Федеральное государственное бюджетное учреждение науки
Институт вычислительной математики РАН

На правах рукописи

Чернышенко Алексей Юрьевич

Технология построения адаптируемых многогранных сеток и численное решение эллиптических уравнений 2-го порядка в трехмерных областях и на поверхностях

05.13.18 – Математическое моделирование,
численные методы и комплексы программ

ДИССЕРТАЦИЯ
на соискание учёной степени
кандидата физико-математических наук

Научный руководитель
д. ф.-м. н.
Василевский Юрий Викторович

Москва – 2013

Содержание

Введение

Процесс решения задач математической физики с использованием ЭВМ можно разделить на три основных этапа: построение расчетной сетки, дискретизация дифференциальных или интегральных уравнений и решение системы алгебраических уравнений. В настоящей работе центральную часть занимает проблема построения качественных расчетных сеток. Кроме этого, будут рассмотрены дискретизации эллиптических уравнений в трехмерных областях и на поверхностях. Различные методы дискретизаций представлены в работах [4, 5, 7]. Методы решения алгебраических уравнений рассматриваются в работах [6, 8, 9, 82].

В трехмерном пространстве среди прочих выделяются три класса сеток: тетраэдральные, треугольные призматические и гексаэдральные. Расчетные сетки должны удовлетворять различным требованиям, и каждый из этих классов сеток имеет свои преимущества и недостатки.

Так, например, сетки должны приближать границу области с достаточным порядком точности. В настоящее время при расчетах приемлемым является второй порядок точности. В консервативных дискретизациях, популярных у инженеров, степени свободы находятся в центрах ячеек, поэтому генераторы сеток должны стремиться минимизировать число ячеек для заданной плотности распределения узлов сетки. При фиксированном количестве вершин $N_{\mathcal{V}}$ сетки известны следующие оценки количества ячеек $N_{\mathcal{C}}$ и граней $N_{\mathcal{F}}$ в сетках разных типов. Для неструктурированных тетраэдральных сеток $N_{\mathcal{C}} \approx 5.5 N_{\mathcal{V}}$, $N_{\mathcal{F}} \approx 11 N_{\mathcal{V}}$, для треугольных призматических сеток $N_{\mathcal{C}} \approx 2 N_{\mathcal{V}}$, $N_{\mathcal{F}} \approx 5 N_{\mathcal{V}}$, для гексаэдральных $N_{\mathcal{C}} \approx N_{\mathcal{V}}$, $N_{\mathcal{F}} \approx 3 N_{\mathcal{V}}$. С этой точки зрения, гексаэдральные сетки являются наиболее выгодными. Получающиеся при дискретизации на гексаэдральных сетках шаблоны являются самыми компактными, а, значит, соответствующие матрицы имеют меньшее количество ненулевых элементов. Такие матрицы являются более экономичными с точки зрения требуемого объема памяти и вычислительной сложности решения системы уравнений. С другой стороны, с точки зрения автоматического построения сеток для сложных областей, гексаэдральные сетки являются самыми трудными и неудобными. Существующие на сегодняшний день методы построения гексаэдральных сеток чаще всего применимы для узкого класса областей (блочно-структурированные сетки). Кроме этого остается открытым вопрос качественной аппроксимации границы подобными сетками. Однако, можно отметить некоторые коммерческие генераторы, позволяющие строить качественные гексаэдральные сетки в сложных областях: генератор HEXPRESS компании Numeca [15], а также генератор Hexotic [20], являющийся частью комплекса MeshGems [21].

Тетраэдральные сетки содержат большое количество ячеек и граней, что увеличивает количество ненулевых значений в матрице системы уравнений. Другим недостатком является то, что для анизотропных областей получаемые неструктурированные сетки могут иметь ячейки с очень большими двугранными углами, что значительно ухудшает качество дискретизации. Однако, с помощью тетраэдральных сеток можно строить расчётные сетки в сколь угодно сложных областях [2, 39]. Существует ряд комплексов программ, позволяющих строить тетраэдральные сетки: открытые Ani3D [17], TetGen [24], NETGEN [23], а также коммерческие TetMesh [25], CUBIT [18] и другие.

В некоторых задачах призматические сетки являются компромиссом между тетраэдральными и гексаэдральными сетками. Они не так дороги с вычислительной точки зрения и могут быть эффективными в анизотропных областях. Они широко применяются в геофизических приложениях в задачах со слоистыми областями. В таких областях призматическую сетку можно представить как тензорное произведение неструктурированной треугольной сетки в плоскости Oxy и одномерной сетки вдоль оси Oz. В Институте вычислительной математики РАН генератор треугольных призматических сеток разработан В.Н.Чугуновым [14]. Известны также коммерческие пакеты FEFLOW [19], MODFLOW-USG [22], а также разрабатываемый расчетный комплекс GeRa [1] [1], позволяющие строить такие сетки.

Перспективным направлением в развитии сеточных генераторов является создание надёжной технологии построения гибридных сеток для сложных областей, состоящих преимущественно из гексаэдров, а также из приграничных многогранных ячеек. В настоящее время можно выделить несколько пакетов программ, которые позволяют строить сетки с преимущественно гексаэдральными ячейками. Закрытый программный пакет ЛОГОС.ПреПост, разрабатываемый в ВНИИЭФ г.Сарова, позволяет строить сетки с шестигранными и многогранными ячейками. Кроме этого, известен коммерческий пакет HEXPRESS/Hybrid компании Numeca [15], позволяющий строить подобные сетки.

В первой главе диссертационной работы представляется технология надёжного построения гибридных сеток на основе восьмеричных деревьев и многогранных *сколотых ячеек*. Сетки типа восьмеричное дерево позволяют иметь быстрый доступ к любой ячейке, ее соседям и вершинам. Кроме того, они требуют небольших расходов памяти при хранении и динамическом перестроении. Технология восьмеричного дерева позволяет легко локально измельчать и разгрублять сетку в необходимых местах. Сгущение сетки к границе области за счет разбиения приграничных ячеек обеспечивает аппроксимацию границы с первым порядком точности. В настоящей работе предлагается метод построения гексаэдральных сеток со сколотыми ячейками, которые позволяют приближать гладкую границу области со вторым порядком точности.

[1] Geomigration of Radionuclides - совместный проект ИВМ РАН и ИБРАЭ РАН в рамках проекта "Прорыв" ГК Росатом

Сколотая ячейка представляет собой часть кубической ячейки, полученную в результате ее среза поверхностной сеткой. Скалывание кубической ячейки может осуществляться различными способами. Так, например, Бретоннет и др. [34] применяют алгоритм полигональных сечений (Polygon clipping algorithm) [87], который используется в коммерческом генераторе HEXPRESS [15]. В настоящей работе в алгоритме скалывания используются поверхностные триангуляции. Существует ряд методов построения поверхностной триангуляции, использующих кубические сетки.

Широко известен метод марширующих кубов [71] (Marching cubes, MC). Однако, при всей своей популярности, известны также проблемы этого метода. Во-первых, в силу неоднозначности выбора триангуляции в кубической ячейке, алгоритм может порождать топологически несвязные сетки. Существует ряд работ, посвященных решению этой проблемы, [36, 53, 64, 73] и др. Во-вторых, классический метод марширующих кубов не может воспроизводить резкие искривления и особенности границы. В своей работе [60] Кобельт и др. предлагают расширенный метод марширующих кубов (Extended marching cubes), который отслеживает сложную геометрию границы, используя дополнительную информацию о нормалях к поверхности. В-третьих, алгоритм MC может быть применён только к равномерным кубическим сеткам. В случае, если размеры двух соседних ячеек отличаются, получаемая триангуляция становится топологически несвязной и может содержать дырки и пустоты. В работах [54, 56, 70, 91] предложены методы для восстановления топологической связности такой триангуляции на сетках типа восьмеричное дерево. Однако, данные алгоритмы также имеют существенные недостатки и сложности реализации, описанные в работе [55]. Там же предложен элегантный алгоритм кубических марширующих квадратов (Cubical marching squares, CMS), который лишен всех описанных недостатков. Данный алгоритм позволяет строить конформную триангуляцию поверхности для сеток типа восьмеричное дерево.

Также известен метод марширующих тетраэдров (Marching tetrahedra) для построения поверхностной триангуляции на кубических сетках. В отличие от метода MC, он лишен проблемы неоднозначности, однако порождает значительно большее количество треугольников в поверхностной триангуляции.

Во многих прикладных задачах, таких как построение сеток для численных моделей подземной гидродинамики или построение трехмерных сеток по данным МРТ или КТ, область разбита на непересекающиеся подобласти с различными физическими свойствами, которые граничат между собой произвольным образом. Построенная сетка должна правильно отражать наличие различных подобластей. Для построения подобных сеток в настоящей работе предложена модификация метода марширующих кубов для областей с несколькими материалами (Multiple material marching cubes, M^3C [93]) с более точным приближением границы. Эта модификация обобщает преимущества методов CMS и M^3C, а, значит, может быть использована для построения сколотых ячеек на сетках типа восьмеричное дерево в областях с несколькими материалами. В результате получается многогранная сетка типа восьмеричное дерево со сколотыми ячейками, которая состоит преимущественно из гексаэдров.

Дискретизации на многогранных сетках также являются сложной задачей. Самые простые и эффективные дискретизации разработаны для конформных сеток. Таким образом, наиболее приемлемыми расчетными сетками представляются гибридные сетки, принадлежащие классу конформных многогранных сеток. Построение консервативных схем дискретизаций, применимых к анизотропным тензорам диффузии на конформных многогранных сетках, является сложной востребованной задачей. Для проверки возможности использования многогранных сеток типа восьмеричное дерево со сколотыми ячейками для приближенного решения краевых задач, во второй главе предлагается трехмерный аналог нелинейной многоточечной схемы, удовлетворяющей *дискретному принципу максимума (ДПМ)*. Двумерная версия этого метода была предложена в работе [68].

Принцип максимума (или минимума) является важным свойством решений линейных и нелинейных уравнений в частных производных. Выполнение дискретного аналога принципа максимума является желаемым свойством для численных схем. К сожалению, общепринятые схемы, удовлетворяющие дискретному принципу максимума, накладывают существенные ограничения на регулярность тетраэдральной сетки [61] и коэффициенты задачи. Нарушение ДПМ может привести к различным численным артефактам, таким как тепловой поток из холодного материала к горячему и др.

В данной работе предлагается трехмерный аналог нелинейного метода конечных объемов для уравнения диффузии с анизотропными коэффициентами, удовлетворяющий ДПМ. Этот метод работает на произвольных многогранных сетках и имеет компактный шаблон.

Классический метод конечных объёмов со степенями свободы в центрах ячеек и линейной двухточечной аппроксимацией диффузионного потока удовлетворяет ДПМ [31, 86]. Однако, главные направления тензора диффузии должны быть ортогональны граням сетки, в противном случае метод не имеет даже первого порядка точности для анизотропных диффузионных задач или на неструктурированных сетках. Тем не менее именно этот метод является наиболее распространённым в моделировании течений в пористых средах в силу своей технологической простоты и монотонности. Многоточечная аппроксимации потока (MPFA – Multipoint flux approximation) имеет второй порядок точности благодаря использованию большего числа точек в шаблоне, однако является условно устойчивой и условно монотонной. Ограничения на устойчивость и монотонность конечно-объёмных схем MPFA рассмотрены в

[26, 59, 75].

Другой класс монотонных схем для произвольных многогранных сеток состоит из нелинейных методов. Первоначальная идея принадлежит ЛеПотье [62], который предложил монотонную двухточечную схему дискретизации потока с коэффициентами, зависящими от концентраций в соседних ячейках. Этот подход развивался в работах [40, 66, 67, 74, 95] (см. также ссылки в этих работах), в которых доказывается неотрицательность решения на произвольных многогранных сетках и тензорах общего вида. Некоторые из подобных схем вводят дополнительные неизвестные на вершинах, ребрах или гранях сетки, значения в которые интерполируются из концентраций в соседних ячейках. Выбор схемы интерполяции оказывает большое влияние на точность нелинейной схемы [65, 95]. Определённый метод интерполяции может оказаться эффективным для одних задач и неэффективным для других. В работах [66, 67] предлагаются и исследуются схемы для уравнений диффузии и конвекции-диффузии, не требующие интерполяции решения в узлы сетки.

Нелинейные многоточечные схемы, удовлетворяющие ДПМ, были предложены в работах [63, 96] для двумерных диффузионных задач. Эти схемы также используют интерполяцию решения в дополнительных неизвестных. В работе [68] предлагается многоточечная нелинейная схема для диффузионного потока, не требующая интерполяции данных. В результате получается схема с минимальным шаблоном (в шаблоне используются только соседи по ребрам двумерной сетки), которая на ортогональных сетках и с диагональным тензором становится классической пятиточечной (для полного тензора шаблон также пятиточечный). В настоящей работе предлагается трехмерный аналог этой схемы для произвольных многогранных сеток с ячейками звездного типа.

Предложенная схема может потребовать интерполяции для некоторых вспомогательных неизвестных, однако бóльшая часть этих неизвестных интерполируется на основе физических принципов, например, на основании непрерывности диффузионного потока на гранях сетки. В некоторых случаях могут понадобиться вспомогательные значения в рёбрах граней, которые получаются с помощью арифметического осреднения значений из соседних граней. Такой выбор интерполяционной схемы использует физически обоснованные значения на гранях и прост в реализации.

В одно время с данной работой независимо от нас была опубликована работа [49], в которой также предлагается нелинейная многоточечная схема дискретизации диффузионного потока на многогранных сетках, удовлетворяющая ДПМ. В настоящей работе не стоит задачи сравнения качества данных схем, однако отметим отличия схем. В работе [49] для аппроксимации диффузионного потока через грань используются *точки гармонического осреднения* [27]. В предлагаемой же схеме такие точки используются только на гранях, в которых происходит *разрыв* тензора, поэтому в ней гораздо меньше интерполяции данных.

Помимо дифференциальных уравнений в трехмерных областях, во многих прикладных задачах возникают уравнения в частных производных на поверхностях. Такие уравнения возникают в математических моделях различных природных явлений: диффузия вдоль межкристаллических границ [72], перенос поверхностно активных веществ на интерфейсах многофазных течений [81], липидные взаимодействия в биомембранах [47], а также во многих инженерных и биомедицинских приложениях: визуализация векторного поля [43], синтез текстур [89], построение развертки мозга [88], моделирование жидкости в легких [52] и многих других. Таким образом, в настоящее время большой интерес представляет развитие методов численного решения уравнений на поверхностях.

Можно выделить несколько основных подходов к численному решению уравнений на поверхностях. Один из них требует построения явных триангуляций поверхности и дискретизации уравнений на получаемых сетках. Развитие численных методов, основанных на поверхностных триангуляциях, началось с работы [45]. В этом классе методов, поверхность аппроксимируется семейством последовательных регулярных триангуляций. Предполагается, что все вершины триангуляций лежат на поверхности. В работе [46] метод из [45] был скомбинирован с Лагранжевым методом отслеживания поверхности и был обобщен для уравнений на движущихся поверхностях. Однако, использование поверхностных триангуляций имеет ряд существенных недостатков, в частности: требуется явная параметризация поверхности, получаются достаточно сложные дискретизации, если поверхность эволюционирует, то триангуляции требуют перестроения, и некоторые другие. Отметим, что для построения поверхностных триангуляций, вообще говоря, может быть использован метод, описанный в первой главе настоящей работы. Однако качество получаемых треугольников не является приемлемым для дискретизаций на таких сетках. В статье [78] был предложен подход для решения уравнений заданных на поверхностях, который не требует параметризации и явной триангуляции поверхности. Метод основан на следах функций из внешнего конечно-элементного пространства на дискретной поверхности, которая может являться нулем кусочно-линейной непрерывной функции уровня и произвольно накладываться на внешнюю сетку. Данный метод получил развитие в [80] и для эволюционирующих поверхностей в [77, 79]. Отметим, однако, что матрицы алгебраических систем, возникающие в данном подходе могут быть плохо обусловленными (см. [76]) и реализация метода требует численного интегрирования вдоль дискретной поверхности. Другой метод, избегающий использования триангуляций поверхности, был предложен в [32]. Он предполагает продолжение дифференциального уравнения на поверхности во множество из $\mathbb{R}^3$ положительной лебеговой меры. В результате получается формулировка уравнения в пространстве большей размерности, однако уравнение может быть решено на сетке, не привязанной к поверхности. В случае движущейся поверхности, этот метод позволяет избежать Лагранжева описания эволюции поверхности, используя Эйлеров подход [94].

Несмотря на преимущества метода, предложенного в [32], он имеет ряд недостатков. В частности, полученные в результате эллиптические или параболические уравнения являются вырожденными, так как отсутствует диффузия в нормальном направлении относительно поверхности. Постановка граничных условий при дискретизациях таких задач также является проблемой. Более подробно плюсы и минусы этого метода описаны в работах [41, 51]. В работе [51] была предложена модификация этого метода для параболических задач, избегающая проблему вырожденности уравнения. Однако, эта модификация также имеет некоторые недостатки, которые будут описаны в разделе 3.2.

В третьей главе настоящей работы предлагается и исследуется новая переформулировка эллиптического уравнения на поверхности, использующая преимущества модификации из работы [51]. Эта переформулировка приводит к невырожденному эллиптическому уравнению в объемной области, содержащей данную поверхность. Она сохраняет все преимущества формулировки из работы [51], однако диффузия вдоль нормального направления добавляется неявно, что позволяет избежать введения дополнительных параметров. Для численного решения полученных уравнений можно использовать различные известные методы. В частности, для численного решения использовалась предложенная нелинейная многоточечная схема метода конечных объемов на многогранных сетках типа восьмеричное дерево со сколотыми ячейками. Кроме того, в случае тетраэдральных сеток используется метод конечных элементов, что позволяет использовать хорошо разработанный аппарат численного анализа. На основе невырожденности новой формулировки в совместной работе [38] доказаны оценки сходимости в L^2 и L^∞ поверхностных нормах. Насколько нам известно, такие оценки ранее не были получены при решении методом конечных элементов уравнения на поверхности с помощью его продолжения в окрестность поверхности.

Актуальность темы. При решении прикладных трехмерных задач в сложных областях возникает необходимость создания технологии построения расчетных сеток, методов дискретизации дифференциальных уравнений на них и способов решения полученных систем алгебраических уравнений. Проблеме построения качественных расчетных сеток для сложных геометрических областей уделяется большое внимание. Особый интерес представляют экономичные гексаэдральные сетки, однако необходима технология для более точного приближения криволинейной границы области такими сетками. Кроме этого, в связи с ограничением вычислительных ресурсов, интересны технологии построения сеток, адаптирующихся к изменению численного решения. Известные на сегодняшний день комплексы программ, позволяющие строить сетки с преимущественно гексаэдральными ячейками, а также многогранными ячейками, являются закрытыми. При этом, используемые алгоритмы не опубликованы, поэтому не представляется возможным судить о их надежности и эффективности.

Дискретизация уравнений математической физики на многогранных сетках является отдельной задачей. Во многих прикладных задачах важно соблюдение определенных физических свойств решения, например, сохранение неотрицательности решения или удовлетворение дискретному принципу максимума. Кроме этого, при моделировании физических процессов часто приходится сталкиваться с анизотропными свойствами среды. Таким образом, в настоящее время особый интерес вызывают монотонные консервативные схемы дискретизации уравнений диффузии для анизотропных сред на многогранных сетках.

Уравнения в частных производных на поверхностях возникают во многих естественных процессах, в компьютерных, инженерных и биомедицинских приложениях. В последнее время интерес представляют численные методы, основанные на расширении уравнения на поверхности в некоторую ее окрестность. В результате полученные уравнения будут решаться в пространстве большей размерности, однако для решения может быть использован широкий набор численных методов в декартовых координатах на различных сетках.

Цель диссертационной работы. Целями диссертационной работы являются разработка технологии построения многогранных сеток с преимущественно гексаэдральными ячейками, разработка нелинейной монотонной схемы дискретизации уравнения диффузии на многогранных сетках, в том числе на предлагаемых сетках, а также разработка метода решения уравнений на поверхностях с помощью продолжения уравнения в окрестность поверхности.

Научная новизна. В работе предложена технология построения многогранных сеток типа восьмеричное дерево со сколотыми ячейками для сложных областей с несколькими материалами; предложена и численно исследована трехмерная версия монотонной нелинейной схемы на основе метода конечных объемов для уравнения диффузии на многогранных сетках; предложена и численно исследована новая переформулировка эллиптических уравнений на поверхности, приводящая к невырожденному эллиптическому уравнению в окрестности поверхности.

Практическая значимость. Практическая значимость диссертационной работы заключается в создании генератора многогранных сеток типа восьмеричное дерево со сколотыми ячейками. Генератор внедрен в расчетный комплекс GeRa, в котором является частью технологической цепочки расчетов геомиграции радионуклидов в слоистых геологических областях. Кроме этого, создана технологическая цепочка на платформе INMOST, включающая в себя построение расчетной многогранной сетки со сколотыми ячейками и численное решение на ней диффузионных задач. С помощью этого комплекса программ также были решены тестовые уравнения на поверхности.

На защиту выносятся следующие основные результаты:

1. Предложен алгоритм и разработана технология надежного построения многогранных сеток типа восьмеричное дерево со сколотыми ячейками для сложных областей с несколькими материалами.

2. Предложена и численно исследована трехмерная версия монотонной нелинейной схемы дискретизации на многогранных сетках уравнения диффузии, удовлетворяющая принципу максимума.

3. Предложена и численно исследована новая формулировка эллиптических уравнений на поверхностях, приводящая к невырожденной эллиптической задаче в окрестности поверхности. Для ее численного решения применялись как метод конечных элементов, так и метод конечных объемов на многогранных сетках.

Апробация работы. Результаты диссертационной работы докладывались автором и обсуждались на научных семинарах Института вычислительной математики РАН, Института прикладной математики РАН им. М. В. Келдыша, Вычислительного центра РАН им. А. А. Дородницына, Института проблем безопасного развития атомной энергетики РАН и на следующих научных конференциях: конференция молодых ученых “Технологии высокопроизводительных вычислений и компьютерного моделирования” (СПбГУ ИТМО, С.-Петербург, апрель 2009); конференция “Лобачевские чтения” (КГУ, Казань, ноябрь 2009); конференции “Тихоновские чтения” (МГУ, октябрь 2012); конференции “Актуальные проблемы прикладной математики и механики” (Абрау-Дюрсо, сентябрь 2012); международная конференция “NUMGRID-2012”

(ВЦ РАН, Москва, июнь 2012); международная конференция “CRC-NAA-2013” (Ростов-на-Дону, июнь 2013); международная конференция “Mathematical modeling of natural disasters and technical hazards” (Сьон, Швейцария, август 2013).

Публикации. Основные материалы диссертации опубликованы в 6 печатных работах, из них 2 статьи в рецензируемых журналах, входящих в перечень ВАК [13, 38], и 4 - в сборниках тезисов конференций [10–12, 37].

Личный вклад автора. Все результаты главы 1 и главы 2 получены автором самостоятельно. В совместной работе [38] вклад автора заключался в разработке новой формулировки эллиптического уравнения на поверхности и реализации численных экспериментов. Программная реализация всех методов и все расчеты выполнены лично автором.

Структура и объем диссертации. Диссертационная работа состоит из введения, обзора используемой терминологии, трех глав, заключения и списка литературы из 97 наименований. Диссертационная работа содержит 31 рисунок и 11 таблиц. Общий объем диссертационной работы – 125 страниц.

Благодарности

Автор диссертационной работы выражает огромную благодарность и глубокую признательность научному руководителю Ю. В. Василевскому за продолжительную поддержку, ценные советы и плодотворное обсуждение вопросов. Автор выражает глубокую признательность М. А. Ольшанскому за совместную работу, помощь и ценные советы. Автор также выражает благодарность А. А. Данилову, К. Д. Никитину, И. В. Капырину, К. М. Терехову и многим другим за активную помощь в обсуждении идей и методов, используемых в диссертационной работе. Автор благодарен своим родителям и друзьям за поддержку.

Работы над диссертацией была частично поддержана грантами РФФИ

11-01-00971, 12-01-33084, 12-01-31223, федеральной целевой программой “Научные и научно-педагогические кадры инновационной России”, грантом Upstream Research Center of ExxonMobil corp. и проектом “Прорыв” ГК “Росатом”.

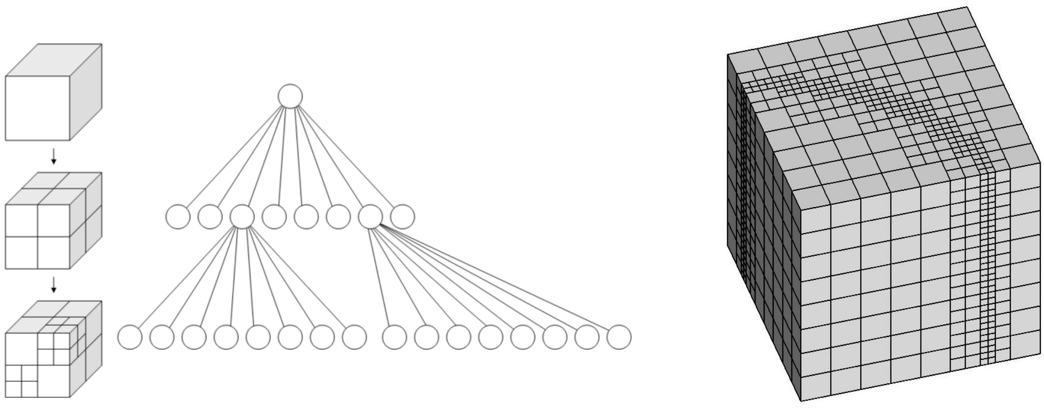
Обзор используемой терминологии

Введем необходимые понятия, которые будут использованы в настоящей диссертационной работе.

В диссертационной работе будем рассматривать *конформные* сетки, любые два элемента которых либо не имеют общих точек, либо имеют ровно одну общую вершину, либо одно общее ребро, либо одну общую грань. Также в первой главе используется понятие *слабо-конформных* сеток, в которых некоторые элементы могут иметь более одной общей грани, в отличие от конформных сеток. В работе используются сетки типа *восьмеричное дерево.* Восьмеричное дерево (или восьмидерево) представляет собой древовидную структуру данных, элементы которой удовлетворяют следующим свойствам:

- Каждый элемент имеет либо ровно восемь потомков, либо не имеет потомков. Элемент, не имеющий потомков, называется *листом*, или листовым элементом;
- Элемент не может быть потомком самого себя;
- Каждый элемент, кроме исходного, называемого *корневым*, является потомком только одного элемента, называемого *родителем*;
- Корневой элемент не имеет родителя, всегда существует и единственен.

Структура восьмеричного дерева состоит из вершин, являющихся родителями и потомками, и ребер, обозначающих связи между родителями и их потомками. Сетка типа восьмеричного дерева представляет собой восьмидерево, элементами которого являются кубы. Она может быть получена путем иерархического измельчения корневой кубической ячейки, см. Рис. 1(a). На каждом уровне измельчения родительская ячейка делится на восемь кубиков-потомков. Пример сетки типа восьмеричное дерево изображен на Рис. 1($б$).



(*а*) (*б*)

Рис. 1. Восьмеричное дерево (*а*). Пример сетки типа восьмеричное дерево (*б*).

В данной работе сетками типа восьмеричное дерево будем также называть сетки, полученные в результате удаления некоторых листовых элементов из сетки типа восьмеричное дерево. При этом, формально, некоторые элементы будут иметь отличное от восьми число потомков.

Грань сетки будем называть *листовой*, если две ячейки, разделяющие эту грань, являются листовыми и имеют одинаковый размер. Заметим, что листовая ячейка может иметь сложную не листовую грань, если какая-либо соседняя ячейка находится на более глубоком уровне измельчения. Такую грань будем называть *переходной*.

Сетки типа восьмеричное дерево с формальной точки зрения не являются конформными. Однако, сетки типа восьмеричное дерево можно отнести к конформным сеткам, если рассматривать их как сетки с многогранными ячейками, у которых некоторые смежные грани могут лежать в одной плоскости.

Для сеток типа восьмеричное дерево в работе вводится дополнительное ограничение: размеры двух соседних ячеек могут отличаться не более чем в два раза. В таком случае измельчение сетки будет более плавным, что существенно при интерполяции сеточных данных и расчетах.

В работе рассматриваются сетки с *ячейками звездного типа* относительно центра масс, в которых каждая грань полностью видна из центра масс ячейки.

Вектор нормали к грани, умноженный на тензор диффузии, называется *вектором конормали* (2 глава). Сетка имеет $\mathbb{K}$-ортогональные ячейки, если вектор, соединяющий точки коллокации соседних ячеек, сонаправлен вектору конормали к общей грани этих ячеек.

Глава 1

Построение многогранных сеток типа восьмеричное дерево со сколотыми ячейками в составных областях

В этой главе будет предложен алгоритм построения конформных многогранных сеток. Сетки являются гибридными и состоят преимущественно из гексаэдров, а также из многогранных ячеек. В алгоритме используются сетки типа восьмеричное дерево. Это позволяет строить гексаэдральные сетки, иерархически сгущающиеся к границам области и к другим ее частям. Для более точного приближения границы используются сколотые многогранные ячейки. Сколотой ячейкой будем называть часть кубической ячейки, полученной в результате ее среза поверхностной сеткой (см. Рис. 1.1). Для построения сколотых ячеек используется алгоритм кубических марширующих квадратов [55].

Во многих прикладных задачах расчетная область разделена на подобласти, что должно быть учтено в алгоритме построения сетки. Будем называть подобные области *составными*, а области, не разделенные на подобласти - *однородными*. Границы между подобластями будем называть *интерфейсами*. Таким образом, каждой точке рассматриваемой области можно сопоставить индекс подобласти, которой она принадлежит. Также такой номер будем называть *материалом*, а составные области будем также называть *областями с несколькими материалами*. Заметим, что область может быть однородной в физическом смысле, однако если она разделена на подобласти, то для алгоритма она будет являться составной. Для построения сколотых ячеек в составных областях предлагается модификация метода марширующих квад-

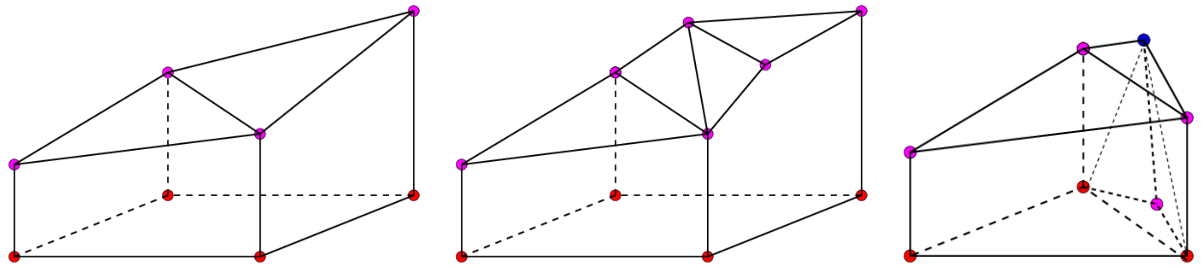


Рис. 1.1. Сколотые ячейки

ратов для областей с несколькими материалами, описанного в [93].

