

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ
И КИБЕРНЕТИКИ

На правах рукописи

Михалев Александр Юрьевич

УДК 519.6

Метод построения блочно-малоранговой
аппроксимации матрицы по её элементам

01.01.07 — Вычислительная математика

ДИССЕРТАЦИЯ

*На соискание учёной степени
кандидата физико-математических наук*

Научный руководитель
д.ф.-м.н. Оселедец И. В.

Москва 2014

Содержание

Введение	2
i.1 Исторический обзор	8
i.2 Содержание работ по главам	11
1 Предварительные сведения	13
1.1 Скелетное разложение	13
1.2 Доминантные подматрицы	17
1.3 Алгоритм maxvol	18
1.4 Мозаичное разбиение матрицы	19
1.5 \mathcal{H} -матрицы	22
1.6 \mathcal{H}^2 -матрицы	24
1.6.1 Матрично-векторные операции в \mathcal{H}^2 -формате	28
1.7 Выводы по главе	32
2 Прямоугольная скелетная аппроксимация	33
2.1 Объём прямоугольных подматриц	33
2.1.1 Оценка l_2 нормы строк матрицы коэффициентов	34
2.1.2 Алгоритм максимизации 2-объёма подматрицы	37
2.2 Прямоугольная псевдоскелетная аппроксимация	40
2.3 Модифицированная скелетная аппроксимация	49
2.4 Вложенное скелетное разложение	51
2.5 Оценка точности аппроксимации вложенными базисами	52
2.6 Выводы по главе	53
3 «Мультизарядовый» метод	55
3.1 «Мультизарядовое» представление \mathcal{H}^2 -матрицы	55

3.2	«Хорошие» строки и столбцы	57
3.2.1	«Хорошие» наборы и своя дальняя зона	58
3.2.2	Вычисление «хороших» наборов	61
3.3	Итерационный алгоритм	61
3.4	Проверка количества итераций и достигаемой точности	63
3.4.1	Программная реализация	64
3.4.2	Проверка на задаче многих тел	65
3.4.3	Проверка на граничном интегральном уравнении	67
3.5	Выводы по главе	68
4	Численные эксперименты	70
4.1	Континуальная модель растворителя	70
4.1.1	Уравнение РСМ и его дискретизация на молекулярной по- верхности	73
4.1.2	Решение уравнения РСМ в программе DISOLV	78
4.1.3	Решение уравнения РСМ при помощи мультитарядового метода в программе MCBHSOLV	79
4.1.4	Численное решение уравнения РСМ	81
4.1.5	Сравнение с мозаично-скелетным методом	92
4.1.6	Выводы по разделу «Континуальная модель растворителя» .	93
	Заключение	94
	Литература	96

Введение

Актуальность темы исследования. Ряд современных математических методов решения различных физических задач требуют решения систем линейных уравнений большой размерности. Этим методам посвящено множество работ [4, 65, 72, 74–83, 87–91]. Для понижения вычислительной сложности работы с возникающими большими матрицами необходимо учитывать *структуру* этих матриц. Учёт этой структуры может сильно ускорить решение конкретного класса задач. Одним из самых распространённых и изученных является класс разреженных матриц [22, 44, 47, 54, 69, 84, 92]. Основным показателем разреженности является количество ненулевых элементов. Этот показатель определяет как память, необходимую для хранения матрицы в памяти компьютера, так и скорость вычисления матрично-векторных произведений, одного из основополагающих элементов линейной алгебры. Кроме этого, класс разреженных матриц довольно глубоко исследован на предмет построения предобуславливателей [7–9, 27, 48–50]. Это позволяет успешно применять различные итерационные методы для быстрого решения систем линейных алгебраических уравнений. Однако, разреженная структура в матрицах – не единственная структура, представление которой требует малого количества параметров. В качестве примера структурированных матриц можно назвать семисепарабельные матрицы. Наддиагональная и поддиагональная части таких матриц являются частями малоранговых матриц, из чего следует существование малопараметрических представлений.

Диссертация посвящена *блочно-малоранговым матрицам*, а именно \mathcal{H} -матрицам [28, 30] и \mathcal{H}^2 -матрицам [11, 31]. Матрицы с данной структурой возникают, например, в задачах электростатики. За счёт того, что кулоновское взаимодействие на дальних расстояниях обладает достаточной гладкостью, потенциал, создаваемый двумя геометрически разделёнными группами заряженных тел (частиц) друг на друга, может быть вычислен приближенно, например, на основе

рядов Тейлора. Количество элементов ряда напрямую зависит от требуемой точности и удалённости групп тел. На этом факте основан *критерий геометрической разделённости*: матрицы, соответствующие взаимодействию геометрически разделённых групп тел, могут быть приближены матрицами малых рангов с любой необходимой точностью. Выделение малоранговых подматриц в исходной матрице составляет основу блочно-малоранговой структуры. Однако матрицы малого ранга возникают не только для кулоновского потенциала: как показано в работе [57], критерий геометрической разделённости выполняется, например, если функция взаимодействия тел является асимптотически гладкой [12]. Таким образом, функциональные матрицы, порождённые асимптотически гладкими функциями, обладают блочно-малоранговой структурой.

На практике оказывается, что критерий геометрической разделённости выполняется и для матриц, порождённых дискретизациями различных сингулярных и гиперсингулярных интегральных уравнений. Это позволяет применять метод дискретных вихрей [75] для решения задач акустики и аэрогидродинамики с большим количеством неизвестных.

Основным результатом диссертации является «мультизарядовый» метод построения \mathcal{H}^2 -приближений матриц. Этот метод относится к алгоритмам типа «серый» ящик: построение \mathcal{H}^2 -приближения опирается лишь на известную блочно-малоранговую структуру. Никакие из ранее построенных методов построения \mathcal{H}^2 -приближений не обладали данным свойством. К алгоритмам типа «серый» ящик относится мозаично-скелетный метод [57]: исходная матрица приближается \mathcal{H} -матрицей по известной блочно-малоранговой структуре. Оба эти метода, мозаично-скелетный и «мультизарядовый», являются чисто алгебраическими, так как человеку, решающему задачу, не нужно производить какие-либо действия с функцией взаимодействия отдельных тел. Отличие этих методов состоит лишь в том, что \mathcal{H}^2 -формат требует меньше памяти компьютера, чем \mathcal{H} -формат, вследствие чего умножение матрицы в \mathcal{H}^2 -формате на вектор потребует меньше времени, чем умножение матрицы в \mathcal{H} -формате на вектор. Алгебраическая натура «мультизарядового» метода позволяет применять его при решении различных задач с минимальными затратами человеческого времени.

Целью диссертационной работы являлась разработка нового быстрого метода построения аппроксимации блочно-малоранговой матрицы по её элементам, основанного на приближении блочных строк и блочных столбцов.

Научная новизна. Введено понятие p -объёма прямоугольных матриц, которое совпадает с модулем определителя матрицы в случае 1-объёма. Приведена явная функция 2-объёма для прямоугольных матриц, обоснованы верхние оценки коэффициентов в l_2 -норме и построен «жадный» алгоритм поиска подматрицы максимального 2-объёма. Оценки псевдоскелетной аппроксимации матриц расширены на случай выбора разного количества базисных строк и столбцов. Разработан и опробован на 2 задачах новый итерационный метод построения \mathcal{H}^2 -аппроксимации матриц.

Практическая ценность. Разработанный в диссертации метод поиска подматриц максимального 2-объёма может быть использован для уменьшения погрешностей скелетных аппроксимаций, построения новых крестовых методов аппроксимации матриц и многомерных тензоров. Предлагаемый в работе «мультизарядовый» метод, в свою очередь, является удобным средством решения различных гиперсингулярных уравнений и может быть, например, использован для решения задач аэро- и гидродинамики методом дискретных вихрей. Для применения «мультизарядового» метода к решению задачи не нужно строить какие-либо аналитические ряды, в отличие от быстрого мультипольного метода.

На защиту выносятся следующие результаты и положения:

1. Предложено новое определение p -объёма прямоугольных матриц и алгоритм поиска прямоугольных экстремальных подматриц. Доказаны оценки на рост коэффициентов, получены результаты по прямоугольной псевдоскелетной аппроксимации матриц.
2. Предложен быстрый итерационный алгоритм построения блочно-малоранговой аппроксимации матрицы в \mathcal{H}^2 -формате по её элементам, который существенно превосходит предыдущие подходы по достижимой точности и не требует дополнительной геометрической информации. Создан программный пакет h2tools, реализующий предложенные алгоритмы.
3. Проведено масштабное тестирование пакета h2tools на суперкомпьютере «Ломоносов» на задаче вычисления энергии десольватации на сетках с сот-

нями тысяч дискретных элементов. Получено ускорение в сотни раз по сравнению с программой DISOLV.

Апробация работы. Результаты диссертационной работы докладывались автором и обсуждались на научных семинарах ИВМ РАН, НИВЦ МГУ, ИПМ РАН и на конференциях: 52-я научная конференция МФТИ (2009), 55-я научная конференция МФТИ (2012), трехсторонний франко-немецко-российский научный семинар «Разделение переменных и приложения» (2010), международная научная конференция студентов, аспирантов и молодых ученых «Ломоносов-2011», научная конференция «Ломоносовские чтения-2011», научная конференция «Ломоносовские чтения-2012», научная конференция «Ломоносовские чтения-2013», научная конференция «Ломоносовские чтения-2014», международная суперкомпьютерная конференция «Научный сервис в сети Интернет: поиск новых решений» (2012), симпозиум «Биоинформатика и компьютерное конструирование лекарств» в рамках XXI Российского национального конгресса «Человек и лекарство» (2014).

Личный вклад. Результаты, описанные в главах 2 и 3: верхние оценки и алгоритм поиска подматриц максимального 2-объёма, оценки прямоугольной псевдоскелетной аппроксимации, итерационный «мультизарядовый» метод и численные эксперименты – получены автором самостоятельно. Кроме этого, автору принадлежат результаты численных экспериментов с применением «мультизарядового» метода, описанные в главе 4.

Публикации.

Статьи в журналах Web of Science

0а. **Oseledets I.V., Mikhalev A.Yu.** Representation of quasiseparable matrices using excluded sums and equivalent charges // *Linear Algebra Appl.* — 2012 . — Vol. 436, Issue 3 — P. 699–708.

Статьи в журналах из списка ВАК

0б. **Михалев А.Ю., Офёркин И.В., Оселедец И.В., Сулимов А.В., Тыртышников Е.Е., Сулимов В.Б.** Применение мультизарядового приближения больших плотных матриц в рамках модели поляризуемого континуума для растворителя // *Вычислительные методы и программирование.* — 2014. — Т. 15. — с. 9–21.

Принято к публикации

Ов. Михалев А.Ю., Оселедец И.В. Прямоугольные подматрицы максимального объема и их вычисление // Доклады академии наук. — 2014.

Основные результаты диссертационной работы опубликованы в 2 статьях: в статье в журнале, входящем в базу цитирования Web of Science [0а], и в статье в журнале из перечня ведущих рецензируемых научных журналов и изданий, рекомендованных ВАК [0б]. Статья [0а] опубликована в соавторстве с Оселедцом И.В. в журнале, входящем в базу цитирования **Web of Science**. В работе [0а] Михалеву А.Ю. принадлежит основная идея метода, программа ЭВМ и численные эксперименты, Оселедцу И.В. принадлежит общая постановка задачи. Статья [0б] опубликована в журнале из списка **ВАК**, автору диссертации принадлежат программа, реализующая предлагаемый в работе метод, и результаты соответствующих численных экспериментов. Статья [0в] принята к публикации в журнале Доклады академии наук, автору принадлежит оценка и метод, постановка задачи выполнялась в соавторстве с Оселедцом И. В.

Объём и структура работы. Работа состоит из 105 страниц и содержит введение, 4 главы, заключение и список литературы. Список литературы включает 92 пункта.

Благодарности. В первую очередь, автор выражает глубочайшую благодарность своим родителям, Михалевой Нине Васильевне и Михалеву Юрию Андреевичу, которые вместе с родительской заботой привили автору любовь к исследованиям и тягу к науке и обеспечили возможность получения высшего профессионального образования в МГУ им.М.В.Ломоносова и ИВМ РАН им.Г.И.Марчука, без сомнения лучших научных учреждениях России.

Нельзя не упомянуть то стечение обстоятельств, благодаря которому автор попал под научное влияние Оселедца Ивана Валерьевича, ставшего не только руководителем и наставником, но и другом. В ходе огромного количества конструктивных споров было придумано и проверено множество идей, большая часть которых оказалась успешной именно благодаря широте научных знаний и интуиции Оселедца Ивана Валерьевича, без участия которого не было бы данной диссертации. Умение подать задачу так, что любой ученик, включая автора, тут же начнёт с интересом её решать – вот то качество Ивана Валерьевича, которому автор благодарен более всего.

Автор выражает благодарность всему преподавательскому составу ИВМ РАН им.Г.И.Марчука, профессионализм которых внушает глубокое уважение и является примером для любого будущего кандидата и доктора физико-математических наук. Собственные методики и многолетний опыт преподавания не только значительно ускоряют образовательный процесс, но и делают его увлекательным. Особую благодарность автор выражает Тыртышникову Евгению Евгеньевичу, Агошкову Валерию Ивановичу, Горейнову Сергею Анатольевичу и Замарашкину Николаю Леонидовичу.

і.1. Исторический обзор

Блочно-малоранговые матрицы чаще всего появляются при решении задачи многих тел в различных постановках и, в частности, при решении граничных интегральных уравнений. Методы численного решения задачи многих тел развиваются уже давно и многие результаты стали классическими. Первыми методами решения гравитационной и электростатической задач многих тел являются метод Barnes-Hut [5] (1986 год) и быстрый мультипольный метод [25] (1987 год) соответственно.

Отличительной чертой метода Barnes-Hut является приближенное вычисление гравитационных взаимодействий при помощи центров масс отдельных групп тел (кластеров). За счет иерархического разбиения всех тел на кластеры сложность метода Barnes-Hut составляет $O(N \log N)$ операций. Однако, приближение взаимодействий отдельных тел центрами масс кластеров сказывается на точности результата.

Основным отличием быстрого мультипольного метода от метода Barnes-Hut является приближение всех попарных взаимодействий тел из двух кластеров при помощи конечных рядов. Простейший пример такого ряда – ряд Тейлора. Если x и y принадлежат кластерам с центрами x_0 и y_0 соответственно, то

$$f(x, y) \approx f(x_0, y_0) + \dots$$

Коэффициенты этого ряда называются мультипольными коэффициентами и, за счёт иерархии кластеров, все они вычисляются рекурсивно. Решение задачи многих тел быстрым мультипольным методом состоит из трех этапов:

1. Иерархическое вычисление мультипольных коэффициентов для источников взаимодействия, в литературе эта операция обозначается как M2M.
2. Вычисление локальных коэффициентов для приёмников взаимодействия по мультипольным коэффициентам. Эта операция так же обозначается M2L.
3. Иерархический пересчёт локальных коэффициентов в приёмники взаимодействия. Эта операция часто обозначается как L2L.

Так как этот метод опирается на приближение функции конечными рядами, то он, как и метод Barnes-Hut, является полностью аналитическим. Легко убедиться, что быстрый мультипольный метод применим не только к гравитационной задаче многих тел, так как единственным условием является ограничение остаточных членов аналитических разложений функции взаимодействия. Это порождает целое семейство быстрых мультипольных методов [1, 18, 20, 26, 37, 38, 40, 41, 62]. В частности, при помощи этих методов решаются задачи электростатики [42], акустики [14] и электромагнитного рассеяния [64]. Самой медленной операцией всех быстрых мультипольных методов является пересчёт локальных коэффициентов по мультипольным коэффициентам (M2L). Эта проблема рассмотрена в работе [13], где M2L операция заменяется вычислением свёрток на основе мультипольных разложений для плоских волн.

В 1993 году Е. Е. Тыртышниковым был разработан алгебраический метод решения задачи многих тел – мозаично-скелетный метод. Иерархическое разбиение всех тел на кластеры, используемое в быстрых мультипольных методах и методе Barnes-Hut, выделяет из матрицы, соответствующей задаче многих тел, подматрицы, близкие к матрицам малых рангов, в соответствии с критерием геометрической разделённости. Это разбиение матрицы называется мозаичным. Суть мозаично-скелетного метода состоит в замене каждой подматрицы мозаичного разбиения близкой к ней матрицу малого ранга. Фактически, исходная матрица приближается матрицей с мозаично-скелетной структурой. Зачастую в литературе матрицы с данной структурой называются \mathcal{H} -матрицами, и мы будем использовать это название как более устоявшееся. Формальное описание данной структуры дано в разделе 1.5. \mathcal{H} -формат является алгебраической структурой и требует только наличие мозаичного разбиения и существование малоранговых аппроксимаций каждой из подматриц данного мозаичного разбиения. Различные

прямые методы построения \mathcal{H} -матриц опираются на малоранговые разложения отдельных матриц, например на крестовую аппроксимацию, как это сделано в работе [58]. В упрощении, что ранги всех подматриц мозаичного разбиения равны r , хранение \mathcal{H} -матрицы требует $O(Nr \log N)$ памяти. Одно умножение матрицы в данном формате на вектор требует $O(Nr \log N)$ операций.

Мозаичное разбиение не только порождает разбиение матрицы на подматрицы малого ранга, но и выделяет иерархию специальных *блочных строк* и *блочных столбцов*. Если все блочные строки аппроксимируются матрицами малого ранга, то строчные базисы блочных строк обладают рекурсивными зависимостями. Этот факт применим и для блочных столбцов. Матрица, обладающая такой структурой, называется \mathcal{H}^2 -матрицей [11, 31] (2000 год). Формально \mathcal{H}^2 -формат описан в разделе 1.6. Иерархическая зависимость базисов блочных строк и блочных столбцов \mathcal{H}^2 -матриц аналогична иерархической зависимости мультипольных коэффициентов быстрого мультипольного метода. Из-за этого \mathcal{H}^2 -матрицы называют алгебраическим аналогом быстрого мультипольного метода. В разделе, посвященном этим матрицам показано, что требование по памяти и количеству операций на одно матрично-векторное произведение в \mathcal{H}^2 -формате составляет $O(Nr)$ (в предположении, что ранги всех блочных строк и блочных столбцов равны r). Основная проблема этого формата состоит в отсутствии прямых методов построения соответствующих разложений по элементам матрицы на основе лишь мозаичного разбиения. Существующие алгебраические методы построения \mathcal{H}^2 -разложений применимы лишь в очень ограниченном количестве случаев, например, если исходная матрица дана в \mathcal{H} -формате. Среди работ по полуаналитическим методам построения \mathcal{H}^2 -разложений можно выделить наиболее значимые работы следующих авторов: W. Hackbusch, S. Börm [29], L. Ying, G. Biros, D. Zorin [61], M. Bebendorf, R. Venn [6]. Первая из этих работ [29] касается решения интегральных уравнений. Подынтегральное ядро приближается полиномиально при помощи специально выбранных точек, на которых значения полинома и ядра совпадают. Оказывается, что полиномы, приближающие ядро на отдельных частях интегрируемой области, обладают иерархическими зависимостями. Это приводит к \mathcal{H}^2 -разложению соответствующей матрицы. В работе [61] предложен метод, использующий «прокси-поверхности» вокруг кластеров тел. На каждой из таких поверхностей специальным образом расставляются «заряженные»

тела, создающие тот же «потенциал» на специальных *проверочных телах* (набор тел, на которых проверяется равенство потенциалов), что и тела внутри кластера. Заряды и расположение дополнительных тел вычисляются рекурсивно, тем самым формируется \mathcal{H}^2 -разложение соответствующей матрицы. Метод, предложенный в работе [6], похож на метод из работы [61], однако тела, индуцирующие тот же «потенциал», что и кластер, выбираются среди тел самого кластера. Для уменьшения вычислительной сложности, авторы работы [6] специальным образом сокращают набор проверочных тел, что накладывает ограничения на функцию взаимодействия тел.

