125 double calcNorm()
{
double res=0,res1=0;
for(int i=0;i<Nx*Ny;i++)
{
130 res+=(-z_exact[i]-z[i])*(-z_exact[i]-z[i]);
res1+=z_exact[i]*z_exact[i];
}
}

```

```

    return sqrt(res/res1);
}
135
void gradient()
{
    //cout<<"OK\n";
    dim3 threads=dim3(16/ndevices,32,1);//2
140    //cout<<"OK\n";
    dim3 blocks=dim3(8,4,1);
    dim3 threads2=dim3(32,1,1);
    dim3 blocks2=dim3((512)/ndevices,1,1);//2
    float zero[1];
145    zero[0]=0;
    float tmp;
    float b1,b2,p1,p21,p22;
    int nb=Nx*Ny*sizeof(double);
    int index=(Nx*Ny/ndevices);//idevice;
150    //cout<<"OK\n";
    for(int i=0;i<MAX_ITER;i++)
    {
        b1=0;b2=0;
        p1=0;p21=0;p22=0;
155        for(int idevice=0;idevice<ndevices;idevice++)
        {
            config_t* config=configs+idevice;
            // cout<<"OK1\n";
            err(cudaSetDevice(idevice));
160            //cout<<"OK2\n";

            double* tmp = config->Sprevdev;
            config->Sprevdev = config->Sdev;
            config->Sdev = tmp;
165

            err(cudaMemcpy(config->zdev,z,nb,cudaMemcpyHostToDevice));
            err(cudaMemcpy(config->b1dev,zero,sizeof(float),
            cudaMemcpyHostToDevice));
            err(cudaMemcpy(config->b2dev,zero,sizeof(float),
            cudaMemcpyHostToDevice));
            //calcAkernel<<<blocks,threads>>>(config->Adev,config->
            Adev,config->Xdev,config->Ydev,config->zdev,idevice,ndevices
            );
170            //cout<<"OK3\n";
            gradient_step1_new_kernel<<<blocks,threads>>>

```



```

    (config->Xdev, config->Ydev, config->fdev, config->Sdev,
    config->Sprevdev, config->Adev, config->b1dev, config->b2dev,
    config->zdev, idevice, ndevices);
    //cout<<"OK4\n";
}
175
for(int idevice=0;idevice<ndevices;idevice++)
{
    //cout<<"OK5\n";
    config_t* config=configs+idevice;
180
    err(cudaSetDevice(idevice));

    err(cudaMemcpy(&tmp, config->b1dev, sizeof(float),
    cudaMemcpyDeviceToHost));
    b1+=tmp;
185
    err(cudaMemcpy(&tmp, config->b2dev, sizeof(float),
    cudaMemcpyDeviceToHost));
    b2+=tmp;
    //err(cudaMemcpy(&f1[index*idevice], &config->Sdev[index*
    idevice], (int)(nb/ndevices), cudaMemcpyDeviceToHost));
}
double bk = max(b1/b2, 0.0f);
190
if(i==0)
{
    bk=0;
}

195
for(int idevice=0;idevice<ndevices;idevice++)
{
    //cout<<"OK5\n";
    config_t* config=configs+idevice;
    err(cudaSetDevice(idevice));

200
    err(cudaMemcpy(config->p1dev, zero, sizeof(float),
    cudaMemcpyHostToDevice));
    err(cudaMemcpy(config->p21dev, zero, sizeof(float),
    cudaMemcpyHostToDevice));
    err(cudaMemcpy(config->p22dev, zero, sizeof(float),
    cudaMemcpyHostToDevice));
    gradient_step2_kernel<<<blocks2, threads2>>> (config->Sdev
    , config->Adev, config->pdev, config->p1dev, config->p21dev,
    config->p22dev, bk, idevice, ndevices);
}

```

```
205     }

    for(int  idevice=0;idevice<ndevices;idevice++)
    {
210     //cout<<"OK5\n";
        config_t*  config=configs+idevice;
        err(cudaSetDevice(idevice));

215
        err(cudaMemcpy(&tmp,config->p1dev,sizeof(float),
cudaMemcpyDeviceToHost));
        p1+=tmp;
        err(cudaMemcpy(&tmp,config->p21dev,sizeof(float),
cudaMemcpyDeviceToHost));
        p21+=tmp;
220        err(cudaMemcpy(&tmp,config->p22dev,sizeof(float),
cudaMemcpyDeviceToHost));
        p22+=tmp;

    }
    double  coefdev = gamma*p1/(p21+alpha*p22);
225    //cout<<f1[10]<<endl;

230

    for(int  idevice=0;idevice<ndevices;idevice++)
    {
        config_t*  config=configs+idevice;
        err(cudaSetDevice(idevice));

235
        gradient_step3_kernel<<<blocks2, threads2>>> (config->pdev
,config->zdev,coefdev,idevice,ndevices);
    }
    for(int  idevice=0;idevice<ndevices;idevice++)
    {
240        config_t*  config=configs+idevice;
        err(cudaSetDevice(idevice));
        //cout<<"OK8\n";
        err(cudaDeviceSynchronize());
```