Сетка типа восьмеричное дерево со сколотыми ячейками для однородной области является конформной многогранной сеткой. В случае, если область является составной, то полученная сетка будет слабо-конформной в том смысле, что некоторые ячейки могут иметь более одной общей грани.

В этой главе будет произведен анализ конечности и вычислительной сложности предложенного алгоритма. Также будут описаны некоторые алгоритмы улучшения качества сеток, а также алгоритм динамического перестроения сетки. Будут продемонстрированы примеры построенных сеток. Предложенная технология построения многогранных сеток типа восьмеричное дерево, ее анализ и результаты экспериментов опубликованы в статье [13].

В разделе 1.6 описывается алгоритм построения сеток для слоистых областей. Такие сетки могут быть построены для областей земной коры, состоящих из геологических слоев. Геологическими слоями называют те массы, ограниченные почти горизонтальными поверхностями, из которых состоит толща осадочных пород. Границу между слоями называют геологическим интерфейсом. Возможны ситуации, когда геологические слои приобретают нулевую толщину, т.е. *выклиниваются*. Кроме этого, в области могут быть геологические разломы. Для простоты они могут быть заданы наклонными поверхностями, разделяющими геологические слои. Алгоритм позволяет строить отображенные сетки для подобных областей, ячейки которых направлены вдоль невыклинивающихся геологических слоев.

1.1. Построение сколотых ячеек

В этом разделе опишем алгоритм кубических марширующих квадратов (CMS) и процесс создания сколотых ячеек на его основе.

1.1.1. Алгоритм кубических марширующих квадратов (CMS)

Входными данными для алгоритма является кубическая сетка, вершинам которой приписан знак характеристической функции области. На ребрах сетки отмечены точные точки пересечения с границей области, а также нормали к границе в этих точках. При этом на каждом ребре должно быть отмечено не более одной точки пересечения. В случае, если на ребре возникает более одной точки пересечения, то все ячейки, разделяющие это ребро, следует измельчить. Как показано в [60], такой тип данных может быть получен из большинства используемых типов входных данных.

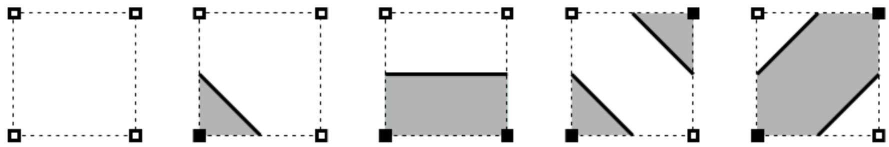
Алгоритм CMS основывается на следующих идеях:

- марширующие кубы могут быть развернуты в марширующие квадраты;
- нормали могут быть использованы для разрешения неоднозначных ситуаций, а также для поиска особенностей границы.

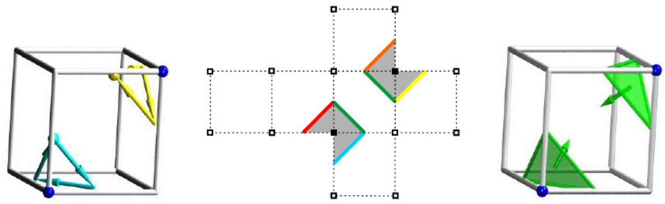
Как показано на Рис. 1.2, *б*), куб может быть развернут в шесть граней. Для каждой грани строится кусочно-линейный контур пересечения с поверхностью границы области с помощью алгоритма марширующих квадратов (MS, Marching squares). Здесь и далее контуром будем называть набор ломаных. Элементами контура будем называть отрезки ломаных. Если грани свернуть обратно в куб, то контуры будут образовывать замкнутые компоненты, которые триангулируются каким-либо способом. Стоит отметить, что в случае равномерной кубической сетки, полученная триангуляция будет похожа на триангуляцию, построенную классическим методом марширующих кубов.

Рассмотрим процесс построения кусочно-линейного контура поверхности на листовой грани. Каждая грань имеет 4 вершины, которым приписан один из двух знаков характеристической функции. Таким образом, всего существует $2^4 = 16$ различных вариантов распределения знаков, которые с учетом симметрии и поворотов могут быть сведены к 5 вариантам, изображенным на Рис. 1.2, *a*). В случаях 4-5 возникает неоднозначность выбора пары точек, которые необходимо соединить контурами. Эту неоднозначность можно решать различными способами. Классический подход предполагает заранее выбрать приоритет связности одного из знаков характеристической функции. Однако, на практике такой подход не всегда отражает действительность. Для более точного решения проблемы неоднозначности можно использовать нормали к границе. В таком случае, для каждой точки на ребре проводится соответствующая касательная в плоскости грани. В зависимости от видов пересечения (или совпадения) таких прямых, можно определить ту пару точек, которые нужно соединить, см. Рис. 1.3, *a*). Для простоты, можно также проверить знак характеристической функции в центре грани, если имеется возможность, и в зависимости от него соединить необходимую пару точек.

Помимо этого, нормали в алгоритме CMS можно использовать для того, чтобы отслеживать особенности поверхности. Рассмотрим грань с двумя точками пересечения на ребрах. Если угол между проекциями нормалей на плоскость грани в этих точках меньше определенного порога, то контуром на данной грани будет отрезок, соединяющий эти точки. В противном случае считается, что поверхность имеет особенность. Тогда из точек на ребрах про-



(*а*)



(*б*)

Рис. 1.2. Марширующие квадраты (*а*). Развертка куба (*б*) . Рисунки из [55].

водятся касательные в плоскости грани, а их точка пересечения добавляется в контур, см. Рис. 1.3, *б*). Очевидно, что полученная точка не обязательно принадлежит поверхности, но полученный контур, вероятно, будет приближать поверхность более качественно. Стоит отметить, что на практике это позволяет точно отслеживать плоские двугранные углы поверхности.

В случае, если грань переходная, то ее кусочно-линейным контуром является объединение контуров смежных с ней граней. Таким образом, любой элемент контура принадлежит ровно двум граням. Это автоматически приводит к отсутствию пустот и топологической несвязности триангуляции даже в случае, когда соседние грани имеют разный размер.

После того, как на всех гранях куба построены контуры, происходит объединение всех контуров. Затем из них выделяются компоненты, которые образуют циклы. Для этого достаточно сделать обход по ломаным, начав с произвольной вершины и двигаясь к соседней. После этого на всех замкнутых компонентах необходимо построить триангуляцию. Для этого можно использовать метод "разделяй и властвуй" (divide-and-conquer triangulation) [83]. Суть метода заключается в следующем. Рассмотрим замкнутую последовательность вершин L, соединенных отрезками. Она рекурсивно делится на две

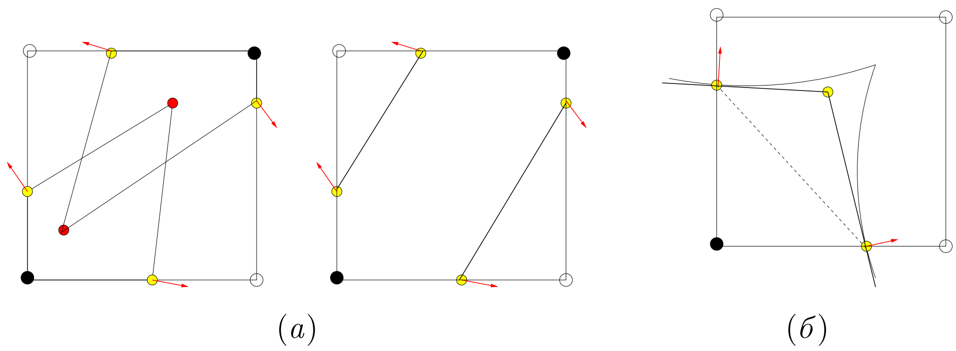


Рис. 1.3. Использование нормалей: (*а*) разрешение неоднозначности, (*б*) отслеживание искривлений границы.

части вдоль *разделяющей прямой*, соединяющей две не соседние вершины из L. Каждая новая часть делится вновь на две части, пока в каждой из частей не останется по три вершины. Так как вершины из L, вообще говоря, не принадлежат одной плоскости, то разделяющую прямую нужно выбрать так, чтобы полученные части не накладывались друг на друга. Кроме этого, разделяющая прямая не должна проходить по граням ячейки, так как на гранях не должны появляться новые отрезки. Для выбора подходящей разделяющей прямой используется *разделяющая плоскость* — плоскость, проходящая через разделяющую прямую и параллельная *средней нормали* к L. При этом, если все вершины последовательности L, находящиеся по разные стороны относительно той пары вершин, через которую проходит разделяющая прямая, находятся также по разные стороны относительно разделяющей плоскости, то полученные части не будут накладываться друг на друга, и разделяющая прямая является подходящей.

Для того, чтобы найти среднюю нормаль, рассмотрим центр масс вершин последовательности. Соединив его последовательно с каждой из вершин, получим набор треугольников. Тогда средняя нормаль $\mathbf{n}_L$ вычисляется по формуле (1.1) как средневзвешенная сумма внешних единичных нормалей $\mathbf{n}_i$ к этим треугольникам:

$$\mathbf{n}_L = \frac{\sum\limits_i area_i \cdot \mathbf{n}_i}{\sum\limits_i area_i}, \tag{1.1}$$

где $area_i$ - площадь i-го треугольника.

Обычно последовательность вершин может быть разделена несколькими способами. В таком случае, можно выбирать наилучшую разделяющую прямую по некоторому критерию. Так, например, в [83] предлагается рассматривать отношение минимального расстояния от вершин последовательности до разделяющей плоскости к длине разделяющего отрезка (отрезка разделяющей прямой, ограниченного соответствующими двумя точками последовательности). Тогда выбирается разделяющая прямая, для которой это отношение является наибольшим.

Триангуляция "разделяй и властвуй" может быть описана в виде алгоритма 1.1. Для простоты в данном алгоритме будем выбирать первую подходящую разделяющую прямую.

Алгоритм 1.1 Триангуляция “разделяй и властвуй”

```
Вход: L                                        ▷ Последовательность вершин
Присвоить N_V количество вершин в L
если N_V = 3 тогда
    Создать треугольник из вершин L
    выход
конец если
Вычислить среднюю нормаль n_L
для i = 1, 2 ... N_V − 2 начало цикла
    для j = i + 2 ... N_V начало цикла
        Положить D_ij - диагональ, соединяющая i и j вершины
        Построить разделяющую плоскость p_ij из D_ij и n_L
        Разделить L на L_left и L_right с помощью D_ij
        если L_left and ℒ_right лежат в разных полупространствах относительно p_ij тогда
            Триангуляция “разделяй и властвуй” (L_left)
            Триангуляция “разделяй и властвуй” (L_right)
        конец если
    конец цикла
конец цикла
```

Наличие нормалей в точках на ребрах позволяет также отслеживать особенности поверхности внутри ячейки. Для поиска особенности поверхности используется идея, предложенная в [60]. Обозначим точки на ребрах за $\mathbf{s}_i$, единичные нормали в этих точках за $\mathbf{n}_i$. Оценим открытый угол конуса, натянутого на $\mathbf{n}_i$, по следующей формуле:

$$\theta = \min_{i,j}(\mathbf{n}_i^T \mathbf{n}_j). \tag{1.2}$$

Если θ меньше некоторой пороговой величины θ_s, то считается, что поверхность имеет особенность. Пусть $\mathbf{n}_0$ и $\mathbf{n}_1$ - нормали, образующие наибольший угол и пусть $\mathbf{n}^* = \mathbf{n}_0 \times \mathbf{n}_1$. Тогда оценим максимальное отклонение нормалей $\mathbf{n}_i$ от плоскости, натянутой на $\mathbf{n}_0$ и $\mathbf{n}_1$. Другими словами, если величина

$$\phi = \max_i |\mathbf{n}_i^T \mathbf{n}^*| \tag{1.3}$$

больше, чем некоторая пороговая величина ϕ_s, то считается, что поверхность имеет особенную точку в ячейке. На практике авторы из [60] используют следующие значения: $\theta_s = 0.9, \phi_s = 0.7$. Искомая точка $\mathbf{p}$ находится как решение системы:

$$[..., \mathbf{n}_i, ...]^T \mathbf{p} = [..., \mathbf{n}_i^T \mathbf{s}_i, ...]. \tag{1.4}$$

В общем случае эта система может быть переопределенной или недоопределенной. Чтобы избежать рассмотрения различных случаев, система решается псевдо-обращением матрицы с использованием сингулярного разложения, и значение $\mathbf{p}$ определяется методом наименьших квадратов.

В случае, если имеется особенная точка $\mathbf{p}$ внутри ячейки, то триангуляция строится методом "веер треугольников" (triangle fan), с центром в точке $\mathbf{p}$. Если точка $\mathbf{p}$ не попадает внутрь ячейки, то считаем, что ячейка не имеет особенной точки.

Стоить отметить, что теоретически возможна ситуация, когда контур достаточно искривленный, но особенная точка не находится, а метод "разделяй и властвуй" не находит подходящую плоскость. В таком случае, триангуляцию можно построить методом "веер треугольников", используя центр масс точек $\mathbf{s}_i$ в качестве центра триангуляции.

Метод поиска особенной точки также позволяет разрешать неоднозначность триангуляции кубической ячейки. Такие неоднозначности могут возникать, когда невозможно определить, нужно ли соединять две замкнутые

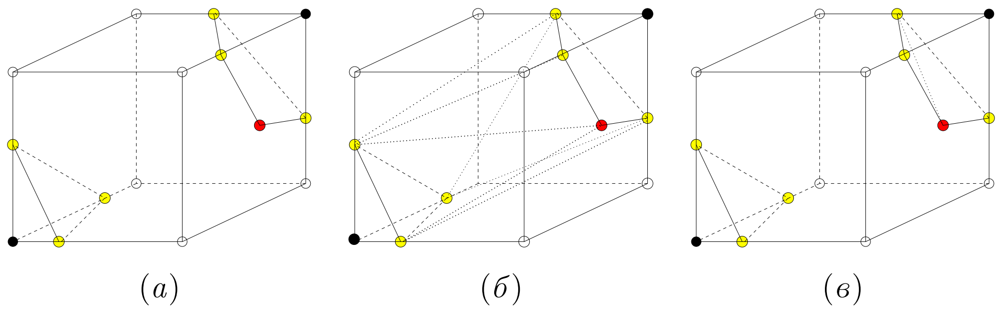


Рис. 1.4. Неоднозначность триангуляции. Желтые точки - на ребрах, красные - на гранях. (*а*) Замкнутые компоненты контура, (*б*) триангуляция объединения контуров, (*в*) триангуляция каждого контура независимо.

компоненты контура или нет, см. Рис. 1.4(*а*). Разрешить такие ситуации в [55] предлагают способом, аналогичным для случая неоднозначности на грани. Для этого, в каждой из компонент находится особенная точка описанным выше способом. Затем в них строится конусообразная поверхность с центром в этой точке. Если полученные поверхности пересекаются, значит, компоненты должны быть связными. Тогда триангуляция, соединяющая эти компоненты, топологически будет цилиндрической поверхностью, см. Рис. 1.4(*б*). В противном случае, для каждой из компонент строится независимые триангуляции, см. Рис. 1.4(*в*). Для простоты можно также проверить знак характеристической функции в центре ячейки, если это возможно, и в зависимости от него определить, соединять ли контуры. Такую проверку также можно сделать, когда система (1.4) не дает решения.

Для произвольной кубической ячейки T здесь и далее будем обозначать множества ее граней, ребер и вершин через $\mathcal{F}_T$, $\mathcal{E}_T$ и $\mathcal{V}_T$, соответственно. Тогда метод CMS для ячейки T может быть представлен в виде следующего алгоритма 1.2:

Алгоритм 1.2 кубических марширующих квадратов

```
1: Вход: ячейка T
2: для f ∈ F_T начало цикла
3:     Получить контур L_f на грани f
4: конец цикла
5: Объединить все контуры L_f и выделить замкнутые компоненты ℒ
6: для L ∈ ℒ начало цикла
7:     Проверить наличие особенной точки для L
8: конец цикла
9: если имеется неоднозначность триангуляции тогда
10:     Разрешить неоднозначность
11: конец если
12: для L ∈ ℒ начало цикла
13:     Триангуляция(L)
14: конец цикла
```

В строке 10 имеется в виду неоднозначность, описанная ранее, см. Рис. 1.4. В строке 13 может быть использована триангуляция “разделяй и властвуй” либо “веер треугольников”, если имеется особенная точка в ячейке.

1.1.2. Построение сколотых ячеек с помощью метода CMS

Подводя итог, можно сказать, что метод CMS позволяет строить триангуляцию поверхности, используя сетки типа восьмеричное дерево. Алгоритм создания сколотой ячейки состоит из следующих шагов. Вершинам кубической ячейки приписывается знак характеристической функции области (Рис. 1.5, *а*) и вычисляются входные данные для алгоритма CMS: точки пересечения на ребрах (Рис. 1.5, *б*) и нормали к поверхности в этих точках. Затем на каждой из граней ячейки строится соответствующий контур. Объ-

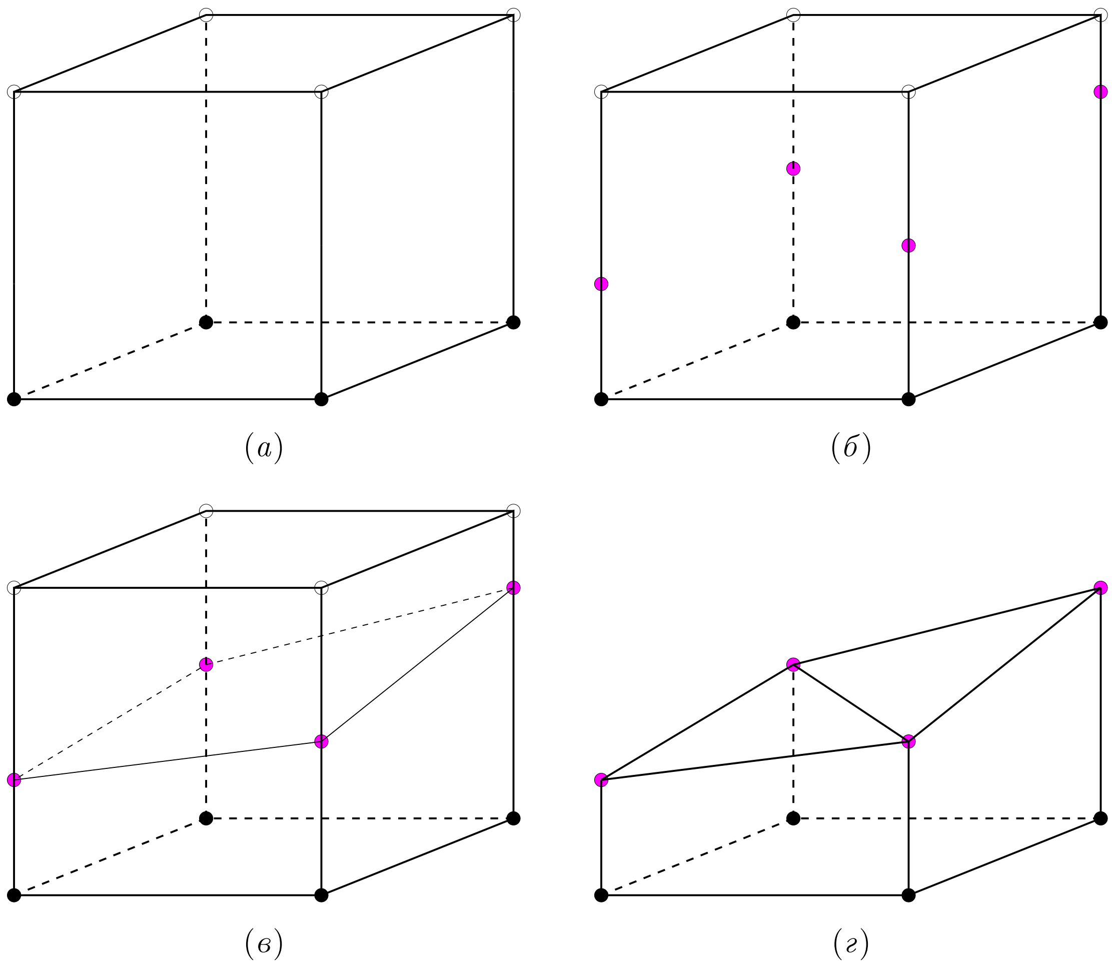


Рис. 1.5. Построение сколотой ячейки

единив полученные контуры в замкнутую ломаную, строится ее триангуляция (Рис. 1.5, *в*). Эта триангуляция делит ячейку на части, причем те части, которые не принадлежат области, удаляются. В результате получается сколотая ячейка (Рис. 1.5, *г*).

Сколотые ячейки аппроксимируют границу области треугольными гранями. Известно, что если поверхность является гладкой, то триангуляция, вершины которой принадлежат поверхности, аппроксимируют ее со вторым порядком точности в C - норме. Если граница области является гладкой, то вершины триангуляции, порождаемые алгоритмом, находятся только на ребрах кубических ячеек, и, соответственно, принадлежат поверхности. Таким образом, для областей с гладкой границей можно утверждать, что сколотые ячейки приближают границу со вторым порядком точности. Если же область имеет особенности, то вообще говоря, особенные точки находятся не точно. Однако, плоские двугранные углы отслеживаются точно.

1.2. Случай составной области

Традиционный метод марширующих кубов, а также упомянутые ранее модификации, могут быть применены только в однородных областях. На практике существует множество задач, в которых область разбита на непересекающиеся подобласти с различными физическими свойствами. Подобласти могут располагаться произвольным образом, и, в таком случае, одна кубическая ячейка может принадлежать нескольким подобластям. Если применить алгоритм MC или CMS для каждой подобласти итерационно, то полученные треугольные интерфейсы будут образовывать пустоты, и сетка получится топологически несвязной. Дотреппе и др. в своей работе [44] предлагают модификацию метода марширующих тетраэдров на случай области с несколькими материалами. В работе [97] предложен алгоритм построения тетраэдральных и гексаэдральных сеток для составных областей, использующий метод дуальных контуров (Dual Contouring [57]). В работе [93] Ву и Салливан предложили модификацию метода марширующих кубов для областей с несколькими материалами. Этот алгоритм позволяет создать топологически связную конформную триангуляцию на равномерной кубической сетке для составных областей. Он использует двумерный метод марширующих квадратов для областей с несколькими материалами (Multiple material marching squares, M^3S). Далее опишем методы марширующих квадратов и кубов для областей с несколькими материалами, а затем опишем предлагаемую модификацию этих алгоритмов.

1.2.1. Метод марширующих квадратов для областей с несколькими материалами

Рассмотрим равномерную квадратную сетку в области, разделенную интерфейсами на непересекающиеся подобласти. Для каждой вершины сетки мы можем определить индекс подобласти, в которой она находится. В случае, если вершина находится на интерфейсе двух или более подобластей, ей присваивается индекс подобласти с наибольшим приоритетом, который определяется заранее. Рассмотрим ребро сетки и две его вершины. Если индексы подобластей в этих вершинах различны, то на середине ребра создается точка, которая считается точкой пересечения ребра с соответствующим интерфейсом. Точку, которую мы считаем точкой пересечения интерфейса с ребром ячейки, будем называть *интерфейсной точкой на ребре*. Будем считать, что интерфейсной точке приписываются оба индекса подобластей вершин отрезка.

Квадратная ячейка имеет четыре вершины, каждая из которых имеет определенный индекс подобласти. Таким образом, всего имеется $4^4 = 256$ вариантов распределения индексов подобластей и построения кусочно-линейного контура на этой грани. Однако, они могут быть сгруппированы в следующие случаи:

- Если количество различных индексов не более двух, то контур получается из обычного метода марширующих квадратов, см. Рис. 1.6, верхние картинки.

- Если количество различных индексов равно трем, причем две вершины с одинаковым индексом расположены по диагонали, то контур строится так, что они будут соединены, см. Рис. 1.6, нижняя средняя картинка.

- В остальных случаях на грани создается центральная точка, которая

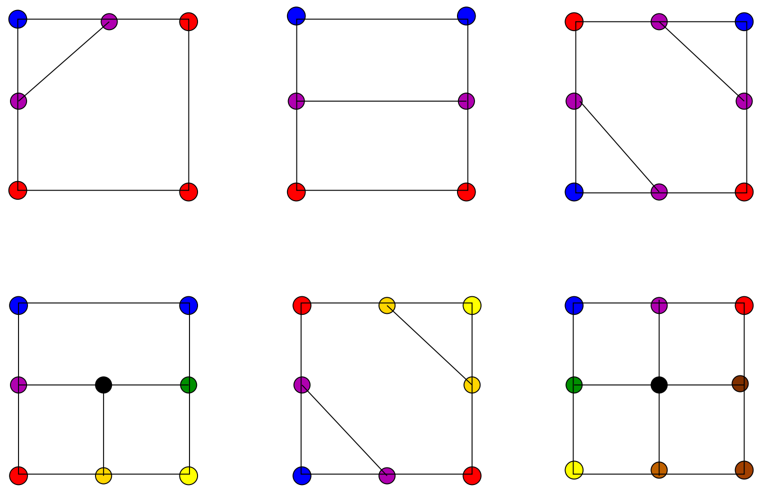


Рис. 1.6. Метод марширующих квадратов для областей с несколькими материалами.

соединяется с интерфейсными точками на ребрах, см. Рис. 1.6, нижние левая и правая картинки. Эту центральную точку будем называть *интерфейсной точкой на грани*.

Обозначим через f - квадратную грань, имеющую ребра $\mathcal{E}_f$ и вершины $\mathcal{V}_f$. Метод M^{3}S может быть описан следующим алгоритмом 1.3:

Алгоритм 1.3 Марширующих квадратов для областей с несколькими материалами ($\mathrm{M}^3\mathrm{S}$)

Вход: Грань f

для $e \in \mathcal{E}_f$ **начало цикла**

    если e имеет различные индексы на концах **тогда**

        Построить точку $\mathbf{p}_e$ в середине e, добавить $\mathbf{p}_e$ в $\mathcal{P}_\mathcal{E}$

    конец если

конец цикла

Присвоить $N_{\mathcal{P}_\mathcal{E}}$ число точек в $\mathcal{P}_\mathcal{E}$

если $N_{\mathcal{P}_\mathcal{E}} = 2$ **тогда**

    Соединить точки $\mathbf{p}_1$ и $\mathbf{p}_2$

иначе если $N_{\mathcal{P}_\mathcal{E}} = 3$ **тогда**

    Построить интерфейсную точку $\mathbf{p}_f$ на f

    Соединить точки $\mathbf{p}_e$ и $\mathbf{p}_f$, $\mathbf{p}_e \in \mathcal{P}_\mathcal{E}$

иначе

    Определить N_M количество различных индексов в вершинах $\mathcal{V}_f$

    если $N_M = 2$ **тогда**

        Разрешить неоднозначность и выбрать пары точек для соединения

    иначе если $N_M = 3$ **тогда**

        Соединить 2 пары точек так, чтобы вершины с одинаковым индексом остались связанными

    иначе

        Построить интерфейсную точку $\mathbf{p}_f$ на f

        Соединить точки $\mathbf{p}_e$ и $\mathbf{p}_f$, $\mathbf{p}_e \in \mathcal{P}_\mathcal{E}$

    конец если

конец если

Интерфейсной точке на грани приписываются индексы подобластей всех вершин грани. Каждому элементу кусочно-линейного контура на грани соответствует ровно два индекса подобластей, которые он разделяет.

1.2.2. Метод марширующих кубов для областей с несколькими материалами

После того, как на каждой грани куба построены кусочно-линейные контуры, куб рассматривается в свернутом виде. Следует отметить, что на ребрах и гранях куба больше не будут добавляться точки или отрезки. Таким образом, общая грань любых двух соседних ячеек всегда будет иметь один и тот же контур. Это означает, что ячейки будут связаны конформно. Объединив все контуры, будем рассматривать его компоненты, которые разделяют одни и те же пары подобластей. Некоторые из таких компонент могут оказаться незамкнутыми. Это происходит, если компонента содержит интерфейсную точку на грани. Такие компоненты необходимо замкнуть и после этого построить соответствующую триангуляцию во всех замкнутых компонентах. Каждая связная компонента имеет ровно два индекса подобластей, которые имеются у всех ее элементов, и отделяет соответствующие подобласти друг от друга.

В зависимости от количества интерфейсных точек на гранях, можно описать алгоритм замыкания контуров:

- Если куб не имеет интерфейсных точек на гранях, то все контуры образуют независимые замкнутые компоненты. В таком случае строится триангуляция методом “разделяй и властвуй”.

- Если куб имеет ровно две интерфейсных точки на гранях (ровно одну он иметь не может, см. [92]), то некоторые компоненты формируют незамкнутые ломаные с концами в этих интерфейсных точках. Соединив их, сделаем все незамкнутые компоненты замкнутыми. Для триангуляции таких компонент также используется метод “разделяй и властвуй”.

- Если имеется более двух интерфейсных точек на гранях, то создается точка в центре ячейки (аналогично, будем называть ее *интерфейсной точкой ячейки*). Это точка соединяется отрезками со всеми интерфейсными точками на гранях, после чего все компоненты становятся связными. Каждая компонента, содержащая интерфейсную точку ячейки, триангулируется с помощью классического метода “веер треугольников” с центром в этой точке. Если компонента не содержит интерфейсную точку ячейки, то она триангулируется методом “разделяй и властвуй”.