«Мультизарядовый» метод, являющийся основным результатом диссертации, позволяет строить \mathcal{H}^2 -приближение матрицы в «мультизарядовом» формате. Физический смысл этого формата опирается на следующую гипотезу: для любой достаточно удалённой точки пространства потенциал, созданный кластером заряженных частиц в эту точку пространства, можно приблизить потенциалом, создаваемым относительно небольшим количеством заряженных тел. На матричном уровне это эквивалентно построению приближения матрицы по некоторым из её строк или столбцов. Выбор таких базисных строк и столбцов может быть осуществлен без использования всей матрицы: достаточно выбрать «хорошую» подматрицу. «Мультизарядовый» метод выбирает такие подматрицы итерационно, независимо от поставленной физической задачи, из-за чего его можно отнести к классу алгебраических, матричных методов. Количество итераций, необходимых для приближения матрицы с требуемой точностью, зависит лишь от поставленной задачи. «Мультизарядового» метод является алгоритмом типа «серый ящик», так как для построения \mathcal{H}^2 -разложения, в рамках критерия геометрической разделённости, необходимым является лишь мозаичное разбиение.

i.2. Содержание работ по главам

Диссертация состоит из 4 глав: «Предварительные сведения», «Прямоугольная скелетная аппроксимация», «Мультизарядовый метод» и «Численные эксперименты». Глава «Предварительные сведения» посвящена малоранговым и блочно-малоранговым структурам. Построение малоранговых скелетных аппроксимаций относительно небольших матриц составляет основу «мультизарядового»

метода, предложенного в данной диссертации. Точность скелетного приближения напрямую зависит от определителя матрицы, расположенной на пересечении базисных строк и столбцов. Алгоритм **maxvol** позволяет быстро находить такие строки и столбцы, что матрица на их пересечении будет являться *доминантной* и, поэтому, будет обладать достаточно «хорошим» определителем.

В главе «Прямоугольная скелетная аппроксимация» оценки погрешностей, возникающих при приближении матриц скелетонами, расширены на случай, если количество выбранных базисных строк и столбцов не совпадают. Выбор дополнительных строк или столбцов в качестве базисных приводит к уменьшению погрешности. Это происходит за счёт уменьшения норм соответствующих матриц коэффициентов. Сформулировано определение 2-объёма матрицы, численно совпадающее с определителем в случае квадратных матриц. Показано, что жадная максимизация 2-объёма приводит к минимизации длин строк или столбцов матрицы коэффициентов, что использовано при оценке точности аппроксимации матрицы вложенными базисными строками.

В главе «Мультизарядовый метод» описан «мультизарядовый» формат \mathcal{H}^2 -матриц, сформулирован принцип «хороших» строк и столбцов и приведён итерационный алгоритм построения \mathcal{H}^2 -приближения матриц, основанный на адаптивном выборе «хороших» наборов строк и столбцов. На примере двух задач показано количество итераций, необходимых для достижения требуемой точности. Приведены графики зависимости времени работы и потребляемой памяти «мультизарядового» алгоритма от размера исходной задачи и параметра точности.

Основу главы «Численные эксперименты» составляет сравнение производительности «мультизарядового» метода для вычисления энергии десольватации молекул в воде с программой DISOLV и мозаично-скелетным методом.

Глава 1

Предварительные сведения

1.1. Скелетное разложение

Скелетное разложение является одним из важнейших разложений в линейной алгебре. Оно основано на выборе линейно-независимых строк и столбцов матрицы. Скелетное разложение для матрицы $A \in \mathbb{C}^{N \times M}$, $\text{rank } A = r$ имеет вид:

$$A = C\hat{A}^{-1}R,$$

где $C \in \mathbb{C}^{N \times r}$ состоит из базисных столбцов матрицы A , $R \in \mathbb{C}^{r \times M}$ – из базисных строк, а $\hat{A} \in \mathbb{C}^{r \times r}$ – подматрица на их пересечении. В качестве \hat{A} может быть выбрана любая невырожденная подматрица размера $r \times r$ из матрицы A . При дополнительном условии $r \ll \max(M, N)$, скелетное разложение позволяет не только снизить память, необходимую для хранения матрицы, но и существенно сократить время матрично-векторных операций.

В практических задачах матрица A никогда не имеет точного ранга, что, в частности, связано с наличием разного рода шумов. Из-за этого выбор любой невырожденной подматрицы \hat{A} может привести к неожиданному результату: ошибки машинных вычислений доминируют над оперируемыми данными. Это преобладание возникает только в тех случаях, когда подматрица \hat{A} оказывается близка вырожденной матрице. Таким образом, необходимо не просто выбирать базисные строки и столбцы, но и оценивать точность аппроксимации:

$$\|A - C\hat{A}^{-1}R\|.$$

Часто оценки скелетной аппроксимации сопровождаются оценками CGR-аппроксимации:

$$\|A - CGR\|.$$

Единственное отличие CGR-аппроксимации состоит в том, что матрица G не обязательно является обратной к подматрице \hat{A}^{-1} . Скелетная и CGR аппроксимации совпадают при условии равенства матриц G и \hat{A}^{-1} .

Различные оценки точности скелетной и CGR аппроксимаций в l_2 -норме и C -норме даны в работах [23, 24]. Эти оценки основываются на *подматрицах максимального объёма*. Объёмом квадратной матрицы A называется модуль её определителя:

$$\text{vol}(A) = |\det(A)|.$$

Соответственно, подматрица максимального объёма это подматрица, объём которой максимален среди всех подматриц размера $r \times r$. Известно, что задача поиска такой подматрицы решается перебором по всем подматрицам, поэтому нужны эффективные методы нахождения подматриц с объёмом, близким к максимальному.

В случае, когда известно, что матрица обладает фиксированным рангом $r \ll \max(M, N)$, подматрицу максимального объёма можно искать по факторам разложения $A = UV$, $U \in \mathbb{C}^{N \times r}$, $V \in \mathbb{C}^{r \times M}$. Искомая подматрица лежит на пересечении строк, соответствующих подматрице максимального объёма в U , и столбцов, соответствующих подматрице максимального объёма в V . Этот факт является следствием того, что определитель произведения двух квадратных матриц равен произведению их определителей:

$$\det(XY) = \det(X)\det(Y).$$

Оценки погрешности в спектральной норме, данные в работе [24], опираются на τ -псевдообратные матрицы и функцию $t(r, n)$. Для любой матрицы $A \in \mathbb{C}^{N \times M}$, $K = \min(N, M)$, псевдообратная матрица A_τ^\dagger определяется из сингулярного разложения следующим способом:

$$A = USV, U \in \mathbb{C}^{N \times K}, V \in \mathbb{C}^{K \times M}, S = \text{diag}(\sigma_1, \dots, \sigma_K),$$

$$A_\tau^\dagger = V^T S_\tau^\dagger U^T, S_\tau^\dagger = \text{diag}(\sigma_1^\dagger, \dots, \sigma_K^\dagger), \sigma_i^\dagger = \begin{cases} \sigma_i^{-1}, & \sigma_i \geq \tau \\ 0, & \sigma_i < \tau \end{cases}.$$

Пусть \mathcal{P} – множество всех ортогональных матриц размера $n \times r$, $\mathcal{M}(U)$ – множество всех подматриц размера $r \times r$ в матрице U . Тогда $t(r, n)$ определяется по формуле

$$t(r, n) = \frac{1}{\min_{U \in \mathcal{P}} \max_{X \in \mathcal{M}(U)} \sigma_{\min}(X)}.$$

Число $t(r, n)$ показывает, что для любой ортогональной матрицы размера $n \times r$ есть такая подматрица, что обратная к ней имеет ограниченную этим числом $t(r, n)$ спектральную норму. Выпишем основные оценки, полученные в рамках работы [24]

Теорема 1.1.1 ([24]). Пусть $A, F \in \mathbb{C}^{N \times M}$, $\text{rank}(A - F) \leq r$, $\|F\|_2 \leq \varepsilon$ при некотором $\varepsilon > 0$. Тогда существуют r строк и r столбцов определяющих CGR аппроксимацию такую, что

$$\|A - CGR\|_2 \leq \varepsilon \left\{ 1 + \left[\sqrt{t(r, N)} + \sqrt{t(r, M)} \right]^2 \right\}.$$

Теорема 1.1.2 ([24]). В рамках предположений предыдущей теоремы существует псевдоскелетная аппроксимация с матрицей G , вычисленной на основе подматрицы \hat{A} , что

$$\|A - CGR\|_2 \leq \varepsilon [2 + 2t(r, N) + 2t(r, M) + 5t(r, N)t(r, M)].$$

Данную оценку можно улучшить до следующей:

$$\|A - CGR\|_2 \leq \varepsilon \sqrt{1 + t^2(r, P)} \left\{ 1 + \left[\sqrt{t(r, N)} + \sqrt{t(r, M)} \right]^2 \right\},$$

где $P = \min(N, M)$.

Кроме этого, нам понадобится оценка C -нормы погрешности аппроксимации из работы [23], которую мы приведём с доказательством.

Теорема 1.1.3 ([23]). Пусть матрица A имеет следующую блочную форму:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

где подматрица $A_{11} \in \mathbb{C}^{r \times r}$ является невырожденной и имеет максимальный объём среди всех подматриц размера $r \times r$. Тогда имеет силу следующая оценка:

$$\|A_{22} - A_{21}A_{11}^{-1}A_{12}\|_C \leq (r + 1)\sigma_{r+1}(A),$$

где $\sigma_{r+1}(A)$ – $(r + 1)$ -ое по старшинству сингулярное число матрицы A .

Доказательство. Так как C -норма показывает максимум модуля, то для получения требуемой оценки достаточно доказать ограничение по модулю одного элемента матрицы A_{22} . Для этого рассмотрим подматрицу \hat{A} , являющуюся расширением подматрицы A_{11} дополнительными столбцом и строкой:

$$\hat{A} = \begin{bmatrix} A_{11} & b \\ c^T & a \end{bmatrix},$$

и параметр $\gamma = a - c^T A_{11}^{-1} b$. Очевидно, этот параметр и является C -нормой погрешности, которую нам необходимо измерить. В случае $\gamma = 0$ оценка C -нормы верна при любых исходных r и $\sigma_{r+1}(A)$. Если $\sigma_{r+1}(A) = 0$, то, очевидно, матрица \hat{A} сингулярна и $\gamma = 0$. Осталось рассмотреть случай $\gamma \neq 0, \sigma_{r+1}(A) > 0$. Воспользуемся известным тождеством для определителей:

$$\det(\hat{A}) = (a - c^T A_{11}^{-1} b) \det(A_{11}) = \gamma \det(A_{11}),$$

$$\gamma^{-1} = \frac{\det(A_{11})}{\det(\hat{A})}.$$

Отметим, что γ^{-1} является элементом матрицы \hat{A}^{-1} . Так как A_{11} имеет максимальное значение модуля среди всех подматриц размера $r \times r$, то, очевидно, все элементы матрицы \hat{A}^{-1} не превосходят по модулю значение γ^{-1} . Из этого следует, что

$$\begin{aligned} \|\hat{A}^{-1}\|_C &= |\gamma^{-1}|, \\ |\gamma| &= \|\hat{A}^{-1}\|_C^{-1}. \end{aligned}$$

Так как \hat{A} является подматрицей матрицы A , то

$$\sigma_{r+1}(A) \geq \sigma_{r+1}(\hat{A}),$$

$$\sigma_{r+1}^{-1}(A) \leq \sigma_{r+1}^{-1}(\hat{A}) = \sigma_1(\hat{A}^{-1}).$$

Оценим старшее сингулярное число матрицы \hat{A}^{-1} :

$$\sigma_1(\hat{A}^{-1}) \leq \|\hat{A}^{-1}\|_F \leq (r+1) \|\hat{A}^{-1}\|_C.$$

Учитывая эти тождества, получаем:

$$\sigma_{r+1}^{-1}(A) \leq (r+1) \|\hat{A}^{-1}\|_C.$$

В итоге, получаем неравенство на γ :

$$|\gamma| = \|\hat{A}^{-1}\|_C^{-1} \leq (r+1)\sigma_{r+1}(A),$$

откуда следует неравенство

$$\|A_{22} - A_{21}A_{11}^{-1}A_{12}\|_C \leq (r+1)\sigma_{r+1}(A),$$

что и требовалось доказать.

Оценку из этой теоремы можно распространить на случай, когда подматрица A_{11} не обладает максимально возможным объёмом. Приведём только формулировку соответствующей теоремы из [23].

Теорема 1.1.4 ([23]). *Пусть, в рамках условий Теоремы 1.1.3, отношение объёма матрицы A_{11} к максимальному объёму не менее α , тогда верно следующее неравенство:*

$$\|A_{22} - A_{21}A_{11}^{-1}A_{12}\|_C \leq \alpha^{-1}(r+1)\sigma_{r+1}(A).$$

Замечание. Совсем не обязательно выбирать отношение объёма подматрицы A_{11} к максимальному объёму. Пусть множество $\mathcal{M}(A_{11})$ обозначает все подматрицы матрицы A , отличающиеся от подматрицы A_{11} не более чем в 1 столбце и не более чем в 1 строке. Как видно из доказательства Теоремы 1.1.3, параметр α есть отношение объёмов подматрицы A_{11} к максимальному объёму подматрицы из множества $\mathcal{M}(A_{11})$.

1.2. Доминантные подматрицы

Пусть дана матрица $A \in \mathbb{C}^{N \times r}$. Будем называть невырожденную подматрицу \hat{A} размера $r \times r$ *доминантной*, если все элементы матрицы $A\hat{A}^{-1}$ по модулю не превосходят 1. В статье [34] показано, что подматрица максимального объёма является доминантной, и приведена наилучшая оценка отношения объёма любой доминантной матрицы к максимальному объёму:

Лемма 1.2.1 ([34]). *Для любой матрицы $A \in \mathbb{C}^{N \times r}$ ранга r : если $\hat{A} \in \mathbb{C}^{r \times r}$ – доминантная подматрица, а A_M – подматрица максимального объёма, то верно следующее отношение:*

$$|\det(\hat{A})| \geq \sqrt{\frac{1}{r^r}} |\det(A_M)|$$

Однако, как указано в замечании к Теореме 1.1.4, оценка на параметр α может быть намного лучше.

Предположим, что мы уже умеем находить доминантные подматрицы размера $r \times r$ в матрицах размера $N \times r$ и $r \times M$. Тогда для поиска подматрицы максимального объёма можно воспользоваться сингулярным разложением:

$$A = \begin{bmatrix} U_Z & U_F \end{bmatrix} \begin{bmatrix} S_Z & 0 \\ 0 & S_F \end{bmatrix} \begin{bmatrix} V_Z \\ V_F \end{bmatrix}, \quad Z = U_Z S_Z V_Z, \quad F = U_F S_F V_F.$$

При этом матрица Z основана на r старших сингулярных векторах, $\|F\|_2 = \sigma_{r+1}(A)$. Вычисляя доминантные подматрицы в матрицах U_Z и V_Z , получаем набор строк и столбцов для построения скелетной аппроксимации, то есть получаем матрицы C и R . В предположении $\sigma_{r+1}(A) \ll \sigma_r(A)$, матрица на пересечении выбранных строк и столбцов, матрица \hat{A} , по объёму будет близка к подматрице максимального объёма из множества подматриц $\mathcal{M}(\hat{A})$. Однако, может случиться и такое, что \hat{A} близка к вырожденной. В статье [23] показано, что вместо обращения матрицы \hat{A} можно подставить специально вычисленную матрицу G , что можно оценить точность CGR аппроксимации:

$$\|A - CGR\|_2 \leq \alpha^{-1}(r+1)\sigma_{r+1}(A),$$

где параметр α вычисляется уже не через отношение объёмов, а на основе элементов матриц CG и GR .

1.3. Алгоритм **maxvol**

Для вычисления доминантных подматриц размера $r \times r$ в матрицах $N \times r$ существует алгоритм **maxvol**. Этот алгоритм сформулирован и применён для задачи поиска максимального по модулю элемента матрицы в статье [34]. Так как этот алгоритм является одним из основных элементов в данной диссертационной работе, сформулируем его в Алгоритме 1. Кратко его суть сводится к следующему: при помощи LU-разложения находим в матрице A невырожденную подматрицу \hat{A} , вычисляем $C = A\hat{A}^{-1}$, если $|C_{ij}| > 1$, то вместо j -ой строки матрицы \hat{A} записываем i -ую строку матрицы A , пересчитывая при этом матрицу C при помощи формулы Шермана-Вудбери-Моррисона [52, 60]. Когда все элементы матрицы C

по модулю не превосходят 1, любая замена строки матрицы \hat{A} на строку матрицы A не увеличивает модуль определителя матрицы \hat{A} . В тексте Алгоритма 1 использовались следующие обозначения:

C_{ij} – максимальный по модулю элемент матрицы C ,

\hat{A}_j – j -ая строка матрицы \hat{A} ,

A_i – i -ая строка матрицы A ,

C_j – j -ый столбец матрицы C ,

C_i – i -ая строка матрицы C ,

e_j – j -ая строка единичной матрицы размера $r \times r$.