```

    //cout<<"OK\n";
245   err(cudaMemcpy(&z[index*idevice],&config->zdev[index*
idevice],(int)(nb/ndevices),cudaMemcpyDeviceToHost));
    //cout<<"OK9\n";
    //if(i%50 == 0)cout<<"Index: "<<index*idevice<<"\t Size:
"<<(int)(nb/ndevices)<<"\t All: "<<nb<<endl;
    }

250
    if(i%5 == 0)
    {
        cout<<"Iteration: "<<i<<" - "<<calcNorm()<<endl;
        if(calcNorm()<eps)break;
255    }
    }

    /*for(int idevice=0;idevice<ndevices;idevice++)
    {
260        config_t* config=configs+idevice;
        err(cudaSetDevice(idevice));

        err(cudaMemcpy(z,config->zdev,nb,cudaMemcpyDeviceToHost));
    }
265 */
    WriteOutputFile();
}

270 void cudaInit()
{
    int nb=Nx*Ny*sizeof(double);
    int nx=Nx*sizeof(double);
    int ny=Ny*sizeof(double);
275 //int ndevices=1;

    for(int idevice=0;idevice<ndevices;idevice++)
    {
280        config_t* config=configs+idevice;
        err(cudaSetDevice(idevice));

        //err(cudaMemcpyToSymbol(&ndevicesDev,&ndevices,sizeof(int),0,
        cudaMemcpyHostToDevice));

```

```

285 //err(cudaMalloc((void**)&config->Adev,nb));
//err(cudaMalloc((void**)&config->Adev,nb));
err(cudaMalloc((void**)&config->Sdev,nb));
err(cudaMalloc((void**)&config->Sprevdev,nb));
err(cudaMalloc((void**)&config->Adev,nb));
290 err(cudaMalloc((void**)&config->pdev,nb));
err(cudaMalloc((void**)&config->fdev,nb));
err(cudaMalloc((void**)&config->zdev,nb));
err(cudaMalloc((void**)&config->Xdev,nx));
err(cudaMalloc((void**)&config->Ydev,ny));
295 err(cudaMalloc((void**)&config->b1dev,sizeof(float));
err(cudaMalloc((void**)&config->b2dev,sizeof(float));
err(cudaMalloc((void**)&config->p1dev,sizeof(float));
err(cudaMalloc((void**)&config->p21dev,sizeof(float));
err(cudaMalloc((void**)&config->p22dev,sizeof(float));
300
err(cudaMemset(config->pdev,0,nb));
err(cudaMemset(config->Sdev,0,nb));
err(cudaMemset(config->Sprevdev,0,nb));
err(cudaMemcpy(config->fdev,f,nb,cudaMemcpyHostToDevice));
305 err(cudaMemcpy(config->zdev,z,nb,cudaMemcpyHostToDevice));
err(cudaMemcpy(config->Xdev,x,nx,cudaMemcpyHostToDevice));
err(cudaMemcpy(config->Ydev,y,ny,cudaMemcpyHostToDevice));
}
}
310
void init(int argc, char* argv[])
{
    if(argc==2)
    {
315     ndevices=atoi(argv[1]);
    }
    else
        ndevices=1;
320
    configs=new config_t[ndevices];

    string fname="field_cube_vertical_128.dat";
325 string fname2="orig128.dat";

```

```

ifstream if1(fname.c_str());
ifstream if2(fname2.c_str());

330 double a;
for(int i=0;i<Ny;i++)
{
    for(int j=0;j<Nx;j++)
    {
335         z[i*(Nx)+j]=Z0;
        if1>>x[j]>>y[i]>>f[j*(Ny)+i];
        if2>>a>>a>>z_exact[j*(Ny)+i];
        //cout<<f[j*(Ny)+i]<<endl;
    }
340 }

    cudaInit();
345 }

void finalize()
{
for(int idevice=0;idevice<ndevices;idevice++)
350 {
    config_t* config=configs+idevice;
    err(cudaSetDevice(idevice));
    //err(cudaFree(config->Adev));
    //err(cudaFree(config->Adev));
355 err(cudaFree(config->Sdev));
    err(cudaFree(config->fdev));
    err(cudaFree(config->zdev));
    err(cudaFree(config->Xdev));
    err(cudaFree(config->Ydev));
360 err(cudaFree(config->b1dev));
    err(cudaFree(config->b2dev));
    err(cudaFree(config->p1dev));
    err(cudaFree(config->p21dev));
    err(cudaFree(config->p22dev));
365 }
    delete [] configs;
}

int main(int argc, char* argv[])

```

```
370 {  
    init(argc,argv);  
    cout<<"GPU: " << ndevices << endl;  
    clock_t t0=clock();  
    gradient();  
375 double t1=double(clock()-t0)/CLOCKS_PER_SEC;  
    cout<<"time = " << t1 << endl;  
    finalize();  
    return 0;  
}
```