Метод M^3C для ячейки T может быть описан следующим алгоритмом 1.4:

Алгоритм 1.4 Марширующих кубов для областей с несколькими материалами

Вход: ячейка T
для $f \in \mathcal{F}_T$ **начало цикла**
    Применить алгоритм M^{3}S для f
    если создана интерфейсная точка $\mathbf{p}_f$ на грани f **тогда**
        Добавить $\mathbf{p}_f$ в $\mathcal{P}_\mathcal{F}$
    конец если
конец цикла
Присвоить $N_{\mathcal{P}_\mathcal{F}}$ число точек в $\mathcal{P}_\mathcal{F}$
если $N_{\mathcal{P}_\mathcal{F}} = 2$ **тогда**
    Соединить две интерфейсные точки на гранях
иначе если $N_{\mathcal{P}_\mathcal{F}} > 2$ **тогда**
    Построить интерфейсную точку $\mathbf{p}_T$ в центре T
    для $\mathbf{p}_f \in \mathcal{P}_\mathcal{F}$ **начало цикла**
        Соединить $\mathbf{p}_f$ и $\mathbf{p}_T$
    конец цикла
конец если
Выделить замкнутые компоненты контуров $\mathcal{L}$
для $L \in \mathcal{L}$ **начало цикла**
    если L содержит $\mathbf{p}_T$ **тогда**
        Построить триангуляцию “веер треугольников” для L
    иначе
        Построить триангуляцию “разделяй и властвуй” для L
    конец если
конец цикла
выход

Таким образом, можно получить триангуляцию контуров, разделяющую различные подобласти, без добавления новых точек и отрезков на гранях и ребрах, а только внутри ячейки.

Авторы метода M^3C используют только середины ребер в качестве интерфейсных точек на ребрах, а также центр грани и ячейки в качестве интерфейсной точки на грани и в ячейке, соответственно. Они объясняют это тем, что в случае использования точных значений точек пересечения могут возникнуть потенциально неправильные ситуации и нереалистичные поверхности. Такой выбор, действительно, упрощает алгоритм, но, в то же время, использование середин ребер вносит ошибку аппроксимации границы порядка $h/2$, где h - размер ребра ячейки, а, значит, сетка приближает гладкую границу с первым порядком точности. Авторы предлагают алгоритм сглаживания сетки после ее построения, однако, было замечено, что изначальный выбор середин ребер может приводить к топологиям сетки, локально отличным от топологии исходной области. Кроме этого, в работе не приводится алгоритм сглаживания для интерфейсных точек на грани и в ячейке. В настоящей работе предложена модификация алгоритма M^3C, в которой используются более точные значения точек пересечения с ребрами и гранями, при этом учтены возникающие в результате сложности.

1.2.3. Модификация метода марширующих кубов для областей с несколькими материалами

Будем считать, что у нас имеется область, заданная функцией, для каждой точки возвращающей индекс подобласти, в которой она находится. Кроме этого будем считать, что построена сетка типа восьмеричное дерево, которую мы можем модифицировать. Итак, вместо точек в серединах ребер, интерфейсные точки на ребрах вычисляются с помощью метода бисекции по заданной функции. Как и в методе CMS, на каждом ребре должно быть не более одной интерфейсной точки. Для этого необходимо измельчить все ячейки, содержащие ребра с более чем одной интерфейсной точкой. В противном случае часть области будет отслежена сеткой недостаточно точно. В качестве интерфейсной точки на грани вместо центральной точки используется более точное положение. Для ее вычисления используется идея, аналогичная поиску особенной точки области внутри ячейки. Обозначим интерфейсные точки на ребрах за $\mathbf{s}_i$, проекции нормалей в этих точках на плоскость грани за $\mathbf{n}_i$. Искомую точку $\mathbf{p}$ будем находить как точку пересечения касательных элементов точек $\mathbf{s}_i$. Другими словами, $\mathbf{p}$ удовлетворяет системе (1.4).

В результате решения системы может получиться точка, не принадлежащая соответствующей грани. Это может быть вызвано неточностью входных данных или наличием особенностей на поверхностях интерфейсов.Кроме этого, это может произойти в случае, если одно ребро ячейки пересекает более чем один интерфейс. Как было отмечено ранее, в таком случае следует иерархически разбить ячейки, содержащие такое ребро. Если интерфейсную точку на грани сразу найти не удалось, то можно сделать несколько уровней иерархического измельчения грани, и попробовать найти ее в одной из полученных дочерних граней. Стоить заметить, что этот процесс измельчения можно применить только для поиска интерфейсной точки на грани, но при этом итоговую сетку оставить без изменений, если измельчение сетки нежелательно. Для простоты можно воспользоваться альтернативным способом выбора интерфейсной точки на грани, используя, например, центр масс точек $\mathbf{s}_i$. Однако, при неточном выборе интерфейсной точки на грани, на сетке могут присутствовать визуальные артефакты, например, появление “зубчиков”, см. Рис. 1.7.

Определенную сложность представляет то, что некоторые интерфейсные точки на ребрах могут совпадать с вершинами ребер. В некоторых случаях

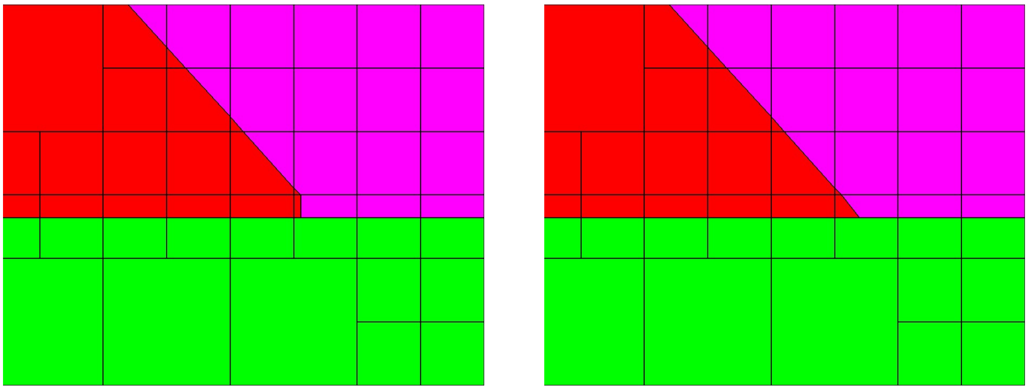


Рис. 1.7. Появление “зубчика” (слева) при неточном поиске интерфейсной точки на грани.

это может привести к тому, что на гранях появляются дополнительные точки или отрезки на этапе замыкания контуров или построения триангуляции контуров. Напомним, что для каждой пары соседних ячеек алгоритм должен создавать одинаковые контуры на их общей грани, иначе получаемая триангуляция будет не конформной. Поэтому, в предлагаемой модификации, если на грани появляется новая точка или отрезок, то их следует учитывать одновременно со стороны обеих ячеек, разделяющих эту грань.

Рассмотрим простой пример, в котором одна из граней приграничной ячейки находится ровно на плоской границе, см. Рис. 1.8. Все вершины данной ячейки имеют одинаковый индекс подобласти, следовательно на ней не будет строиться триангуляция. Однако, если рассмотреть ячейку, соседнюю с ней по грани на границе, то в ней строится триангуляция, которая полностью попадает на общую грань этих ячеек. Объем части этой ячейки, попавшей в область, равен нулю, поэтому она удаляется из сетки, однако при этом мы оставляем триангуляцию для унификации следа сетки на границах и интерфейсах. Таким образом, эта триангуляция добавляется в рассматриваемую приграничную ячейку. Это приводит к тому, что след сетки на поверхности границы области, а также на интерфейсах, представляет собой триангуляцию. Поэтому внешне сетка всегда выглядит как поверхностная триангуля-

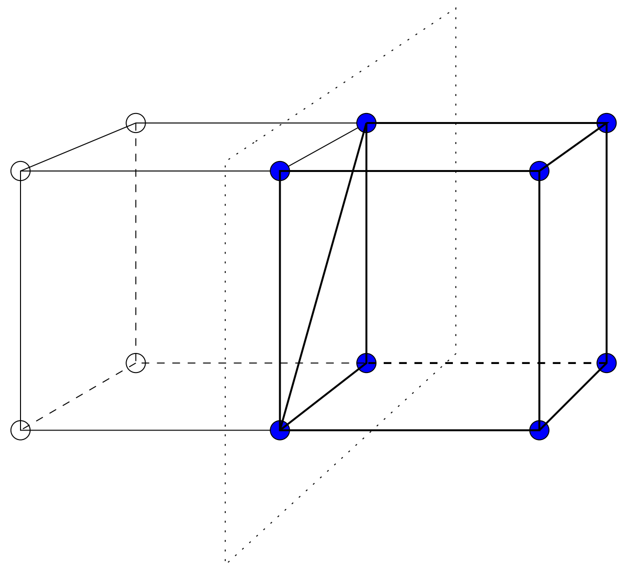
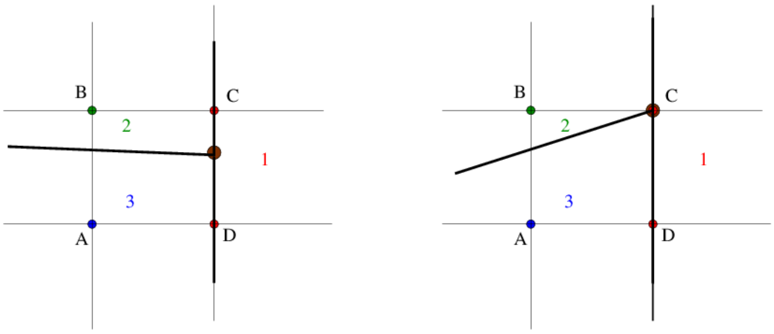


Рис. 1.8. Пример триангуляции на граничной грани

ция, хотя и состоит из кубических и сколотых ячеек. Заметим, что в случае плоских сколов, треугольники могут быть легко заменены на многогранники, что приведет к уменьшению количества граней в сетке.

Рассмотрим две соседние ячейки и их общую грань после применения к ней алгоритма M^{3}S. Заметим, что если со стороны одной из ячеек на грани нужно добавить отрезок, вершины которого уже имеются на грани, то это можно сделать без каких-либо дополнительных изменений со стороны второй ячейки. Действительно, добавление нового отрезка лишь разделяет на части грань, соответствующую одной подобласти.

Когда некоторые интерфейсные точки на ребрах попадают в вершины грани, интерфейсные точки на грани также могут совпасть с вершиной грани, либо попасть на ее ребро. Действительно, для примера рассмотрим грань, в которой две соседние вершины имеют индекс подобласти 1, две другие - индексы 2 и 3 соответственно, см. Рис. 1.9. Если интерфейсная точка на ребре BC совпадет с C, а интерфейсная точка на ребре AD совпадет с D, то, очевидно, интерфейсная точка на грани $ABCD$ будет лежать на ребре CD. Причем, если интерфейсная точка на ребре CD совпадет, скажем, с вершиной



(*а*) (*б*)

Рис. 1.9. Попадание интерфейсной точки на грани на ребро 1.9, *а* и в вершину 1.9, *б*.

C, то тогда интерфейсная точка на грани также совпадет с вершиной C, см. Рис. 1.9 (*б*).

В большинстве случаев попадание интерфейсной точки грани на ребро или в вершину грани не приводит к изменениям алгоритма. Однако, в некоторых ситуациях это может привести к появлению новых точек или отрезков на гранях. Это может произойти после того, как мы проводим дополнительные отрезки, чтобы контуры стали замкнутыми. Изучим такие ситуации подробнее.

Рассмотрим две соседние грани f и g одной кубической ячейки и их общее ребро e. Обозначим за $\mathbf{p}_f$ интерфейсную точку на грани f. Пусть оказалось, что $\mathbf{p}_f \in e$. Если концы ребра e имеют одинаковый индекс, то на нем не должно быть интерфейсной точки, следовательно, $\mathbf{p}_f$ будет новой точкой на этом ребре. Об этой точке должны “узнать” остальные ячейки, разделяющие ребро e. Для этого проводится постпроцессинг, который проходит по всем треугольникам, содержащим ребро e. Каждый такой треугольник разделяется на два треугольника отрезком, соединяющим точку $\mathbf{p}_f$ и третью вершину треугольника, не лежащую на ребре e. Для остальных граней, не являющихся треугольными, и содержащих e, точка $\mathbf{p}_f$ просто увеличивает число вершин грани на единицу.

Далее, в зависимости от расположения интерфейсной точки на второй грани g, рассмотрим несколько вариантов:

- Если у g нет интерфейсной точки, то на грани g не будут построены дополнительные отрезки.

- Пусть $\mathbf{p}_g$ - интерфейсная точка на грани g и $\mathbf{p}_g \in e$. Так как на одном ребре может быть не более одной интерфейсной точки, то $\mathbf{p}_f = \mathbf{p}_g$, а, значит, общее количество интерфейсных точек на гранях ячейки уменьшится на один. Покажем, что это не приведет к изменению алгоритма замыкания контуров. Отличие может возникнуть, если новое количество интерфейсных точек на гранях стало равно одному. Однако, это эквивалентно ситуации, когда их число равно нулю, то есть все компоненты контуров являются замкнутыми, см., например, Рис. 1.10(a).

- Пусть $\mathbf{p}_g \notin e$. Если концы e имеют разные индексы, то на нем есть интерфейсная точка, в которую и попала точка $\mathbf{p}_f$. Если общее количество интерфейсных точек на гранях в ячейке равно двум, то, согласно алгоритму, мы должны соединить отрезком интерфейсные точки $\mathbf{p}_g$ и $\mathbf{p}_f$. Но $\mathbf{p}_f$ уже была соединена с $\mathbf{p}_g$ (как интерфейсная точка на ребре с интерфейсной точкой на грани), а значит отрезок, соединяющий $\mathbf{p}_g$ и $\mathbf{p}_f$, не является новым для грани g, см. Рис. 1.10($б$).

  Если концы ребра e имеют одинаковый индекс, то на нем нет интерфейсной точки. Если нужно соединить точки $\mathbf{p}_g$ и $\mathbf{p}_f$, то этот отрезок будет новым для грани g, см. Рис. 1.10($в$). Этот отрезок нужно учитывать со стороны соседней по грани g ячейки. Как было сказано ранее, это не приводит к изменениям алгоритма для этой ячейки, однако нужно учесть наличие новой точки на ребре с помощью постпроцессинга, описанного ранее.

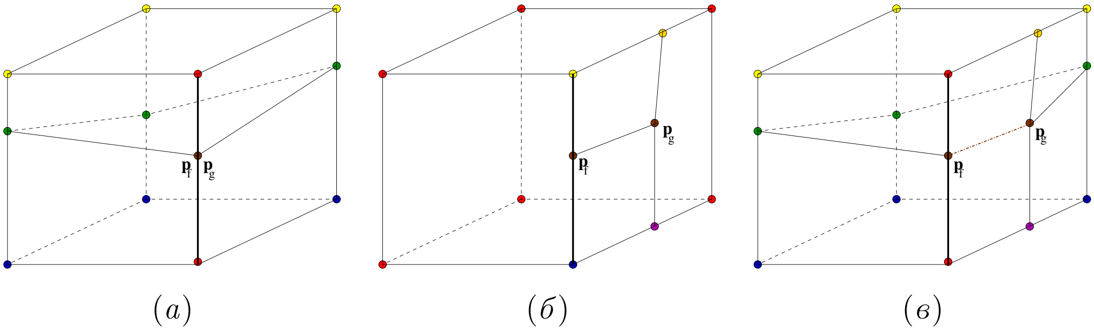


Рис. 1.10. Попадание на ребро интерфейсной точки на грани

Случай, когда интерфейсная точка на грани попадает в вершину, аналогичен случаю попадания на ребро, при этом новые точки на ребрах не появляются.

Интерфейсная точка ячейки также выбирается более точно. Она считается аналогично интерфейсной точке на грани, по формуле (1.4). При этом в формулу входят все интерфейсные точки на ребрах. На практике возникают случаи, когда полученная точка не принадлежит рассматриваемой ячейке. Тогда для поиска интерфейсной точки в ячейке можно воспользоваться иерархическим измельчением ячейки, как и в аналогичном случае для интерфейсной точки на грани. Если точность сетки не важна, то, для простоты, можно выбрать альтернативный вариант, например, принять за интерфейсную точку ячейки центр масс интерфейсных точек на гранях.

На положение интерфейсной точки ячейки также влияет попадание на ребра и в вершины ячейки интерфейсных точек на гранях. Когда на одну грань попадает три или более интерфейсных точек на грани, интерфейсной точкой ячейки автоматически становится интерфейсная точка этой грани. При этом случаи появления новых отрезков на грани аналогичны рассмотренным ранее.

1.3. Общий алгоритм построения сетки

Заметим, что описанная выше модификация метода M^3C может быть использована на неравномерных сетках типа восьмеричное дерево. Отличия в алгоритме могут возникнуть лишь на переходных гранях. Возможна ситуация, когда переходная грань ячейки имеет более одной интерфейсной точки на грани. В этом случае можно модифицировать алгоритм замыкания контуров, однако лучшим решением будет разбить такую ячейку. Это связано с тем, что в таких случаях область представляет собой локально довольно сложную конструкцию. Отметим, что при этом минимальный размер ячеек сетки не изменяется, так как соседние по переходной грани ячейки имеют меньший размер.

Таким образом, метод CMS является частным случаем этой модификации метода M^3C. Соответственно, полученная модификация имеет преимущества обоих методов:

- возможность работы на сетках типа восьмеричное дерево,
- возможность работы в составных областях.

Модифицированный метод будем называть методом кубических марширующих квадратов для областей с несколькими материалами (Multiple material cubical marching squares, MMCMS). Опишем общий алгоритм построения сетки типа восьмеричное дерево со сколотыми ячейками для составной области Ω.

Будем считать, что входными данными является функция, которая для любой точки возвращает индекс подобласти, которой она принадлежит, а также нормаль к внутренним или внешним границам, если точка находится вблизи границы. Такую функцию можно восстановить из различных типов входных данных. Например, если область задана поверхностной триангуляцией, то функцию можно получить с помощью алгоритма отслеживания лучей (ray-tracing). Стоит отметить, что вершины этой триангуляции, вообще говоря, не переносятся на итоговую сетку. Если точка находится вне области Ω, ей присваивается фиктивный индекс, обозначающий внешнюю подобласть.

Сначала строится кубическая сетка типа восьмеричное дерево. В данной работе процесс построения сетки типа восьмеричное дерево описываться не будет. Сетка может сгущаться к границе области, интерфейсам, вблизи особенностей и резких изменений области, либо к другим участкам. При этом сгущение может быть неравномерным для разных участков границы и интерфейсов. Как было отмечено ранее, если ячейка имеет ребро, которое пересекается с интерфейсами более чем в одной точке, то ее следует измельчить. В противном случае, будет учтена только одна из точек пересечения, и часть области будет потеряна.

В каждой вершине сетки известен индекс подобласти, которой она принадлежит. Для всех ячеек, у которых число различных индексов в вершинах больше одного, применяется метод MMCMS, описанный ранее. В результате, некоторые ячейки будут сколоты либо разбиты на части с соответствующими индексами подобластей. Ячейки, имеющие фиктивный индекс, удаляются из сетки. Отметим, что после применения метода MMCMS сетка может быть дополнительно локально измельчена или разгрублена.

Построение сетки типа восьмеричное дерево со сколотыми ячейками для области Ω может быть представлено в виде следующего алгоритма 1.5:

Алгоритм 1.5 Построения сетки типа восьмеричное дерево со сколотыми ячейками

Построить сетку $\mathcal{T}$ типа восьмеричное дерево для Ω
Присвоить вершинам сетки индексы подобластей
для $T \in \mathcal{T}$ **начало цикла**
    если Число различных индексов в T больше 1 **тогда**
        Применить алгоритм MMCMS для T
        Разделить ячейку T на части с помощью триангуляции
    конец если
конец цикла
Сделать постпроцессинг ▷ Описан в разделе 1.2.3
Удалить ячейки с фиктивным индексом

Стоить отметить, что после того, как мы делим кубическую ячейку на отдельные ячейки, то, формально, сетка перестает быть восьмеричным деревом, так как некоторые ячейки имеют отличное от 8 число потомков. Однако, для сохранения структуры восьмеричного дерева, будем считать, что листовые ячейки, разделенные на части, остаются листовыми с точки зрения структуры восьмеричного дерева, а сколотые подъячейки будем считать множеством, объединением которого является листовая ячейка. Такая формальная структура понадобится при динамическом перестроении сетки.

Также отметим, что ориентация получаемых треугольников, вообще говоря, не является согласованной, даже в случае однородной области. Поэтому, при необходимости, производится соответствующее переупорядочивание вершин треугольников при создании сколотых ячеек.

Напомним, что сетку типа восьмеричное дерево можно рассматривать как конформную многогранную сетку. Сетка со сколотыми ячейками также принадлежит классу конформных многогранных сеток, если область не раз-

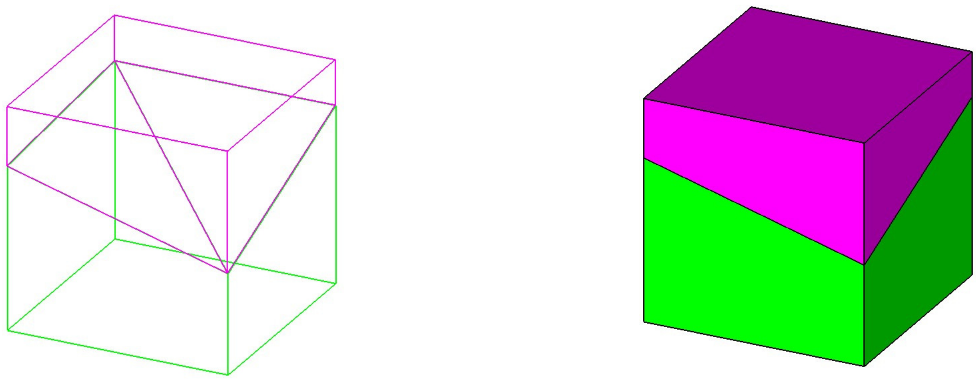


Рис. 1.11. Кубическая ячейка, поделенная на две части триангуляцией

деляется на подобласти. В случае, если область разделена на подобласти, некоторые соседние ячейки могут иметь более одной общей грани. Действительно, когда кубическая ячейка разбивается на части методом MMCMS, граница между этими частями представляет собой набор из нескольких треугольников, см. Рис. 1.11. Поэтому сетку со сколотыми ячейками для составных областей можно отнести к слабо-конформным многогранным сеткам.

1.4. Анализ алгоритма

Проведем анализ Алгоритма 1.5, покажем конечность его работы и оценим вычислительную сложность.

1.4.1. Конечность работы алгоритма

Процесс построения сетки типа восьмеричное дерево описан в различной литературе, см., например, [69, 84]. Он конечен при ограниченной глубине измельчения дерева. Процесс дополнительного измельчения сетки также ограничен заданной глубиной измельчения дерева.

Рассмотрим цикл по ячейкам в строке 3. Для каждой ячейки в алгоритме MMCMS в цикле по граням применяется модифицированный алгоритм M^3S.

В нем необходимо искать интерфейсную точку на грани. Как было сказано ранее, она может быть найдена решением линейной системы, либо с помощью измельчения сетки или альтернативным выбором точки на грани. Так как можно ограничить число уровней измельчения сетки при этом поиске, то этот процесс конечен. Далее, в некоторых ячейках нужно найти интерфейсную точку ячейки. Аналогично случаю интерфейсной точки на грани, процесс ее поиска является конечным.

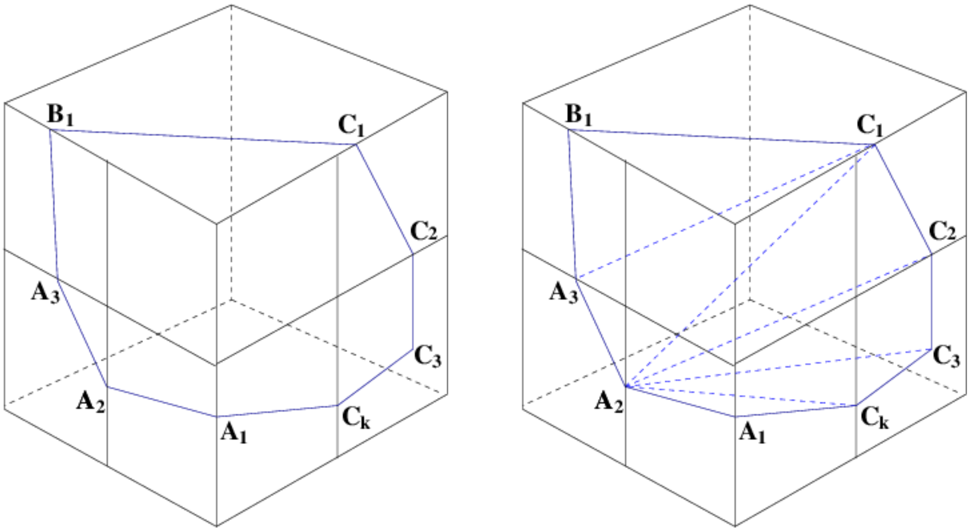
После этого находятся все замкнутые компоненты контура. Этот процесс конечен, так как каждая вершина контура имеет ограниченное число соседей. Далее, для каждого замкнутого контура строится триангуляция. Очевидно, что метод триангуляции “веер треугольников” конечен, так как требует лишь соединения центра триангуляции с остальными не соседними вершинами. Метод триангуляции “разделяй и властвуй” требует поиска подходящей разделяющей плоскости. Как отмечено в [83], для произвольных последовательностей вершин существуют примеры, в которых разделяющая плоскость не будет найдена данным алгоритмом. Однако, в [50] автор указывает, что такие случаи могут возникать лишь в более сложных последовательностях вершин, которые не возникают в методе M^{3}C. В любом случае докажем, что если не удалось построить триангуляцию методом “разделяй и властвуй”, то ее можно построить альтернативным способом.

Утверждение 1.4.1. *Пусть имеется ячейка сетки типа восьмеричное дерево, причем сетка удовлетворяет условиям, накладываемым алгоритмом MMCMS. Тогда процесс построения триангуляции для замкнутого контура, полученного алгоритмом, конечен.*

Доказательство.

Сначала рассмотрим случай, когда внутри ячейки не создается интерфейсная точка. Предположим, что не удалось построить триангуляцию методом “разделяй и властвуй”. Тогда она строится методом “веер треугольников”, где в качестве центра триангуляции берется особенная точка ячейки, либо центр масс интерфейсных точек. Если же ячейка имеет внутри интерфейсную точку, то все контуры, проходящие через нее, триангулируются этим же способом. Осталось рассмотреть случай, когда ячейка имеет интерфейсную точку, но контур не проходит через нее. Докажем, что если контур не проходит через интерфейсную точку ячейки, то он не проходит также ни через какую интерфейсную точку на грани. Предположим, что интерфейсная точка грани принадлежит последовательности и рассмотрим две ее соседние точки. Ни одна из них не является интерфейсной точкой ячейки, так как контур не проходит через нее. В то же время, ни одна из них не является другой интерфейсной точкой грани, так как такие точки не могут быть соединены в данном случае. Следовательно, оба ее соседа - это интерфейсные точки на ребрах, причем на этой же грани. Этим точкам приписаны ровно по два индекса подобластей, которые они разделяют, причем эти пары индексов не совпадают. Но тогда эти две точки не могут принадлежать одному контуру, так как контуру соответствует только одна пара индексов. Таким образом, контур не проходит через интерфейсные точки на гранях, и, следовательно, не проходит через грани, на которых есть такая интерфейсная точка. Так как имеется интерфейсная точка внутри ячейки, то согласно алгоритму замыкания контуров, число интерфейсных точек на гранях не меньше трех. Следовательно, число граней, на которых нет интерфейсных точек, не больше трех.

Рассмотрим случай, когда таких граней ровно три, и контур проходит через них. Очевидно, что они имеют одну общую вершину, иначе контур был бы незамкнутый. Для наглядности обозначим последовательно вершины за $A_1, A_2, ..., A_n, B_1, B_2, ..B_m, C_1, ..., C_k$, причем A_1, B_1, C_1 - интерфейсные точки на ребрах, см. Рис. 1.12. Рассмотрим первую вершину, не совпадающую с



(*а*) (*б*)

Рис. 1.12. Замкнутый контур (*а*), его триангуляция (*б*).

A_1, B_1, C_1. Не умаляя общности можно считать, что это A_2. Соединим вершину A_2 со всеми остальными вершинами последовательности, не лежащими с ней на одной грани: $B_2, ..., B_m, C_1, ..., C_k$. Далее, соединив B_2 (может совпадать с C_1) с оставшимися вершинами $A_3, ..., A_n$, мы получим подходящую триангуляцию последовательности.

Если таких граней две, то рассмотрим их общее ребро. Так как контур замкнутый, то он пересекает это ребро в двух точках (как минимум одна из них совпадает с вершиной ребра), которые разделяют контур на две части. Соединив последовательно пары вершин из разных частей, получим искомую триангуляцию. □

Таким образом, если алгоритм "разделяй и властвуй" не находит подходящую разделяющую плоскость и не может быть выполнен, то всегда можно построить триангуляцию альтернативным способом. Однако, на практике оказывается достаточным использовать алгоритм "разделяй и властвуй". Условия для сетки, упоминаемые в Утверждении 1.4.1, связаны с необходимым измельчением сетки и описаны ранее в алгоритме.

1.4.2. Вычислительная сложность алгоритма

Проведем краткий анализ вычислительной сложности работы Алгоритма 1.5. Будем считать, что область задается функцией $M : \mathbb{R}^3 \to \mathbb{Z}$, возвращающую индекс подобласти, которой принадлежит точка (x, y, z). В зависимости от типа входных данных, вызов этой функции может быть относительно дорогой, поэтому будем учитывать ее сложность cM.

Максимальная сложность построения сетки типа восьмеричное дерево - порядка $O(N \ln N)$, где N - число вершин дерева.

Пусть имеется сетка типа восьмеричное дерево, $N_{\mathcal{V}}$ - число вершин сетки, $N_{\mathcal{T}}$ - число ячеек. Для оценки вычислительной сложности построения сколотых ячеек рассмотрим следующие основные операции:

1. Поиск интерфейсной точки на ребре и нормали к ней;

2. Поиск интерфейсной точки на грани и в ячейке;

3. Построение триангуляции замкнутого контура.

Рассмотрим ребро ячейки, концы которого имеют разные индексы. Для нахождения интерфейсной точки на ребре необходимо сделать Nb шагов метода бисекции, где Nb зависит от желаемой точности сетки. На каждом шаге один раз вызывается функция M, значит общая сложность равна $Nb \cdot cM$. В гексаэдральной сетке типа восьмеричное дерево имеется следующая оценка на количество ребер: $N_{\mathcal{E}} \approx 3N_{\mathcal{V}}$. Предположим, что на каждом ребре может находиться интерфейсная точка, тогда максимальная сложность поиска всех интерфейсных точек пропорциональна $Nb \cdot cM \cdot 3N_{\mathcal{V}} = O(N_{\mathcal{V}} \cdot cM)$. Для вычисления нормали, в зависимости от типа входных данных, используется один из способов, описанных в [60]. Так, например, если область задается поверхностной триангуляцией, то используется нормаль к ближайшему треугольнику.