Алгоритм 1 maxvol [34]

Дано: Полноранговая матрица $A \in \mathbb{C}^{N \times r}$, $N > r$

Вычислить: Доминантную подматрицу \hat{A} и матрицу коэффициентов C такую, что $A = C\hat{A}$

- 1: Найти невырожденную подматрицу \hat{A} в матрице A
 - 2: $C := A\hat{A}^{-1}$, $C_{ij} = \max(C)$
 - 3: **while** $C_{ij} > 1$ **do**
 - 4: $\hat{A}_j := A_i$
 - 5: $C := C - C_j(C_i - e_j)/C_{ij}$
 - 6: $C_{ij} = \max(C)$
 - 7: **end while**
 - 8: **return** C, \hat{A}
-

1.4. Мозаичное разбиение матрицы

Пусть I – множество индексов строк матрицы A . Построим следующую структуру: t_0 – инициализируем множеством I , t_1 и t_2 есть непересекающиеся подмножества множества t_0 , в объединении дающие множество t_0 . Далее каждый из t_1 и t_2 разбиваем на непересекающиеся подмножества t_{11}, t_{12}, t_{21} и t_{22} так, что $t_1 = t_{11} \cup t_{12}$ и $t_2 = t_{21} \cup t_{22}$. Применяя непересекающееся разбиение для $t_{11}, t_{12}, t_{21}, t_{22}$ и так далее, получаем иерархическую структуру. Эта структура называется *иерархическим кластерным деревом* строк матрицы A , так как исходные строки делятся на кластеры строк иерархически. Для простоты будем считать де-

рево бинарным (т.е. у каждого нелистьевого узла есть ровно два сына), однако подход обобщается на произвольное количество детей.

N_I – количество строк матрицы A ,

N_J – количество столбцов матрицы A ,

I – множество номеров строк матрицы A ,

J – множество номеров столбцов матрицы A ,

\mathcal{T}_I – иерархическое кластерное дерево строк матрицы A ,

\mathcal{T}_J – иерархическое кластерное дерево столбцов матрицы A ,

$\text{ind}(t)$ – номера (index) строк или столбцов, относящихся к узлу t ,

$A(t, s)$ – подматрица матрицы A на строках $\text{ind}(t)$ и столбцах $\text{ind}(s)$,

$\chi(t)$ – диагональная матрица такая, что $\chi(t)_{ii} = 1$, если $i \in \text{ind}(t)$ и 0 в противном случае,

$\chi(t)$ имеет размер $N_I \times N_I$, если $t \in \mathcal{T}_I$, или $N_J \times N_J$, если $t \in \mathcal{T}_J$,

$\mathbf{s}(t)$ – (sons) узлы-сыновья узла t ,

$\mathbf{p}(t)$ – (parent) узел-отец узла t ,

$\mathbf{pp}(t)$ – (parents) множество узлов, составленных по правилу $\mathbf{pp}(t) = \{\mathbf{p}(t), \mathbf{pp}(\mathbf{p}(t))\}$.

Пусть нам уже даны иерархические кластерные деревья строк \mathcal{T}_I и столбцов \mathcal{T}_J . Выберем I_0 – корень дерева \mathcal{T}_I и J_0 – корень дерева \mathcal{T}_J и запустим для них следующую процедуру: если подматрица, основанная на строках и столбцах, соответствующих выбранным узлам деревьев, приближается матрицей малого ранга, то отмечаем саму подматрицу и пару выбранных узлов меткой «F», иначе запускаем эту процедуру для каждой пары, составленной из узла-сына каждого из выбранных узлов. Если у какого-то из узлов нет узлов-сыновей, то такую пару, как и соответствующую подматрицу, просто отмечаем меткой «C». Таким образом, получаем набор отмеченных пар узлов кластерных деревьев. Заметим, что отмеченные подматрицы не имеют общих элементов. Таким образом, все элементы матрицы A попадают ровно в 1 отмеченную подматрицу.

Формализуем понятия, объявленные в абзаце выше:

$\mathbf{F}(A)$ – все пары узлов, отмеченные меткой «F»,

$\mathbf{C}(A)$ – все пары узлов, отмеченные меткой «C».

В случае, если каждая подматрица с меткой «F» имеет малый по сравнению с размерами подматрицы ранг, то описанное выше разбиение матрицы называется

мозаичным разбиением. В данном случае, разбиение матрицы подразумевает раскрашивание всей матрицы в соответствии с метками. Матрицы, имеющие мозаичное разбиение, называются \mathcal{H} -матрицами. Мозаичное разбиение так же называется \mathcal{H} -форматом. Один из простейших примеров мозаичного разбиения матрицы приведён в Рисунке 1.1

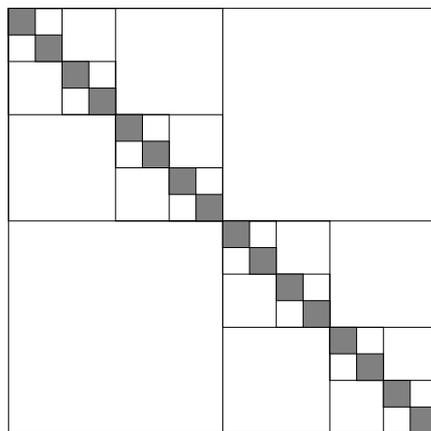


Рисунок 1.1: Простейшее мозаичное разбиение (белым обозначены подматрицы малого ранга, серым – плотные подматрицы).

Задача поиска мозаичного разбиения в данной матрице A не представляет интереса, так как, без какой-либо информации о происхождении этой матрицы, эта задача может быть решена только перебором (как минимум, необходимо выделить все подматрицы малого ранга). Методы, предложенные в данной диссертационной работе предполагают, что мозаичное разбиение исходной матрицы уже дано. Учитывая, что плотная матрица A состоит из малоранговых подматриц, можем предположить, что матрица A может быть порождена дискретизацией интегрального или граничного интегрального уравнения или соответствует системе взаимодействующих друг с другом частиц.

Один из возможных способов построения мозаичного разбиения матрицы — по геометрической информации, известной из постановки исходной задачи. Основное предположение состоит в том, что если объекты, соответствующие выбранным строкам матрицы, и объекты, соответствующие выбранным столбцам матрицы, геометрически разделены, то подматрица на выбранных строках и столбцах обладает малым рангом. Пример применения геометрического критерия при построении мозаичного разбиения очень прост: кластер частиц геометрически сам с собой не разделён, делим его на 2 подкластера и проверяем критерий

геометрической разделённости к подкластерам. Процедура идентична описанной ранее с той лишь разницей, что пары узлов отмечаем в случае их геометрической разделённости. Такое построение мозаичного разбиения является классическим и применено, например, в мозаично-скелетном методе [57, 58].

Для иллюстрации рассмотрим следующие задачи:

Одномерная задача. На отрезке $[0, 1]$ равномерно расположены частицы, необходимо построить мозаичное разбиение матрицы взаимодействия частиц.

Двумерная задача. В квадрате $[0, 1]^2$ равномерно расположены частицы, необходимо построить мозаичное разбиение матрицы взаимодействия частиц.

В одномерной задаче для построения иерархических деревьев будем использовать деление частиц по середине отрезка, содержащего эти частицы. В двумерной задаче для построения иерархических деревьев будем использовать деление частиц на две половины по одному из измерений. Примерные мозаичные разбиения, получающиеся в ходе построения, приведены на Рисунке 1.2.

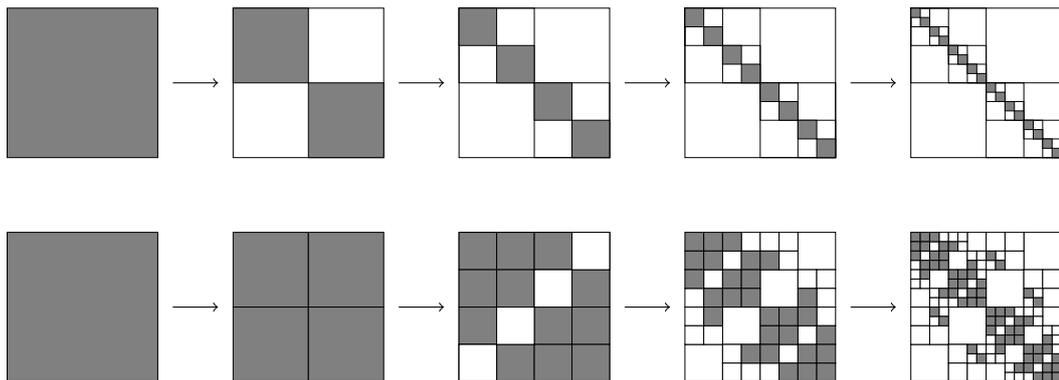


Рисунок 1.2: Примеры мозаичного разбиения для одномерной (сверху) и двумерной (снизу) задач, описанных ранее. Белым обозначены геометрически разделённые подматрицы, серым – близкие.

В соответствии с геометрической разделённостью, пары узлов мозаичного разбиения и подматрицы, отмеченные меткой «F», будем называть *дальними*, а пары узлов и подматрицы, отмеченные меткой «C», будем называть *близкими*.

1.5. \mathcal{H} -матрицы

Так как все элементы \mathcal{H} -матрицы принадлежат только одной подматрице с меткой «F» или «C», то в рамках ранее введённых обозначений имеет место следу-

ющая запись:

$$A = \sum_{(t,s) \in \mathbf{F}(A), t \in \mathcal{T}_I, s \in \mathcal{T}_J} \chi(t)A\chi(s) + \sum_{(t,s) \in \mathbf{C}(A), t \in \mathcal{T}_I, s \in \mathcal{T}_J} \chi(t)A\chi(s).$$

По предположению о мозаичном разбиении имеем:

$$\text{rank}(\chi(t)A\chi(s)) \leq \min(|\text{ind}(t)|, |\text{ind}(s)|),$$

где $|\text{ind}(t)|$ и $|\text{ind}(s)|$ – мощности множеств $\text{ind}(t)$ и $\text{ind}(s)$ (количество строк узла t и количество столбцов узла s). Из этого следует, что все дальние подматрицы можно заменить их малоранговыми приближениями, от чего уменьшится как память, необходимая для хранения самой матрицы в \mathcal{H} -формате, так и количество операций, необходимых для вычисления матрично-векторных произведений. Пусть для каждой дальней подматрицы известно её разложение:

$$\chi(t)A\chi(s) = \chi(t)U(t, s)V(t, s)\chi(s),$$

где $U(t, s)$ и $V(t, s)$ – факторы малорангового разложения. Очевидно, что все $U(t, s)$ независимы друг от друга для всевозможных пар (t, s) , это же самое касается и матриц $V(t, s)$. В предположении, что ранги всех дальних подматриц равны r , получаем требование по памяти $O(Nr \log N)$. Этот формат хранения дальних подматриц был предложен в работе [58] по мозаично-скелетному методу, где малоранговое разложение каждой дальней подматрицы строится при помощи крестовой аппроксимации.

В случае построения скелетного разложения дальней подматрицы, количество необходимой памяти сократится до $O(Nr)$, так как вместо матриц размера $|\text{ind}(t)| \times r$ будем хранить матрицы размера $r \times r$. Это улучшение по памяти, однако, увеличивает трудоёмкость матрично-векторных произведений за счёт того, что при каждой матрично-векторной операции необходимо вычислять подматрицы, соответствующие базисным строкам и базисным столбцам скелетного разложения. Такой формат хранения дальних подматриц был предложен в работе [57].

Разумное предположение на ближние подматрицы состоит в следующем: память, необходимая для хранения ближних подматриц, не превосходит память, необходимую для хранения дальних подматриц в малоранговых форматах. В рамках этого предположения, оценки памяти, необходимой для хранения всей \mathcal{H} -матрицы, как и оценки сложности матрично-векторных произведений, зависят только от оценок на дальние подматрицы.

Умножение одной дальней подматрицы на вектор требует $O(r|\text{ind}(t)|)$ операций, независимо от формата малорангового разложения этой подматрицы. Это приводит к тому, что общая сложность умножения \mathcal{H} -матрицы на вектор составляет $O(Nr \log N)$ операций.

1.6. \mathcal{H}^2 -матрицы

В мозаичном разбиении можно выделить еще одну структуру, *блочные строки* и *блочные столбцы*. Для определения блочных строк и блочных столбцов дополнительно введём следующие понятия:

$\mathcal{F}(t)$ – множество всех узлов, дальних с узлом t ,

$\mathcal{F}_p(t)$ – (\mathcal{F} of parents) множество узлов, дальних с одним из узлов из множества $\mathbf{pp}(t)$,

$\mathcal{F}_f(t)$ – (full \mathcal{F}) множество узлов, дальних с самим узлом или с одним из узлов из множества $\mathbf{pp}(t)$.

Параметры $\mathcal{F}(t)$, $\mathcal{F}_p(t)$ и $\mathcal{F}_f(t)$ так же называются *своя дальняя зона*, *родительская дальняя зона* и *полная дальняя зона* узла t соответственно. Блочными строками $\mathbf{R}(t)$, $\mathbf{Rp}(t)$ и $\mathbf{Rf}(t)$ узла $t \in \mathcal{T}_{I(A)}$ будем называть следующие матрицы:

$$\mathbf{R}(t) = \chi(t)A \sum_{s \in \mathcal{F}(t)} \chi(s),$$

$$\mathbf{Rp}(t) = \chi(t)A \sum_{s \in \mathcal{F}_p(t)} \chi(s),$$

$$\mathbf{Rf}(t) = \chi(t)A \sum_{s \in \mathcal{F}_f(t)} \chi(s) = \mathbf{R}(t) + \mathbf{Rp}(t).$$

Таким образом, \mathbf{R} есть матрица A , в которой все подматрицы, кроме дальних для узла t , занулили. $\mathbf{Rp}(t)$ есть матрица A , в которой все подматрицы, кроме дальних для отцов узла t , занулили. $\mathbf{Rf}(t)$ есть матрица A , в которой все подматрицы, кроме дальних для узла t или отцов узла t , занулили. Матрицы $\mathbf{R}(t)$, $\mathbf{Rp}(t)$ и $\mathbf{Rf}(t)$ называются *своя дальняя матрица*, *родительская дальняя матрица* и *полная дальняя матрица* узла t соответственно. Аналогичным образом вводятся блочные столбцы $\mathbf{C}(t)$, $\mathbf{Cp}(t)$ и $\mathbf{Cf}(t)$, называемые *своя дальняя матрица*, *родительская дальняя*

матрица и полная дальняя матрица соответственно, для узла $t \in \mathcal{T}_J$:

$$\mathbf{C}(t) = \sum_{s \in \mathcal{F}(t)} \chi(s) A_{\chi}(t),$$

$$\mathbf{C}_p(t) = \sum_{s \in \mathcal{F}_p(t)} \chi(s) A_{\chi}(t),$$

$$\mathbf{C}_f(t) = \sum_{s \in \mathcal{F}_f(t)} \chi(s) A_{\chi}(t) = \mathbf{C}(t) + \mathbf{C}_p(t).$$

Примеры блочных строк приведены на Рисунке 1.3.

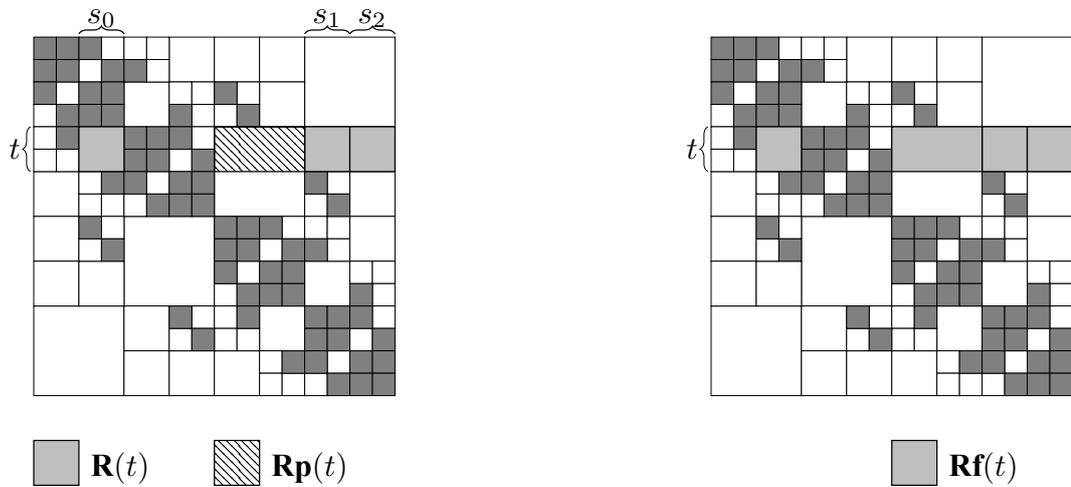


Рисунок 1.3: Различные блочные строки узла t . $\mathbf{R}(t)$, $\mathbf{R}_p(t)$, $\mathbf{R}_f(t)$.

Такое выделение подматриц имеет смысл только в условиях следующего предположения: ранг любой полной дальней подматрицы ограничен сверху некоторой константой, много меньшей размеров матрицы A . Дальнейшие рассуждения проводятся в рамках данного предположения.