Для поиска интерфейсной точки на грани необходимо решить систему линейных уравнений с матрицей порядка 3×2 или 4×2. В случае, если требуется сделать дополнительно не более k измельчений сетки, то необходимо решить еще k подобных систем. Для поиска интерфейсной точки ячейки необходимо решить систему линейных уравнений с матрицей порядка не более 12×3, либо дополнительно k систем, если необходимо измельчать сетку.

При построения триангуляции контура методом "разделяй и властвуй" имеется два вложенных цикла, каждый из которых ограничен количеством N_L точек в контуре, а также рекурсия для каждой части контура. На каждой из 6 граней ячейки восьмеричного дерева может находиться не более 4 отрезков контура для листовой грани и не более 8 отрезков для переходной. Следовательно, число N_L не превосходит 48, хотя существует более точная оценка.

Заметим, что применение алгоритма MMCMS является локальным, и его сложность для одной ячейки не зависит от общего числа ячеек сетки. При построении сколотой ячейки используются только соседние ячейки. Таким образом, общая сложность алгоритма пропорциональна $c_1 N_{\mathcal{V}} + c_2 N_{\mathcal{V}} \ln N_{\mathcal{V}}$, причем c_1, c_2 линейно зависят от cM. Причем сложность построения сколотых ячеек является линейной.

1.5. Дополнительные операции с сеткой

В этом разделе опишем некоторые дополнительные операции над полученной сеткой. Во-первых, будут рассмотрены способы улучшения качества сетки. Во-вторых, будет описан алгоритм динамического перестроения сетки.

1.5.1. Улучшение качества полученной сетки

После построения сетки, некоторые сколотые ячейки могут иметь очень маленький объем. Это отрицательно сказывается при дискретизациях на таких сетках. Поэтому предлагается алгоритм объединения маленьких ячеек с соседними по граням. Кроме критерия маленького объема может быть какой-либо другой, например, толщина ячейки. Рассмотрим ячейку, которую мы хотим объединить с соседней. Среди соседних ячеек выбираются те, у которых такой же индекс подобласти с данной ячейкой. Далее выбирается ячейка, с которой их общая грань имеет наибольшую площадь. Эта ячейка объединяется с рассматриваемой простым удалением их общей грани. При этом ребра удаленной грани остаются в ячейке. Стоит отметить, что, вообще говоря, у объединяемых ячеек могут быть разные родительские ячейки. Это следует учитывать при динамическом перестроении сетки.

Следующим способом улучшения сетки является удаление излишних граней. Когда кубическая ячейка разделяется на части, то полученные ячейки граничат между собой некоторым количеством треугольных граней. Очевидно, что если общие треугольные грани двух ячеек лежат в одной плоскости, то их можно заменить на многоугольную грань. Аналогично, в случае плоских сколов граничных ячеек, триангулированные сколы могут быть заменены многоугольными. В таком случае, в частности, поверхностная сетка для простого куба будет выглядеть как обычная квадратная сетка, а не треугольная, см. Рис. 1.13*(а,б)*. Кроме визуальных эффектов, это приводит к упрощению дискретизаций на таких сетках. Заметим, что можно также объединять треугольники, почти лежащие в одной плоскости, то есть угол между нормалями которых меньше некоторого порога, если в сетке допускаются не плоские грани. На Рис. 1.13*(в,г)* изображены сетки для шара с треугольными и многоугольными гранями на границе. При этом объединялись соседние грани, у

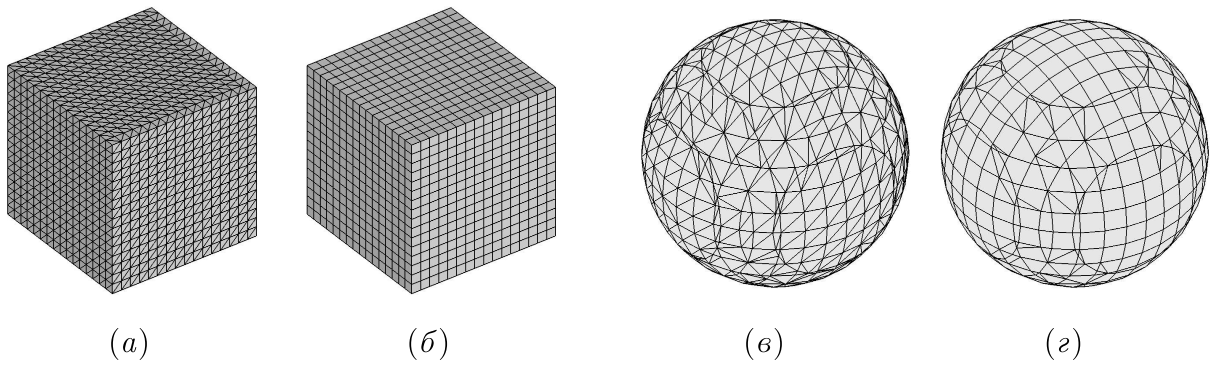


Рис. 1.13. Сетки для куба и шара с (*а*),(*в*) треугольными и (*б*),(*г*) многоугольными гранями на границе.

которых единичные нормали $\mathbf{n}_1, \mathbf{n}_2$ удовлетворяют условию $(\mathbf{n}_1, \mathbf{n}_2) \geq 0.99$. Количество граней на границе равно 1208 и 708, соответственно.

1.5.2. Динамическое перестроение сетки

Для получения высокой точности решения численных задач расчетная сетка должна адаптироваться к изменению сеточной функции на каждом шаге по времени. При этом могут быть использованы различные критерии адаптации сетки. Это может быть вариация функции в ячейках, изменение градиента функции, кинетической энергии поля и др.

Одним из основных преимуществ сеток типа восьмеричное дерево является возможность локального динамического перестроения сетки. В таких сетках можно измельчить ячейку или объединить несколько ячеек, при этом сетка изменится только локально в окрестности этих ячеек. Сетки типа восьмеричное дерево со сколотыми ячейками также могут быть динамически перестроены. Кратко опишем процесс динамического перестроения сеток типа восьмеричное дерево со сколотыми ячейками.

Пусть имеется сетка типа восьмеричное дерево со сколотыми ячейками и отмечены листовые ячейки, которые должны быть измельчены, а также родительские ячейки, потомки которых должны быть объединены. Напомним также, что при перестроении сетки типа восьмеричное дерево необходимо учитывать то, что размеры любых двух соседних ячеек не могут отличаться более чем в два раза. На этом этапе сетка рассматривается без учета сколотых ячеек, а только как сетка типа восьмеричное дерево. При измельчении ячейки в такой сетке должны быть перестроены также все соседние по граням и ребрам ячейки, имеющие размер, больший либо равный размеру данной ячейки, так как на них появляются новые вершины и ребра. Если эта ячейка имела сколотые подъячейки, то при ее измельчении они также удаляются, а при создании новых, меньших ячеек, к каждой из них применяется алгоритм скалывания. При этом будут получены некоторые новые вершины, ребра и грани, однако они будут также учтены у всех соседних ячеек. Действительно, некоторые из них будут также перестроены, а остальные имеют такой же размер, как и новые ячейки, значит, все контуры на общих гранях и ребрах будут одинаковыми. Аналогично, при объединении листовых ячеек, также перестраиваются все их соседние ячейки большего размера, так как у них пропадают некоторые вершины, ребра и грани. Если эти листовые ячейки имели сколотые подъячейки, то они удаляются, а к их родительской ячейке применяется алгоритм скалывания. В этом случае, аналогично, все новые элементы сетки будут учтены соседними ячейками.

Как было отмечено ранее, при динамическом перестроении сетки необходимо следить за объединенными ячейками, см. раздел 1.5.1. Если объединенные ячейки имели разных родителей, то когда один из них перестраивается, то также должен быть перестроен второй, после чего можно снова объединять полученные новые ячейки.

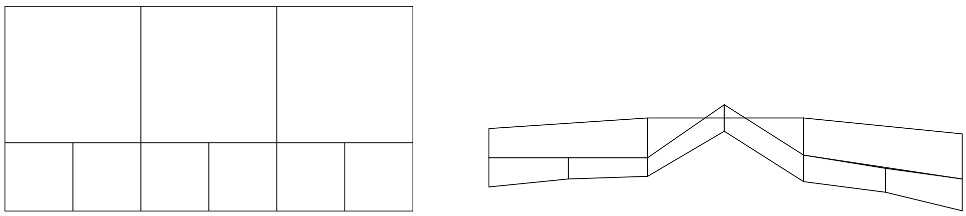
1.6. Сетки для слоистых областей

В данном разделе опишем алгоритм построения сеток для слоистых областей специального вида. Сетки получаются в результате отображения сетки типа восьмеричное дерево со сколотыми ячейками вдоль оси Oz. В результате получается сетка, ячейки которой располагаются вдоль искривленных слоев области. Такие сетки могут быть построены для геологических областей пластовых сред, допускающих выклинивания слоев и разломы.

Рассмотрим область $\mathcal{D}$, состоящую из почти горизонтальных слоев. Введем оператор проектирования $\mathbf{P} : \mathbb{R}^3 \to \mathbb{R}^2$ на плоскость $z = 0$. Обозначим двумерную область $\Gamma = \mathbf{P}(\mathcal{D})$ и ее границу $\partial\Gamma$. Будем считать, что область $\mathcal{D}$ представлена набором интерфейсов $\mathcal{I}$ и слоев $\mathcal{L}$. Каждый слой $L_j \in \mathcal{L}$ находится между интерфейсами I_j и $I_{j+1} \in \mathcal{I}$. Кроме этого, в области могут быть разломы, представляющие собой наклонные поверхности $\mathcal{R}$. Будем считать, что каждый интерфейс I_j задан некоторой функцией fI_j, которая для любой точки $(x, y) \in \Gamma$ возвращает z-координату соответствующей точки на интерфейсе:

$$z = fI_j(x, y).$$

Геологические интерфейсы могут пересекаться, а соответствующие геологические слои выклиниваться. В нашей модели сначала будем рассматривать только не выклинивающиеся слои. Если слой выклинивается, то будем его временно объединять с соседним слоем, временно удаляя интерфейс между ними. Таким образом, будем рассматривать только непересекающиеся интерфейсы. После построения основной части сетки, удаленные интерфейсы будут аппроксимированы с помощью сколотых ячеек. Стоит отметить, что можно расширить алгоритм построения сетки, сразу учитывая наличие выклинивающихся слоев, однако в полученной сетке будет сложно сохранить структуру восьмеричного дерева, что не позволит ее динамически перестраивать опи-



(a) ($б$)

Рис. 1.14. (a) Прообраз сетки (в двумерном случае). ($б$) Отображенная сетка с вывернутой ячейкой.

санным ранее способом.

Алгоритм построения сетки для такой области будет состоять из следующих шагов. Сначала из всех интерфейсов выбираются непересекающиеся. Обозначим их количество за N_I. Остальные интерфейсы маркируем как пересекающиеся, они будут использованы позже. Затем строится сетка типа восьмеричное дерево со сколотыми ячейками в области, состоящей из $N_I - 1$ ровных горизонтальных слоев $p\mathcal{L}$, разделенных интерфейсами $p\mathcal{I}$. Вообще говоря, сетка может сгущаться к различным местам области. Однако, для сильно искривленных слоев $\mathcal{L}$ алгоритм будет надежным только в случае, если измельчение сетки типа восьмеричное дерево в вертикальном направлении происходит полностью. Другими словами, сетка должна быть представима в виде тензорного произведения двумерной сетки типа четвертичное дерево в плоскости Oxy и одномерной сетки вдоль оси Oz. Такое ограничение обусловлено тем, что, в противном случае, в результате кусочно-линейного отображения этой сетки вдоль оси Oz, некоторые ячейки могут оказаться вывернутыми, см. Рис. 1.14. Область является составной, где каждый из слоев является подобластью. Каждый интерфейс $pI_j \in p\mathcal{I}$ принадлежит плоскости $z = z_j$ для некоторого числа z_j. Боковой границей области является цилиндрическая поверхность $\partial\Gamma \times [z_1, z_{N_I}]$. Для аппроксимации боковой границы используются сколотые ячейки. Полученную сетку назовем *прообразом* сетки.

После этого мы можем построить семейство кусочно-линейных отображений F для z-координат вершин прообраза сетки, переводящее ровные слои $p\mathcal{L}$ в искривленные $\mathcal{L}$. Определим это отображение следующим образом. Рассмотрим точку из прообраза $(x, y, z) \in pL_j$, тогда $z \in [z_j, z_j + 1)$, где z_i определяется положением интерфейса pI_i. Тогда F линейно по z на отрезке $[z_j, z_j + 1]$, причем $F(x, y, z_j) = fI_j(x, y)$ и $F(x, y, z_{j+1}) = fI_{j+1}(x, y)$. Таким образом, точка (x, y, z) получит новую координату $\tilde{z} = F(x, y, z)$. В результате такого отображения, все интерфейсы pI_j прообраза сетки попадут на интерфейсы I_j.

В результате получится сетка, состоящая из гексаэдров, а также из многогранных сколотых ячеек, аппроксимирующих боковую границу области. После этого можно добавить в сетку маркированные интерфейсы, а также разломы. Поверхности этих интерфейсов и разломов можно аппроксимировать сеткой, применив алгоритм скалывания для ячеек, пересекаемых этими поверхностями. Если они пересекают некоторую ячейку, то генератор будет считать, что точки, лежащие по разные стороны относительно каждой из поверхностей, имеют различный индекс подобласти.

Описанный выше метод может быть сформулирован в виде алгоритма 1.6 с использованием введенных ранее обозначений.

Алгоритм 1.6 Построения сетки в слоистой области

Вход: Интерфейсы $\mathcal{I}$, разломы $\mathcal{R}$, боковая граница $\partial\Gamma$
$N_I = 0$
для $I \in \mathcal{I}$ **начало цикла**
    если I не пересекается с другими интерфейсами **тогда**
        $N_I = N_I + 1$
    иначе
        Маркировать I как пересекающийся
    конец если
конец цикла
Создать сетку типа восьмеричное дерево из $N_I - 1$ горизонтальных слоев $p\mathcal{L}$ и N_I интерфейсов $p\mathcal{I}$
Сколоть ячейки, пересекающие $\partial\Gamma \times [z_1, z_{N_I}]$
для $V \in \mathcal{V}$ **начало цикла** ▷ Вершины сетки
    Получить (x, y, z) - координаты V
    Найти номер j такой что $V \in pL_j$
    Присвоить z_j и z_{j+1} - z координаты pI_j и pI_{j+1}
    Найти линейную функцию F, используя $F(x, y, z_j) = fI_j(x, y)$, $F(x, y, z_{j+1}) = fI_{j+1}(x, y)$
    Найти $\tilde{z} = F(x, y, z)$ и присвоить V новые координаты $(x, y, \tilde{z})$
конец цикла
для $T \in \mathcal{C}$ **начало цикла** ▷ Ячейки сетки
    если T пересекается с маркированным $I \in \mathcal{I}$ или $R \in \mathcal{R}$ **тогда**
        Применить алгоритм MMCMS для T
    конец если
конец цикла

Преимуществом таких сеток является то, что количество многогранных

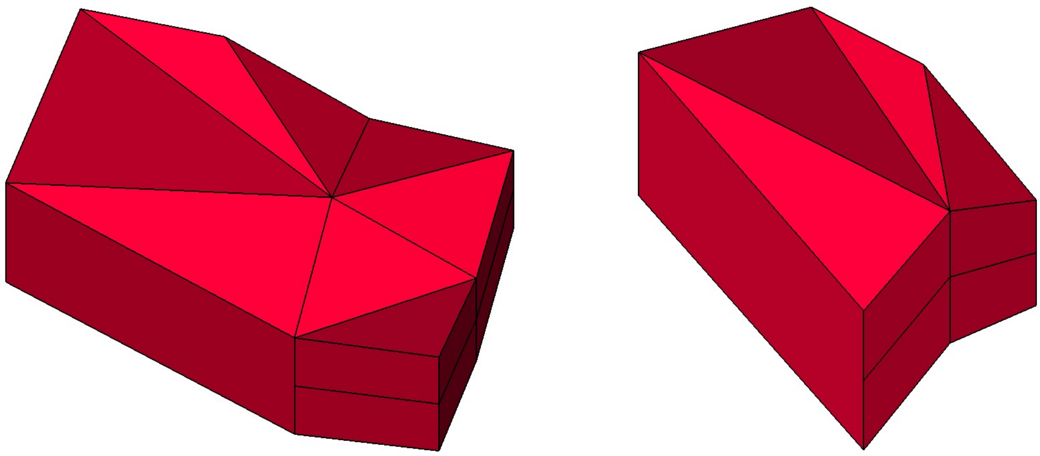


Рис. 1.15. Ячейки с неплоскими горизонтальными гранями

сколотых ячеек получается меньше, чем в сетках без отображений. Также они могут быть построены для областей с тонкими анизотропными слоями без измельчения сетки типа восьмеричное дерево. Сколотые ячейки используются только для аппроксимации боковых границ, а также пересекающихся интерфейсов и разломов. Однако в результате получается сетка с ячейками, горизонтальные грани которых, вообще говоря, не являются плоскими, см. Рис 1.15.

1.7. Примеры сеток

На основе Алгоритма 1.5 был программно реализован сеточный генератор, написанный на языке C++. В этом разделе представлены некоторые примеры сеток, построенных с помощью этого генератора. В данных примерах не использовался алгоритм объединения граней, лежащих в одной плоскости, поэтому сетка на поверхности всегда является треугольной. Кроме этого была создана версия генератора на платформе INMOST [1]. Преимущества этой платформы заключаются в том, что она предоставляет возможности для параллельного построения сеток, а также дискретизаций и решения алгебраических систем. Сетки, построенные этой версией генератора, будут также

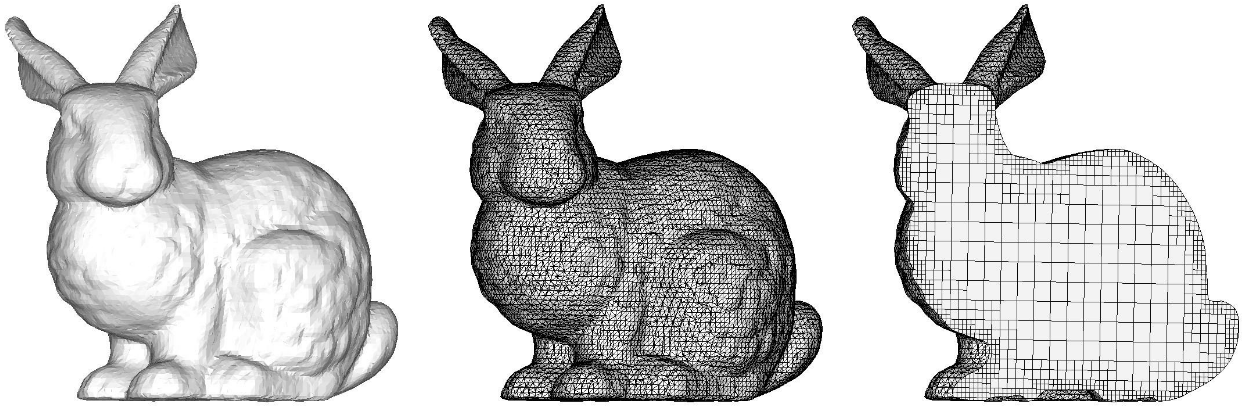


Рис. 1.16. Сетка для модели кролика, ее срез.

продемонстрированы.

1.7.1. Однородная область

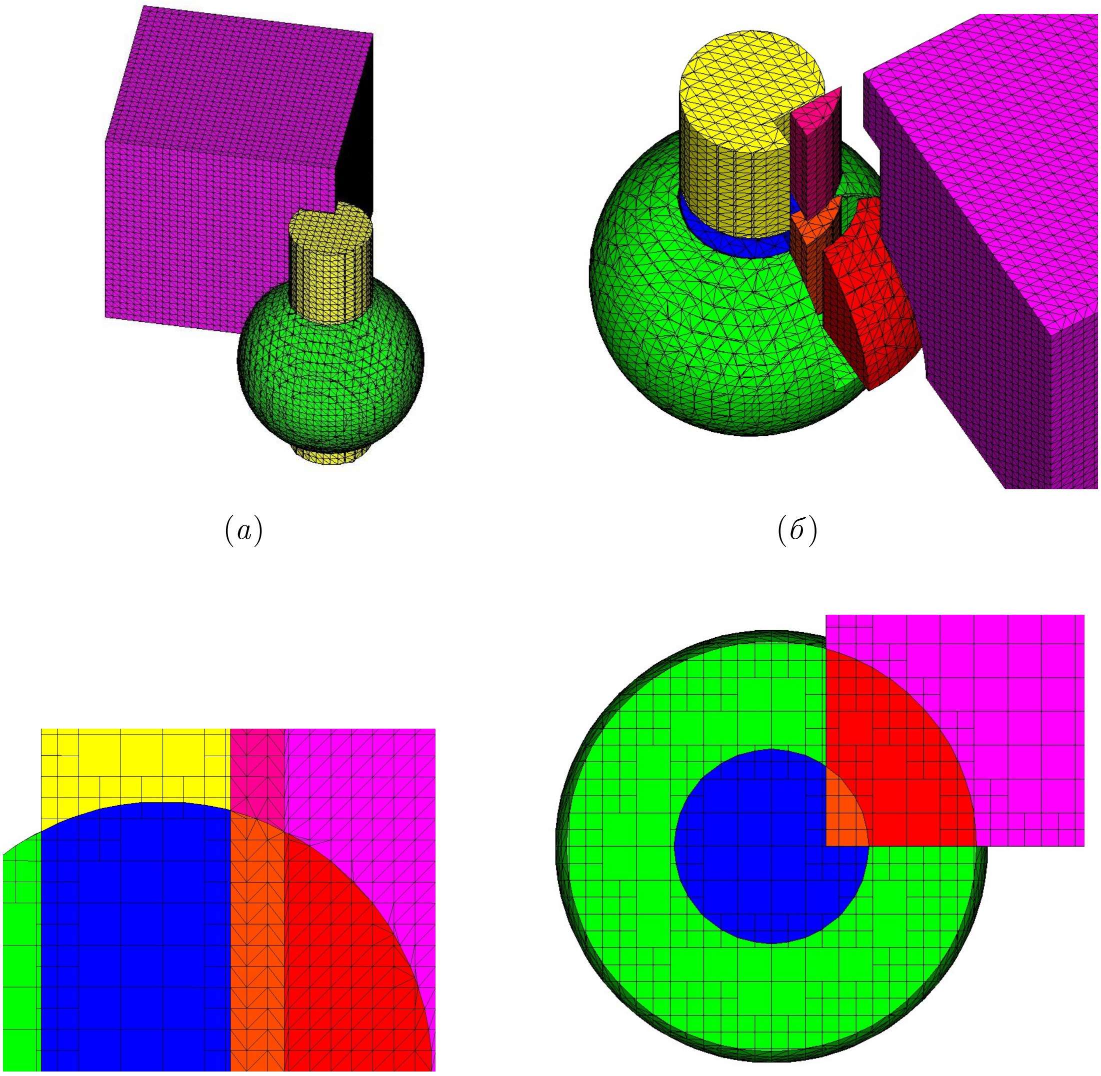
В качестве в однородной области рассмотрим модель Стэнфордского кролика [16], см. Рис. 1.16. Входными данными является поверхностная триангуляция области, с помощью которой была восстановлена характеристическая функция области. В полученной сетке 49997 ячеек, из них 14742 сколотых, 68244 вершины и 191059 граней. Сетка сгущается к границе области.

1.7.2. Составная область

Больший интерес представляют составные области, ведь построение сеток в таких областях является ключевым результатом работы.

Рассмотрим область, состоящую из пересечения примитивов: шара, параллелепипеда и цилиндра. Будем рассматривать области пересечения фигур как отдельные подобласти.

Подобласти задаются аналитическими функциями, поэтому сложность cM пренебрежительно мала. Построим последовательность сеток, которые на каждом шаге имеют более глубокий уровень измельчения приграничных ячеек. В данном примере используются следующие параметры, описанные в



(*а*) (*б*) (*в*) (*г*)

Рис. 1.17. Сетка для пересечения примитивов (*а*), ее внутренняя структура (*б*), увеличения у вертикального (*в*) и горизонтального (*г*) срезов.

Таблица 1.1. Параметры сетки для области из пересечения примитивов и время построения

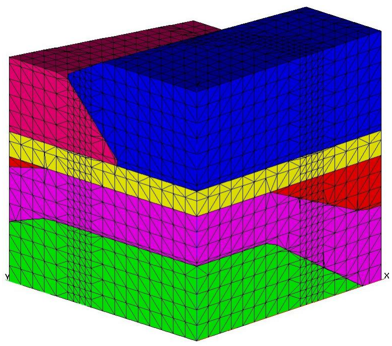
Уровень	Время, сек	# ячеек	# сколотых ячеек	# граней	# вершин
1	0.75	22073	7792	88645	29786
2	3.25	99206	31422	391859	134526
3	14.62	421 894	119306	1724179	604021

разделе 1.4.2: $Nb = 10, k = 1$. В таблице 1.1 представлены параметры полученных сеток, а также время их построения. В данном примере основное время работы алгоритма занимает построение сколотых ячеек. Можно заметить квазилинейную зависимость между числом сколотых ячеек и временем работы алгоритма. На Рис. 1.17 изображена сетка на первом уровне измельчения, а также некоторые ее срезы. Разные подобласти изображены разным цветом.

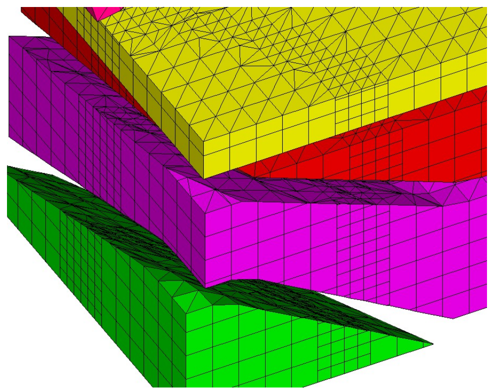
Далее, рассмотрим область, состоящую из шести геологических слоев. Область задается в виде поверхностной триангуляции каждого слоя, поэтому сложность cM в данном случае существенна. Построенная сетка сгущается к границе области и интерфейсам. Кроме этого, имеется дополнительное измельчение сетки вдоль некоторого фронта, см. Рис. 1.18. Для проверки вычислительной сложности генератора также была построена последовательность сеток с разными уровнями измельчения, измерено время работы алгоритма. Результаты представлены в Таблице 1.2. Из таблицы можно увидеть, что время работы алгоритма растет пропорционально росту количества вершин. На Рис. 1.18 изображена построенная сетка с первым уровнем измельчения, а также ее внутренняя структура. На нижних картинках можно увидеть интерфейсные точки на гранях.

Таблица 1.2. Параметры сетки для области из 6 слоев.

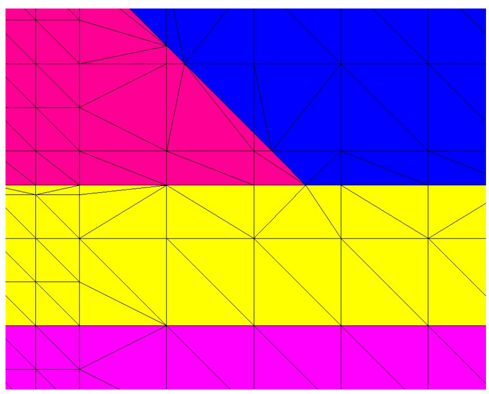
Уровень	Время, сек	# ячеек	# сколотых ячеек	# граней	# вершин
1	1.14	8293	3370	34354	11159
2	6.38	54846	13422	202180	66394
3	38.96	199991	66978	768287	439839



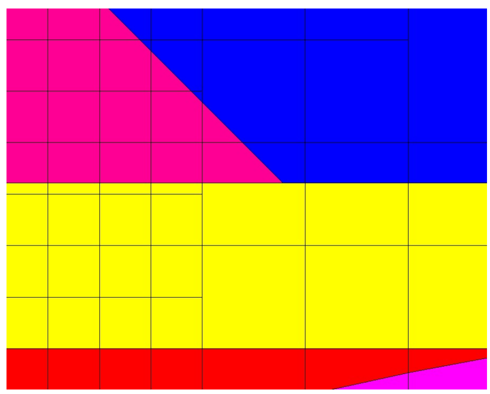
(a)



($б$)



($в$)



($г$)

Рис. 1.18. Сетка для 6 слоев (a), ее срез ($б$). Увеличения сетки в окрестности интерфейсной точки на грани ($в$),($г$).

1.7.3. Адаптирующиеся сетки

Для модели из шести слоев можно построить отображенную сетку, описанную в разделе 1.6. Построим набор динамически изменяющихся сеток такого вида.

Будем считать, что сетка адаптируется к положению фронта некоторого решения. Для примера фронт представляет собой часть окружности с центром в начале координат с увеличивающимся радиусом. На рисунке 1.19 показаны срезы сетки, динамически адаптирующейся к положению фронта. Данные сетки были построены с помощью генератора, реализованного на платформе INMOST. Также в данной версии генератора учитываются алгоритмы улучшения сетки. Поэтому на этих сетках поверхностная сетка является многоугольной, а не треугольной.

1.7.4. Приложения сеток

Генератор сеток типа восьмеричное дерево со сколотыми ячейками для слоистых областей используется в расчетном комплексе GeRa (Geomigration of Radionuclides - совместный проект ИВМ и ИБРАЭ РАН в рамках проекта "Прорыв" ГК Росатом). В этом комплексе он является частью технологической цепочки расчетов. Кроме этих сеток в данном комплексе используются треугольные призматические сетки. На Рис. 1.20 изображено окно графического интерфейса комплекса GeRa. В нем можно видеть контур границы области, а также отмеченную на ней скважину. Справа изображена поверхность сетки типа восьмеричное дерево со сколотыми ячейками. Сетка сгущается к границе области, а также к скважине.

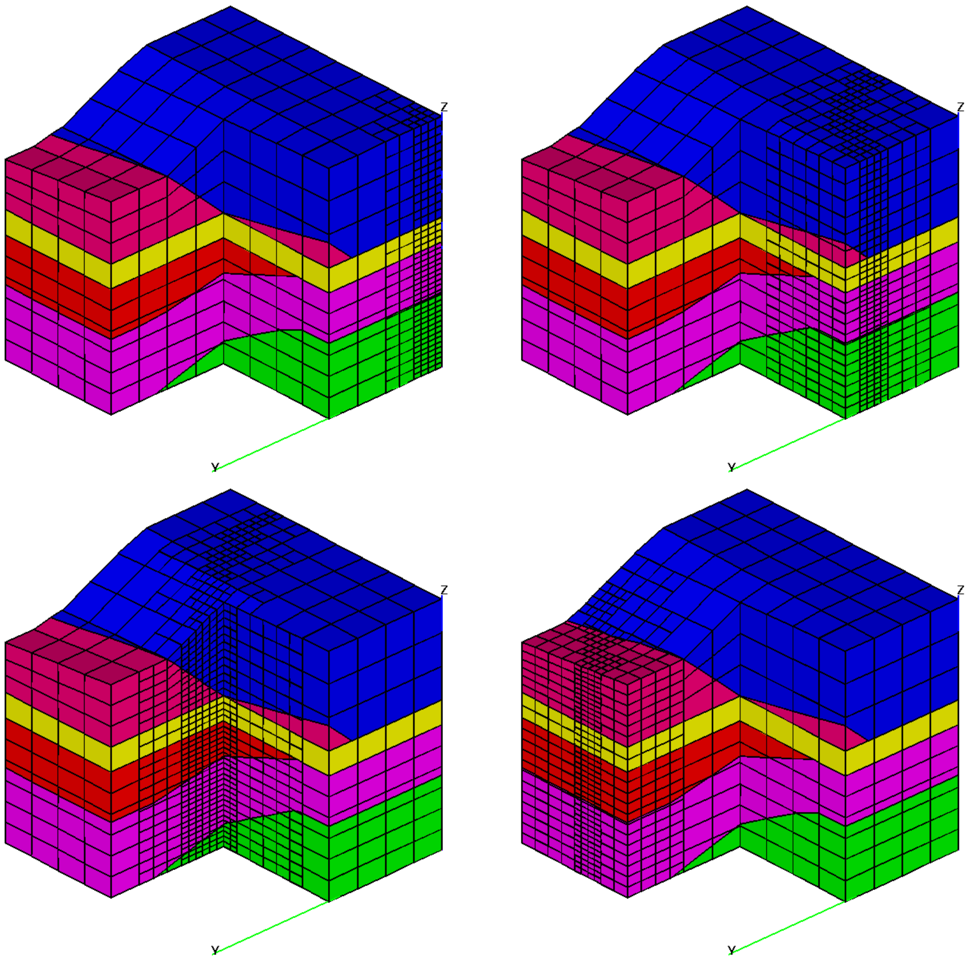


Рис. 1.19. Срезы динамически перестраивающейся сетки

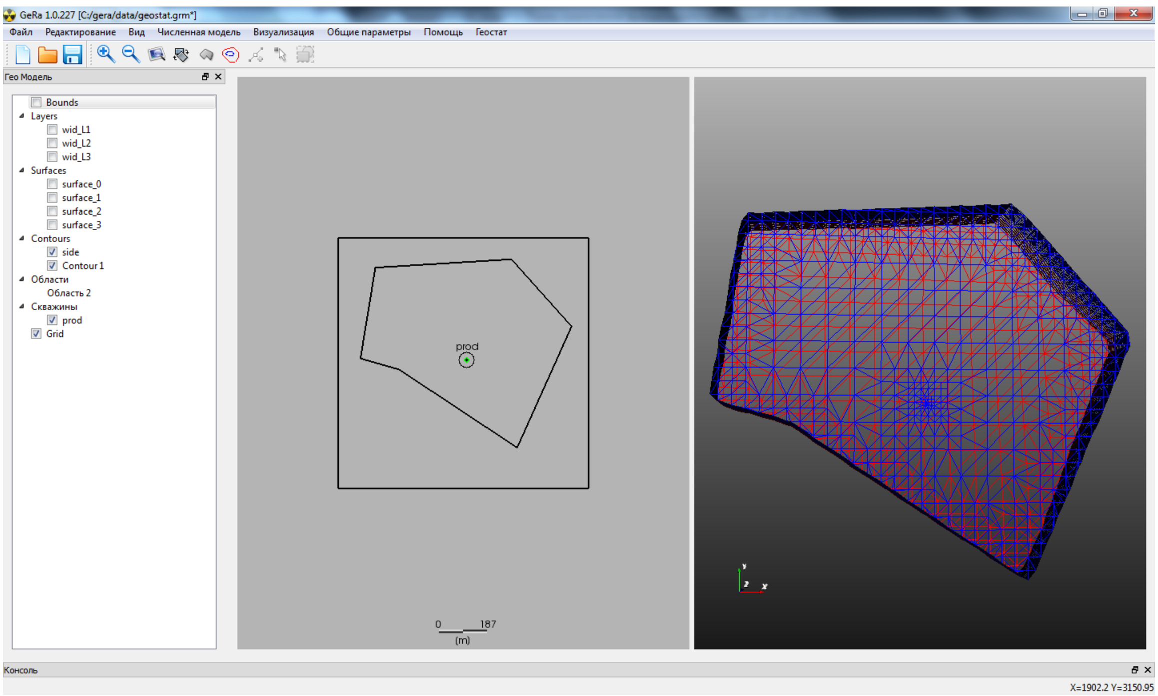


Рис. 1.20. Окно графического интерфейса комплекса GeRa.

1.8. Выводы к первой главе

В данной главе представлена технология построения сеток типа восьмеричное дерево со сколотыми ячейками. Сетки состоят преимущественно из гексаэдров, а также из многогранных ячеек. Они приближают границу области со вторым порядком точности для гладких границ. Алгоритм может быть применен к однородным и составным областям. Полученная сетка принадлежит классу конформных многогранных сеток в случае, если область является однородной. Если область разбита на подобласти, то сетка является слабо-конформной. Была доказана конечность алгоритма и показана его линейная сложность, а также приведены примеры работы алгоритма, подтверждающие данную оценку. Предложены алгоритмы улучшения сетки и ее динамического перестроения. Также предложен алгоритм построения сеток для слоистых областей специального вида.

Глава 2

Монотонный метод конечных объемов для трехмерной задачи диффузии

В этой главе будет предложена новая монотонная нелинейная схема конечных объемов для решения уравнения диффузии на многогранных сетках. Будет проведен теоретический анализ метода, а также проведены эксперименты для подтверждения дискретного принципа максимума и оценки порядка сходимости предложенной схемы. В экспериментах будут также использоваться сетки типа восьмеричное дерево со сколотыми ячейками.

2.1. Стационарное уравнение диффузии

Пусть Ω – трехмерная многогранная область, граница которой состоит из двух частей, $\Gamma = \Gamma_N \cup \Gamma_D$, причем множество Γ_D замкнуто и непусто, $\Gamma_D = \bar{\Gamma}_D$, $\Gamma_D \neq \emptyset$. Рассмотрим задачу диффузии для неизвестной концентрации c:

$$\begin{aligned} -\mathrm{div}(\mathbb{K}\nabla c) &= g && \text{в} && \Omega, \\ c &= g_D && \text{на} && \Gamma_D, \\ -\mathbf{n}\cdot\mathbb{K}\nabla c &= g_N && \text{на} && \Gamma_N, \end{aligned} \tag{2.1}$$

здесь $\mathbb{K}(\mathbf{x}) = \mathbb{K}^{\mathrm{T}}(\mathbf{x}) > 0$ – полный анизотропный диффузионный тензор, g – внешние источники, g_D и g_N - граничные условия Дирихле и Неймана, соответственно, и $\mathbf{n}$ – внешний нормальный вектор.

Принцип максимума заключается в том, что для $g < 0$ концентрация $c(\mathbf{x})$ не может достигать локального максимума в $\mathbf{x}$. Аналогично, принцип минимума утверждает, что для $g > 0$ концентрация $c(\mathbf{x})$ не может достигать локального минимума в $\mathbf{x}$. Эти два принципа обычно называют принципом максимума. Следствием этого является то, что для $g \leq 0$ в Ω концентрация

c достигает своего максимума на $\partial\Gamma$, либо если $g \geq 0$ в Ω, тогда c достигает своего минимума на $\partial\Gamma$.

Пусть везде $g \leq 0$ и $g_N \geq 0$ (выток). Если c достигает локальный максимум в $\mathbf{x} \in \Gamma_N$, то это будет точка втока. Следовательно, максимум будет достигаться на Γ_D, концентрация на которой известна заранее. Аналогично, можно сделать вывод, что при $g \geq 0$ и $g_N \leq 0$ (вток) минимум достигается на Γ_D.

Однако, если $g \geq 0$ везде и $g_N > 0$ где-либо, то минимум будет достигаться на границе, но заранее он не известен и может зависеть от g и g_N. Аналогичное утверждение для максимума можно сформулировать для случая $g \leq 0$ и $g_N < 0$.

2.2. Нелинейный метод конечных объемов на сетках с многогранными ячейками

Рассмотрим конформную сетку $\mathcal{T}$ из многогранников с плоскими гранями. Будем предполагать, что каждая ячейка является трехмерной ячейкой звездного типа по отношению к центру масс ячейки. Поэтому, для использования предлагаемого метода, пользователю следует предварительно разбить ячейки не звездного типа на подъячейки звездного типа (или на выпуклые), если такие имеются в сетке. Будем считать, что сетка $\mathcal{T}$ является связной по граням, то есть не может быть разбита на две подсетки, не имеющих общих граней. Также будем предполагать, что тензор $\mathbb{K}(\mathbf{x})$ в пределах одной ячейки меняется незначительно, однако он может значительно отличаться при переходе через грань от одной ячейки к другой, и, в частности, менять направления своих главных осей. В таких случаях мы будем говорить, что тензор является разрывным на соответствующей грани.

Обозначим через $\mathcal{F}_I$ и $\mathcal{F}_B$ множества внутренних и внешних граней, соответственно. Пусть подмножество $\mathcal{F}_J \subset \mathcal{F}_I$ содержит все грани, на которых тензор является разрывным. Будем говорить, что область имеет однородный материал, если тензор меняется незначительно, и неоднородный - в противном случае. Множество $\mathcal{F}_B$ в свою очередь разбивается на два подмножества $\mathcal{F}_B^D$ и $\mathcal{F}_B^N$, в которых накладываются граничные условия типа Дирихле и Неймана, соответственно. Для произвольной ячейки T обозначим множества ее граней и ребер как $\mathcal{F}_T$ и $\mathcal{E}_T$, соответственно, а множество ребер грани f обозначим как $\mathcal{E}_f$.

Метод конечных объемов использует одну степень свободы C_T для ячейки T, расположенную в ее центре масс $\mathbf{x}_T$. Для каждой грани $f \in \mathcal{F}_B \cup \mathcal{F}_I \setminus \mathcal{F}_J$ обозначим ее центр масс как $\mathbf{x}_f$, который также будем считать точкой коллокации. Точки коллокации для граней с разрывным тензором будут введены позднее. Также определим точки коллокации для ребер $e \in \mathcal{E}_f$ в их серединах $\mathbf{x}_e$. Точки коллокации в центрах масс ячеек мы будем называть *первичными* точками коллокации, дискретные значения концентрации в этих точках образуют вектор неизвестных в системе алгебраических уравнений. Точки коллокации на гранях и на ребрах будем называть *вспомогательными* точками коллокации. Они введены для удобства записи математических выражений в дальнейшем изложении, и не войдут в конечную систему алгебраических уравнений, хотя и будут влиять на коэффициенты системы.

Обозначим через $\mathbf{q} = -\mathbb{K}\nabla c$ диффузионный поток, который удовлетворяет уравнению баланса:

$$\operatorname{div} \mathbf{q} = g \quad \text{в} \quad \Omega. \tag{2.2}$$

Интегрируя уравнение (2.2) по многогранной ячейке T и используя формулу Грина, мы получаем:

$$\int\limits_{\partial T} \mathbf{q} \cdot \mathbf{n}_T \, \mathrm{d}s = \int\limits_{T} g \, \mathrm{d}x, \tag{2.3}$$

здесь $\mathbf{n}_T$ обозначает единичную внешнюю нормаль к ∂T. Для грани f ячейки T соответствующий нормальный вектор обозначим как $\mathbf{n}_f$. В дальнейшем, если в контексте участвует только одна ячейка T, то будем считать, что $\mathbf{n}_f$ соответствует внешнему по отношению к T нормальному вектору. Во всех остальных случаях мы будем явно указывать ориентацию $\mathbf{n}_f$. Будет удобно считать, что $|\mathbf{n}_f| = |f|$, где $|f|$ – площадь грани f. В таком случае уравнение (2.3) запишется в следующем виде:

$$\sum_{f \in \partial T} \mathbf{q}_f \cdot \mathbf{n}_f = \int\limits_T g \, \mathrm{d}x, \tag{2.4}$$

где $\mathbf{q}_f$ – средняя плотность диффузионного потока для грани f.

2.2.1. Диффузионный поток в однородной анизотропной среде

Для каждой ячейки T определим множество Σ_T соседних точек коллокации следующим образом. Для каждой грани $f \in \mathcal{F}_T \setminus (\mathcal{F}_B \cup \mathcal{F}_J)$ добавляется точка коллокации $\mathbf{x}_{T'_f}$ ячейки T'_f, соседствующей с T через грань f. Для внешних граней $f \in \mathcal{F}_T \cap \mathcal{F}_B$ мы добавляем точки коллокации $\mathbf{x}_f$.

Опишем схему для случая однородной области. Предположим, что для каждой грани f ячейки T_i существуют три точки $\mathbf{x}_{f,j}$, $\mathbf{x}_{f,k}$, и $\mathbf{x}_{f,l}$ из множества Σ_{T_i}, такие, что выполнено следующее условие (см. Рис. 2.1): вектор конормали $\ell_i = \mathbb{K}_{T_i}\mathbf{n}_f$, отложенный от $\mathbf{x}_{T_i}$, лежит в трехгранном угле, образованном векторами

$$\mathbf{t}_{ij} = \mathbf{x}_{f,j} - \mathbf{x}_{T_i}, \quad \mathbf{t}_{ik} = \mathbf{x}_{f,k} - \mathbf{x}_{T_i}, \quad \mathbf{t}_{il} = \mathbf{x}_{f,l} - \mathbf{x}_{T_i},$$

и

$$\ell_i = \alpha_{ij}\mathbf{t}_{ij} + \alpha_{ik}\mathbf{t}_{ik} + \alpha_{il}\mathbf{t}_{il}, \tag{2.5}$$

где $\alpha_{ij} \geq 0$, $\alpha_{ik} \geq 0$, $\alpha_{il} \geq 0$. Будем считать, что первая точка, $\mathbf{x}_{f,j}$, является точкой коллокации ячейки, соседней с T_i по грани f (либо точкой коллокации грани f, если $f \in \mathcal{F}_B$).

Тройку векторов $(\mathbf{t}_{ij}, \mathbf{t}_{ik}, \mathbf{t}_{il})$ будем называть триплетом. Алгоритм поиска триплета, а также коэффициентов α_{i*}, подробно описан в работе [40]. В общем случае для произвольной многогранной сетки необходимого триплета может не существовать. В этом случае множество соседних точек коллокации расширяется, добавляя центры соседних ячеек по ребрам и середины ребер.

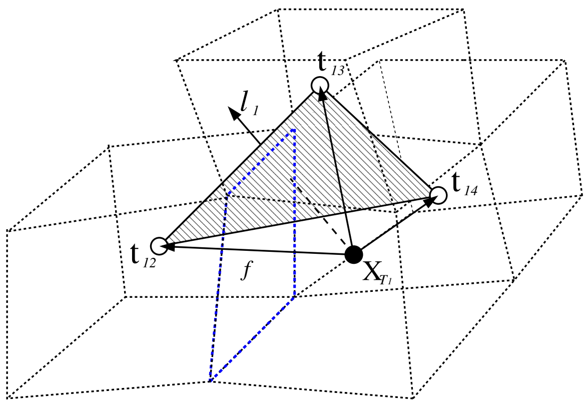


Рис. 2.1. Вектор конормали и векторы триплета.

Используя определение диффузионного потока и заменив производные по направлению на конечные разности, мы получим:

$$\begin{aligned} \mathbf{q}_f^h \cdot \mathbf{n}_f &= -\nabla c \cdot (\mathbb{K}_{T_i} \mathbf{n}_f) = -\alpha_{ij} \nabla c \cdot \mathbf{t}_{ij} - \alpha_{ik} \nabla c \cdot \mathbf{t}_{ik} - \alpha_{il} \nabla c \cdot \mathbf{t}_{il} \\ &= -\alpha_{ij}(C_{T_j} - C_{T_i}) - \alpha_{ik}(C_{T_k} - C_{T_i}) - \alpha_{il}(C_{T_l} - C_{T_i}) + O(|f|). \end{aligned} \tag{2.6}$$

Дискретный диффузионный поток q_f получается путем отбрасывания члена $O(|f|)$ в уравнении 2.6.

Для простоты, положим $i = 1$, $j = 2$, $k = 3$, $l = 4$ в (2.6) и получим дискретный поток $q_f^{(1)}$ из ячейки T_1 к T_2 через их общую грань f. Аналогично, положив $i = 2$, $j = 1$, $k = 5$, $l = 6$ в (2.6), получим другой дискретный поток $q_f^{(2)}$ в противоположном направлении. Итоговый дискретный поток является линейной комбинацией этих двух потоков:

$$
\begin{aligned}
q_f &= \mu_1 q_f^{(1)} + \mu_2(-q_f^{(2)}) \\
&= \mu_1(\alpha_{12}(C_{T_1} - C_{T_2}) + \alpha_{13}(C_{T_1} - C_{T_3}) + \alpha_{14}(C_{T_1} - C_{T_4})) \\
&\quad -\mu_2(\alpha_{21}(C_{T_2} - C_{T_1}) + \alpha_{25}(C_{T_2} - C_{T_5}) + \alpha_{26}(C_{T_2} - C_{T_6})).
\end{aligned} \tag{2.7}
$$

Наложим условие на веса μ_1 и μ_2, чтобы сбалансировать относительные вклады левого и правого потоков в общий поток:

$$
q_f^{(1)}\mu_1 + q_f^{(2)}\mu_2 = 0.
$$

Второе условие на веса накладывается из соображения аппроксимации настоящего потока:

$$
\mu_1 + \mu_2 = 1.
$$

Если $|q_f^{(1)}| + |q_f^{(2)}| = 0$, то решение системы из этих двух уравнений не является единственным. В этом случае положим $\mu_1 = \mu_2 = 1/2$. В противном случае, рассмотрим два случая. В первом случае $q_f^{(1)} q_f^{(2)} \leq 0$, тогда решением будет

$$
\mu_1 = \frac{|q_f^{(2)}|}{|q_f^{(1)}| + |q_f^{(2)}|},\ \mu_2 = \frac{|q_f^{(1)}|}{|q_f^{(1)}| + |q_f^{(2)}|}.
$$

Таким образом

$$
q_f = \frac{2q_f^{(1)}|q_f^{(2)}|}{|q_f^{(1)}| + |q_f^{(2)}|} = -\frac{2q_f^{(2)}|q_f^{(1)}|}{|q_f^{(1)}| + |q_f^{(2)}|}
$$

и дискретный диффузионный поток имеет два эквивалентных алгебраических представления:

$$
\begin{aligned}
q_f &= 2\mu_1(\alpha_{12}(C_{T_1} - C_{T_2}) - \alpha_{13}(C_{T_1} - C_{T_3}) - \alpha_{14}(C_{T_1} - C_{T_4})) \\
&= A_{12}(C_{T_1} - C_{T_2}) + A_{13}(C_{T_1} - C_{T_3}) + A_{14}(C_{T_1} - C_{T_4})
\end{aligned} \tag{2.8}
$$

$$
\begin{aligned}
-q_f &= 2\mu_2(\alpha_{21}(C_{T_2} - C_{T_1}) - \alpha_{25}(C_{T_2} - C_{T_5}) - \alpha_{26}(C_{T_2} - C_{T_6})) \\
&= A_{21}(C_{T_2} - C_{T_1}) + A_{25}(C_{T_2} - C_{T_5}) + A_{26}(C_{T_2} - C_{T_6})
\end{aligned} \tag{2.9}
$$

с неотрицательными коэффициентами $A_{12}, A_{13}, A_{14}, A_{21}, A_{25}$ и A_{26}. Заметим, что эти коэффициенты зависят от концентраций в соседних ячейках.

Во втором случае, когда $q_f^{(1)} q_f^{(2)} > 0$, получаем потенциально вырожденный диффузионный поток. Для того, чтобы избежать вырожденности, авторы из [96] предлагают перегруппировать члены в уравнении (2.7) следующим образом:

$$q_f = \mu_1 \tilde{q}_f^{(1)} + \mu_2(-\tilde{q}_f^{(2)}) + (\mu_1 \alpha_{12} + \mu_2 \alpha_{21})(C_{T_1} - C_{T_2}), \tag{2.10}$$

где $\tilde{q}_f^{(1)} = \alpha_{13}(C_{T_1} - C_{T_3}) + \alpha_{14}(C_{T_1} - C_{T_4})$, $\tilde{q}_f^{(2)} = \alpha_{25}(C_{T_2} - C_{T_5}) + \alpha_{26}(C_{T_2} - C_{T_6})$. Коэффициенты μ_1 и μ_2 находятся из соображений баланса модифицированных дискретных потоков и аппроксимации полного потока:

$$\begin{aligned} \tilde{q}_f^{(1)} \mu_1 + \tilde{q}_f^{(2)} \mu_2 &= 0, \\ \mu_1 + \mu_2 &= 1. \end{aligned}$$

Снова, если решение этой системы не является единственным, то берем $\mu_1 = \mu_2 = 1/2$. Для случая $\tilde{q}_f^{(1)} \tilde{q}_f^{(2)} \le 0$ получим:

$$\begin{aligned} q_f &= 2\mu_1 \tilde{q}_f^{(1)} + (\mu_1 \alpha_{12} + \mu_2 \alpha_{21})(C_{T_1} - C_{T_2}) \\ &= A_{13}(C_{T_1} - C_{T_3}) + A_{14}(C_{T_1} - C_{T_4}) + A_{12}(C_{T_1} - C_{T_2}) \\ &= -2\mu_2 \tilde{q}_f^{(2)} - (\mu_1 \alpha_{12} + \mu_2 \alpha_{21})(C_{T_2} - C_{T_1}) \\ &= -A_{25}(C_{T_2} - C_{T_5}) - A_{26}(C_{T_2} - C_{T_6}) - A_{21}(C_{T_2} - C_{T_1}), \end{aligned} \tag{2.11}$$

где $A_{12} = A_{21} = \mu_1 \alpha_{12} + \mu_2 \alpha_{21}$. В случае, если $\tilde{q}_f^{(1)} \tilde{q}_f^{(2)} > 0$, то

$$q_f = (\mu_1 \alpha_{12} + \mu_2 \alpha_{21})(C_{T_1} - C_{T_2}) = A_{12}(C_{T_1} - C_{T_2}). \tag{2.12}$$

Коэффициенты $A_{12}, A_{13}, A_{14}, A_{21}, A_{25}$ и A_{26} из (2.11), (2.12) являются неотрицательными по построению.

В нашей схеме конечных объемов будем использовать граничные условия типа Дирихле на гранях $f \in \mathcal{F}_B^D$, $C_f = \int_f g_D ds/|f|$, как известные значения концентраций в точках $\mathbf{x}_f$. Для граничных условий типа Неймана поток на внешних гранях $f \in \mathcal{F}_B^N$ известен из формулы $q_f = \overline{g}_{N,f}$, где $\overline{g}_{N,f}$ – среднее значение g_N на грани f.

2.2.2. Диффузионный поток в неоднородной анизотропной среде

Теперь рассмотрим область с неоднородным материалом. Рассмотрим грань $f \in \mathcal{F}_J$ между ячейками T_1 и T_2, в которых $\mathbb{K}_{T_1}(\mathbf{x}_f)$ и $\mathbb{K}_{T_2}(\mathbf{x}_f)$ отличаются. Обозначим через p_f плоскость, проходящую через f и рассмотрим непрерывную кусочно-линейную функцию $\mathcal{R}(\mathbf{x})$, такую что:

$$\mathcal{R}(\mathbf{x}_{T_1}) = C_{T_1}, \quad \mathcal{R}(\mathbf{x}_{T_2}) = C_{T_2},$$

и диффузионный поток непрерывен:

$$\mathbb{K}_{T_1} \nabla \mathcal{R}(\mathbf{x})|_{T_1} \cdot \mathbf{n}_f = \mathbb{K}_{T_2} \nabla \mathcal{R}(\mathbf{x})|_{T_2} \cdot \mathbf{n}_f.$$

Тогда существуют точка гармонического осреднения $\mathbf{y}_f \in p_f$ и коэффициент α_f, не зависящие от $\mathcal{R}$, такие что [27]:

$$\mathbf{y}_f = \alpha_f \mathbf{y}_1 + (1 - \alpha_f)\mathbf{y}_2 + \frac{d_{1,f} d_{2,f}}{d_{2,f} \mathbf{n}_f \cdot (\mathbb{K}_{T_1} \mathbf{n}_f) + d_{1,f} \mathbf{n}_f \cdot (\mathbb{K}_{T_2} \mathbf{n}_f)} (\mathbf{K}_1 - \mathbf{K}_2),$$

где $\mathbf{y}_i$ - проекция $\mathbf{x}_{T_i}$ на плоскость p_f, $d_{i,f}$ - расстояние от точки $\mathbf{x}_{T_i}$ до плоскости p_f, $\mathbf{K}_i = (\mathbb{K}_{T_i} - \mathbf{n}_f \cdot (\mathbb{K}_{T_i} \mathbf{n}_f) \mathbf{I}) \mathbf{n}_f$, и

$$\alpha_f = \frac{d_{2,f} \mathbf{n}_f \cdot (\mathbb{K}_{T_1} \mathbf{n}_f)}{d_{2,f} \mathbf{n}_f \cdot (\mathbb{K}_{T_1} \mathbf{n}_f) + d_{1,f} \mathbf{n}_f \cdot (\mathbb{K}_{T_2} \mathbf{n}_f)},$$

Тогда

$$C_f \equiv \mathcal{R}(\mathbf{y}_f) = \alpha_f C_{T_1} + (1 - \alpha_f) C_{T_2}, \quad 0 \le \alpha_f \le 1. \tag{2.13}$$

Тогда метод, описанный в предыдущем разделе, может быть расширен на случай разрывных тензоров с помощью точек гармонического осреднения. Заметим, что аппроксимация производной по направлению $\nabla c \cdot \mathbf{t}_{ij}$ точна только в однородных подобластях. Это ограничивает множество точек коллокации для построения триплетов. Однако, используя точку гармонического осреднения как дополнительную точку коллокации, можно построить триплеты для ячеек T_1 и T_2. Формула для итогового потока будет включать концентрации C_{T_i} и C_f, однако последняя может быть исключена, используя уравнение (2.13), без увеличения шаблона и удовлетворяя ДПМ. Так, например, формула (2.8) может быть записана в виде:

$$\begin{aligned} q_f &= A_{12}(C_{T_1} - C_f) + A_{13}(C_{T_1} - C_{T_3}) + A_{14}(C_{T_1} - C_{T_4}) \\ &= A_{12}(1-\alpha_f)(C_{T_1} - C_{T_2}) + A_{13}(C_{T_1} - C_{T_3}) + A_{14}(C_{T_1} - C_{T_4}) \end{aligned} \tag{2.14}$$

Напомним, что точки гармонического осреднения используются в работе [49] для аппроксимации потока на всех гранях сетки. В предлагаемой дискретизации они используются только для граней со скачком тензора.

2.2.3. Свойства системы уравнений

Пусть $\mathbf{C}$ - вектор неизвестных из всех точек коллокации в ячейках. Заменяя потоки в уравнениях (2.4) на их аппроксимации, мы получим систему нелинейных уравнений

$$\mathbf{M}(\mathbf{C})\mathbf{C} = \mathbf{F}(\mathbf{C}). \tag{2.15}$$

с М-матрицей $\mathbf{M}$ с элементами, зависящими от $\mathbf{C}$, имеющей диагональное преобладание в строках, и вектором правой части $\mathbf{F}$. Эта система решается с помощью метода Пикара. Итоговое решение, а также решения на каждой пикаровской итерации $\mathbf{C}^k = \mathbf{M}(\mathbf{C}^{k-1})^{-1}\mathbf{F}(\mathbf{C}^{k-1})$, удовлетворяют дискретному принципу максимума [68]. Линейные системы решаются с помощью стабилизированного метода бисопряженных градиентов (BiCGStab) с использованием предобуславливателя ILU2 [58]. Шаблон этой матрицы является компактным. В частности, на регулярных кубических сетках порождается 7-точечный шаблон, а для оператора Лапласа получается классический линейный шаблон 6-1-1-1-1-1-1.

2.3. Результаты численных экспериментов

Проверим сходимость метода и свойство монотонности на некоторых численных экспериментах. В численных экспериментах будем использовать гексаэдральные сетки, сетки типа восьмеричное дерево, а также многогранные сетки типа восьмеричное дерево со сколотыми ячейками.

Будем использовать дискретные относительные L_2-нормы для оценки ошибок дискретизации для концентрации c и для диффузионного потока $\mathbf{q}$:

$$\varepsilon_2^C = \left[\frac{\sum\limits_{T\in\mathcal{T}} \left(c(\mathbf{x}_T) - C_T\right)^2 |T|}{\sum\limits_{T\in\mathcal{T}} \left(c(\mathbf{x}_T)\right)^2 |T|}\right]^{1/2} \quad \text{и} \quad \varepsilon_2^q = \left[\frac{\sum\limits_{f\in\mathcal{F}_I\cup\mathcal{F}_B} \left((\mathbf{q}_f - \mathbf{q}_f^h)\cdot\mathbf{n}_f\right)^2 |V_f|}{\sum\limits_{f\in\mathcal{F}_I\cup\mathcal{F}_B} \left(\mathbf{q}_f\cdot\mathbf{n}_f\right)^2 |V_f|}\right]^{1/2},$$

где $|V_f|$ – характерный объем грани f (среднее арифметическое объемов ячеек с гранью f). Нелинейные итерации прекращались при достижении относительной нормы невязки $\varepsilon_{\text{non}} = 10^{-9}$. Критерием остановки решения системы линейных уравнений было выбрано достижение относительной нормы невязки $\varepsilon_{\text{lin}} = 10^{-12}$.