Отметим иерархическую вложенность блочных строк и блочных столбцов:

$$\mathbf{R}_f(t) = \sum_{s \in \mathbf{s}(t)} \mathbf{R}_p(s),$$

$$\mathbf{C}_f(t) = \sum_{s \in \mathbf{s}(t)} \mathbf{C}_p(s).$$

Из этого факта следует, что факторы малоранговых разложений блочных строк и блочных столбцов имеют иерархические зависимости, то есть могут быть вычислены в нужный момент и их не нужно хранить. Докажем этот факт. Пусть даны

матрицы A_0 , A_1 и A_2 , размеров $n_0 \times m$, $n_1 \times m$ и $n_2 \times m$ соответственно, при этом матрица A_0 задана следующим образом:

$$A_0 = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}.$$

Предположим, что все эти матрицы имеют точные ранги:

$$A_0 = U_0 V_0, \text{rank}(A_0) = r_0, U_0 \in \mathbb{C}^{n_0 \times r_0}, V_0 \in \mathbb{C}^{r_0 \times m},$$

$$A_1 = U_1 V_1, \text{rank}(A_1) = r_1, U_1 \in \mathbb{C}^{n_1 \times r_1}, V_1 \in \mathbb{C}^{r_1 \times m},$$

$$A_2 = U_2 V_2, \text{rank}(A_2) = r_2, U_2 \in \mathbb{C}^{n_2 \times r_2}, V_2 \in \mathbb{C}^{r_2 \times m}.$$

Из этого следует, что строки матриц V_1 и V_2 лежат в пространстве строк матрицы V_0 :

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = M_0 V_0.$$

Откуда получаем иерархическую зависимость матриц U_0 , U_1 и U_2 :

$$U_0 = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} M_0.$$

Матрица M , при этом, называется *матрицей перехода*. Определим матрицы B_0 , B_1 и B_2 , размеров $M \times N_0$, $M \times N_1$ и $M \times N_2$ соответственно, по аналогии с матрицами A_0 , A_1 и A_2 :

$$B_0 = \begin{bmatrix} B_1 & B_2 \end{bmatrix}.$$

Предположим, что все эти матрицы имеют точные ранги:

$$B_0 = X_0 Y_0, \text{rank}(B_0) = r r_0, X_0 \in \mathbb{C}^{N_0 \times r r_0}, Y_0 \in \mathbb{C}^{r r_0 \times M},$$

$$B_1 = X_1 Y_1, \text{rank}(B_1) = r r_1, X_1 \in \mathbb{C}^{N_1 \times r r_1}, Y_1 \in \mathbb{C}^{r r_1 \times M},$$

$$B_2 = X_2 Y_2, \text{rank}(B_2) = r r_2, X_2 \in \mathbb{C}^{N_2 \times r r_2}, Y_2 \in \mathbb{C}^{r r_2 \times M}.$$

Тогда

$$Y_0 = \tilde{M}_0 \begin{bmatrix} Y_1 & 0 \\ 0 & Y_2 \end{bmatrix}.$$

Пусть матрицы A_0 и B_0 пересекаются по подматрице C , тогда существуют такие \tilde{V}_0 и \tilde{X}_0 , подматрицы матриц V_0 и X_0 соответственно, что:

$$C = U_0 \tilde{V}_0 = \tilde{X}_0 Y_0.$$

Это означает, что существует такая матрица \hat{C} , что:

$$C = U_0 \hat{C} Y_0 = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} M_0 \hat{C} \tilde{M}_0 \begin{bmatrix} Y_1 & 0 \\ 0 & Y_2 \end{bmatrix}.$$

Выберем любую дальнюю пару (t, s) , $\mathbf{s}(t) = (t_1, t_2)$, $\mathbf{s}(s) = (s_1, s_2)$, тогда соответствующая дальняя подматрица есть пересечение полных дальних матриц узлов t и s . Значит, существует такая матрица $\hat{A}(t, s)$, что:

$$A(t, s) = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} M(t) \hat{A}(t, s) M(s) \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix},$$

где U_1, U_2, V_1 и V_2 – факторы малоранговых разложений родительских дальних матриц узлов t_1, t_2, s_1 и s_2 соответственно, а $M(t)$ и $M(s)$ – матрицы переходов узлов t и s соответственно. Учитывая, что U_1, U_2, V_1 и V_2 могут быть вычислены только при помощи матриц перехода иерархически (если узел не имеет узлов-сыновей, то в качестве матрицы перехода можно задать фактор малорангового разложения), введём следующие обозначения:

$P(t)$ – глобальная матрица перехода узла t , то есть если $t \in \mathcal{T}_I$, то

$$P(t) = \begin{bmatrix} P(t_1) & 0 \\ 0 & P(t_2) \end{bmatrix} M(t),$$

иначе

$$P(t) = M(t) \begin{bmatrix} P(t_1) & 0 \\ 0 & P(t_2) \end{bmatrix}.$$

В случае, если множество st пусто, то принимаем $P(t) = M(t)$. Тогда любую дальнюю подматрицу можно записать в следующем виде:

$$A(t, s) = P(t) \hat{A}(t, s) P(s). \quad (1.1)$$

Матрица $\hat{A}(t, s)$ называется *матрицей взаимодействия* узлов t и s .

Не умаляя общности, можем расширить матрицу $P(t)$ нулевыми столбцами, а матрицу $P(s)$ нулевыми строками так, что:

$$\chi(t) A \chi(s) = P(t) \hat{A}(t, s) P(s). \quad (1.2)$$

При этом немного поменяется иерархическая зависимость матриц $P(t)$:

$$P(t) = \begin{bmatrix} P(t_1) & P(t_2) \end{bmatrix} M(t), t \in \mathcal{T}_I,$$

$$P(t) = M(t) \begin{bmatrix} P(t_1) \\ P(t_2) \end{bmatrix}, t \in \mathcal{T}_J.$$

1.6.1. Матрично-векторные операции в \mathcal{H}^2 -формате

Предположим, что все матрицы перехода и взаимодействия уже известны, необходимо вычислить матрично-векторное произведение $Ax = f$. Так как вся матрица A есть сумма дальних и ближних подматриц, то умножение на вектор можно записать следующим способом:

$$Ax = \sum_{(t,s) \in \mathbf{F}(A), t \in \mathcal{T}_I, s \in \mathcal{T}_J} P(t) \hat{A}(t,s) P(s)x + \sum_{(t,s) \in \mathbf{C}(A), t \in \mathcal{T}_I, s \in \mathcal{T}_J} \chi(t) A \chi(s)x.$$

Покажем как можно вычислять первое слагаемое за наименьшее время. Данный процесс можно разделить на 3 этапа. Первый этап состоит в вычислении всевозможных $a(s) = P(s)x$. Так как $P(s)$ – иерархически вычисляемая, то и $P(s)x$ тоже можно вычислять иерархически. Выглядит это следующим образом:

$$a(s) = P(s)x = M(s) \begin{bmatrix} P(s_1)x \\ P(s_2)x \end{bmatrix} = M(s) \begin{bmatrix} a(s_1) \\ a(s_2) \end{bmatrix}.$$

Когда вычислены все $a(s)$ можно вычислять следующие суммы:

$$\forall t \in \mathcal{T}_I : b(t) = \sum_{s \in \mathcal{F}(t)} \hat{A}(t,s) P(s)x.$$

Последний этап вычисления первого слагаемого использует вспомогательные вектора $b(t)$, вычисленные в ходе 2 этапа:

$$\sum_{(t,s) \in \mathbf{F}(A), t \in \mathcal{T}_I, s \in \mathcal{T}_J} P(t) \hat{A}(t,s) P(s)x = \sum_{t \in \mathcal{T}_I} P(t)b(t).$$

Так как $P(t)$ имеют такие же иерархические зависимости, что и $P(s)$, то несложно вывести следующую закономерность:

$$P(t)b(t) = \begin{bmatrix} P(t_1) & P(t_2) \end{bmatrix} M(t)b(t).$$

Разобьём матрицу $M(t)$ на две подматрицы $M_1(t)$ и $M_2(t)$ следующим образом:

$$M(t) = \begin{bmatrix} M_1(t) \\ M_2(t) \end{bmatrix}.$$

Тогда получаем

$$P(t)b(t) = P(t_1)M_1(t)b(t) + P(t_2)M_2(t)b(t).$$

Введём дополнительные вектора $c(t)$, проинициализировав их векторами $b(t)$. Из последнего уравнения следует, что вектор $c(t)$ можно перекинуть в вектора $c(t_1)$ и $c(t_2)$:

$$c(t_1) = c(t_1) + M_1(t)c(t), \quad c(t_1) = c(t_2) + M_2(t)c(t), \quad c(t) = 0.$$

Это преобразование можно записать в матричном виде:

$$\begin{bmatrix} c(t_1) \\ c(t_2) \end{bmatrix} = \begin{bmatrix} c(t_1) \\ c(t_2) \end{bmatrix} + M(t)c(t), \quad c(t) = 0.$$

При этом неизменной останется следующая сумма:

$$\sum_{t \in \mathcal{T}_I} P(t)c(t).$$

Учитывая, что таким образом можно занулить все $c(t)$, кроме тех, для которых $|\mathbf{s}(t)| = 0$, получаем:

$$\sum_{(t,s) \in \mathbf{F}(A), t \in \mathcal{T}_I, s \in \mathcal{T}_J} P(t)\hat{A}(t,s)P(s)x = \sum_{t \in \mathcal{T}_I, |\mathbf{s}(t)|=0} P(t)c(t).$$

Важно заметить, что при $|\mathbf{s}(t)| = 0$, $P(t) = M(t)$ на ненулевых строках, поэтому будем считать это равенство верным (дополняем матрицу $M(t)$ нулевыми строками).

Построенный таким образом алгоритм, приведённый в книге [11], напоминает быстрые мультипольные методы: на первом этапе снизу вверх вычисляются мультипольные заряды для всех $s \in \mathcal{T}_J$, далее вычисляются мультипольные потенциалы для всех $t \in \mathcal{T}_I$, а в конце эти потенциалы распределяются на все тела иерархически сверху вниз. Из-за этого факта \mathcal{H}^2 -формат так же называют алгебраическим мультипольным методом.

Приведённый в данном разделе алгоритм формализован в Алгоритме 2. В качестве сокращения обозначений, в Алгоритме 2 использованы следующие конструкции:

$$\mathbf{s}(t) = t_1, \dots, t_k,$$
$$\left[a(\mathbf{s}(t)) \right] = \begin{bmatrix} a(t_1) \\ \vdots \\ a(t_k) \end{bmatrix},$$
$$\left[b(\mathbf{s}(t)) \right] = \begin{bmatrix} b(t_1) \\ \vdots \\ b(t_k) \end{bmatrix}.$$

Алгоритм 2 Быстрое умножение дальних подматриц \mathcal{H}^2 -матрицы на вектор [11]

Дано: матрицы перехода $M(t), t \in \mathcal{T}_I \cup \mathcal{T}_J$, взаимодействия $\hat{A}(t, s), (t, s) \in \mathbf{F}(A)$, вектор x

Вычислить: $y = \sum_{(t,s) \in \mathbf{F}(A)} \chi(t)A\chi(s)x$

```

1: for  $t \in \mathcal{T}_J$  по дереву снизу вверх do
2:   если  $\left[ a(\mathbf{s}(t)) \right]$  еще не известно, то нельзя вычислять  $a(t)$ 
3:   if  $|\mathbf{s}(t)| = 0$  {у узла  $t$  нет узлов-детей} then
4:      $a(t) = M(t)x$ 
5:   else
6:      $a(t) = M(t) \left[ a(\mathbf{s}(t)) \right]$ 
7:   end if
8: end for
9: for  $t \in \mathcal{T}_I$  do
10:   $b(t) = 0$ 
11:  for  $s \in \mathcal{F}(t)$  {для каждой дальней пары  $(t, s)$ } do
12:     $b(t)+ = \hat{A}(t, s)a(s)$ 
13:  end for
14: end for
15:  $y = 0$  {обнуляем вектор, аккумулирующий решение}
16: for  $t \in \mathcal{T}_I$  по дереву сверху вниз do
17:  вместо  $c(t)$  будем использовать  $b(t)$ 
18:  if  $|\mathbf{s}(t)| = 0$  {у узла  $t$  нет узлов-детей} then
19:     $y+ = M(t)b(t)$ 
20:  else
21:     $\left[ b(\mathbf{s}(t)) \right] + = M(t)b(t)$ 
22:  end if
23: end for
24: return  $y$ 

```

Оценим количество операций, необходимых для одного матрично-векторного произведения. Очевидно, оно эквивалентно количеству элементов всех матриц перехода $M(t)$ и всех матриц взаимодействий \hat{A} . Для простоты оценки полагаем, что ранги всех блочных столбцов и строк порядка r , а коли-

чество строк нелистьевых узлов t порядка $2r$. Получаем $O(Mr^2)$ операций, где M – количество узлов иерархических кластерных деревьев строк и столбцов. Очевидно, $M = O(N/r)$, следовательно, общее количество операций на одно матрично-векторное произведение составляет $O(Nr)$ операций.

1.7. Выводы по главе

Данная глава является вводной и содержит описание ранее известных структур, оценок погрешности и алгоритмов. В частности, приведена структура \mathcal{H}^2 -матриц, являющаяся ключевой в следующих главах диссертации. Любые мало-параметрические представления \mathcal{H}^2 -матриц требуют малоранговых приближений специальных блочных строк и блочных столбцов, для построения которых может быть использована представленная в данной главе скелетная аппроксимация. Точность скелетной аппроксимации напрямую зависит от определителя матрицы, расположенной на пересечении базисных строк и столбцов. Алгоритм **maxvol** позволяет быстро находить такие строки и столбцы, что матрица на их пересечении будет являться *доминантной* и, поэтому, будет обладать достаточно «хорошим» определителем.

Глава 2

Прямоугольная скелетная аппроксимация

2.1. Объём прямоугольных подматриц

Обобщим понятие объёма на случай прямоугольных подматриц. Пусть $A \in \mathbb{C}^{K \times r}$, $K \geq r$, определим K -мерное множество $G(A)_p$ следующим образом:

$$G(A)_p = \{v \in \mathbb{R}^r : \exists c \in \mathbb{R}^K, \|c\|_p \leq 1 \mid v = cA\}.$$

Тогда p -объёмом $\text{vol}_p(A)$ матрицы A будем называть объём множества $G(A)_p$. Заметим, что определение 1-объёма совпадает с ранее введённым определением объёма квадратной матрицы ($\text{vol}_1(A) = |\det(A)|$). Рассмотрим сингулярное разложение матрицы $A = USV$, $U \in \mathbb{C}^{K \times K}$, $S \in \mathbb{R}^{K \times r}$, $V \in \mathbb{C}^{r \times r}$. Очевидно, $G(A)_2 = G(S)_2$, а $G(S)_2$ представляет собой r -мерный эллипс в K -мерном пространстве, поэтому 2-объём матрицы A есть произведение объёма единичной сферы в r -мерном пространстве и всех сингулярных чисел матрицы A . В свою очередь, произведение сингулярных чисел легко вычислять по следующей формуле:

$$\prod_{i=1}^r \sigma_i = \sqrt{\det A^* A}.$$

Поэтому, в качестве 2-объёма можно ввести следующий функционал:

$$\text{vol}_2(A) = \sqrt{\det(A^* A)}.$$

Из алгоритма **maxvol**, описанного в разделе 1.3, следует, что C -норма матрицы коэффициентов не превосходит 1. Из этого ограничения получаем оценки норм строк матрицы C :

$$\|C_i\|_1 \leq r, \|C_i\|_2 \leq \sqrt{r}.$$

Эти оценки являются неулучшаемыми, например, для матриц Адамара.

Возникает вопрос: можно ли увеличить количество строк матрицы \hat{A} так, что сохранится равенство

$$A = C\hat{A},$$

а норма матрицы C станет меньше. Очевидно, C -норма этой матрицы никогда не станет меньше 1, однако, это не касается других норм.

2.1.1. Оценка l_2 нормы строк матрицы коэффициентов

Введём понятие *прямоугольных доминантных подматриц* по аналогии с квадратными доминантными подматрицами. Пусть дана матрица $A \in \mathbb{C}^{N \times r}$, $N \geq r$, $\text{rank} A = r$, из которой выбрана подматрица $\hat{A} \in \mathbb{C}^{K \times r}$, $K \geq r$. Обозначим за $\mathcal{M}(\hat{A})$ множество всех подматриц размера $K \times r$ матрицы A , отличающихся от \hat{A} ровно в 1 строке. Прямоугольной доминантной подматрицей будем называть такую подматрицу \hat{A} , что её 2-объём не меньше 2-объёма любой подматрицы из множества $\mathcal{M}(\hat{A})$.

Напомним теорему Бине-Коши [21]:

Теорема 2.1.1 (Теорема Бине-Коши). *Произведение двух матриц A и B даёт квадратную матрицу порядка t , если матрица A имеет n столбцов и t строк, а матрица B имеет n строк и t столбцов. Миноры матриц A и B одинакового порядка, равного наименьшему из чисел n и t , называются соответствующими друг другу, если они стоят в столбцах (матрицы A) и строках (матрицы B) с одинаковыми номерами. Тогда определитель матрицы AB равен нулю, если $n < t$, и равен сумме попарных произведений соответствующих друг другу миноров порядка t , если $n \geq t$.*

$$\det(AB) = \sum_i \det(A_i) \det(B_i),$$

где $\det(A_i)$ и $\det(B_i)$ – соответствующие миноры.

Из теоремы Бине-Коши следует лемма:

Лемма 2.1.2 (Результат автора). *В рамках условия теоремы Бине-Коши, будем называть подматрицы соответствующими, если они стоят в столбцах (матрицы A) и строках (матрицы B) с одинаковыми номерами. Тогда, если $n > m$, то определитель матрицы AB равен одной $(n - m)$ -ной суммы определителей попарных произведений соответствующих подматриц размеров $(n - 1) \times m$ (подматрицы матрицы B) и $m \times (n - 1)$ (подматрицы матрицы A):*

$$\det(AB) = \frac{1}{n - m} \sum_{i=1}^n \det(A_i B_i),$$

где A_i получается из A вырезанием i -ого столбца, а B_i получается из B вырезанием i -ой строки.

Доказательство. Обозначим A_i – подматрица матрицы A без i -ого столбца, B_i – подматрица матрицы B без i -ой строки. По теореме Бине-Коши, определитель произведения $A_i B_i$ есть сумма произведений всех соответствующих миноров матриц A_i и B_i . То есть определитель произведения $A_i B_i$ есть сумма произведений всех соответствующих миноров матриц A и B , не содержащих i -ый столбец (для миноров матрицы A) и i -ую строку (для миноров матрицы B). Таким образом, сумма определителей произведений $A_i B_i$ есть сумма произведений соответствующих миноров матриц A и B с точностью до множителя. Искомый множитель легко вычисляется из следующего утверждения: каждое произведение соответствующих миноров встречается в сумме определителей произведений $A_i B_i$ ровно $(n - m)$ раз. Так как каждый минор основан на m столбцах или строках, то любой минор отсутствует ровно в m из всех произведений $A_i B_i$ и содержится в $(n - m)$ оставшихся произведениях. Таким образом, сумма определителей произведений $A_i B_i$ содержит каждой произведение соответствующих миноров ровно $(n - m)$ раз, откуда получаем следующую формулу:

$$\det(AB) = \frac{1}{n - m} \sum_i \det(A_i B_i). \quad \square$$

Пусть матрица W есть расширение подматрицы \hat{A} при помощи строки a :

$$W = \begin{bmatrix} \hat{A} \\ a \end{bmatrix}.$$

Тогда из леммы выше следует, что

$$\det(W^*W) = \frac{1}{K+1-r} \sum_i \det(W_i^*W_i),$$

где W_i есть матрица W без i -ой строки.

$$\det(W^*W) = \frac{1}{K+1-r} \left(\det(\hat{A}^*\hat{A}) + \sum_{X \in \mathcal{M}(\hat{A})} \det(X^*X) \right),$$

$$\det(W^*W) \leq \frac{K+1}{K+1-r} \det(\hat{A}^*\hat{A}).$$

С другой стороны,

$$\det(W^*W) = \det(\hat{A}^*\hat{A} + a^*a) = \det(\hat{A}^*\hat{A}) \det(I + (\hat{A}^*\hat{A})^{-1}a^*a),$$

$$\det(W^*W) = \det(\hat{A}^*\hat{A})(1 + a(\hat{A}^*\hat{A})^{-1}a^*).$$

Пусть a – строка матрицы A , тогда ей соответствует строка $c = a\hat{A}^\dagger$ матрицы C . Очевидно, столбцы матрицы \hat{A} линейно-независимы, из чего следует, что $\hat{A}^\dagger = (\hat{A}^*\hat{A})^{-1}\hat{A}^*$. Учитывая $a(\hat{A}^*\hat{A})^{-1}a^* = cc^*$, получаем:

$$\det(W^*W) = \det(\hat{A}^*\hat{A})(1 + \|c\|_2^2).$$

Составляя 2 оценки на определитель матрицы W^*W вместе, получаем:

$$\|c\|_2 \leq \sqrt{\frac{r}{K+1-r}}. \quad (2.1)$$

Отсюда получаем **принцип максимального 2-объёма подматрицы**: если подматрица обладает максимальным 2-объёмом среди всех подматриц размера $K \times r$ в матрице размера $N \times r$, то существует такая матрица коэффициентов, что длина любой её строки ограничена сверху выражением $\sqrt{\frac{r}{K+1-r}}$. Отметим, что этот принцип выполняется не только для подматриц максимального 2-объёма, но и для прямоугольных доминантных подматриц. Подобная оценка ранее была получена в работе [19], в которой кроме самой оценки приведены примеры задач поиска «хорошей» подматрицы.

2.1.2. Алгоритм максимизации 2-объёма подматрицы

Построим «жадный» алгоритм максимизации 2-нормы подматрицы путём увеличения количества строк. Сначала выбираем невырожденную $r \times r$ подматрицу \hat{A} , а потом на каждом шаге выбираем новую строку A_j матрицы A так, чтобы 2-объём расширенной матрицы

$$\hat{A} := \begin{bmatrix} \hat{A} \\ A_j \end{bmatrix}$$

был максимален. При этом, матрица коэффициентов с минимальной второй нормой строк вычисляется как $C = A\hat{A}^\dagger$, где \hat{A}^\dagger – псевдообратная к матрице \hat{A} .

Рассмотрим процесс добавления строки к матрице $H = \hat{A}$:

$$W = \begin{bmatrix} H \\ a \end{bmatrix}.$$

Основное уравнение тогда можно переписать в виде $A = C_H H = C_W W$. Предполагаем, что столбцы матрицы H линейно-независимы, тогда столбцы матрицы W также линейно-независимы. Так как минимизируем 2-объём матриц C_H и C_W , то $C_H = AH^\dagger$ и $C_W = AW^\dagger$. Из линейной независимости столбцов H и W имеем $W^\dagger = (W^*W)^{-1}W^*$ и $H^\dagger = (H^*H)^{-1}H^*$.

$$W^*W = H^*H + a^*a.$$

Выражаем обратную к матрице W^*W по формуле Шермана-Вудберри-Моррисона [52, 60]:

$$(W^*W)^{-1} = (H^*H)^{-1}(H^*H - (1 + a(H^*H)^{-1}a^*)^{-1}a^*a)(H^*H)^{-1}.$$

Строка a содержится в A , ей соответствует строка c из C_H , то есть $a = cH$ и $c = aH^\dagger$.

$$(H^*H)^{-1}a^* = (H^*H)^{-1}H^*c^* = H^\dagger c^* \Rightarrow a(H^*H)^{-1}a^* = cc^*,$$

$$(W^*W)^{-1} = \left(I - \frac{(H^*H)^{-1}a^*a}{1 + cc^*} \right) (H^*H)^{-1},$$

$$(W^*W)^{-1} = \left(I - \frac{H^\dagger c^* c H}{1 + cc^*} \right) (H^*H)^{-1},$$

$$\begin{aligned}
A(W^*W)^{-1} &= \left(A - \frac{C_H c^* c H}{1 + c c^*} \right) (H^* H)^{-1}, \\
C_W &= A W^\dagger = A(W^*W)^{-1} W^*, \\
C_W &= \left(A - \frac{C_H c^* c H}{1 + c c^*} \right) (H^* H)^{-1} \begin{bmatrix} H^* & a^* \end{bmatrix}, \\
C_W &= \left(A - \frac{C_H c^* c H}{1 + c c^*} \right) (H^* H)^{-1} H^* \begin{bmatrix} I & c^* \end{bmatrix}, \\
C_W &= \left(A H^\dagger - \frac{C_H c^* a H^\dagger}{1 + c c^*} \right) \begin{bmatrix} I & c^* \end{bmatrix}, \\
A H^\dagger - \frac{C_H c^* a H^\dagger}{1 + c c^*} &= C_H - \frac{C_H c^* c}{1 + c c^*}, \\
C_W &= \begin{bmatrix} C_H - \frac{C_H c^* c}{1 + c c^*} & C_H c^* - \frac{C_H c^* c c^*}{1 + c c^*} \end{bmatrix}, \\
C_W &= \begin{bmatrix} C_H - \frac{C_H c^* c}{1 + c c^*} & \frac{C_H c^*}{1 + c c^*} \end{bmatrix}.
\end{aligned}$$

Таким образом, если матрица C_H – матрицы размера $N \times K$, то C_W может быть вычислена на основе C_H за $O(NK)$ операций.

В случае, если W отличается от H всего в одной строке, можно определить вектора a , c и e такие, что:

$$W = H + e^* a, c = a H^\dagger.$$

При этом вектор e есть некоторая строка единичной матрицы. Тогда при помощи преобразований, аналогичных применённым ранее, получаем зависимость C_W и C_H :

$$C_W = C_H - \frac{C_H c^* c}{1 + c c^*} + \frac{C_H c^* e}{1 + c c^*}.$$

Таким образом, при расширении матрицы \hat{A} i -ой строкой матрицы A , матрица коэффициентов C пересчитывается через свою i -ую строку следующим образом:

$$C := \left[C - \frac{C C_i^* C_i}{1 + C_i C_i^*} \quad \frac{C C_i^*}{1 + C_i C_i^*} \right].$$

Тогда для любой строки матрицы C выполняется следующее равенство:

$$\forall j : \|C_j\|_2^2 := \|C_j\|_2^2 - \frac{|C_j C_i^*|^2}{1 + C_i C_i^*}.$$

При замене j -ой строки матрицы \hat{A} на i -ую строку матрицы A , матрица коэффициентов C пересчитывается следующим образом:

$$C := C - \frac{CC_i^*C_i}{1 + C_iC_i^*} + \frac{CC_i^*e_j}{1 + C_iC_i^*},$$

где e_j – j -ая строка единичной матрицы размера $K \times K$ (K – количество строк матрицы \hat{A}).

Покажем, какую именно строку a необходимо выбирать для максимизации 2-объёма матрицы W . Воспользуемся тождеством для определителей:

$$\det(W^*W) = \det(H^*H + a^*a) = \det(H^*H) \det(I + (H^*H)^{-1}a^*a),$$

$$\det(W^*W) = \det(H^*H)(1 + a(H^*H)^{-1}a^*).$$

В формулах выше было показано, что $a(H^*H)^{-1}a^* = cc^*$, то есть

$$\det(W^*W) = \det(H^*H)(1 + \|c\|_2^2).$$

Отсюда следует, что строка a , при помощи которой мы расширяем матрицу H до матрицы W , должна соответствовать строке матрицы C_H с максимальной l_2 -нормой.

Критерий останова алгоритма расширения матрицы \hat{A} может быть, например, следующий: l_2 -норма каждой строки матрицы C не превосходит заданный параметр τ . Итоговый алгоритм является результатом работы автора и формализован в Алгоритме 3.

На практике алгоритм оказался эффективным. Для случайных матриц разного размера ограничение нормы

$$\forall i : \|C_i\|_2 \leq 2,$$

достигается на подматрицах размера $K \times r$ уже при $K \approx 1.2r$ (что соответствует оценке из неравенства (2.1)). При этом ограничение нормы

$$\forall i : \|C_i\|_2 \leq 1,$$

достигается на подматрицах размера $K \times r$ уже при $K \approx 2r$ (что также соответствует оценке из неравенства (2.1)).

Алгоритм 3 (Результат автора) «Жадная» максимизация 2-объёма подматрицы \hat{A}

Дано: Полноранговая матрица $A \in \mathbb{C}^{N \times r}$, $N > r$, параметр τ критерия останова

Вычислить: Подматрицу \hat{A} и матрицу коэффициентов C такие, что $A =$

$$C\hat{A}, \|C_i\|_2 \leq \tau$$

- 1: Найти невырожденную подматрицу \hat{A} в матрице A
 - 2: $C := A\hat{A}^{-1}, \forall i : L_i = \|C_i\|_2^2$
 - 3: $i = \operatorname{argmax}(L)$
 - 4: **while** $L_i > \tau^2$ **do**
 - 5: $\hat{A} := \begin{bmatrix} \hat{A} \\ A_i \end{bmatrix}$
 - 6: $C := \begin{bmatrix} C - \frac{CC_i^*C_i}{1+C_iC_i^*} & \frac{CC_i^*}{1+C_iC_i^*} \\ \frac{C_jC_i^*}{1+C_iC_i^*} & \end{bmatrix}$
 - 7: $\forall j : L_j := L_j - \frac{|C_jC_i^*|^2}{1+C_iC_i^*}$
 - 8: $i = \operatorname{argmax}(L)$
 - 9: **end while**
 - 10: **return** C, \hat{A}
-

2.2. Прямоугольная псевдоскелетная аппроксимация

В работах про скелетную и псевдоскелетную аппроксимации, упомянутых ранее, было показано, что точность аппроксимации зависит от норм матриц коэффициентов. Уменьшение нормы матрицы коэффициентов за счет выбора дополнительных базисных строк было показано в предыдущей секции данной главы. Выбор дополнительных базисных строк и столбцов приводит к тому, что подматрица на их пересечении оказывается прямоугольной. Скелетную аппроксимацию на основе прямоугольной подматрицы построить невозможно, из-за невозможности применения операции обращения к прямоугольной подматрице. Оценки скелетных аппроксимаций из работ [23, 24], приведённые ранее, доказаны в источниках только для выбора квадратных подматриц. Расширим эти оценки на случай выбор прямоугольных подматриц.

Теорема 2.2.1 (Результат автора, точность τ -псевдоскелетной аппроксимации). Пусть матрица $A \in \mathbb{C}^{N \times M}$, такая что $A = Z + F$, $\operatorname{rank} Z = r$, $\|F\|_2 \leq \varepsilon$, имеет

следующую блочную запись:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

Тогда существует τ -псевдоскелетный аппроксимант \tilde{A} , построенный на подматрице A_{11} , такой, что

$$\|A - \tilde{A}\|_2 \leq \varepsilon(4 + 2s + 3s^2 + 2(1 + s)\sqrt{1 + s^2}),$$

где s – максимальная из l_2 норм матриц $Z_{21}Z_{11}^\dagger$ и $Z_{11}^\dagger Z_{12}$.

Доказательство. Пусть $A_{11} \in \mathbb{C}^{n \times m}$. Напомним, что τ -псевдоскелетный аппроксимант тогда будет выглядеть следующим образом:

$$\tilde{A} = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} (A_{11})_\tau^\dagger \begin{bmatrix} A_{11} & A_{12} \end{bmatrix}. \quad (2.2)$$

Разобьём матрицу Z на части по аналогии с матрицей A :

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix}.$$

Пусть выбранная подматрица A_{11} обеспечивает существование таких матриц C_u и C_v , что

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = C_u \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} C_v. \quad (2.3)$$

Так как количество строк u_1 может быть больше количества столбцов u_1 , то матрица C_u определяется не единственным способом. Поиск коэффициентов C_u и C_v является отдельной задачей оптимизации в некоторой, например l_2 , норме. Если распишем матрицу C_u через подматрицы:

$$C_u = \begin{bmatrix} C_{u1} \\ C_{u2} \end{bmatrix},$$

то получим уравнение

$$u_1 = C_{u1} u_1.$$

Как уже было сказано ранее, подматрица C_{u1} определяется не единственным способом, т.е. очевидное на первый взгляд тождество

$$C_{u1} = I, \quad (2.4)$$

не всегда является верным. Однако, в дальнейших оценках кроме нормы самой матрицы C_{u1} важную роль будет играть норма матрицы $I - C_{u1}$. Очевидно,

$$\|C_{u1}\|_2 \geq 1,$$

поэтому замена C_{u1} на единичную не увеличит нормы матриц C_{u1} и $I - C_{u1}$. Далее будем считать, что тождество (2.4) является верным. Разобьём матрицу C_v на подматрицы по аналогии:

$$C_v = \begin{bmatrix} C_{v1} & C_{v2} \end{bmatrix}, \quad C_{v1} = I.$$

Получаем выражения для матрицы Z :

$$Z = C_u Z_{11} C_v,$$

$$C_{u1} Z_{11} C_{v1} = C_{u1} Z_{11} = Z_{11} C_{v1} = Z_{11}.$$

Учитывая $A = Z + F$, получаем:

$$A = C_u A_{11} C_v - C_u F_{11} C_v + F. \quad (2.5)$$

Распишем это равенство для подматриц $\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}$ и $\begin{bmatrix} A_{11} & A_{12} \end{bmatrix}$:

$$\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} = C_u A_{11} - C_u F_{11} + \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix} = C_u A_{11} + g_1, \quad (2.6)$$

$$g_1 = \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix} - C_u F_{11} = \begin{bmatrix} 0 \\ F_{21} - C_{u2} F_{11} \end{bmatrix}$$

$$\begin{bmatrix} A_{11} & A_{12} \end{bmatrix} = A_{11} C_v - F_{11} C_v + \begin{bmatrix} F_{11} & F_{12} \end{bmatrix} = A_{11} C_v + g_2. \quad (2.7)$$

$$g_2 = \begin{bmatrix} F_{11} & F_{12} \end{bmatrix} - F_{11} C_v = \begin{bmatrix} 0 & F_{12} - F_{11} C_{v2} \end{bmatrix}$$

Совмещая уравнения (2.2), (2.6) и (2.7), получаем:

$$\tilde{A} = C_u A_{11} (A_{11})_{\tau}^{\dagger} A_{11} C_v + g_1 (A_{11})_{\tau}^{\dagger} A_{11} C_v + C_u A_{11} (A_{11})_{\tau}^{\dagger} g_2 + g_1 (A_{11})_{\tau}^{\dagger} g_2. \quad (2.8)$$

Введём дополнительные обозначения:

$$h = A_{11} - (A_{11})_{\tau},$$

$$H_u = A_{11} (A_{11})_{\tau}^{\dagger}, H_v = (A_{11})_{\tau}^{\dagger} A_{11}.$$

Очевидно,

$$\left\| (A_{11})_{\tau}^{\dagger} \right\|_2 \leq \tau^{-1}, \|h\|_2 \leq \tau, \|H_u\|_2 \leq 1, \|H_v\|_2 \leq 1.$$

Перепишем уравнение (2.8):

$$\tilde{A} = C_u (A_{11})_{\tau} C_v + g_1 H_v C_v + C_u H_u g_2 + g_1 (A_{11})_{\tau}^{\dagger} g_2. \quad (2.9)$$

Оценим точность приближения матрицы A аппроксимантом \tilde{A} :

$$E = A - \tilde{A}. \quad (2.10)$$

Подставляя уравнения (2.5) и (2.9) в уравнение (2.10), получаем:

$$E = C_u h C_v - C_u F_{11} C_v + F - g_1 H_v C_v - C_u H_u g_2 - g_1 (A_{11})_{\tau}^{\dagger} g_2.$$

Тогда l_2 норму аппроксимации можно оценить следующим образом:

$$\|E\|_2 \leq \|C_u\|_2 \|C_v\|_2 (\tau + \|F_{11}\|_2) + \varepsilon + \|C_v\|_2 \|g_1\|_2 + \|C_u\|_2 \|g_2\|_2 + \frac{\|g_1\|_2 \|g_2\|_2}{\tau}. \quad (2.11)$$

Известно, что старшее сингулярное число любой подматрицы меньше старшего сингулярного числа самой матрицы, то есть имеют силу следующие утверждения:

$$\forall i, j : \|F_{ij}\|_2 \leq \|F\|_2.$$

Кроме этого, воспользуемся выражением

$$\|F\|_2^2 \leq \sum_{i,j} \|F_{ij}\|_2^2. \quad (2.12)$$

Ограничим нормы матриц C_{u2} и C_{v2} :

$$s = \max(\|C_{u2}\|_2, \|C_{v2}\|_2) = \max(\|Z_{21} Z_{11}^{\dagger}\|_2, \|Z_{11}^{\dagger} Z_{12}\|_2).$$

Из этого ограничения и неравенства (2.12) следуют оценки на нормы матриц C_u и C_v :

$$\|C_u\|_2 \leq \sqrt{1 + s^2}, \quad \|C_v\|_2 \leq \sqrt{1 + s^2}.$$

Тогда норму матрицы E можно записать в следующем виде:

$$\|E\|_2 \leq (1 + s^2)(\tau + \varepsilon) + \varepsilon + \sqrt{1 + s^2}(\|g_1\|_2 + \|g_2\|_2) + \frac{\|g_1\|_2 \|g_2\|_2}{\tau}.$$

Имеем очевидные оценки на нормы матриц g_1 и g_2 :

$$\|g_1\|_2 \leq \varepsilon(1 + s),$$

$$\|g_2\|_2 \leq \varepsilon(1 + s).$$

Собирая полученные оценки, получаем:

$$\|E\|_2 \leq (1 + s^2)\tau + (2 + s^2 + 2(1 + s)\sqrt{1 + s^2})\varepsilon + (1 + s)^2 \frac{\varepsilon^2}{\tau}.$$

Подставляя $\tau = \varepsilon$, получаем итоговое:

$$\|A - \tilde{A}\|_2 \leq (4 + 2s + 3s^2 + 2(1 + s)\sqrt{1 + s^2})\varepsilon. \quad \square$$

Пусть U – ортогональная матрица размера $n \times r$. Обозначим через $\mathcal{M}_k(U)$ все подматрицы размера $k \times r$. Введём следующее определение:

$$t(r, n, k) = \frac{1}{\min_U \max_{P \in \mathcal{M}_k(U)} \sigma_{\min}(P)}.$$

Это определение идентично определению, введённому в работе [24], и является расширением на случай прямоугольных подматриц. Очевидно, при $k \geq r$:

$$t(r, n, k) \leq t(r, n, r) = t(r, n).$$

Теорема 2.2.2 (Результат автора, точность τ -псевдоскелетной аппроксимации). *Для любой матрицы $A \in \mathbb{C}^{N \times M}$, такой что $A = Z + F$, $\text{rank} Z = r$, $\|F\|_2 \leq \varepsilon$, существует τ -псевдоскелетный аппроксимант \tilde{A} , построенный на n строках и m столбцах, такой, что*

$$\|A - \tilde{A}\|_2 \leq (5t(r, N, n)t(r, M, m) + 2t(r, N, n) + 2t(r, M, m) + 2)\varepsilon.$$

Доказательство. Напомним, что

$$A \in \mathbb{C}^{N \times M}, \quad A_{11} \in \mathbb{C}^{n \times m}.$$

Перепишем оценку $\|A - \tilde{A}\|_2$ в рамках расширенного определения $t(r, n, k)$. Для этого выпишем оценки каждого слагаемого из уравнения (2.11):

$$\|C_u\|_2 \leq t(r, N, n),$$

$$\|C_v\|_2 \leq t(r, M, m),$$

$$\|g_1\|_2 \leq (t(r, N, n) + 1)\varepsilon,$$

$$\|g_2\|_2 \leq (t(r, M, m) + 1)\varepsilon.$$

Принимая $\tau = \varepsilon$, получаем:

$$\|A - \tilde{A}\|_2 \leq (5t(r, N, n)t(r, M, m) + 2t(r, N, n) + 2t(r, M, m) + 2)\varepsilon. \quad \square$$

Заметим, что данная оценка совпадает с оценкой, упомянутой ранее и представленной в работе [24], с точностью до функции $t(r, n, k)$.