2.3.1. Сходимость на гладком решении

В этом разделе мы проведем исследование скорости сходимости предложенной схемы на гладком решении. Рассмотрим тест, описанный в [2, 40]. Пусть $\Omega = (0,1)^3$, $\Gamma_D = \partial\Omega$, и g получена подстановкой в уравнение (2.1) точного решения

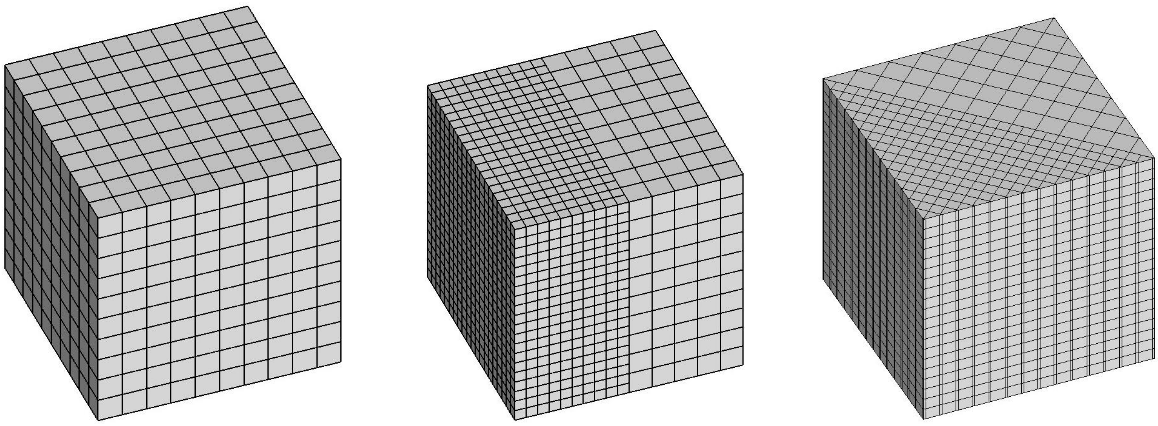
$$c(x,y,z) = \frac{1}{3\pi^2}\sin(\pi x)\sin(\pi y)\sin(\pi z). \tag{2.16}$$

Граничные условия Дирихле, g_D, совпадают со следом $c(x, y, z)$ на Γ_D. Анизотропный тензор диффузии зададим в виде:

$$\mathbb{K} = k(x, y, z) \cdot \mathrm{diag}(1, 10, 100).$$

где $k(x, y, z) = 1 + 0.25 \cos(x + y - z)$, и получим непрерывный диффузионный тензор без резких скачков коэффициентов и направлений собственных векторов.

В Таблице 2.1 представлены погрешности на гексаэдральной сетке, на сетке типа восьмеричное дерево, полученной в результате измельчения одной половины куба, а также на сетке типа восьмеричное дерево со сколотыми ячейками, полученной в результате поворота куба на угол $\pi/4$ вдоль вертикальной оси и измельчения одной половины куба (см. Рис. 2.2). В последнем случае на боковых границах получаются сколотые многогранные ячейки разного типа. Эта сетка имеет не $\mathbb{K}$-ортогональные ячейки. Под h в таблице понимается размер наибольшей гексаэдральной ячейки сетки.



(*а*) (*б*) (*в*)

Рис. 2.2. (*а*) кубическая сетка, $h = 1/10$ (*б*) сетка типа восьмеричное дерево, $h_{max} = 1/10$ (*б*) сетка типа восьмеричное дерево со сколотыми ячейками, $h_{max} = 1/10$

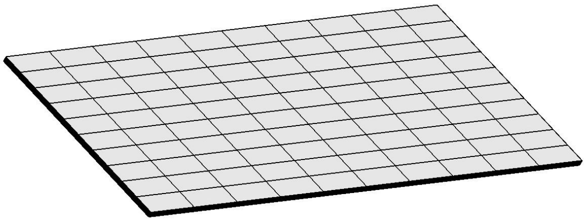
Таблица 2.1. Погрешности для задачи с гладким решением и переменным непрерывным тензором диффузии.

Тип сетки	Гексаэдры		Восьмидерево		Многогранники	
h	ε_2^C	ε_2^q	ε_2^C	ε_2^q		
1/10	8.402e-3	4.287e-3	6.234e-3	4.335e-3	8.969e-3	8.560e-3
1/20	2.093e-3	1.067e-3	1.564e-3	1.106e-3	2.426e-3	2.499e-3
1/40	5.228e-4	2.665e-4	3.921e-4	2.786e-4	6.416e-4	7.289e-4
порядок	2.00	2.00	2.00	1.99	1.88	1.77

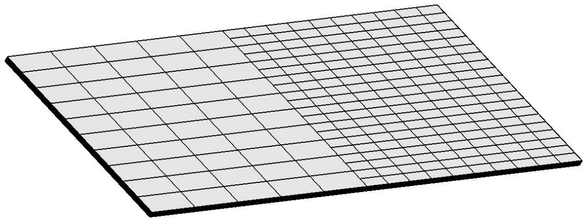
Для гексаэдральной сетки и для сетки типа восьмеричное дерево получился второй порядок сходимости неизвестной C, а также нормальных составляющих диффузионного потока на гранях сетки. Для сетки типа восьмеричное дерево со сколотыми ячейками имеем почти второй порядок сходимости неизвестной C, а также близкий ко второму порядок сходимости диффузионного потока.

2.3.2. Сходимость на анизотропных сетках

Рассмотрим задачу (2.1) с единичным тензором диффузии на сильно анизотропных сетках. Работоспособность численной схемы на таких сетках является важным показателем, так как в реальных приложениях распространены именно анизотропные сетки. Рассмотрим сжатие единичного куба вдоль оси Oz в 100 раз и сдвиговый наклон его yz граней на угол $\frac{\pi}{6}$. Рассмотрим три типа сеток, описанные в предыдущем примере. После предложенного аффинного преобразования мы получим анизотропные сетки с отношением анизотропии 0.01 (см. Рис. 2.3).



(*а*)



(*б*)

Рис. 2.3. Анизотропные сетки с $h = 1/10$, отношение сторон 0.01, угол наклона – $\frac{\pi}{6}$. (*а*) кубическая сетка, (*б*) сетка типа восьмеричное дерево.

Точным решением является функция $c(x, y, z) = \cosh(\pi x)\cos(\pi z)$, порождающая нулевую правую часть и ненулевые граничные условия типа Дирихле.

Погрешности представлены в Таблице 2.2.

Таблица 2.2. Погрешности для задачи с анизотропными сетками.

	Гексаэдры				Восьмидерево				Многогранники			
h	ε_2^C	пор.	ε_2^q	пор.	ε_2^C	пор.	ε_2^q	пор.	ε_2^C	пор.	ε_2^q	пор.
1/10	4.58e-4		7.63e-1		2.85e-4		7.27e-1		2.44e-4		6.41e-1	
1/20	2.69e-4	0.77	5.80e-1	0.40	1.19e-4	1.26	4.62e-1	0.65	9.67e-5	1.34	3.84e-1	0.73
1/40	1.02e-4	1.40	3.25e-1	0.83	3.66e-5	1.70	2.51e-1	0.88	2.57e-5	1.91	2.28e-1	0.75
1/80	2.80e-5	1.86	1.63e-1	0.99	9.63e-6	1.92	1.25e-1	0.98	6.79e-5	1.92	1.27e-1	0.84

Можно заметить, что асимптотический порядок сходимости неизвестной C лежит в интервале $(1, 2)$, также наблюдается асимптотическое стремление к первому порядку для сходимости потоков.

2.3.3. Тесты на монотонность

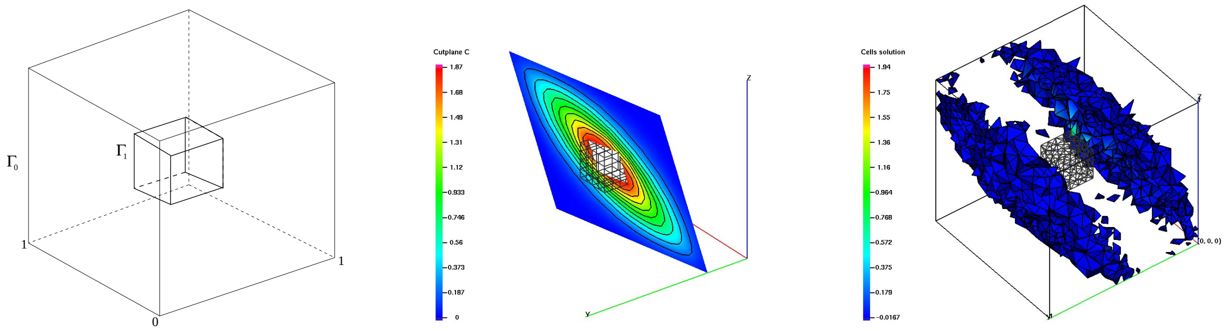
Задача Дирихле

Рассмотрим задачу (2.1) в единичном кубе с кубической дыркой, $\Omega = (0, 1)^3/[0.4, 0.6]^3$. Граница области Ω состоит из двух несвязных частей, внешней Γ_0 и внутренней Γ_1 (см. Рис. 2.4, *а*). Положим $\Gamma_N = \emptyset$, $g = 0$, $g_D = 0$

на Γ_0, $g_D = 2$ на Γ_1, в качестве диффузионного тензора $\mathbb{K}$ выберем полный тензор с сильной анизотропией

$$\mathbb{K} = R_z(-\theta_z)R_y(-\theta_y))R_x(-\theta_x)\mathrm{diag}(k_1, k_2, k_3)R_x(\theta_x)R_y(\theta_y)R_z(\theta_z), \qquad (2.17)$$

где $k_1 = 100, k_2 = 10, k_3 = 1$, $\theta_x = \pi/3$, $\theta_y = \pi/4$, $\theta_z = \pi/6$, и $R_a(\alpha)$ – матрица поворота на угол α в плоскости, ортогональной оси oa.



(a) ($б$) ($в$)

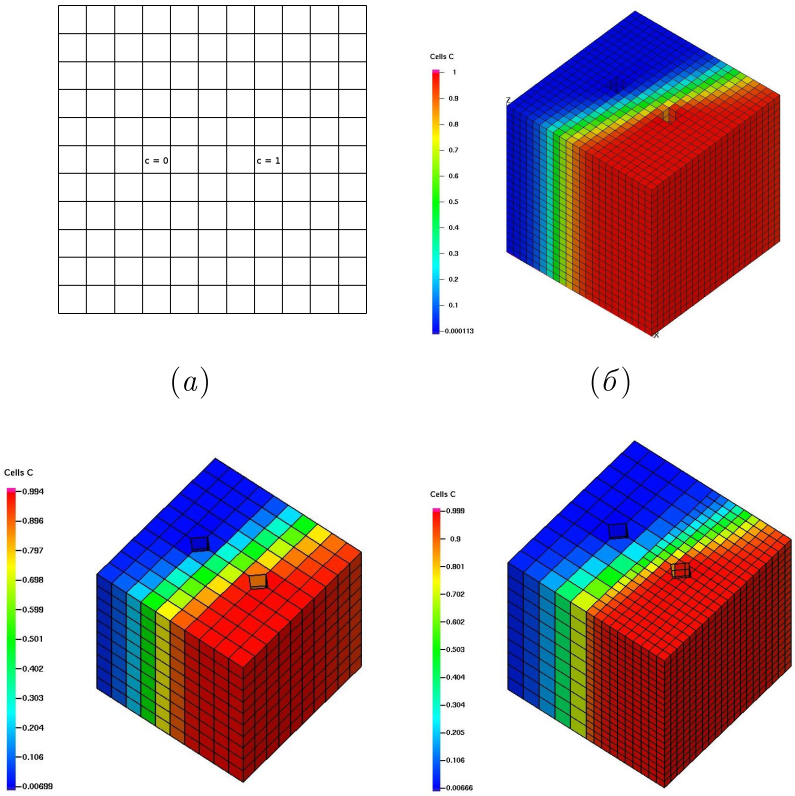
Рис. 2.4. Тест 1 на монотонность, задача Дирихле: (a) эскиз области, ($б$) сечение полученного решения с помощью нелинейного метода конечных объемов, ($в$) элементы, значения в которых меньше чем -10^{-3}, при решении методом смешанных конечных элементов.

Согласно принципу максимума для эллиптических уравнений в частных производных, решение задачи должно лежать между 0 и 2. Сеточное решение, полученное с помощью предлагаемого нелинейного метода конечных объемов, принадлежит данному интервалу. Отметим, что дискретизация на основе смешанных конечных элементов Равьера-Тома низшего порядка порождает большие области с отрицательным решением (см. Рис. 2.4, $в$). Аналогичное наблюдение верно и в двумерном случае задачи (2.1) для дискретизаций MFE и MPFA.

Однородные условия типа Неймана на части границы

Рассмотрим единичный куб с двумя вертикальными дырками P_1, P_2, $\Omega = (0,1)^3 \setminus (P_1 \cup P_2)$, где $P_i = S_i \times (0,1)$, $i = 1,2$, $S_1 = [3/11, 4/11] \times [5/11, 6/11]$,

$S_2 = [7/11, 8/11] \times [5/11, 6/11]$. Граница области разбита на внешнюю часть Γ_N, на которой накладываются однородные условия типа Неймана, и две внутренних части $\Gamma_{D,1} = \partial P_1 \bigcap \partial\Gamma$, $\Gamma_{D,2} = \partial P_2 \bigcap \partial\Gamma$, на которых накладываются условия типа Дирихле: $g_D(\mathbf{x}) = 0$, $\mathbf{x} \in \Gamma_{D,1}$, $g_D(\mathbf{x}) = 1$, $\mathbf{x} \in \Gamma_{D,2}$ (см. Рис. 2.5, *а*). Анизотропный тензор задан формулой (2.17) с коэффициентами $k_1 = k_3 = 1, k_2 = 10^{-3}$, $\theta_x = \theta_y = 0$, $\theta_z = 67.5°$. Согласно принципу максимума для эллиптических уравнений, точное решение должно лежать между 0 и 1.



(*а*) (*б*)

(*в*) (*г*)

Рис. 2.5. Тест 2 на монотонность, смешанные краевые условия: (*а*) эскиз области (плоскость xy), полученое решение на сетке (*б*) кубической ($h = 1/22$), (*в*) кубической со сколотыми ячейками ($h = 1/10$), (*г*) восьмидерево со сколотыми ячейками ($h_{max} = 1/10$).

В Таблице 2.3 преведены максимальные значения концентраций, полученные с помощью данной схемы. Результаты из [40], полученные с помощью двухточечной нелинейной схемы, для сравнения также приведены в таблице. Из таблицы можно видеть, что в предложенной схеме выполнен принцип максимума, в отличие от схемы, предложенной в [40].

Далее, рассмотрим эту же задачу, однако вертикальные дырки повернем на угол $\pi/4$. В этом случае для аппроксимации их границ используются сколотые ячейки. Построим равномерную кубическую сетку со сколотыми ячейками, а также сетку типа восьмеричное дерево со сколотыми ячейками (см. Рис. 2.5). Расчеты на таких сетках также приводят к решению, удовлетворяющему ДПМ. На Рис.2.5 представлены визуализации сеточного решения на некоторых описанных сетках.

Таблица 2.3. Максимальное значение концентрации для предложенной схемы и схемы с нелинейным двухточечным потоком из [40].

h	$\max C$	$\max C$[40]
1/11	0.992	2.163
1/22	0.999	1.765
1/44	0.999	1.136

2.4. Выводы к второй главе

В данной главе предложена и численно исследована трехмерная версия нелинейной многоточечной схемы конечных объемов для задачи диффузии на многогранных сетках, удовлетворяющая дискретному принципу максимума. Шаблон метода является компактным и содержит в большинстве случаев только соседние ячейки по граням. Численные эксперименты подтверждают монотонность схемы и близкий ко второму порядок сходимости по концентрациям на многогранных сетках, в том числе на сетках типа восьмеричное дерево со сколотыми ячейками. Таким образом показано, что сетки, предложенные в главе 1, могут быть успешно использованы при дискретизациях эллиптических уравнений в трехмерных областях.

Глава 3

Метод приближенного решения эллиптических уравнений 2-го порядка на поверхностях

В данной главе диссертационной работы предлагается и исследуется новая переформулировка эллиптического уравнения на поверхности, приводящая к невырожденному эллиптическому уравнению в окрестности поверхности.

3.1. Предварительные сведения и обозначения

Предположим, что Ω является открытым подмножеством в $\mathbb{R}^N$, $N = 2, 3$ и Γ является связной C^2 компактной гиперповерхностью, находящейся в Ω. Для достаточно гладкой функции $g : \Omega \to \mathbb{R}$ *тангенциальный градиент* (вдоль Γ) определяется как

$$\nabla_\Gamma g = \nabla g - \nabla g \cdot \mathbf{n}_\Gamma \, \mathbf{n}_\Gamma. \tag{3.1}$$

Обозначим через Δ_Γ - оператор Лапласа-Бельтрами на Γ, $\Delta_\Gamma = \nabla_\Gamma \cdot \nabla_\Gamma$.

В качестве базового эллиптического уравнения, рассмотрим уравнение Лапласа-Бельтрами:

$$-\Delta_\Gamma u + \alpha\, u = f \quad \text{на } \Gamma, \tag{3.2}$$

для неотрицательной функции $\alpha \in L^\infty(\Gamma)$. Тогда соответствующая слабая формулировка уравнения (3.2) будет следующей: для данной функции $f \in L^2(\Gamma)$ найти $u \in H^1(\Gamma)$ такую что

$$\int\limits_\Gamma (\nabla_\Gamma u \nabla_\Gamma v + \alpha\, uv)\, \mathrm{d}\mathbf{s} = \int\limits_\Gamma f v \, \mathrm{d}\mathbf{s} \qquad \text{для всех } \; v \in H^1(\Gamma). \tag{3.3}$$

Для корректности (3.3) достаточно предположить, чтобы коэффициент α был строго положителен в подмножестве $\mathcal{A} \in \Gamma$ положительной поверхностной меры:

$$\mathrm{meas}_{\mathbf{s}}\{\mathbf{x} \in \mathcal{A} \,:\, \alpha(\mathbf{x}) \geq \alpha_0\} > 0, \tag{3.4}$$

при $\alpha_0 > 0$. В таком случае, соответствующее неравенство типа Фридрихса [85] (см., также Лемму 3.1 в [78])

$$\|v\|^2_{L^2(\Gamma)} \leq C_F(\|\nabla_\Gamma v\|^2_{L^2(\Gamma)} + \|\sqrt{\alpha} v\|^2_{L^2(\Gamma)}) \quad \forall\, v \in H^1(\Gamma) \tag{3.5}$$

верно для константы C_F, зависящей от α_0 и $\mathcal{A}$.

Решение u задачи (3.3) является единственным, причем $u \in H^2(\Gamma)$ и $\|u\|_{H^2(\Gamma)} \leq c\|f\|_{L^2(\Gamma)}$, где константа c не зависит от f, (см. [45]). Отметим, что случай $\alpha = 0$ также охватывается результатами данной работы. В этом случае неравенство Фридрихса (3.5) верно для всех $v \in H^1(\Gamma)$ с нулевым средним. Следовательно, если $\alpha = 0$, то мы предполагаем, что $\int_\Gamma f\,\mathrm{d}\mathbf{s} = 0$ и слабое решение (3.3) должно удовлетворять условию $\int_\Gamma u\,\mathrm{d}\mathbf{s} = 0$.

Обозначим через Ω_d область, состоящую из всех точек, находящихся на расстоянии от Γ, меньшем чем $d > 0$:

$$\Omega_d = \{\, \mathbf{x} \in \mathbb{R}^N \;:\; \mathrm{dist}(\mathbf{x}, \Gamma) < d \,\}. \tag{3.6}$$

Пусть $\phi : \Omega_d \to \mathbb{R}$ является функцией расстояния со знаком, $|\phi(x)| = \mathrm{dist}(\mathbf{x}, \Gamma)$ для всех $\mathbf{x} \in \Omega_d$. Тогда поверхность Γ является изоповерхностью нуля функции уровня ϕ:

$$\Gamma = \{\mathbf{x} \in \mathbb{R}^N \,:\, \phi(\mathbf{x}) = 0\}. \tag{3.7}$$

Будем считать, что $\phi < 0$ внутри Γ и $\phi > 0$ во внешней части. Определим $\mathbf{n}(\mathbf{x}) := \nabla\phi(\mathbf{x})$ для всех $\mathbf{x} \in \Omega_d$. Таким образом, $\mathbf{n}$ является внешним нормальным вектором Γ, $\mathbf{n}_\Gamma = \nabla\phi$ на Γ, и $|\mathbf{n}(\mathbf{x})| = 1$ для всех $\mathbf{x} \in \Omega_d$. Гессиан функции ϕ обозначим через $\mathbf{H}$:

$$\mathbf{H}(\mathbf{x}) = D^2\phi(\mathbf{x}) \in \mathbb{R}^{3\times 3} \quad \text{для всех } \ \mathbf{x} \in \Omega_d. \tag{3.8}$$

Собственные значения $\mathbf{H}(\mathbf{x})$ обозначим через $\kappa_1(\mathbf{x}), \kappa_2(\mathbf{x})$ и 0. Для $\mathbf{x} \in \Gamma$, собственные значения $\kappa_i(\mathbf{x})$, $i = 1, 2$, являются главными кривизнами.

Нам понадобится следующий оператор проектирования:

$$\mathbf{P}(\mathbf{x}) = \mathbf{I} - \mathbf{n}(\mathbf{x}) \otimes \mathbf{n}(\mathbf{x}) \quad \text{для всех} \;\; \mathbf{x} \in \Omega_d.$$

Тогда тангенциальный градиент может быть записан в виде $\nabla_\Gamma g(\mathbf{x}) = \mathbf{P}\nabla g(\mathbf{x})$ для $\mathbf{x} \in \Gamma$. Введем локально ортогональную координатную систему, используя проектор $\mathbf{p} : \Omega_d \to \Gamma$:

$$\mathbf{p}(\mathbf{x}) = \mathbf{x} - \phi(\mathbf{x})\mathbf{n}(\mathbf{x}) \quad \text{для всех} \;\; \mathbf{x} \in \Omega_d.$$

Предположим, что разложение $\mathbf{x} = \mathbf{p}(\mathbf{x}) + \phi(\mathbf{x})\mathbf{n}(\mathbf{x})$ единственно для всех $\mathbf{x} \in \Omega_d$. Для функции v на Γ определим ее продолжение, или расширение:

$$v^e(\mathbf{x}) := v(\mathbf{p}(\mathbf{x})) \quad \text{для всех} \;\; \mathbf{x} \in \Omega_d. \tag{3.9}$$

Таким образом, v^e является продолжением v вдоль нормалей к Γ и удовлетворяет $\mathbf{n} \cdot \nabla v^e = 0$ в Ω_d, другими словами, v^e постоянна вдоль нормалей к Γ.

3.2. Расширение уравнения на поверхности

В этом разделе будет определено расширение уравнения на поверхности (3.2) на приповерхностный слой. Используя (3.7) и то, что $\nabla_\Gamma u = \mathbf{P}\nabla u$ на Γ, запишем слабую формулировку для (3.2) на нулевой поверхности функции уровня ϕ:

$$\int\limits_\Gamma (\nabla_\Gamma u \cdot \nabla_\Gamma v + \alpha\, uv - fv)\, d\mathbf{s} = \int\limits_{\{\phi=0\}} (\mathbf{P}\nabla u \cdot \mathbf{P}\nabla v + \alpha\, uv - fv)\, d\mathbf{s} = 0.$$

В [32] предложена идея продолжить (3.3) с помощью глобально определенных величин $\mathbf{n}$ и $\mathbf{P}$ на все изоповерхности (линии) уровня функции ϕ, пересекающие Ω_d. Новая формулировка состоит в нахождении $u \in H_P$, такой что

$$
\begin{aligned}
0 &= \int\limits_{-d}^{+d} \int\limits_{\{\phi=r\}} (\mathbf{P}\nabla u \cdot \mathbf{P}\nabla v + \alpha^e\, uv - f^e v)\, d\mathbf{s}\, dr \\
&= \int\limits_{\Omega_d} (\mathbf{P}\nabla u \cdot \mathbf{P}\nabla v + \alpha^e\, uv - f^e v)|\nabla\phi| d\mathbf{x} \quad \text{для всех } v \in H_P,
\end{aligned}
\tag{3.10}
$$

где

$$
H_P = \{v \in L^2(\Omega_d) \,:\, \mathbf{P}\nabla v \in (L^2(\Omega_d))^N\}.
$$

В работе [35] была показана корректность этой слабой формулировки. Уравнение на поверхности (3.3) включается в (3.10) и решение на каждой изоповерхности уровня ϕ не зависит от данных в окрестности этой изоповерхности. Действительно, можно рассматривать (3.10) как набор взаимно независимых поверхностных задач на каждой изоповерхности уровня ϕ. Следовательно, ограниченное на Γ гладкое решение (3.10) является решением исходной задачи Лапласа-Бельтрами (3.2). Для простоты, будем обозначать через u решения как уравнения на поверхности, так и расширенного в Ω_d уравнения.

Запишем соответствующую сильную формулировку (3.10):

$$
-\,|\nabla\phi|^{-1}\operatorname{div}|\nabla\phi|\mathbf{P}\nabla u + \alpha^e\, u = f^e \quad \text{в } \Omega_d. \tag{3.11}
$$

Заметим, что (3.10) и (3.11) остаются корректными расширениями (3.3) и (3.2), когда ϕ является произвольной гладкой функцией уровня и $\nabla\phi \neq 0$, причем она не обязательно должна быть функцией расстояния со знаком, а α^e, f^e не обязательно постоянны вдоль нормальных направлений. Если граница области Ω_d не является изоповерхностью уровня ϕ, то уравнение (3.11) должно быть дополнено граничными условиями. Это могут быть естественные граничные условия

$$
(\mathbf{P}\nabla u)\cdot \mathbf{n}_{\partial\Omega_d} = 0 \quad \text{на } \partial\Omega_d, \tag{3.12}
$$

где $\mathbf{n}_{\partial\Omega_d}$ - внешний нормальный вектор к $\partial\Omega_d$.

Основное преимущество расширенной формулировки заключается в том, что можно решить (3.11)–(3.12) в области Ω_d, используя стандартные методы дискретизаций (например, метод конечных разностей в декартовых координатах), а затем определить след полученного решения на Γ (или на аппроксимации Γ). Численные эксперименты в работах [32, 35, 51, 94] показывают, что полученные таким образом решения являются хорошими аппроксимациями решения поверхностной задачи (3.2). Однако, анализ метода является неполным: оценки ошибки для конечно-элементных методов решения (3.10) показаны лишь в работах [35, 41]. Оценка ошибки в работе [35] получена только в интегральной объемной норме

$$\|v\|_{H_P}^2 = \|v\|_{L^2(\Omega_d)}^2 + \|\mathbf{P}\nabla v\|_{L^2(\Omega_d)}^2,$$

а не в поверхностной норме для Γ. В работе [41] доказана оценка сходимости с первым порядком в H^1 норме на поверхности, в случае, если шаг сетки является величиной порядка ширины слоя d в (3.6), а также если используется квази-равномерная триангуляция области Ω. Для линейных элементов эта оценка является оптимальной.

Несмотря на удобства возможных численных методов, расширенная формулировка имеет некоторые недостатки, описанные в [32] и рассмотренные в работах [41, 51]. Объемная формулировка (3.11) определяется в области, имеющей на единицу большую размерность, чем исходная формулировка. Это приводит к появлению дополнительных степеней свободы в численных методах. Если Ω_d является узкой полосой вокруг Γ, то учет граничных условий (3.12) может повлиять на точность дискретного решения. Эта проблема может возникать на сетках, не отслеживающих изоповерхности уровня ϕ на $\partial\Omega_d$. Численная устойчивость требует продолжения данных согласно (3.9), и в схемах с временным шагом для параболических задач необходимо периодически реинициализировать u через расширение из Γ по формуле (3.9). Другой проблемой для расширенной формулировки (3.11) является то, что оператор, входящий в уравнение, является вырожденным, так как отсутствует диффузия в нормальном направлении к изоповерхностям уровня ϕ. Численное решение вырожденных эллиптических и параболических уравнений не является хорошо изученной задачей.

Попытка преодолеть вырожденность и некоторые другие проблемы подхода из [32] была сделана Гриром в [51], где рассматривалось уравнение теплопроводности

$$\frac{\partial u}{\partial t} - \Delta_\Gamma u = 0, \quad u|_{t=0} = u_0 \tag{3.13}$$

на поверхности Γ. Он предложил метод расширения уравнения (3.13) в окрестности Γ, накладывая следующие условия:

1. ϕ является функцией расстояния со знаком;

2. u_0 продолжается в окрестность Γ согласно (3.9), т.е. является постоянной вдоль нормалей;

3. проектор $\mathbf{P}$ заменяется на следующий неортогональный проектор

$$\widetilde{\mathbf{P}} = (\mathbf{I} - \phi\mathbf{H})^{-1}\mathbf{P} \tag{3.14}$$

   на тангенциальные плоскости изоповерхностей уровня ϕ. Предполагается, что область Ω_d является такой, что новый проектор $\widetilde{\mathbf{P}}$ определен. Для гладких Γ для этого нужно выбрать достаточно малое $d > 0$.

При описанных условиях доказано, что решение расширенного уравнения теплопроводности

$$\frac{\partial u}{\partial t} - (\widetilde{\mathbf{P}}\nabla) \cdot \widetilde{\mathbf{P}}\nabla u = 0, \quad u|_{t=0} = u_0^e \quad \text{в} \quad \Omega_d \tag{3.15}$$

является постоянным вдоль нормалей:

$$\mathbf{n} \cdot \nabla u = 0 \quad \text{в } \Omega_d \tag{3.16}$$

для всех $t > 0$.