Оценка, полученная для τ -псевдоскелетных аппроксимаций, на первый взгляд, является достаточно хорошей. Однако, с практической точки зрения намного проще в уравнении (2.2) вместо $(A_{11})_\tau^\dagger$ использовать $(Z_{11})_\tau^\dagger$. Такая аппроксимация называется CGR-аппроксимацией с ядром $(Z_{11})_\tau^\dagger$.

Теорема 2.2.3 (Результат автора, точность CGR аппроксимации). *Для любой матрицы $A \in \mathbb{C}^{N \times M}$, такой что $A = Z + F$, $\text{rank} Z = r$, $\|F\|_2 \leq \varepsilon$, существует CGR аппроксимация \tilde{A} , построенная на n строках и m столбцах, такая, что*

$$\|A - \tilde{A}\|_2 \leq \left\{ 1 + \left[\sqrt{t(r, N, n)} + \sqrt{t(r, M, m)} \right]^2 \right\} \varepsilon.$$

Доказательство. Перепишем основные уравнения в рамках нового ядра:

$$A = C_u Z_{11} C_v + F,$$

$$\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} = C_u Z_{11} + \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix},$$

$$\begin{bmatrix} A_{11} & A_{12} \end{bmatrix} = Z_{11} C_v + \begin{bmatrix} F_{11} & F_{12} \end{bmatrix},$$

$$\begin{aligned} CGR &= C_u Z_{11} (Z_{11})_\tau^\dagger Z_{11} C_v + C_u Z_{11} (Z_{11})_\tau^\dagger \begin{bmatrix} F_{11} & F_{12} \end{bmatrix} + \\ &+ \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix} (Z_{11})_\tau^\dagger Z_{11} C_v + \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix} (Z_{11})_\tau^\dagger \begin{bmatrix} F_{11} & F_{12} \end{bmatrix}. \end{aligned} \quad (2.13)$$

Откуда сразу следует оценка погрешности:

$$\|A - CGR\|_2 \leq \tau \|C_u\|_2 \|C_v\|_2 + \varepsilon + \varepsilon \|C_u\|_2 + \varepsilon \|C_v\|_2 + \frac{\varepsilon^2}{\tau}.$$

Минимум этого уравнения достигается при

$$\tau = \varepsilon (\|C_u\|_2 \|C_v\|_2)^{-\frac{1}{2}}.$$

При этом значении параметра τ получается следующая оценка:

$$\|A - CGR\|_2 \leq \varepsilon \left(\left(\sqrt{\|C_u\|_2} + \sqrt{\|C_v\|_2} \right)^2 + 1 \right). \quad (2.14)$$

Заметим, что существует такой выбор подматрицы A_{11} , что

$$\|C_u\|_2 \leq t(r, N, n), \quad \|C_v\|_2 \leq t(r, M, m).$$

Значит, имеет силу следующая оценка:

$$\|A - CGR\|_2 \leq \left\{ 1 + \left[\sqrt{t(r, N, n)} + \sqrt{t(r, M, m)} \right]^2 \right\} \varepsilon. \quad \square \quad (2.15)$$

Заметим, что данная оценка в точности совпадает с упомянутой ранее оценкой для квадратной подматрицы A_{11} , приведённой в статье [24]. Последнюю оценку можно улучшить определив новую матрицу шума.

Теорема 2.2.4 (Результат автора). *Для любой матрицы $A \in \mathbb{C}^{N \times M}$, такой что $A = Z + F$, $\text{rank} Z = r$, $\|F\|_2 \leq \varepsilon$, существует τ -псевдоскелетный аппроксимант \tilde{A} , построенный на n строках и m столбцах, $m \geq n$, такой, что*

$$\|A - \tilde{A}\|_2 \leq \varepsilon t(n, M, m) \sqrt{1 + t^2(r, N, n)}.$$

Доказательство. Определим строчный аппроксимант \tilde{A}_R :

$$\tilde{A}_R = C_u \begin{bmatrix} A_{11} & A_{12} \end{bmatrix},$$

$$C_u = \begin{bmatrix} I \\ Z_{21} Z_{11}^\dagger \end{bmatrix}.$$

Введём матрицу \hat{F} по следующему правилу:

$$\hat{F} = A - \tilde{A}_R, \quad \text{rank } \tilde{A}_R \leq n, \quad (2.16)$$

$$\hat{F} = Z + F - C_u \begin{bmatrix} Z_{11} & Z_{12} \end{bmatrix} - C_u \begin{bmatrix} F_{11} & F_{12} \end{bmatrix},$$

$$\hat{F} = F - C_u \begin{bmatrix} F_{11} & F_{12} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -C_{u2} & I \end{bmatrix} F, \quad (2.17)$$

$$\|C_u\|_2 \leq t(r, N, n).$$

Напомним, что

$$Z = uv, u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, C_u u_1 = u, u_1 \in \mathbb{C}^{n \times r}.$$

Так как u_1 – подматрица ортогональной матрицы, то все её сингулярные числа не превосходят 1. Подберём $w \in \mathbb{C}^{n \times (n-r)}$ такую, что:

$$u_1^T w = 0, w^T w = I.$$

Введём дополнительные обозначения:

$$d = \begin{bmatrix} u_1 & w \end{bmatrix}, D = C_u d.$$

Так как u_1 и w ортогональны друг другу и все сингулярные числа u_1 не превосходят сингулярных чисел w , то верно следующее утверждение:

$$\forall i = 1..r : \sigma_i(u_1) = \sigma_{n-r+i}(d). \quad (2.18)$$

Проведём преобразования со строчным аппроксимантом \tilde{A}_R :

$$\tilde{A}_R = C_u \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} = D \left(d^{-1} \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} \right) = D \hat{U} \hat{S} \hat{V}, \hat{U} \in \mathbb{C}^{n \times n}.$$

Так как \hat{U} – квадратная унитарная матрица, то умножение на неё не меняет сингулярные числа ни одной подматрицы размера $n \times n$ в матрице D . Далее покажем, что D – матрица с ортонормированными столбцами.

$$D = \begin{bmatrix} u_1 & w \\ u_2 & C_{u2} w \end{bmatrix},$$

так как C_{u2} – решение уравнения

$$C_{u2} u_1 = u_2$$

в смысле наименьших квадратов, то есть

$$C_{u2} = u_2 (u_1)^\dagger = u_2 (u_1^T u_1)^{-1} u_1^T,$$

то при условии ортогональности w и u_1 получаем:

$$C_{u2} w = u_2 (u_1^T u_1)^{-1} (u_1^T w) = 0.$$

Из этого следует, что матрица D состоит из ортонормированных столбцов и имеет следующий вид:

$$D = \begin{bmatrix} u_1 & w \\ u_2 & 0 \end{bmatrix}.$$

Из равенства сингулярных чисел в уравнении (2.18) следует, что

$$\|d^\dagger\|_2 = \|(u_1)^\dagger\|_2.$$

Это означает, что при построении CGR аппроксимации по первым n строкам имеет силу следующее соотношение:

$$C_d = C_u \Rightarrow \|C_d\|_2 = \|C_u\|_2 \leq t(r, N, n).$$

Матрица коэффициентов C_v строится на основе матрицы $\begin{bmatrix} A_{11} & A_{12} \end{bmatrix}$, $\text{rank} \left(\begin{bmatrix} A_{11} & A_{12} \end{bmatrix} \right) \leq n$. Из этого можно заключить существование такого набора столбцов, что

$$\|C_v\|_2 \leq t(n, M, m).$$

Не умаляя общности, будем считать, что базисными выбраны первые m столбцов. Воспользуемся уравнением (2.17) и распишем τ -псевдоскелетную аппроксимацию \tilde{A} из уравнения (2.13):

$$\tilde{A} = C_d A_{11} (A_{11})_\tau^\dagger A_{11} C_v + \begin{bmatrix} 0 & 0 \\ -C_{u2} & I \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix} (A_{11})_\tau^\dagger A_{11} C_v.$$

Из этого следует оценка

$$\|A - \tilde{A}\|_2 \leq \tau \|C_d\|_2 \|C_v\|_2 + \varepsilon \|C_v\|_2 \sqrt{1 + \|C_{u2}\|_2^2}.$$

Заменяя τ -псевдообратную к подматрице A_{11} на просто псевдообратную, получаем следующие выкладки:

$$\begin{aligned} (A_{11}) \tau^\dagger &= (A_{11})^\dagger, \\ A_{11} (A_{11}) \tau^\dagger A_{11} &= A_{11}. \end{aligned}$$

Это приводит к следующей оценке:

$$\|A - \tilde{A}\|_2 \leq \varepsilon \|C_v\|_2 \sqrt{1 + \|C_{u2}\|_2^2}.$$

Подставляя вместо $\|C_v\|_2$ и $\|C_{u2}\|_2$ ранее полученные оценки, завершаем доказательство теоремы.

2.3. Модифицированная скелетная аппроксимация

Определим модифицированную скелетную аппроксимацию матрицы A по подматрице \hat{A} следующим образом:

$$\tilde{A} = \hat{C}\hat{A}\hat{R}.$$

Здесь матрицы \hat{C} и \hat{R} являются матрицами коэффициентов, а не подматрицами A как в скелетной аппроксимации. В частном случае, когда подматрица \hat{A} обратима, матрицы коэффициентов могут быть вычислены из скелетного разложения:

$$A = C\hat{A}^{-1}R = (C\hat{A}^{-1})\hat{A}(\hat{A}^{-1}R) = \hat{C}\hat{A}\hat{R}.$$

Напомним, что

$$A = Z + F, \text{rank}Z = r, Z = uv, \|F\|_2 \leq \varepsilon.$$

Покажем как будет вести себя разность самой матрицы и аппроксиманта при условии построения матриц \hat{C} и \hat{R} по матрице Z . Вычисление матриц \hat{C} и \hat{R} по матрице Z основывается на выборе матрицы u_1 , состоящей из строк матрицы u , и на выборе матрицы v_1 , состоящей из столбцов матрицы v . Не умаляя общности, будем считать, что \hat{A} основано на подматрице A_{11} из блочного разбиения матрицы A :

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}, F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix},$$

Как и ранее, матрицы коэффициентов будем обозначать C_u и C_v .

$$u = C_u u_1, v = v_1 C_v, Z = C_u Z_{11} C_v.$$

В качестве матриц \hat{C} и \hat{R} выберем матрицы C_u и C_v , тогда аппроксимант будет выглядеть следующим образом:

$$\tilde{A} = C_u A_{11} C_v = C_u Z_{11} C_v + C_u F_{11} C_v = Z + C_u F_{11} C_v.$$

При этом разность матрицы A и аппроксиманта \tilde{A} примет вид:

$$A - \tilde{A} = F - C_u F_{11} C_v.$$

Очевидно, верна оценка:

$$\|A - \tilde{A}\| \leq (t(r, N, n)t(r, M, m) + 1)\varepsilon.$$

Однако, мы не ограничимся этой оценкой. Большой интерес представляет ошибка аппроксимации в C -норме:

$$\|A - \tilde{A}\|_C \leq (\|C_u\|_\infty \|C_v^T\|_\infty + 1)\varepsilon.$$

Так как u_1 и v_1 можно выбирать не квадратными, то уравнения

$$u = C_u u_1, v = v_1 C_v$$

будут иметь лишние степени свободы для минимизации некоторых норм матриц C_u и C_v . Пример таких норм уже был описан при оценке ошибки аппроксимации в C -норме.

Построим строчную аппроксимацию на строках $\begin{bmatrix} A_{11} & A_{12} \end{bmatrix}$ по аналогии с модифицированной скелетной аппроксимацией:

$$\tilde{A}_R = C_u \begin{bmatrix} A_{11} & A_{12} \end{bmatrix},$$

где матрица C_u вычисляется из малоранговой матрицы Z . Очевидно,

$$\tilde{A}_R = Z + C_u \begin{bmatrix} F_{11} & F_{12} \end{bmatrix}.$$

Ранее уже было показано, что матрица шума в таком случае будет вычисляться из следующего выражения:

$$\hat{F} = A - \tilde{A}_R = \begin{bmatrix} 0 & 0 \\ -C_{u2} & I \end{bmatrix} F.$$

Из этого равенства была получена оценка для l_2 -нормы:

$$\|A - \tilde{A}_R\|_2 \leq \sqrt{1 + t^2(r, N, n)}\varepsilon.$$

Рассматривая C -норму ошибки аппроксимации, получаем:

$$\|A - \tilde{A}_R\|_C \leq (\|C_u\|_\infty + 1)\varepsilon.$$

Таким образом, получили ошибку аппроксимации выбранными строками в C -норме.

2.4. Вложенное скелетное разложение

Пусть даны матрицы A_0 , A_1 и A_2 размеров $n_0 \times m$, $n_1 \times m$ и $n_2 \times m$ соответственно, при этом матрица A_0 задана следующим образом:

$$A_0 = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}.$$

Предположим, что все эти матрицы имеют точные ранги. Таким образом, для каждой из этих матриц можно воспользоваться скелетным разложением через базисные строки:

$$A_0 = \hat{C}_0 R_0,$$

$$A_1 = \hat{C}_1 R_1,$$

$$A_2 = \hat{C}_2 R_2.$$

Так как R_1 и R_2 являются строками матрицы A_0 , их можно выразить через базисные строки матрицы R_0 :

$$\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = M_0 R_0.$$

Для того, чтобы вычислить матрицу M_0 , необходимо из R_0 выбрать несингулярную подматрицу X_0 размера $m \times m$. Пусть X_1 и X_2 – подматрицы матриц R_1 и R_2 на тех же столбцах, что и матрица X_0 . Тогда последнее уравнение преобразуется к следующему:

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = M_0 X_0.$$

Учитывая, что X_0 – несингулярная, получаем:

$$M_0 = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} X_0^{-1}.$$

Заметим, что матрица \hat{C}_0 вычисляется из следующего выражения:

$$\hat{C}_0 = \begin{bmatrix} \hat{C}_1 & 0 \\ 0 & \hat{C}_2 \end{bmatrix} M_0.$$

Из этой записи следует, что последовательность вложенных друг в друга матриц может быть представлена в виде последовательности соответствующих матриц M .

2.5. Оценка точности аппроксимации вложенными базисами

Рассмотрим простейший случай иерархически вложенных подматриц:

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}.$$

Предполагаем, что каждая из матриц A_1 и A_2 близка к матрице малого ранга:

$$\exists F_1 : \|F_1\| \leq \varepsilon \|A_1\|, \text{rank}(A_1 - F_1) \leq r,$$

$$\exists F_2 : \|F_2\| \leq \varepsilon \|A_2\|, \text{rank}(A_2 - F_2) \leq r,$$

где r – некоторое ограничение на ранг, а ε – параметр относительной точности. Из этого следует, что существуют матрицы M_1 и M_2 коэффициентов разложения всех строк матриц $(A_1 - F_1)$ и $(A_2 - F_2)$ по соответствующим базисным строкам:

$$(A_1 - F_1) = M_1 P_1 (A_1 - F_1),$$

$$(A_2 - F_2) = M_2 P_2 (A_2 - F_2),$$

где матрицы P_1 и P_2 – подматрицы единичных матриц, соответствующие выбору базисных строк. Для краткости следующих формул, введём матрицы P , M и F :

$$P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}, M = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}, F = \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix}.$$

Из этих замен получаем:

$$A = MPA + F.$$

Матрица PA есть вертикальное объединение базисных строк матриц A_1 и A_2 , выберем из них вложенные базисные строки:

$$(PA) = M_0 P_0 (PA) + F_0, \|F_0\| \leq \varepsilon \|PA\|,$$

где P_0 – подматрица единичной матрицы, соответствующая выбору базисных строк из матрицы (PA) . Построим следующую аппроксимацию:

$$\tilde{A} = M(M_0 P_0 (PA)),$$

являющуюся аппроксимацией базисными строками иерархически вложенных подматриц. Оценим ошибку аппроксимации:

$$E = A - \tilde{A}.$$

Так как

$$M_0 P_0(PA) = (PA) - F_0, MPA = A - F,$$

то ошибку аппроксимации можно переписать в следующем виде:

$$E = F + MF_0.$$

Оценим C -норму матрицы E :

$$\|E\|_C \leq \|F\|_C + \|MF_0\|_C.$$

Ранее, в подразделе 2.1.1 было показано, что базисные строки матриц A_1 и A_2 можно выбирать так, что l_2 -норма любой строки матриц M_1 и M_2 не будет превышать значение 1. Из этого факта следует, что:

$$\|MF_0\|_C \leq \|F_0\|_2,$$

$$\|E\|_C \leq \|F\|_2 + \|F_0\|_2.$$

В случае, если каждая из матриц A_1 и A_2 , в свою очередь, так же имеет вложенные подматрицы со своими базисными строками, то ошибку аппроксимации можно обобщить следующим образом:

$$\|E\|_C \leq \sum_{i=1}^h \|F_i\|_2, \quad (2.19)$$

где F_i – матрица шума на i -ом уровне вложенности, h – количество уровней вложенности.

2.6. Выводы по главе

В данной главе автором предложено обобщенное понятия p -объёма матриц, применимое и к прямоугольным матрицам. Сформулирована и доказана лемма об определителе произведения матриц, следствием которой является предложенный

автором принцип максимального 2-объёма. Построен и опробован «жадный» алгоритм поиска подматрицы максимального 2-объёма. Выведены оценки погрешности псевдоскелетных аппроксимаций на основе прямоугольных матриц, являющиеся обобщением ранее известных оценок.

Глава 3

«Мультизарядовый» метод

3.1. «Мультизарядовое» представление \mathcal{H}^2 -матрицы

Матрица принадлежит классу \mathcal{H}^2 -матриц, если все блочные строки и блочные столбцы имеют малый ранг. В разделе 1.6, посвященном \mathcal{H}^2 -матрицам, было показано, что это приводит к существованию следующих малоранговых разложений дальних подматриц:

$$\forall(t, s) \in \mathbf{F}(A) : \chi(t)A\chi(s) = P(t)\hat{A}(t, s)P(s).$$

Заметим, что эта запись походит на запись модифицированного скелетного разложения матрицы $\chi(t)A\chi(s)$. Рассмотрим строчные аппроксимации блочных строк:

$$R(t) = \hat{P}(t)\hat{R}(t), t \in \mathcal{T}_I,$$

где $\hat{R}(t)$ – базисные строки матрицы $R(t)$, и столбцовые аппроксимации блочных столбцов:

$$C(t) = \hat{C}(t)\hat{P}(t), t \in \mathcal{T}_J,$$

где $\hat{C}(t)$ – базисные столбцы матрицы $C(t)$. При этом матрицы $\hat{P}(t)$ – матрицы коэффициентов разложения всей блочной строки или всего блочного столбца по базисным строкам или базисным столбцам. Обозначим базисные строки матриц $R(t)$ и базисные столбцы матриц $C(t)$ следующим образом:

$\text{basis}(t)$ – номера базисных строк или столбцов, соответствующих узлу t ,

$\hat{\chi}(t)$ – подматрица единичной матрицы, если $t \in \mathcal{T}_I$, то $\hat{\chi}(t)$ есть набор строк с номерами $\text{basis}(t)$ единичной матрицы размера $N_I \times N_I$, иначе $\hat{\chi}(t)$ есть набор

столбцов с номерами $\text{basis}(t)$ единичной матрицы размера $N_J \times N_J$. Далее набор $\text{basis}(t)$ и соответствующую матрицу $\hat{\chi}(t)$ будем называть базисным набором узла t . Тогда, для любой пары разделённых узлов (t, s) , выражение $\chi(t)A\chi(s)$ можно переписать в виде:

$$\begin{aligned}\chi(t)A\chi(s) &= R(t)\chi(s), \\ R(t) &= \hat{P}(t)\hat{\chi}(t)R(t), \\ \chi(t)A\chi(s) &= \hat{P}(t)\hat{\chi}(t)R(t)\chi(s) = \hat{P}(t)\hat{\chi}(t)C(s), \\ \chi(t)A\chi(s) &= \hat{P}(t)\hat{\chi}(t)A\hat{\chi}(s)\hat{P}(s).\end{aligned}$$

Ранее введённые в разделе 1.6 матрицы перехода $M(t)$ обозначали коэффициенты пересчёта базисов узлов-детей по базису узла t . В разделе 2.4, на примере матриц \hat{C} , было показано, что матрицы $\hat{P}(t)$ могут быть вычислены рекурсивно именно через такие матрицы $M(t)$. Тогда в качестве $M(t)$ выберем матрицы коэффициентов разложения базисных строк или столбцов узлов-детей по базисным строкам или столбцам узла t . Далее, принимая

$$\hat{A}(t, s) = \hat{\chi}(t)A\hat{\chi}(s),$$

получаем полноценное \mathcal{H}^2 -представление матрицы A .

Так как матрицы $\hat{A}(t, s)$ являются подматрицами исходной матрицы A , а матрица A порождена некоторой функцией, то подматрицы $\hat{A}(t, s)$ можно вычислять по мере необходимости. Это означает, что хранить матрицы взаимодействий $\hat{A}(t, s)$ не обязательно. В предположении, что все ранги блочных строк и блочных столбцов равны r , на каждый узел t приходится матрица перехода $M(t)$ размера $(|\mathbf{s}(t)|r) \times r$ и $|\mathcal{F}(t)|$ матриц взаимодействий размера $r \times r$. Учитывая, что $|\mathbf{s}(t)| \ll |\mathcal{F}(t)|$, в случае вычисления матриц взаимодействий по мере необходимости, получаем эффективную по памяти структуру. Если принять ранги каждой блочной строки и каждого блочного столбца равными 1, получаем немного изменённый классический алгоритм Барнс-Хата. Если посмотреть на данное разложение с точки зрения задачи многих тел (вычисление потенциалов, созданных облаком частиц на самих себя), то каждый кластер частиц заменяется несколькими базисными частицами. Вычисление потенциалов происходит в 3 этапа, подобных этапам, описанным в разделе 1.6.1. Сначала при помощи матриц перехода

$M(t), t \in \mathcal{T}_I$, вычисляются заряды базисных частиц всех кластеров. Потом при помощи матриц взаимодействий $\hat{A}(t, s)$ вычисляются потенциалы, созданные базисными частицами-источниками на базисные частицы-приёмники. В конце, при помощи матриц перехода $M(t), t \in \mathcal{T}_J$, вычисленные потенциалы пересчитываются с базисных на исходные частицы. Начиная с этого момента, будем называть такое представление \mathcal{H}^2 -матрицы *мультизарядовым представлением*. Это представление характеризуется матрицами перехода $M(t)$ и базисными наборами $\text{basis}(t)$.

3.2. «Хорошие» строки и столбцы

Описанное в предыдущем разделе разложение использует готовые наборы базисных строк и базисных столбцов. Основная проблема состоит в том, как адаптивно выбирать эти базисные наборы в каждом конкретном случае, когда заданная матрица приближается \mathcal{H}^2 -матрицей с некоторой точностью. Предыдущие подходы [6, 33, 61] используют специальные геометрические конструкции для выбора базисных строк и столбцов. В данной диссертации предложен чисто алгебраический метод выбора искомых строк и столбцов в духе алгоритма построения подматрицы максимального объёма.

Покажем, как вычислять базисные наборы и матрицы переходов для каждого узла t кластерного дерева строк. Пусть $t_0 = t$ – нелистьевой узел (узел, у которого есть узлы-дети), а t_1 и t_2 – его узлы-сыновья. Так как $\mathbf{Rf}(t) = \mathbf{Rp}(t_1) + \mathbf{Rp}(t_2)$ из определения, то базисные строки $\mathbf{Rf}(t)$ можно выбрать из объединения базисных строк матриц $\mathbf{Rf}(t_1)$ и $\mathbf{Rf}(t_2)$ (так как $\mathbf{Rp}(t_1)$ и $\mathbf{Rp}(t_2)$ являются их подматрицами):

$$\hat{\mathbf{Rf}}(t) = \hat{\chi}(t_1)\mathbf{Rp}(t_1) + \hat{\chi}(t_2)\mathbf{Rp}(t_2).$$

Матрица $\hat{\mathbf{Rf}}(t)$ есть объединение базисных строк матриц $\mathbf{Rp}(t_1)$ и $\mathbf{Rp}(t_2)$ и содержит ненулевую подматрицу размера $(r_1 + r_2) \times N_t$, где $N_t = |\text{ind}(\mathcal{F}_f(t))|$, $r_1 = |\text{basis}(t_1)|$, $r_2 = |\text{basis}(t_2)|$. Таким образом, вычисление базисных строк узла t по матрице $\hat{\mathbf{Rf}}(t)$ потребует $O(N_t)$ операций. Складывая эту сложность по всем узлам кластерного дерева строк, получаем итоговую сложность $O(N^2)$ операций. Поэтому, для сокращения количества операций, необходимо использовать лишь часть столбцов матрицы $\hat{\mathbf{Rf}}(t)$, то есть выбирать *хорошие* столбцы.

Предположим, что «хорошие» строки и столбцы известны для каждого узла. Для удобства, снова воспользуемся диагональными матрицами для их обозначения:

$\text{repr}(t)$ – (representors) номера «хороших» строк или столбцов для узла t ,

$\psi(t)$ – диагональная матрица такая, что $\psi(t)_{ii} = 1$, если $i \in \text{repr}(t)$, и 0 в противном случае.

Если известен набор «хороших» столбцов для узла t , то базисные строки могут быть вычислены по матрице $\hat{\mathbf{Rf}}(t)\psi(t)$. Эта операция, наряду с вычислением матрицы перехода для узла t , может быть выполнена с высокой эффективностью при помощи процедуры **maxvol** [35], приведённой в разделе 1.3. Общая схема алгоритма вычисления всех базисных наборов и матриц перехода по данным «хорошим» наборам, приведена в Алгоритме 4.

Алгоритм 4 (Результат автора) Вычисление базисных наборов и матриц перехода по данным наборам «хороших» строк и столбцов.

Дано: $\psi(t)$ для каждого узла $t \in \mathcal{T}_I \cup \mathcal{T}_J$, параметр относительной точности τ

Вычислить: $\hat{\chi}(t)$ и $M(t)$ для каждого узла $t \in \mathcal{T}_I \cup \mathcal{T}_J$

- 1: **for** $t \in \mathcal{T}_I \cup \mathcal{T}_J$, по деревьям снизу вверх **do**
 - 2: **if** $|\mathbf{s}(t)| > 0$ {если у t есть узлы-дети} **then**
 - 3: $Z = \hat{\chi}(\mathbf{s}(t))\mathbf{Rf}(t)\psi(t)$
 - 4: **else**
 - 5: $Z = \mathbf{Rf}(t)\psi(t)$
 - 6: **end if**
 - 7: {truncated svd + **maxvol**}
 - 8: $U, S, V = \text{svd}(Z, \tau); M(t), \hat{\chi}(t) = \mathbf{maxvol}(U)$
 - 9: **end for**
-

3.2.1. «Хорошие» наборы и своя дальняя зона

Так как полная дальняя зона состоит из своей дальней зоны и родительской дальней зоны, то, предположим, что «хорошие» наборы родительской дальней зоны $\psi_p(t)$ каждого узла t уже даны. Как выбирать «хорошие» наборы для своей дальней зоны? Для этого будем использовать базисные наборы своей дальней

зоны. Таким образом, мы задаем следующее:

$$\psi(t) = \psi_p(t) + \hat{\chi}(\mathcal{F}(t)).$$

Это определение иерархическое и, более того, требует чтобы базисные наборы каждого узла из $\mathcal{F}(t)$ были уже известны. Так как для каждого узла $s \in \mathcal{F}(t)$, узел $t \in \mathcal{F}(s)$, то $\psi(t)$ зависит от $\hat{\chi}(s)$ и $\psi(s)$ зависит от $\hat{\chi}(t)$. Учитывая, что $\hat{\chi}(t)$ вычисляется на основе $\psi(t)$, получаем «мёртвый цикл» зависимостей:

$$\psi(t) \leftarrow \hat{\chi}(s) \leftarrow \psi(s) \leftarrow \hat{\chi}(t) \leftarrow \psi(t).$$

Эту проблему можно решить довольно просто: заменив базисные наборы своей дальней зоны на объединение базисных наборов узлов-сыновей своей дальней зоны, что можно записать в следующем виде:

$$\psi(t) = \psi_p(t) + \hat{\chi}(\mathbf{s}(\mathcal{F}(t))).$$

В итоге, мы получаем алгоритм, который вычисляет базисные наборы и матрицы переходов для обоих кластерных деревьев постепенно, уровень за уровнем.

Обозначим за *level_count* количество уровней (высоту) обоих кластерных деревьев (для простоты, считаем что их высоты совпадают). Предложенный алгоритм имеет сложность $O(N)$ и приведен в виде блок-таблицы в Алгоритме 5. Отметим, что при достаточно высокой требуемой точности и нулевом значении всех $\psi_p(t)$, алгоритм 5 не может получить требуемую точность, так как наборы «хороших» строк и столбцов ограничены своей дальней зоной каждого узла кластерных деревьев. Более подробные примеры приведены в численных экспериментах.

Алгоритм 5 (Результат автора) Вычисление базисных наборов и матриц перехода

Дано: «хорошие» наборы родительских дальних зон $\psi_p(t)$, параметр относительной точности τ

Вычислить: $\hat{\chi}(t)$ и $M(t)$ для каждого узла $t \in \mathcal{T}_I \cup \mathcal{T}_J$

```

1: for current_level = level_count to 1 do
2:   for  $t \in \mathcal{T}_J$  {на уровне current_level} do
3:      $\hat{\psi}(t) = \psi_p(t)$  {инициализируем  $\hat{\psi}(t)$ }
4:     for  $s \in \mathcal{F}(t)$  do
5:       if  $|\mathbf{s}(s)|! = 0$  {если у узла  $s$  есть узлы-дети} then
6:          $\hat{\psi}(t)+ = \hat{\chi}(\mathbf{s}(s))$ 
7:       else
8:          $\hat{\psi}(t)+ = \hat{\chi}(s)$ 
9:       end if
10:    end for
11:    if  $|\mathbf{s}(t)|! = 0$  {если у узла  $t$  есть узлы-дети} then
12:       $Z = \hat{\psi}(t)A\hat{\chi}(\mathbf{s}(t))$ 
13:    else
14:       $Z = \hat{\psi}(t)A\chi(t)$ 
15:    end if
16:     $U, S, V = \text{svd}(Z, \tau); M(t), \hat{\chi}(t) = \mathbf{maxvol}(V)$  {truncated svd + maxvol}
17:  end for
18:  for  $t \in \mathcal{T}_I$  {на уровне current_level} do
19:     $\hat{\psi}(t) = \psi_p(t)$  {инициализируем  $\hat{\psi}(t)$ }
20:     $\hat{\psi}(t)+ = \hat{\chi}(\mathcal{F}(t))$ 
21:    if  $|\mathbf{s}(t)|! = 0$  {если у узла  $t$  есть узлы-дети} then
22:       $Z = \hat{\chi}(\mathbf{s}(t))A\hat{\psi}(t)$ 
23:    else
24:       $Z = \chi(t)A\hat{\psi}(t)$ 
25:    end if
26:     $U, S, V = \text{svd}(Z, \tau); M(t), \hat{\chi}(t) = \mathbf{maxvol}(U)$  {truncated svd + maxvol}
27:  end for
28: end for

```

3.2.2. Вычисление «хороших» наборов

Если все $\psi(t)$ вычислены, то $\psi_p(t)$ можно задать следующим способом:

$$\psi_p(t) = \psi(\mathbf{p}(t)).$$

Покажем, как можно быстро вычислить «хороший» набор узла, если «хороший» набор его узла-отца уже известен. Пусть t_1 – узел-сын узла t , $\psi(t)$ – «хороший» набор узла t , тогда «хороший» набор узла t_1 можно выбирать из *потенциально хорошего* набора, соответствующего следующей матрице:

$$\hat{\psi}(t_1) = \psi(t) + \hat{\chi}(\mathcal{F}(t_1)).$$

Данный набор является «хорошим» для узла t_1 . Однако, этот набор можно сократить, используя процедуру **maxvol** и ранее вычисленный базисный набор $\hat{\chi}(t_1)$. Итоговый «хороший» набор для узла $t_1 \in \mathcal{T}_I$ вычисляется по следующей формуле:

$$\psi(t_1) = \mathbf{maxvol} \left(\hat{\chi}(t_1) A \hat{\psi}(t_1) \right),$$

где **maxvol** применяется для нахождения базисных столбцов матрицы $\hat{\chi}(t_1) A \hat{\psi}(t_1)$. Аналогичная формула получается и для вычисления «хорошего» набора узла $t \in \mathcal{T}_J$:

$$\psi(t) = \mathbf{maxvol} \left(\hat{\psi}(t_1) A \hat{\chi}(t_1) \right),$$

но здесь **maxvol** применяется для вычисления базисных строк. Если же у узла t_1 нет узла-родителя, то просто заменяем $\psi(t)$ из определения $\hat{\psi}(t_1)$ на нулевую матрицу. Таким образом, «хорошие» наборы можно вычислять по базисным наборам иерархически, проходя кластерное дерево «сверху вниз». Соответствующий алгоритм формализован в виде блок-таблицы в Алгоритме 6.

3.3. Итерационный алгоритм

Основная идея «мультизарядового» метода состоит в поочередном вычислении базисных наборов и «хороших» наборов для каждого узла кластерных деревьев строк \mathcal{T}_I и столбцов \mathcal{T}_J . Из «хороших» наборов получаем базисные наборы, по ним пересчитываем «хорошие» наборы, и так далее, пока не будет выполнен некоторый критерий останова. Позже мы покажем, что количество итераций,

Алгоритм 6 (Результат автора) Вычисление «хороших» наборов по базисным наборам

Дано: кластерное дерево \mathcal{T} , базисные наборы $\hat{\chi}(t)$ каждого узла $t \in \mathcal{T}$

Вычислить: «хорошие» наборы $\psi(t)$

```

1: for  $current\_level = 1$  to  $level\_count$  do
2:   for  $t \in \mathcal{T}$  {на уровне  $current\_level$ } do
3:     if  $current\_level = 1$  {инициализируем  $\hat{\psi}(t)$ } then
4:        $\hat{\psi}(t) = 0$ 
5:     else
6:        $\hat{\psi}(t) = \psi(\mathbf{p}(t))$ 
7:     end if
8:      $\hat{\psi}(t)_+ = \hat{\chi}(\mathcal{F}(t))$ 
9:     if  $\mathcal{T} = \mathcal{T}_I$  {если дано кластерное дерево строк} then
10:       $\psi(t) = \mathbf{maxvol}(\hat{\chi}(t)A\hat{\psi}(t))$ 
11:    else
12:       $\psi(t) = \mathbf{maxvol}(\hat{\psi}(t)A\hat{\chi}(t))$ 
13:    end if
14:  end for
15: end for

```

необходимых для достижения требуемой точности, само по себе мало, но зависит от требуемой точности и физической задачи, порождающей матрицу A .

Предложенный автором алгоритм является итерационным и представлен в Алгоритме 7. В качестве инициализации обнуляем все «хорошие» наборы и используем Алгоритм 5 для вычисления базисных наборов. Каждый шаг итерации состоит из двух этапов: вычисление «хороших» наборов при помощи Алгоритма 6 и вычисление обновлённых базисных наборов и матриц перехода при помощи Алгоритма 5. Каждая итерация использует всю информацию с предыдущей итерации и не ухудшает точность аппроксимации. Численные эксперименты показали что итерационный алгоритм сходится за 1 – 2 итерации.

Алгоритм 7 (Результат автора) Итерационный алгоритм МСВН

Дано: иерархические кластерные деревья $\mathcal{T}_I, \mathcal{T}_J$, параметр относительной точности τ , матрица A

Вычислить: базисные наборы и матрицы перехода для каждого узла $t \in \mathcal{T}_I \cup \mathcal{T}_J$

- 1: $\forall t \in \mathcal{T}_I \cup \mathcal{T}_J : \psi(t) = 0$
 - 2: **while** количество итераций меньше заданного **do**
 - 3: вычислить «хорошие» наборы Алгоритмом 6
 - 4: $\forall t \in \mathcal{T}_I \cup \mathcal{T}_J : \psi_p(t) = \psi(\mathbf{p}(t))$
 - 5: вычислить базисные наборы и матрицы переходов Алгоритмом 5
 - 6: **end while**
-

3.4. Проверка количества итераций и достигаемой точности

Изучим поведение относительной ошибки «мультизарядового» приближения матрицы A в различных условиях. В качестве тестовых примеров были выбраны две задачи. Первая – задача многих тел, которая состоит в вычислении потенциалов, созданных облаком частиц самим на себя. Формально эту задачу можно сформулировать следующим способом: даны случайно распределённые в единичном кубе $[0, 1]^3$ точечные заряды, взаимодействующие по закону Кулона $f(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{x}-\mathbf{y}|}$. Вторая тестовая задача – задача вычисления полярной составляющей энергии сольватации молекулы, которая является составной частью сложной

системы моделирования и дизайна лекарств. Подробное описание и сравнение результатов различных методов решения этой задачи дано в разделе 4.1. В рамках модели РСМ эта задача сводится к решению граничного интегрального уравнения.

$$f(\mathbf{r}) = \int_{\Omega} \frac{\sigma(\mathbf{r}') ((\mathbf{r} - \mathbf{r}') \cdot \mathbf{n})}{|\mathbf{r} - \mathbf{r}'|} dS',$$

где Ω есть замкнутая поверхность в пространстве \mathbb{R}^3 . При дискретизации вся интегральная поверхность разбивается на отдельные дискретные элементы поверхности, а интеграл по каждому из таких дискретных элементов заменяется формулой прямоугольников. Таким образом, задача сводится к решению линейной системы с большой плотной матрицей A , элементы которой вычисляются по следующему закону:

$$A_{ij} = \begin{cases} \frac{(\varepsilon-1)}{4\pi(1+\varepsilon)} \frac{((\mathbf{r}_i - \mathbf{r}_j) \cdot \mathbf{n}_i) S_i}{|\mathbf{r}_i - \mathbf{r}_j|^3}, & i \neq j \\ \frac{\varepsilon}{1+\varepsilon} - \sum_{k \neq j} A_{kj}, & i = j \end{cases},$$

где ε – диэлектрическая проницаемость растворителя (диэлектрическая проницаемость молекулы полагается равной 1), \mathbf{r}_i – центр i -ого дискретного элемента поверхности, \mathbf{n}_i – нормаль к поверхности молекулы в точке \mathbf{r}_i , направленная внутрь растворителя, S_i – площадь i -ого дискретного элемента поверхности.