Свойство (3.16) является ключевым, так как оно позволяет добавить диффузию вдоль нормального направления без изменения решения. Таким образом, получается невырожденный эллиптический оператор. При этом, вместо (3.15) в работе [51] предлагается рассматривать следующую параболическую задачу:

$$\frac{\partial u}{\partial t} - (\widetilde{\mathbf{P}}\nabla) \cdot \widetilde{\mathbf{P}}\nabla u - c_n^2 \operatorname{div}(\mathbf{n} \otimes \mathbf{n})\nabla u = 0, \quad u|_{t=0} = u_0 \quad \text{в } \ \Omega_d, \tag{3.17}$$

с некоторым коэффициентом c_n^2. В плоском случае, $\Omega_d \in \mathbb{R}^2$, автор рекомендует использовать $c_n = (1-\phi\kappa_0)$, $\kappa_0 = \kappa(p(\mathbf{x}))$, где κ является кривизной Γ (Γ является кривой в этом случае). Однако, для случая поверхности, вложенной в $\mathbb{R}^3$, рекомендации для выбора c_n не приводятся.

Формально, описанный выше подход решает проблему вырожденности и предполагает, что условие (3.16), наложенное на решение на $\partial\Omega_d$, является подходящим граничным условием. Однако, требуется определять параметр c_n. Кроме этого, новая расширенная формулировка включает гессиан $\mathbf{H}$. Если Γ представлена только некоторой аппроксимацией, например изоповерхностью нуля дискретной функции уровня ϕ_h, то вычисление $\mathbf{H}$ является весьма деликатной задачей, чувствительной к численной реализации.

Далее мы введем новую формулировку задачи на поверхности, также основанную на продолжении в объемную область. Новая постановка задачи “автоматически” добавляет диффузию вдоль нормального направления, приводя к эллиптической или параболической задаче в Ω_d.

3.2.1. Новая расширенная формулировка уравнения на поверхности

Снова будем рассматривать уравнение Лапласа-Бельтрами (3.2) вместо уравнения теплопроводности. Предположим, что теперь все расширения данных из Γ удовлетворяют (3.9). Рассмотрим уравнение Лапласа-Бельтрами, расширенное в Ω_d:

$$-(\widetilde{\mathbf{P}}\nabla)\cdot\widetilde{\mathbf{P}}\nabla u + \alpha^e\, u = f^e \quad \text{в} \ \ \Omega_d. \tag{3.18}$$

Для того, чтобы оператор $\widetilde{\mathbf{P}}$ был определен, достаточно, чтобы матрица $(\mathbf{I}-\phi\mathbf{H})$ была положительно определеной в Ω_d. Таким образом, предположим, что Ω_d удовлетворяет ограничению:

$$|\phi(\mathbf{x})| = \operatorname{dist}(\mathbf{x},\Gamma) \le \frac{1}{2}\|\mathbf{H}(\mathbf{x})\|^{-1} \quad \forall\,\mathbf{x}\in\Omega_d. \tag{3.19}$$

Для этого нужно выбрать d достаточно малым. Для более точной оценки, рассмотрим следующую формулу для собственных значений $\mathbf{H}$ из работы [42]:

$$\kappa_i(\mathbf{x}) = \frac{\kappa_i(\mathbf{p}(\mathbf{x}))}{1+\phi(\mathbf{x})\kappa_i(\mathbf{p}(\mathbf{x}))} \quad \text{для } \mathbf{x}\in\Omega_d.$$

Следовательно, ограничение (3.19) выполнено, если параметр d из (3.6) удовлетворяет

$$d \le \Big(4\max_{\mathbf{x}\in\Gamma}(|\kappa_1(\mathbf{x})| + |\kappa_2(\mathbf{x})|)\Big)^{-1}.$$

Так как $\Gamma\in C^2$ и Γ компактна, то главные кривизны Γ ограничены и d может быть выбран достаточно малым положительным числом.

Слабая формулировка задачи (3.18) является следующей: найти $u\in H_P$, удовлетворяющую

$$\int\limits_{\Omega_d}\widetilde{\mathbf{P}}\nabla u\cdot\widetilde{\mathbf{P}}\nabla v + \alpha^e\, uv\, d\mathbf{x} = \int\limits_{\Omega_d} f^e v d\mathbf{x} \quad \forall\, v\in H_P. \tag{3.20}$$

Если выполнено (3.19), то существование и единственность решения (3.20) следует из леммы Лакса-Мильграма. Если решение (3.20) является гладким, то оно является решением (3.2) ($\mathbf{P} = \widetilde{\mathbf{P}}$ на Γ). Более того, гладкое решение (3.20) удовлетворяет (3.16). Действительно, применим оператор $(\mathbf{n}\cdot\nabla)$ к уравнению (3.18), используя следующее свойство коммутативности (см. Лемму 1 в [51]):

$$(\mathbf{n}\cdot\nabla)((\widetilde{\mathbf{P}}\nabla)\cdot\widetilde{\mathbf{P}}\nabla) = ((\widetilde{\mathbf{P}}\nabla)\cdot\widetilde{\mathbf{P}}\nabla)(\mathbf{n}\cdot\nabla).$$

Используя $(\mathbf{n}\cdot\nabla)\alpha^e = (\mathbf{n}\cdot\nabla)f^e = 0$, получим

$$-(\widetilde{\mathbf{P}}\nabla)\cdot\widetilde{\mathbf{P}}\nabla v_n + \alpha^e\, v_n = 0 \quad \text{в} \;\; \Omega_d.$$

для $v_n := (\mathbf{n}\cdot\nabla)u$. Единственность решения уравнения влечет $v_n = 0$, а следовательно выполняется (3.16).

С помощью равенства $\mathbf{HP} = \mathbf{PH}$ получим

$$(\mathbf{I} - \phi\mathbf{H})^{-1}\mathbf{P} = \mathbf{P}(\mathbf{I} - \phi\mathbf{H})^{-1}. \tag{3.21}$$

Используя (3.21) и $\mathbf{P}^2 = \mathbf{P} = \mathbf{P}^T$, перепишем (3.20) следующим образом:

$$\int\limits_{\Omega_d} ((\mathbf{I} - \phi\mathbf{H})^{-1}\mathbf{P}\nabla u\cdot(\mathbf{I} - \phi\mathbf{H})^{-1}\nabla v + \alpha^e\, uv)\, d\mathbf{x} = \int\limits_{\Omega_d} f^e v\, d\mathbf{x} \quad \forall\, v\in H^1(\Omega_d). \tag{3.22}$$

Далее, согласно равенствам $|(\mathbf{I} - \mathbf{P})\nabla u|^2 = |(\mathbf{n}\cdot\nabla u)\mathbf{n}|^2 = (\mathbf{n}\cdot\nabla u)^2$, мы перепишем равенство (3.16) для решения (3.22) в следующей форме:

$$(\mathbf{I} - \mathbf{P})\nabla u = 0.$$

Благодаря (3.21), имеем:

$$\begin{aligned}(\mathbf{I} - \phi\mathbf{H})^{-1}\nabla u &= \mathbf{P}(\mathbf{I} - \phi\mathbf{H})^{-1}\nabla u + (\mathbf{I} - \mathbf{P})(\mathbf{I} - \phi\mathbf{H})^{-1}\nabla u\\ &= (\mathbf{I} - \phi\mathbf{H})^{-1}\mathbf{P}\nabla u + (\mathbf{I} - \phi\mathbf{H})^{-1}(\mathbf{I} - \mathbf{P})\nabla u\\ &= (\mathbf{I} - \phi\mathbf{H})^{-1}\mathbf{P}\nabla u\end{aligned}$$

для u - решения (3.22). В результате слабая формулировка (3.22) может быть записана следующим образом: найти $u \in H^1(\Omega_d)$, такую, что

$$\int\limits_{\Omega_d} ((\mathbf{I} - \phi\mathbf{H})^{-2}\nabla u \cdot \nabla v + \alpha^e\, uv)\, d\mathbf{x} = \int\limits_{\Omega_d} f^e v\, d\mathbf{x} \quad \text{для всех } v \in H^1(\Omega_d). \tag{3.23}$$

Найдем теперь сильную формулировку для (3.23). Для слагаемых, возникающих после интегрирования по частям и соответствующим граничным условиям, используем равенство $\mathbf{n} = \mathbf{n}_{\partial\Omega_d}$, т.к. граница области Ω_d является изоповерхностью уровня функции ϕ. Более того, из равенства $\mathbf{H}\mathbf{n} = 0$ следует $(\mathbf{I} - \phi\mathbf{H})^{-1}\mathbf{n} = \mathbf{n}$, а также

$$((\mathbf{I} - \phi\mathbf{H})^{-1}\nabla v) \cdot \mathbf{n} = (\nabla v) \cdot ((\mathbf{I} - \phi\mathbf{H})^{-1}\mathbf{n}) = (\mathbf{n} \cdot \nabla)v.$$

Таким образом, получим (3.23) в сильной форме:

$$\begin{aligned} -\operatorname{div}(\mathbf{I} - \phi\mathbf{H})^{-2}\nabla u + \alpha^e\, u = f^e \quad &\text{в } \Omega_d \\ (\mathbf{n} \cdot \nabla)u = 0 \quad &\text{на } \partial\Omega_d. \end{aligned} \tag{3.24}$$

Формулировка (3.24) имеет следующие преимущества относительно (3.11), (3.17) и (3.18): уравнения (3.24) являются невырожденными и эллиптическими, расширенная задача не вводит дополнительных параметров, граничные условия соответствуют свойству решения (3.16).

При выполнении описанных ранее условий на Ω_d, докажем следующую теорему:

Теорема 3.1. *При условии* (3.19)*, выполнено:*

(i) Задача (3.24) *имеет единственное слабое решение* $u \in H^1(\Omega_d)$*, удовлетворяющее* $\|u\|_{H^1(\Omega_d)} \le C\, \|f^e\|_{L^2(\Omega_d)}$*, где константа* C *зависит только от* α*,* Γ *и* d*;*

(ii) Предположим дополнительно, что $\Gamma \in C^3$*, тогда* $u \in H^2(\Omega_d)$ *и*

$$\|u\|_{H^2(\Omega_d)} \le C\, \|f^e\|_{L^2(\Omega_d)},$$

где константа C зависит только от α, Γ и d.

Доказательство. Сначала проверим, что билинейная форма

$$a(u,v) := \int\limits_{\Omega_d} ((\mathbf{I}-\phi\mathbf{H})^{-2}\nabla u \cdot \nabla v + \alpha^e\, uv)\, d\mathbf{x}$$

непрерывна и эллиптична на $H^1(\Omega_d)$.

Условие (3.19) приводит к следующему ограничению на спектр:

$$\mathrm{sp}(\mathbf{I}-\phi\mathbf{H}) \in \left[\tfrac{1}{2}, \tfrac{3}{2}\right] \;\Rightarrow\; \mathrm{sp}\left((\mathbf{I}-\phi\mathbf{H})^{-2}\right) \in \left[\tfrac{4}{9}, 4\right] \quad \text{для всех } \mathbf{x} \in \Omega_d.$$

Следовательно, выполнено

$$\frac{4}{9}\|\nabla u\|^2_{L^2(\Omega_d)} \le \int\limits_{\Omega_d} (\mathbf{I}-\phi\mathbf{H})^{-2}\nabla u \cdot \nabla u\, d\mathbf{x}, \tag{3.25}$$

$$\int\limits_{\Omega_d} (\mathbf{I}-\phi\mathbf{H})^{-2}\nabla u \cdot \nabla v\, d\mathbf{x} \le 4\|\nabla u\|_{L^2(\Omega_d)}\|\nabla v\|_{L^2(\Omega_d)}. \tag{3.26}$$

Используя оценку (3.26) и то, что $\|\alpha^e\|_{L^\infty(\Omega_d)} = \|\alpha\|_{L^\infty(\Gamma)}$, получим оценку непрерывности

$$\begin{aligned} a(u,v) &\le 4\|\nabla u\|_{L^2(\Omega_d)}\|\nabla v\|_{L^2(\Omega_d)} + \|\alpha^e\|_{L^\infty(\Omega_d)}\|u\|_{L^2(\Omega_d)}\|v\|_{L^2(\Omega_d)} \\ &\le (4 + \|\alpha\|_{L^\infty(\Gamma)})\|u\|_{H^1(\Omega_d)}\|v\|_{H^1(\Omega_d)}. \end{aligned}$$

Обозначим

$$\mu(\mathbf{x}) := \big(1 - d(\mathbf{x})\kappa_1(\mathbf{x})\big)\big(1 - d(\mathbf{x})\kappa_2(\mathbf{x})\big), \quad \mathbf{x} \in \Omega_d.$$

Из (2.20), (2.23) работы [42] имеем $\mu(\mathbf{x})\mathrm{d}\mathbf{x} = \mathrm{d}r\mathrm{d}\mathbf{s}(\mathbf{p}(\mathbf{x}))$, для $\mathbf{x} \in \Omega_d$, где $\mathrm{d}\mathbf{x}$ является мерой в Ω_d, $\mathrm{d}\mathbf{s}$ — поверхностная мера на Γ, и r — локальная координата для $\mathbf{x} \in \Gamma$ в нормальном направлении. Используя (3.19), получим $\frac{1}{4} \le \mu(\mathbf{x}) \le \frac{9}{4}$ для всех $\mathbf{x} \in \Omega_d$. Из этого, а также из (3.9) и (3.4), получим, что α^e строго положительна на подмножестве Ω_d положительной меры:

$$\widetilde{\mathcal{A}} := \mathrm{meas}_{\mathbf{x}}\{\mathbf{x} \in \Omega_d \,:\, \alpha^e(\mathbf{x}) \ge \alpha_0\} \ge \frac{8}{9}d\; \mathrm{meas}_{\mathbf{s}}\{\mathbf{x} \in \Gamma \,:\, \alpha(\mathbf{x}) \ge \alpha_0\} > 0,$$

при $\alpha_0 > 0$. Следовательно, аналогично неравенству на поверхности (3.5), верно неравенство типа Фридрихса

$$\|v\|^2_{L^2(\Omega_d)} \le \widetilde{C}_F(\|\nabla v\|^2_{L^2(\Omega_d)} + \|\sqrt{\alpha}v\|^2_{L^2(\Omega_d)}) \quad \forall\ v \in H^1(\Omega_d), \tag{3.27}$$

где константа $\widetilde{C}_F$ зависит от α_0 и $\widetilde{\mathcal{A}}$.

Неравенства (3.25) и (3.27) означают эллиптичность билинейной формы: $a(u, u) \ge c\|u\|^2$ для всех $u \in H^1(\Omega_d)$, где константа c зависит только от $\widetilde{C}_F$ из (3.27). Таким образом, часть (i) теоремы следует из леммы Лакса-Мильграма.

Для проверки части (ii) теоремы, заметим, что из $\Gamma \in C^3$ следует $\phi \in C^3$, см. [48], и $\partial\Omega_d \in C^3$. Следовательно, элементы матрицы "диффузии" $(\mathbf{I} - \phi\mathbf{H})^{-2}$ принадлежат C^1 и $\alpha \in L^\infty(\Gamma) \Rightarrow \alpha^e \in L^\infty(\Omega_d)$. Такой гладкости данных достаточно, чтобы эллиптическая задача была H^2-регулярной [3, 28], откуда следует утверждение части (ii) теоремы.

$\square$

Замечание 3.2.1. Теорема 3.1 показывает одно теоретическое преимущество новой расширенной формулировки (3.24) над (3.18) и (3.11): Для достаточно гладких данных может быть применена теория регулярности Агмона-Дуглиса-Ниренберга [28]. В частности, для $\Gamma \in C^3$, имеем $u \in H^2(\Omega_d)$, а из теоремы о следах (см. [3]) следует, что $u|_\Gamma \in H^1(\Gamma)$. Это позволяет рассматривать след u как слабое решение (3.2).

3.3. Численные методы

Для численного решения предложенной переформулировки уравнения на поверхности будем использовать метод конечных элементов и метод конечных объемов. Для решения методом конечных объемов используется предложенная многоточечная схема дискретизации диффузионного потока на многогранных сетках типа восьмеричное дерево со сколотыми ячейками. Метод конечных элементов на треугольных и тетраэдральных сетках позволяет применить хорошо разработанный аппарат численного анализа. В результате были получены оценки сходимости метода в различных нормах. Полный анализ метода и доказательства теорем приводятся в работе [38].

3.3.1. Метод конечных объемов

Для численного решения расширенного уравнения предлагается метод конечных объемов с использованием нелинейной схемы, описанной в главе 2, на сетках типа восьмеричное дерево со сколотыми ячейками. Будем рассматривать систему (3.24) как уравнение реакции-диффузии. Особое внимание в данном случае представляют граничные условия. В постановке (3.24) используются только граничные условия типа Неймана, причем в нормальном направлении, а не в конормальном. Поэтому, будем использовать следующую технику. Рассмотрим ячейку T и грань $f \in T$, на которой задано граничное условие типа Неймана. Сначала строится аппроксимация диффузионного потока на грани f в нормальном направлении, из которой можно получить выражение для C_f, зависящее от соседних ячеек. Это значение затем используется при аппроксимации диффузионного потока на грани f в конормальном направлении.

3.3.2. Метод конечных элементов

Пусть $\Gamma \in C^2$ и фиксирована область Ω_d шириной d, удовлетворяющей (3.19). Рассмотрим тетраэдризацию (триангуляцию) $\mathcal{T}$ области Ω_d на тетраэдральные (треугольные) элементы и определим область $\overline{\Omega}_h = \bigcup_{T\in\mathcal{T}} \overline{T}$, аппроксимирующую Ω_d.

В некоторых приложениях поверхность Γ может быть не известна явно, а задана лишь приближенно, например как изоповерхность нуля конечно-элементной функции уровня ϕ_h. В таких случаях, вместо гессиана $\mathbf{H} = \nabla^2\phi$ необходимо использовать дискретный гессиан $\mathbf{H}_h \approx \mathbf{H}$, который получается из ϕ_h с помощью какого-либо метода восстановления дискретного гессиана, см., например, [29, 30, 90]. Мы предполагаем, что ϕ_h и $\mathbf{H}_h$ удовлетворяют условию (3.19).

Пусть $V_h \subset H^1(\Omega_h)$ является пространством конечно-элементных функций. Тогда метод конечных элементов формулируется следующим образом: найти $u_h \in V_h$, удовлетворяющую

$$\int\limits_{\Omega_h} \left[\big((\mathbf{I} - \phi_h \mathbf{H}_h)^{-2} \nabla u_h \big) \cdot \nabla v_h + \alpha^e \, u_h v_h \right] \, d\mathbf{x} = \int\limits_{\Omega_h} f^e v_h \, d\mathbf{x} \quad \forall \, v_h \in V_h. \tag{3.28}$$

Анализ метода (3.28) проводится для специального случая, когда $\Omega_h = \Omega_d$, $\phi_h = \phi$, и $\mathbf{H}_h = \mathbf{H}$. Однако, численные эксперименты, описанные в следующем разделе, будут представлены и для общего случая.

Так как тензор диффузии $(\mathbf{I} - \phi\mathbf{H})^{-2}$ является положительно определенным и ограниченным, то верна следующая оптимальная оценка (см., например, [33]):

Теорема 3.2. *Пусть $\Omega_h = \Omega_d$, $\phi_h = \phi$, and $\mathbf{H}_h = \mathbf{H}$. Пусть u и u_h являются решениями* (3.24) *и* (3.28), *соответственно. Тогда верно*

$$\|u - u_h\|_{H^1(\Omega_d)} \leq C \inf_{v_h \in V_h} \|u - v_h\|_{H^1(\Omega_d)},$$

где константа C может зависеть только от α, Γ и d.

Теорема 3.2 и теорема о следах приводят к простой оценке ошибки на поверхности:

$$\|u - u_h\|_{L^2(\Gamma)} \leq C \inf_{v_h \in V_h} \|u - v_h\|_{H^1(\Omega_d)}. \tag{3.29}$$

Однако, эта оценка ошибки в L^2 норме не является оптимальной и может быть улучшена. Далее приведем без доказательства улучшенную оценку в L^2 норме.

Обозначим через

$$h = \sup_{T \in \mathcal{T}} \operatorname{diam}(T)$$

наибольший диаметр элементов и предположим, что V_h удовлетворяет

$$\inf_{v_h \in V_h} \|v - v_h\|_{H^1(\Omega_d)} \le C_a\, h \|v\|_{H^2(\Omega_d)}, \quad \forall\, v \in H^2(\Omega_d). \tag{3.30}$$

Для тетраэдра (треугольника) T обозначим через $\rho(T)$ диаметр вписанного шара. Пусть $\mathcal{T}_\Gamma$ является множеством всех тетраэдров, пересекающихся с Γ. Обозначим

$$\beta = \sup_{T \in \mathcal{T}_\Gamma} \operatorname{diam}(T)/\rho(T). \tag{3.31}$$

Будем считать, что тетраэдры (треугольники) из $\mathcal{T}_\Gamma$ являются регулярными, т.е. β не сильно велико.

Тогда можно доказать следующую оценку:

Теорема 3.3. *Пусть $\Omega_h = \Omega_d$, $\phi_h = \phi$, $\mathbf{H}_h = \mathbf{H}$, и $\Gamma \in C^3$. Пусть u и u_h являются решениями* (3.24) *и* (3.28), *соответственно. Тогда верно*

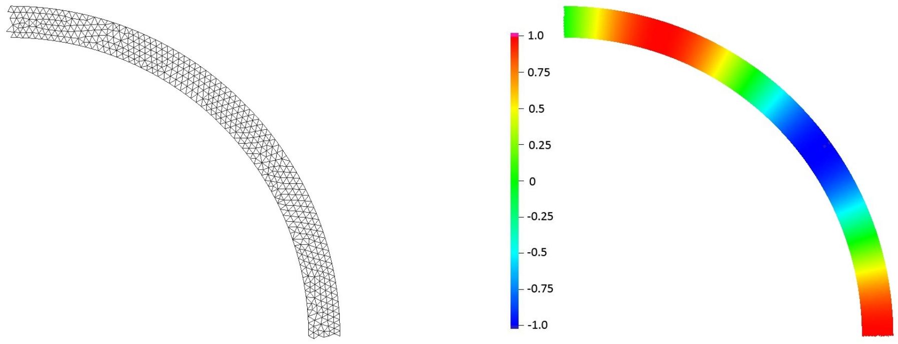
$$\|u - u_h\|_{L^2(\Gamma)} \le C\, h^{\frac{1}{2}} \inf_{v_h \in V_h} \|u - v_h\|_{H^1(\Omega_d)},$$

где константа C может зависеть только от α, Γ, d, констант C_a из (3.30), *и β из* (3.31).

Подробное доказательство этой оценки, а также оценки в L^∞ норме представлены Ольшанским в работе [38].

3.4. Численные эксперименты

В этом разделе будут представлены результаты некоторых численных экспериментов. Экперименты будут проведены для двумерного и трехмерного случаев. В численных экспериментах по умолчанию будем использовать точное аналитическое значение гессиана $\mathbf{H}$.



(a) (*б*)

Рис. 3.1. (a) Триангуляция Ω_d после одного шага измельчения для примера с окружностью и (*б*) визуализация решения расширенной задачи

3.4.1. Двумерный случай

Рассмотрим уравнение Лапласа-Бельтрами (3.2) в единичной окружности $\Gamma = \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x}\|_2 = 1\}$ с известным аналитически решением. В этом случае мы можем посчитать ошибку между точным и дискретными решениями. Положим $\alpha = 1$ и рассмотрим

$$u(r, \phi) = \cos(5\phi)$$

в полярных координатах, аналогично примеру 5.1 из [41].

Построим квази-равномерную триангуляцию Ω_d при $d = 0.05$ и применим процесс равномерного измельчения сетки. Сетка всегда привязывается к границе Ω_d, в результате $\partial\Omega_h$ является аппроксимацией $\partial\Omega_d$ с ошибкой порядка $O(h^2)$. На Рис. 3.1, a) показана верхняя правая часть сетки после одного шага измельчения.

Решение, полученное на сетке с тремя уровнями измельчения, показано на Рис. 3.1, *б*). Визуально решение выглядит постоянным вдоль нормального направления, каким и должно быть решение расширенной задачи. Далее, найдем ошибку численного решения на поверхности. Для оценки этой ошиб-

Таблица 3.1. Нормы ошибок для примера с окружностью с точным и приближенным гессианом. Ω_d фиксирована, $d = 0.1$. #d.o.f — число степеней свободы, # Iter — число итераций метода BCGstab

	h	#d.o.f.	L^2-norm	Порядок	C-norm	Порядок	# Iter.
$\mathbf{H}$	0.0417	610	0.318e-2		0.345e-2		13
	0.0208	2058	0.662e-3	2.26	0.148e-2	1.22	28
	0.0104	7351	0.179e-3	1.89	0.308e-3	2.26	60
	0.0052	27954	0.409e-4	2.13	0.812e-4	1.92	142
	0.0026	109576	0.983e-5	2.06	0.195e-4	2.06	325
$\mathbf{H}_h$	0.0417	610	0.449e-2		0.451e-2		13
	0.0208	2058	0.147e-2	1.61	0.182e-2	1.31	28
	0.0104	7351	0.390e-3	1.91	0.420e-3	2.16	63
	0.0052	27954	0.124e-3	1.65	0.160e-3	1.39	137
	0.0026	109576	0.325e-4	1.93	0.425e-4	1.91	297

ки, рассмотрим кусочно-линейные аппроксимации Γ и оценим ошибки вдоль них. Для проверки оценок из теорем, описанных в предыдущем разделе, в таблице 3.1 показаны нормы ошибок в L^2 и C-нормах на поверхности. Виден второй порядок сходимости в обеих нормах, когда в (3.28) используется точный аналитический гессиан. Также проведены эксперименты, когда ϕ_h является кусочно-линейным лагранжевым интерполянтом ϕ, а $\mathbf{H}_h$ является кусочно линейным непрерывным тензором, восстановленным из ϕ_h вариационным методом, см., например, [29]. В этом случае ошибки метода получаются больше, однако скорость сходимости остается близкой ко второму порядку. Дискретные задачи были решены с использованием метода BCGstab с переобуславливателем ILU2 [58] с относительной невязкой 10^{-9}. Также в правой колонке таблицы приведено число итераций метода.

Таблица 3.2. Нормы ошибок для примера со сферой с точным и приближенным гессианом. Ω_d фиксирована, $d = 0.1$.

	#d.o.f.	L^2-norm	Порядок	C-norm	Порядок	# Iter.
	1026	0.6085e-1		0.9033e-1		9
H	8547	0.1503e-1	1.98	0.1523e-1	2.52	23
	63632	0.3990e-2	1.98	0.3971e-2	2.01	47
	1026	0.8095e-1		0.1032e+00		9
$\mathbf{H}_h$	8547	0.2144e-1	1.88	0.1909e-1	2.39	23
	63632	0.5114e-2	2.14	0.4529e-2	2.15	46

3.4.2. Трехмерный случай

В трехмерном случае сначала рассмотрим уравнение Лапласа-Бельтрами на единичной сфере:

$$-\Delta_\Gamma u + u = f \quad \text{на } \Gamma,$$

с $\Gamma = \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\|_2 = 1\}$. Правая часть f выбирается так, чтобы решением уравнения являлась функция

$$u(\mathbf{x}) = \frac{12}{\|\mathbf{x}\|^3}\left(3x_1^2 x_2 - x_2^3\right), \quad \mathbf{x} = (x_1, x_2, x_3) \in \Omega.$$

Заметим, что u и f являются постоянными вдоль нормалей к Γ.

Для различных значений ширины d области Ω_d были построены квазиравномерные тетраэдральные сетки с помощью пакета Ani3D [17]. Сетка аппроксимирует границу Ω_d со вторым порядком точности. Полученные в результате алгебраические системы также решаются методом BCGStab с переобуславливателем ILU2 с относительной невязкой 10^{-9}.

В Таблицах 3.2–3.3 показаны нормы ошибок полученного решения на поверхности и число итераций предобусловленного метода BCGstab. Ошибки на поверхности посчитаны с помощью Γ_h — кусочно линейной аппроксимации

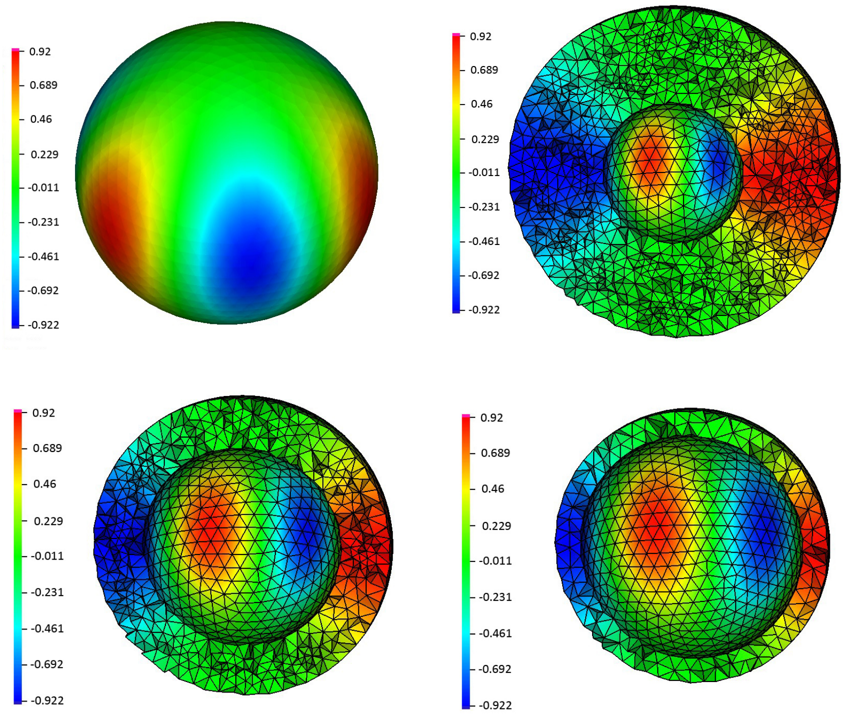


Рис. 3.2. Визуализация решения на сфере и срезы объемных сеток для Ω_d при $d = \{0.1,\, 0.2,\, 0.4\}$.

Таблица 3.3. Зависимость норм ошибок от ширины d для примера со сферой.

d	#d.o.f.	L^2-norm	C-norm	# Iter.
0.4	80442	0.7700e-2	0.6986e-2	46
0.2	34305	0.9389e-2	0.9560e-2	42
0.1	13560	0.9579e-2	0.1025e-1	35

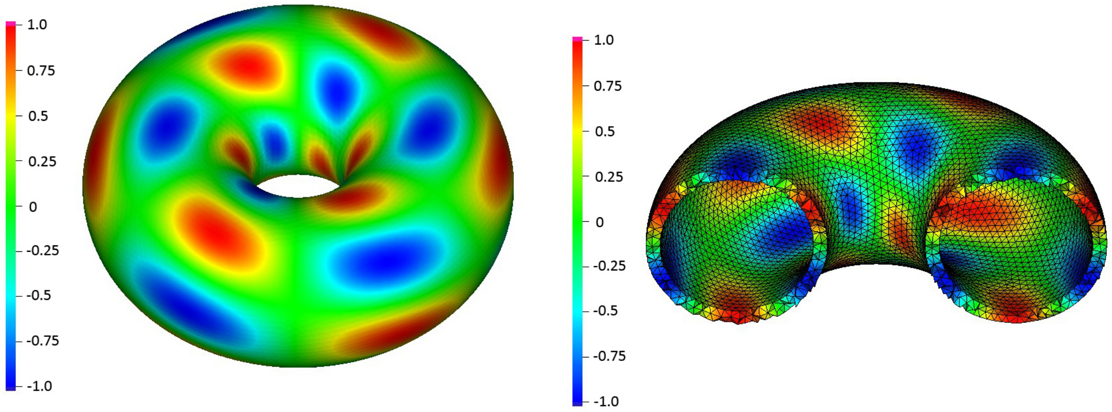


Рис. 3.3. Визуализация решения на торе и срез объемной сетки для Ω_d при $d = 0.1$.

Г. В Таблице 3.2 показаны нормы ошибок для случая фиксированной области Ω_d и набора измельчающихся сеток. Формальный порядок сходимости p рассчитывается по формуле

$$p = 3 \log \left(err_1 / err_2 \right) / \log \left((\#d.o.f.)_2 / (\#d.o.f.)_1 \right).$$

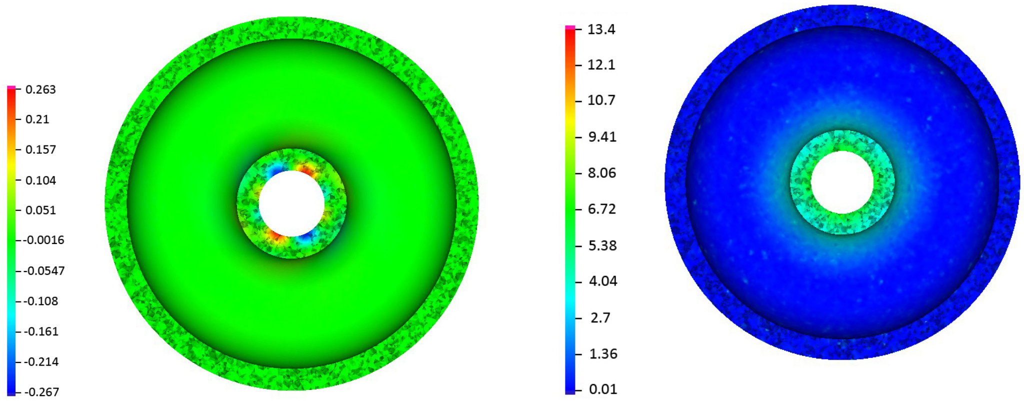
Виден второй порядок сходимости как в случае с точной функцией расстояния ϕ и аналитическим гессианом $\mathbf{H}$, так и в случае их дискретных аналогов ϕ_h и $\mathbf{H}_h$. Как и в двумерном случае, ϕ_h является кусочно-линейным лагранжевым интерполянтом ϕ, и $\mathbf{H}_h$ является кусочно-линейным непрерывным тензором, восстановленным из ϕ_h вариационным методом. В Таблице 3.3 показаны результаты для случаев, когда область Ω_d сужается к поверхности, а шаг сетки h остается примерно одинаковым для всех сеток.

На Рис. 3.2 показана визуализация решений для различных значений ширины области Ω_d. Можно наблюдать, что дискретное решение является постоянным вдоль нормального направления к поверхности.

Далее, повторим предыдущий эксперимент, только вместо сферы возьмем тор. Пусть $\Gamma = \{\mathbf{x} \in \Omega \mid r^2 = x_3^2 + (\sqrt{x_1^2 + x_2^2} - R)^2\}$. Примем $R = 1$ и

Таблица 3.4. Нормы ошибок для примера с тором с точным и восстановленным гессианом. Ω_d фиксирована, $d = 0.1$.

	#d.o.f.	L^2-norm	Порядок	C-norm	Порядок	# Iter.
	26257	0.7826e-1		0.1405e+00		18
$\mathbf{H}$	174021	0.2843e-1	1.61	0.8400e-1	0.82	42
	1511742	0.7780e-2	1.80	0.1077e-1	2.85	98
	26257	0.1144e+00		0.1708e+00		20
$\mathbf{H}_h$	174021	0.7680e-1	0.63	0.1569e+00	0.13	43
	1511742	0.6888e-1	0.15	0.9679e-1	0.67	94



(a) (б)

Рис. 3.4. (a) Ошибка численного решения с восстановленным гессианом. (б) Ошибка между точным и восстановленным гессианом.

$r = 0.6$. В системе координат (ρ, ϕ, θ), имеем

$$\mathbf{x} = R \begin{pmatrix} \cos\phi \\ \sin\phi \\ 0 \end{pmatrix} + \rho \begin{pmatrix} \cos\phi\cos\theta \\ \sin\phi\cos\theta \\ \sin\theta \end{pmatrix},$$

направление ρ является нормальным к Γ, $\frac{\partial \mathbf{x}}{\partial \rho} \perp \Gamma$ для $\mathbf{x} \in \Gamma$. Соответствующие решение u и правая часть f постоянны вдоль нормального направления:

$$\begin{aligned} u(\mathbf{x}) &= \sin(3\phi)\cos(3\theta + \phi), \\ f(\mathbf{x}) &= r^{-2}(9\sin(3\phi)\cos(3\theta+\phi)) \\ &\qquad - (R + r\cos(\theta))^{-2}(-10\sin(3\phi)\cos(3\theta+\phi) - 6\cos(3\phi)\sin(3\theta+\phi) \\ &\qquad - (r(R + r\cos(\theta))^{-1}(3\sin(\theta)\sin(3\phi)\sin(3\theta+\phi)). \end{aligned} \tag{3.32}$$

В Таблице 3.4 представлены нормы ошибок на поверхности тора. На Рис. 3.3 представлена визуализация решения. Снова, при использовании точного гессиана, порядок сходимости метода близок ко второму. Однако, когда точный гессиан заменяется на восстановленный, сходимость значительно ухудшается. Более точный анализ показывает, что ошибка сконцентрирована во внутреннем кольце тора, где Гауссова кривизна является отрицательной, см. Рис. 3.4, *а*). Однако, можно заметить, что эта ошибка связана с плохой точностью восстановления гессиана. Рассмотрим ошибку дискретного гессиана: $|\mathbf{H} - \mathbf{H}_h| = \left(\sum_{i,j=1}^{3} (\mathbf{H} - \mathbf{H}_h)_{i,j}^2\right)^{\frac{1}{2}}$. На Рис. 3.4, *б*) показана эта ошибка и можно заметить, что она также велика возле внутреннего кольца тора. В этой части Ω_d гессиан имеет ненулевые собственные значения разных знаков.

3.4.3. Тесты для метода конечных объемов

Вернемся снова к тесту со сферой, в котором мы будем использовать только точное аналитическое значение гессиана $\mathbf{H}$. Построим набор сеток

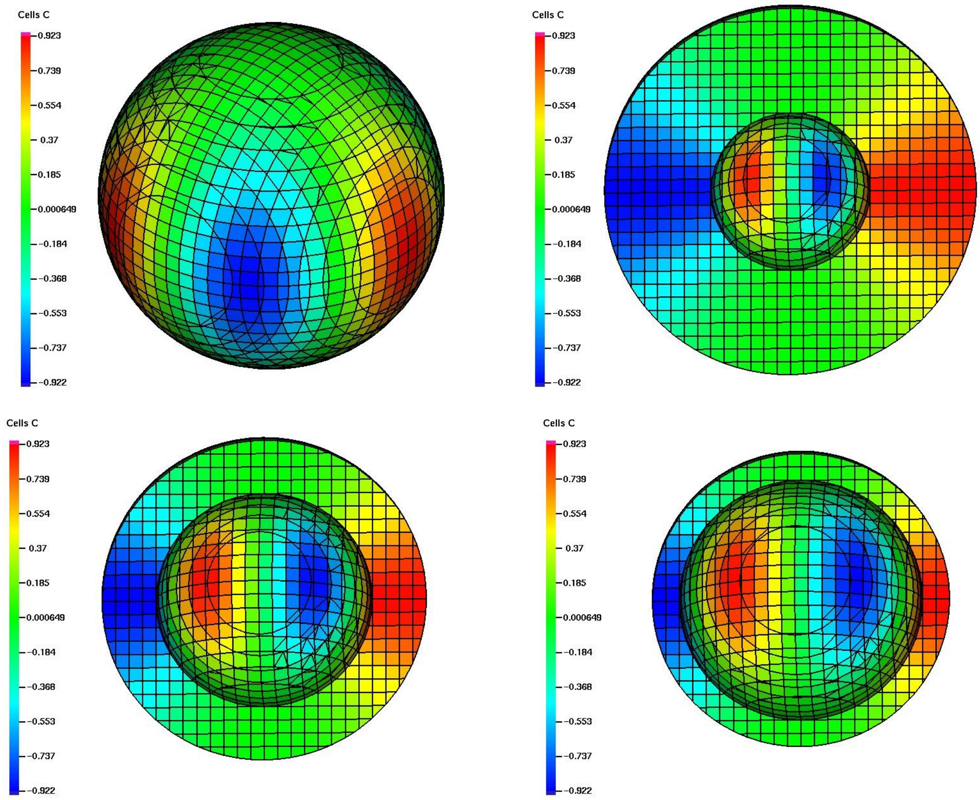


Рис. 3.5. Визуализация решения на сетке и срезы объемных сеток для Ω_d при $d = \{0.1, 0.2, 0.4\}$.

типа восьмеричное дерево со сколотыми ячейками для различной ширины d области Ω_d. Чтобы не измельчать сетку одновременно к границам области и к центру области, для простоты будем использовать равномерные сетки.

В Таблице 3.5 показаны нормы ошибок полученного решения на поверхности сферы для набора измельчающихся сеток в Ω_d при $d = 0.2$. Ошибки на поверхности посчитаны с помощью Γ_h — триангуляции Γ, полученной с помощью метода CMS. Для интерполяции численного решения на Γ_h использовалось кусочно-линейное восполнение кусочно-постоянного решения, порожденного методом конечных объемов. Из таблицы можно видеть второй порядок сходимости в L^2 норме, и порядок ниже второго в C - норме. В Таблице 3.6 показаны результаты для случаев, когда область Ω_d сужается к поверхности, а шаг сетки h остается фиксированным. Из таблицы можно видеть,

Таблица 3.5. Нормы ошибок для примера со сферой с точным гессианом. Ω_d фиксирована, $d = 0.2$.

h	#d.o.f.	L^2-norm	Порядок	C-norm	Порядок
1/20	1120	1.202e-2		4.555e-2	
1/40	6936	3.167e-3	1.92	1.795e-2	1.34
1/80	48472	7.859e-4	2.01	4.983e-3	1.84

Таблица 3.6. Зависимость норм ошибок от ширины d для примера со сферой, $h = 1/40$.

d	#d.o.f.	L^2-norm	C-norm
0.4	12744	2.878e-3	1.767e-2
0.2	6936	3.167e-3	1.795e-2
0.1	4312	2.989e-3	1.792e-2

что нормы ошибок примерно одинаковы для различных значений ширины d, следовательно можно выбирать d достаточно маленьким, чтобы минимизировать число степеней свободы и упростить решение задачи. На Рис. 3.5 показана визуализация решений для различных значений ширины области Ω_d. Можно также наблюдать, что дискретное решение является постоянным вдоль нормального направления к поверхности.

3.5. Выводы к третьей главе

В данной главе была предложена новая формулировка для решения эллиптического уравнения на гиперповерхностях в $\mathbb{R}^N$, $N = 2, 3$. Формулировка использует продолжение дифференциального уравнения в окрестность поверхности. В отличие от оригинального метода из [32], предложенное расширение приводят к невырожденным эллиптическим задачам в объемной области. Это позволяет напрямую использовать стандартные методы дискретизаций и применить хорошо разработанный аппарат численного анализа для эллиптических уравнений в частных производных. В работе используется метод конечных элементов на треугольных и тетраэдральных сетках, для которого в [38] доказываются оценки сходимости в L^2 и L^∞ нормах на поверхности, а также метод конечных объемов на многогранных сетках типа восьмеричное дерево со сколотыми ячейками. Однако, предлагаемый подход требует хорошую аппроксимацию гессиана функции расстояния со знаком.

Заключение

В диссертационной работе получены следующие результаты.

Разработана технология надежного построения гибридных конформных сеток типа восьмеричное дерево со сколотыми ячейками. Они состоят преимущественно из гексаэдров, а также из многогранных ячеек, приближающих криволинейные гладкие границы области со вторым порядком точности. Сетки могут динамически адаптироваться к изменениям численного решения. В работе доказана конечность алгоритма скалывания и показана его линейная сложность, а также приведены примеры сеток, подтверждающие данную оценку.

Для дискретизации уравнения диффузии на неструктурированных многогранных сетках предложена трехмерная версия двумерной монотонной нелинейной схемы метода конечных объемов [68], удовлетворяющая дискретному принципу максимума. Проведено экспериментальное исследование скорости сходимости предложенной схемы на разных типах сеток, в том числе на предложенных сетках типа восьмеричное дерево со сколотыми ячейками.

Для решения эллиптических уравнений на поверхностях предложена новая переформулировка, приводящая к расширенному невырожденному эллиптическому уравнению в окрестности поверхности. Это позволяет напрямую использовать стандартные методы дискретизаций в декартовых координатах. В работе используется предложенная нелинейная схема метода конечных объемов на сетках типа восьмеричное дерево со сколотыми ячейками, а также стандартный метод конечных элементов, для которого в [38] доказываются оценки сходимости.

Литература

1. Василевский Ю., Коньшин И., Копытов Г., Терехов К. INMOST Программная платформа и графическая среда для разработки параллельных численных моделей на сетках общего вида. М.: Издательство Московского Университета, 2013.

2. Данилов А. Технология построения неструктурированных сеток и монотонная дискретизация уравнения диффузии: Кандидатская диссертация. ИВМ РАН. Москва, 2010.

3. Ладыженская О. А. Краевые задачи математической физики. М.: Наука, 1973.

4. Марчук Г. И. Методы вычислительной математики. М.: Наука, 1989.

5. Марчук Г. И., Агошков В. И. Введение в проекционно-сеточные методы. М.: Наука, 1981.

6. Марчук Г. И., Кузнецов Ю. А. Итерационные методы и квадратичные функционалы // Методы вычислительной математики. Новосибирск: Наука, 1972.

7. Самарский А. А. Теория разностных схем. М.: Наука, 1982.

8. Самарский А. А. Введение в численные методы: Учебн. пособие для вузов. М.: Наука, 1987.

9. Тыртышников Е. Е. Методы численного анализа. М.: Издательский центр "Академия", 2007.

10. Чернышенко А. Численное решение уравнений Навье-Стокса на сетках

типа восьмеричное дерево со сколотыми ячейками // Труды Математического центра им. Н.И. Лобачевского. Т. 39. 2009. С. 391–393.

11. Чернышенко А. Алгоритм генерации многогранных сеток типа восьмеричное дерево в областях с несколькими материалами // Тезисы конференции "Тихоновские чтения". 2012. С. 23.

12. Чернышенко А. Генерация сеток типа восьмеричное дерево со сколотыми ячейками в многоматериальных областях // Тезисы VI Российской конференции "Актуальные проблемы прикладной математики и механики". 2012. С. 85–86.

13. Чернышенко А. Построение сеток типа восьмеричное дерево со сколотыми ячейками в неоднородных областях // Вычислительные методы и программирование. 2013. Т. 14. С. 229–245.

14. Чугунов В. Н. Алгоритм построения конформной квази-иерархической треугольной сетки, слабо δ-аппроксимирующей заданные ломаные // Ж. Выч. Мат. и Мат. Физ. 2009. Т. 49, № 5. С. 874–878.

15. Электронный ресурс: генератор HEXPRESS компании Numeca. `http://www.numeca.com/`.

16. Электронный ресурс: модель Стэнфордского кролика. `http://www-graphics.stanford.edu/data/3Dscanrep/`.

17. Электронный ресурс: Advanced Numerical Instruments 3D. `http://sourceforge.net/projects/ani3d/`.

18. Электронный ресурс: CUBIT. `http://cubit.sandia.gov/`.

19. Электронный ресурс: FEFLOW. `http://www.feflow.info/`.

20. Электронный ресурс: Hexotic. http://www-roc.inria.fr/gamma/gamma/Membres/CIPD/Loic.Marechal/Research/Hexotic.html.

21. Электронный ресурс: MeshGems: Suite of Reliable Meshing Software Components for High-Demanding CAD/CAE Developers. http://meshgems.com/volume-meshing-meshgems-hexa.html.

22. Электронный ресурс: MODFLOW-USG. http://water.usgs.gov/ogw/mfusg/.

23. Электронный ресурс: Netgen Mesh Generator. http://sourceforge.net/projects/netgen-mesher/.

24. Электронный ресурс: TetGen: A Quality Tetrahedral Mesh Generator. http://tetgen.berlios.de/.

25. Электронный ресурс: TetMesh-GHS3D. http://www.distene.com/build/meshing.html.

26. Aavatsmark I., Eigestad G., Mallison B., Nordbotten J. A compact multipoint flux approximation method with improved robustness // Num. Meth. for Part. Diff. Eqs. 2008. Vol. 24, no. 5. Pp. 1329–1360.

27. Agelas L., Eymard R., Herbin R. A nine-point finite volume scheme for the simulation of diffusion in heterogeneous media // C. R. Acad. Sci. Paris, Ser. I. 2009. Vol. 347. Pp. 673–676.

28. Agmon S., Douglis A., Nirenberg L. Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions // Communications on Pure and Applied Mathematics. 1995. Vol. 12. Pp. 623–727.

29. Agouzal A., Lipnikov K., Vassilevski Y. Adaptive generation of quasi-optimal tetrahedral meshes // East-West J. Numer. Math. 1999. Vol. 7. Pp. 223–244.

30. Agouzal A., Vassilevski Y. On a discrete Hessian recovery for P1 finite elements // Journal of Numerical Mathematics. 2002. Vol. 10. Pp. 1–12.

31. Berman A., Plemmons R. J. Nonnegative Matrices in the Mathematical Sciences. New York: Academic Press, 1979.

32. Bertalmio M., Cheng L., Osher S., Sapiro G. Variational problems and partial differential equations on implicit surfaces: The framework and examples in image processing and pattern formation // J. Comput. Phys. 2001. Vol. 174. Pp. 759–780.

33. Braess D. Finite elements: Theory, fast solvers, and applications in solid mechanics. Cambridge University Press, 2001.

34. Bretonnet L., Li Y., Hirsch C. 3D Navier-Stokes Cutcell Solver for Octree Meshes // Academy Colloquium on Immersed Boundary Methods. 2009.

35. Burger M. Finite element approximation of elliptic partial differential equations on implicit surfaces // Comp. Vis. Sci. 2009. Vol. 12. Pp. 87–100.

36. Chernyaev E. Marching Cubes 33: Construction of Topologically Correct Isosurfaces // Tech. Rep. CN/95-17. CERN, 1995.

37. Chernyshenko A. Generation of octree meshes with cut-cells for domains with multiple materials // Proc. of Second China-Russia Conf. on Num. Algebra and Applications. 2013. Pp. 59–60.

38. Chernyshenko A., Olshanskii M. Non-degenerate Eulerian finite element method for solving PDEs on surfaces // Russ. J. Numer. Anal. Math. Modelling. 2013. Vol. 28, no. 2. Pp. 101–124.

39. Danilov A. Unstructured tetrahedral mesh generation technology // Ж. Выч. Мат. и Мат. Физ. 2010. Т. 50, № 1. С. 146–163.

40. Danilov A., Vassilevski Y. A monotone nonlinear finite volume method for diffusion equations on conformal polyhedral meshes // Russ. J. Numer. Anal. Math. Modelling. 2009. Vol. 24, no. 3. Pp. 207–227.

41. Deckelnick K., Dziuk G., Elliott C. M., Heine C. An h-narrow band finite-element method for elliptic equations on implicit surfaces // IMA J. Numer. Anal. 2010. Vol. 30. Pp. 351–376.

42. Demlow A., Dziuk G. An adaptive finite element method for the Laplace-Beltrami operator on implicitly defined surfaces // SIAM J. Numer. Anal. 2007. Vol. 45. Pp. 421–442.

43. Diewald U., Preufer T., Rumpf M. Anisotropic diffusion in vector field visualization on Euclidean domains and surfaces // IEEE Trans. Visualization Comput. Graphics. 2000. Vol. 6. Pp. 139–149.

44. D'Otreppe V., Boman R., Ponthot J.-P. Generating smooth surface meshes from multi-region medical images // Int. J. Numer. Meth. Biomed. Engng. 2011. Pp. 709–749.

45. Dziuk G. Finite elements for the Beltrami operator on arbitrary surfaces // Partial Differential Equations and Calculus of Variations (S. Hildebrandt & R. Leis eds). Lecture Notes in Mathematics. Berlin: Springer, 1988. Vol. 1357. Pp. 142–155.

46. Dziuk G., Elliott C. M. Finite elements on evolving surfaces // IMA J. Numer. Anal. 2007. Vol. 27. Pp. 262–292.

47. Elliott C. M., Stinner B. Modeling and computation of two phase geometric biomembranes using surface finite elements // Journal of Computational Physics. 2010. Vol. 229. Pp. 6585–6612.

48. Foote R. L. Regularity of the distance function // Proc. Amer. Math. Soc. 1984. Vol. 92. Pp. 153–155.

49. Gao Z.-M., Wu J.-M. A small stencil and extremum-preserving scheme for anisotropic diffusion problems on arbitrary 2D and 3D meshes // J. Comp. Phys. 2013. Vol. 250. Pp. 308–331.

50. Golodetz S. Seeing Things Differently // Overload. 2007. Vol. 87. Pp. 4–9.

51. Greer J. B. An improvement of a recent Eulerian method for solving PDEs on general geometries // J. Sci. Comput. 2006. Vol. 29. Pp. 321–352.

52. Halpern D., Jensen O., Grotberg J. A theoretical study of surfactant and liquid delivery into the lung // J. Appl. Physiol. 1998. Vol. 85. Pp. 333–352.

53. Hansen C., Johnson C. The Visualisation Handbook. Oxford: Elsevier, 2005.

54. Hekhar S., Fayyad E., Yagel R., Cornhill J. F. Octree-based decimation of marching cubes surfaces // Proc. of IEEE Visualization. 1996. Pp. 335–342.

55. Ho C.-C., Wu F.-C., Chen B.-Y. et al. Cubical Marching Squares: Adaptive Feature Preserving Surface Extraction from Volume Data // EUROGRAPHICS. 2005. Vol. 24, no. 3.

56. Hu S., Hen C. Z., Ankanhalli K. Adaptive marching cubes // The Visual Computer. 1995. Vol. 11. Pp. 202–217.

57. Ju T., Losasso F., Schaefer S., Warren J. Dual Contouring of Hermite Data // Proceedings of SIGGRAPH. 2002. Pp. 339–346.

58. Kaporin I. High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU+U^TR+R^TU$-decomposition // Numer. Linear Algebra Appl. 1998. Vol. 5. Pp. 483–509.

59. Klause R. A., Winther R. Convergence of multipoint flux approximations on quadrilateral grids // Numer. Meth. Part. Diff. Eq. 2006. Vol. 22. Pp. 1438–1454.

60. Kobbelt L., Botsch M., U. Schwanecke H.-P. S. Marching cubes: A high resolution 3d surface construction algorithm // ACM SIGGRAPH. 2001. Pp. 57–66.

61. Korotov S., Krizek M., Neittaanmaki P. Weakened acute type condition for tetrahedral triangulations and the discrete maximum principle // Math. Comp. 2001. Vol. 70, no. 233. Pp. 107–119.

62. LePotier C. Schema volumes finis monotone pour des operateurs de diffusion fortement anisotropes sur des maillages de triangle non structures // C. C. Acad. Sci. Paris, 2005. Vol. 341. Pp. 787–792.

63. LePotier C. Finite volume scheme satisfying maximum and minimum principles for anisotropic diffusion operators // Finite Volumes for Complex Applications / Ed. by R. Eymard, J.-M. Hérard. 2008. Pp. 103–118.

64. Lewiner T., Lopes H., Vieira A. W., Tavares G. Efficient implementation of marching cubes' cases with topological guarantees // Journal of Graphics Tools. 2003. Vol. 8, no. 2. Pp. 1–15.

65. Lipnikov K., Svyatskiy D., Shashkov M., Vassilevski Y. Monotone finite volume schemes for diffusion equations on unstructured triangular and shape-regular polygonal meshes // J. Comp. Phys. 2007. Vol. 227. Pp. 492–512.

66. Lipnikov K., Svyatskiy D., Vassilevski Y. Interpolation-free monotone finite volume method for diffusion equations on polygonal meshes // J. Comp. Phys. 2009. Vol. 228, no. 3. Pp. 703–716.

67. Lipnikov K., Svyatskiy D., Vassilevski Y. A monotone finite volume method for advection–diffusion equations on unstructured polygonal meshes // J. Comp. Phys. 2010. Vol. 229. Pp. 4017–4032.

68. Lipnikov K., Svyatskiy D., Vassilevski Y. Minimal stencil finite volume scheme with the discrete maximum principle // Russ. J. Numer. Anal. Math. Modelling. 2012. Vol. 27, no. 4. Pp. 369–385.

69. Livnat Y., Shen H., Johnson C. R. A near optimal isosurface extraction algorithm using the span space // IEEE Trans. Vis. Comp.Graphics. 1996. Vol. 2. Pp. 73–84.

70. Lopes A., Brodlie K. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing // IEEE Transactions on Visualization & Computer Graphics. 2003. Vol. 9, no. 1. Pp. 16–29.

71. Lorensen W., Cline H. Marching cubes: A high resolution 3d surface construction algorithm // ACM SIGGRAPH. 1987. Vol. 21. Pp. 189–207.

72. Mullins W. W. Mass transport at interfaces in single component system // Metallurgical and Materials Trans. 1995. Vol. 26. Pp. 1917–1925.

73. Neilson G., Hamann B. The asymptotic decider: resolving the ambiguity in marching cubes // Proceedings of Visualisation. IEEE Computer Society Press, 1991. Pp. 83–91.

74. Nikitin K., Vassilevski Y. A monotone nonlinear finite volume method for

advection–diffusion equations on unstructured polyhedral meshes in 3D // Russ. J. Numer. Anal. Math. Modelling. 2010. Vol. 25, no. 4. Pp. 335–358.

75. Nordbotten J. M., Aavatsmark I., Eigestad G. T. Monotonicity of control volume methods // Numer. Math. 2007. Vol. 106, no. 2. Pp. 255–288.

76. Olshanskii M., Reusken A. A finite element method for surface PDEs: Matrix properties // Numer. Math. 2010. Vol. 114. Pp. 491–520.

77. Olshanskii M., Reusken A. Error analysis of a space-time finite element method for solving PDEs on evolving surfaces: Tech. rep.: Department of Mathematics, University of Houston, 2013.

78. Olshanskii M., Reusken A., Grande J. A Finite Element method for elliptic equations on surfaces // SIAM J. Numer. Anal. 2009. Vol. 47. Pp. 3339–3358.

79. Olshanskii M., Reusken A., X.Xu. An Eulerian space-time finite element method for diffusion problems on evolving surfaces: Tech. rep.: Department of Mathematics, University of Houston, 2013.

80. Olshanskii M., Reusken A., X.Xu. A stabilized finite element method for advection-diffusion equations on surfaces // IMA Journal of Numerical Analysis. 2013.

81. S. Gross A. R. Numerical methods for two-phase incompressible flows. Springer-Verlag, 1975. Vol. 40.

82. Saad Y. Iterative method for sparse Linear Systems. Second edition. Society for Industrial and Applied Mathematics, 2003.

83. Schroeder W., Zarge J., Lorensen W. Decimation of Triangle Meshes // Comput. Graph. 1992. Vol. 26. Pp. 65–70.

84. Shephard M., Georges M. Automatic three-dimensional mesh generation by the finite octree technique // Int. J. Numer. Methods Eng. 1991. Vol. 32. Pp. 709–749.

85. Sobolev S. Some Applications of Functional Analysis in Mathematical Physics. Third Edition. AMS, 1991.

86. Stoyan G. On maximum principles for monotone matrices // Linear Algebra and Its Applications. 1986. Vol. 78. Pp. 147–161.

87. Sutherland I., Hogdman G. W. Reentrant Polygon Clipping // Communication of the ACM, Graphics and Image Processing. 1974. Vol. 17, no. 1. Pp. 32–42.

88. Toga A. Brain Warping. New York: Academic Press, 1998.

89. Turk G. Generating textures on arbitrary surfaces using reaction–diffusion // Comput. Graphics. 1991. Vol. 25. Pp. 289–298.

90. Vallet M.-G., Manole C.-M., Dompierre J. et al. Numerical comparison of some Hessian recovery techniques // International Journal for Numerical Methods in Engineering. 2007. Vol. 72. Pp. 987–1007.

91. Wilhelms J., Gelder A. Octrees for faster isosurface generation // ACM Transactions on Graphics. 1992. Vol. 11, no. 3. Pp. 201–227.

92. Wu Z. Accurate and efficient three-dimensional mesh geenration for biomedical engineering applications: Ph.D. thesis. Worcester Polytechnic Institute, 2001.

93. Wu Z., Sullivan J. Multiple material marching cubes algorithm // Int. J. Numer. Meth. Engng. 2003. Vol. 58. Pp. 189–207.

94. Xu J., Zhao H.-K. An Eulerian formulation for solving partial differential equations along a moving interface // J. Sci. Comput. 2003. Vol. 19. Pp. 573–594.

95. Yuan A., Sheng Z. Monotone finite volume schemes for diffusion equations on polygonal meshes // J. Comp. Phys. 2008. Vol. 227, no. 12. Pp. 6288–6312.

96. Yuan G., Sheng Z. The finite volume scheme preserving extremum principle for diffusion equations on polygonal meshes // J. Comp. Phys. 2011. Vol. 230, no. 7. Pp. 2588–2604.

97. Zhang Y., Hughes T., Bajaj C. Three-Dimensional Mesh Generation by Finite Octree Technique // International Journal for Numerical Methods in Engineering. 1991. Vol. 32. Pp. 709–749.