3.4.1. Программная реализация

Алгоритм 7 реализован на языке программирования Python, а наиболее трудоёмкие узкие части программы переписаны на языке C и соединены с основной программой посредством пакета Cython. Для совершения базовых операций линейной алгебры использовалась библиотека Intel Math Kernel Library. Программный комплекс доступен по адресу <http://bitbucket.org/muxas/h2tools>. Для построения иерархических кластерных деревьев применялось стандартное рекурсивное инерциальное биение. При вычислении базисных наборов и матриц перехода при помощи Алгоритма 5, в качестве параметра относительной точности было выбрано значение $\tau/level_count$, где τ – требуемая относительная точность аппроксимации, а $level_count$ – высота кластерных деревьев. Вместе с этим, мы проверили несколько ранги, полученные с помощью итерационного алгоритма, оптимальны, и провели оптимизацию рангов \mathcal{H}^2 -матрицы с

помощью алгоритма дожимания сингулярным разложением, приведённого в книге [11]. Это дожимание было применено для \mathcal{H}^2 -приближения матрицы A , полученного алгоритмом 7 на 1 итерации при $\tau = 10^{-8}$. Рисунки, подписанные “Approximation memory, MB”, показывают количество мегабайт памяти, необходимых для хранения всех матриц перехода и взаимодействий. Относительная ошибка приближения матриц в спектральной норме меряется при помощи пакета [PROPACK](#).

Тесты проводились на компьютере с 2 процессорами Intel(R) Xeon(R) CPU E5504 @ 2.00GHz с общей оперативной памятью в 72 гигабайта. Не смотря на то, что библиотека MKL является многопоточной, все рутинны линейной алгебры запускались в режиме 2-х потоков (количество потоков по умолчанию равно 2). Интерпретатор языка Python, пакет Cython и библиотека MKL взяты из [Enthought Python Distribution \(EPD 7.3-1, 64-bit\)](#). Версии программ следующие: Python – “2.7.3”, Cython – “0.16”, MKL – “10.3-1”.

3.4.2. Проверка на задаче многих тел

На рисунке 3.1 показана зависимость относительной ошибки аппроксимации дальних подматриц от количества итераций и параметра относительной точности «мультизарядового» метода.

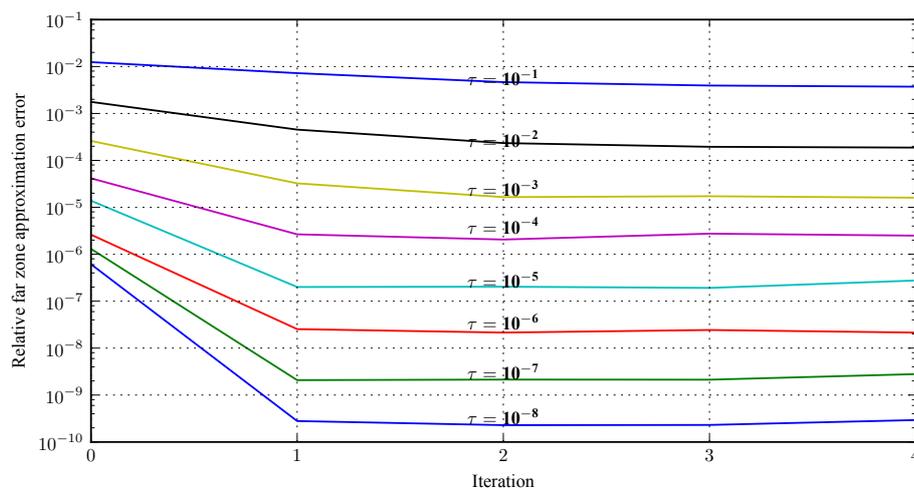


Рисунок 3.1: Зависимость относительной точности приближения дальней зоны от количества итераций (слева) и параметра относительной точности (слева) для задачи многих тел, 100000 частиц.

На следующих рисунках показаны время построения «мультизарядового» приближения и необходимая для хранения аппроксимации память. Численные эксперименты для задачи многих тел проводились с 20000, 40000, 60000, 80000 и 100000 частиц на 1 итерации, при этом параметр точности τ принимал значения 10^{-1} , 10^{-2} , 10^{-3} , 10^{-4} , 10^{-5} , 10^{-6} , 10^{-7} и 10^{-8} . Для каждого набора параметров были проведены по 100 экспериментов со случайной конфигурацией частиц в единичном кубе $[0; 1]^3$. На графиках представлены средние и максимальные значения времени расчётов и памяти для хранения \mathcal{H}^2 -аппроксимации.

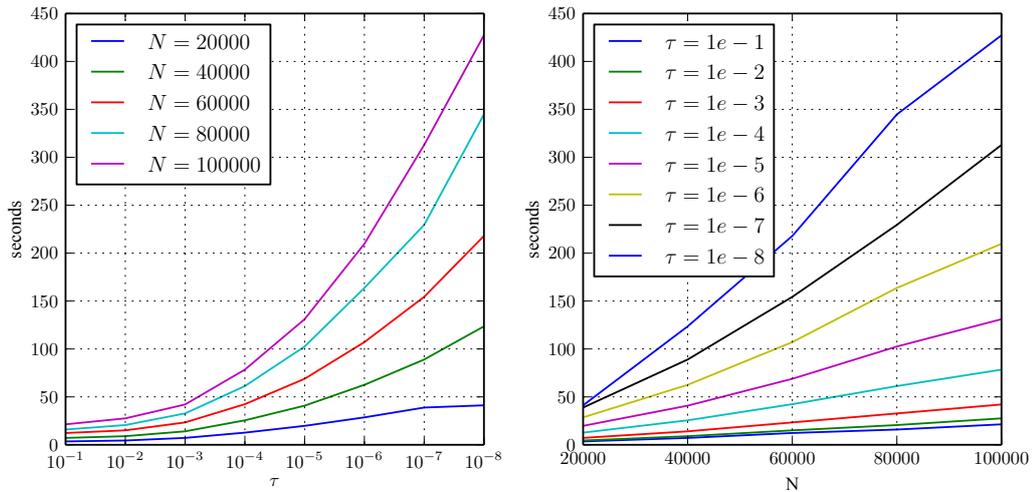


Рисунок 3.2: Зависимость среднего времени построения разложения от параметра точности (слева) и количества частиц (справа).

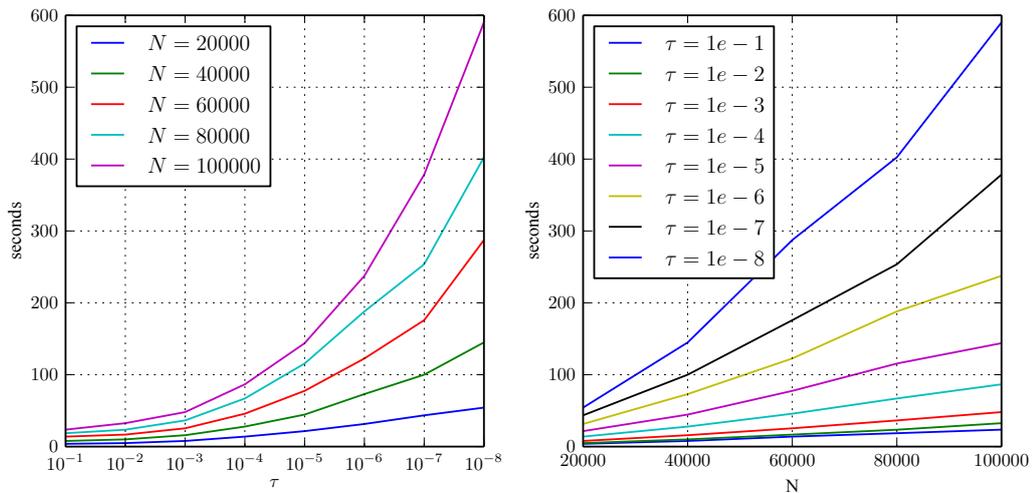


Рисунок 3.3: Зависимость максимального времени построения разложения от параметра точности (слева) и количества частиц (справа).

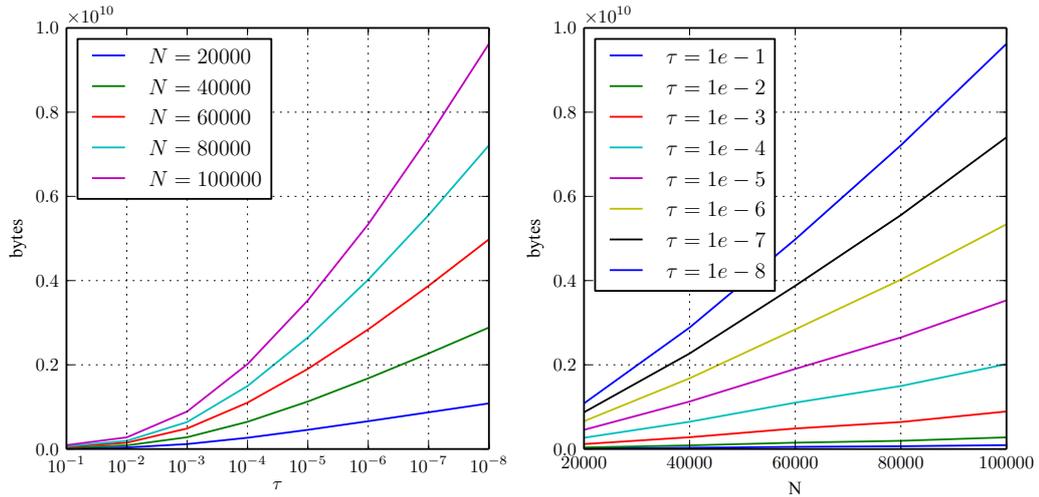


Рисунок 3.4: Зависимость среднего значения памяти для хранения \mathcal{H}^2 -аппроксимации от времени построения разложения от параметра точности (слева) и количества частиц (справа).

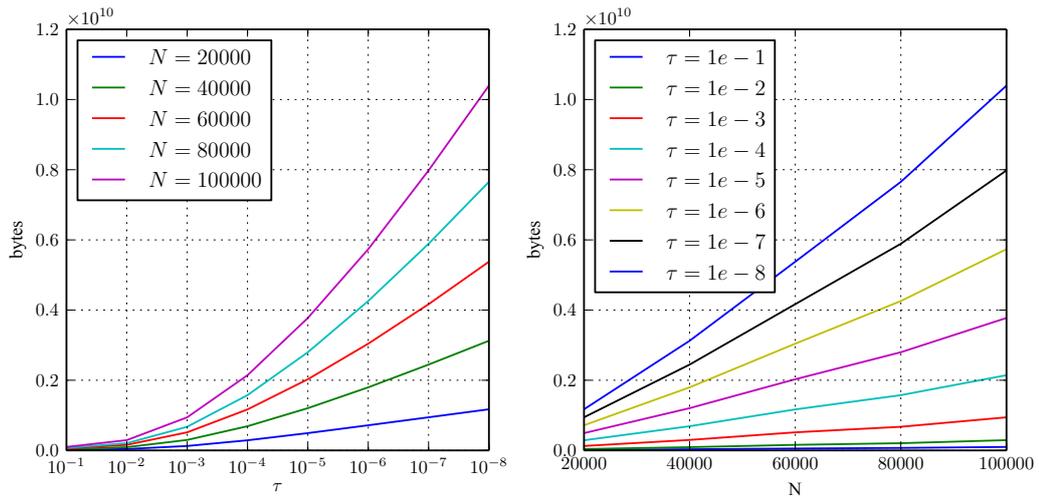


Рисунок 3.5: Зависимость максимального значения памяти для хранения \mathcal{H}^2 -аппроксимации от времени построения разложения от параметра точности (слева) и количества частиц (справа).

3.4.3. Проверка на граничном интегральном уравнении

На следующих рисунках показана относительная ошибка аппроксимации дальних подматриц, время построения аппроксимации и памяти «мультизарядового» метода для граничного интегрального уравнения.

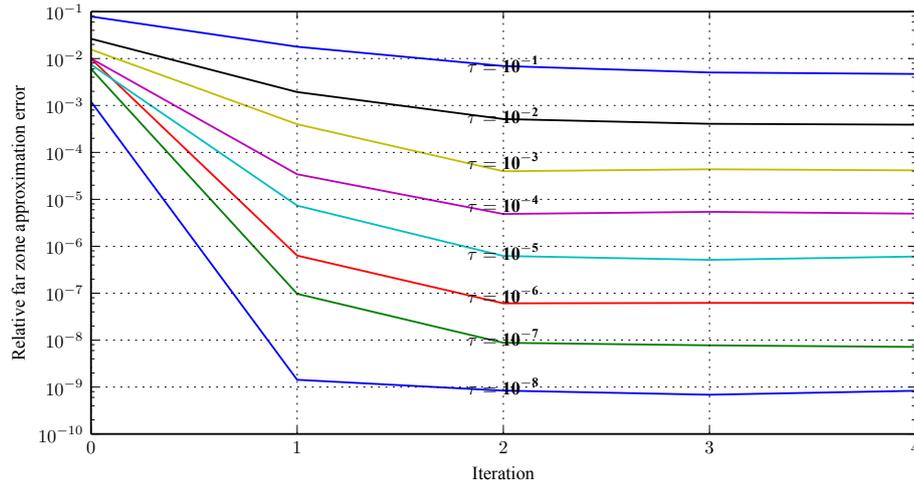


Рисунок 3.6: Зависимость относительной точности приближения дальней зоны от количества итераций (слева) и параметра относительной точности (слева) для граничного интегрального уравнения, 222762 дискретных элемента поверхности.

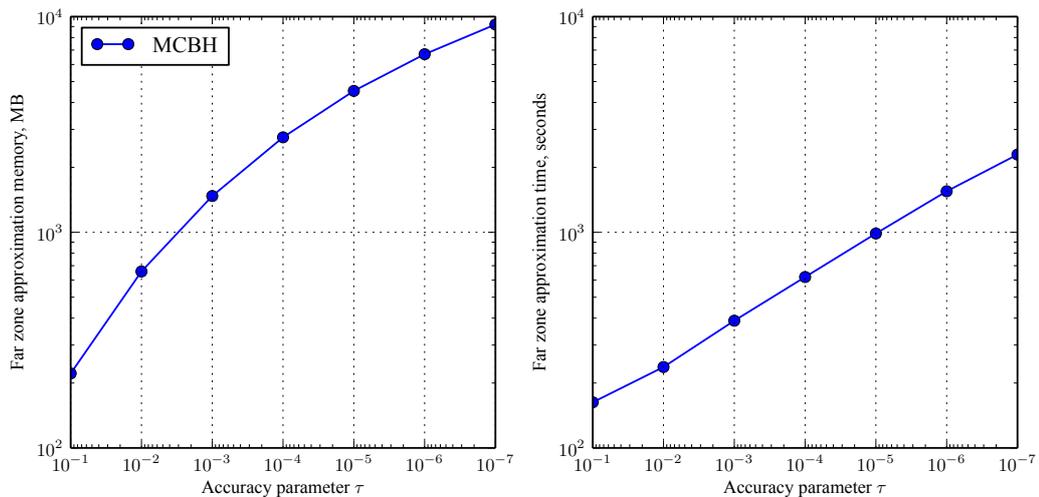


Рисунок 3.7: Зависимость времени работы (справа) и требуемой памяти (слева) от параметра относительной точности для граничного интегрального уравнения, 222762 дискретных элемента поверхности.

3.5. Выводы по главе

Данная глава посвящена основному результату диссертации – «мультизарядовому» методу приближения больших плотных матриц \mathcal{H}^2 -матрицами. Иерархическая структура \mathcal{H}^2 -матриц в «мультизарядовом» формате основывается на

выборе базисных строк и базисных столбцов в специальных подматрицах. Выбор базисных строк любой матрицы можно свести к поиску базисных строк подматрицы, основанной на «хороших» столбцах. Предлагаемый «мультизарядовый» метод выбирает «хорошие» строки и столбцы адаптивно при помощи итерационного процесса. Показано, что для приближения матриц, возникающей в задачах электростатики и вычисления энергии десольватации молекул в воде, достаточно 1-2 итераций.

Глава 4

Численные эксперименты

4.1. Континуальная модель растворителя

В настоящее время рациональная разработка лекарств становится не только всё более актуальной, но и возможной благодаря интенсивному развитию молекулярной биологии, а также методов рентгеноструктурного анализа белков. Для многих патологий сравнительно быстро выявляются белки-мишени, блокирование работы которых позволяет остановить развитие заболевания, определяются их трёхмерные атомно-молекулярные структуры, а в них – так называемые активные центры, ответственные за функционирование этих белков. Блокировка работы этих белков-мишеней осуществляется путём избирательного связывания с их активными центрами сравнительно небольших молекул-ингибиторов, которые и становятся основой новых лекарств. Поиск или разработка таких ингибиторов представляет собой и ключевой, и наиболее сложный этап разработки новых лекарств. Существенную помощь в решении этой непростой задачи может дать применение методов компьютерного молекулярного моделирования [86]. Способность ингибитора блокировать работу данного белка-мишени определяет энергия его связывания с белком – чем она больше, тем выше эффективность лекарства на основе этого ингибитора.

В компьютерной разработке новых ингибиторов первое место занимает докинг лигандов (см., например, [2, 70, 85]), т.е. молекул-кандидатов в ингибиторы, в активные центры белков-мишеней, в результате которого происходит позиционирование лигандов в этих центрах и вычисление свободной энергии их связывания

с белком-мишенью. Точность вычисления этой величины – ключевой параметр, определяющий надёжность предсказания ингибирующей способности лигандов и напрямую влияющий на эффективность применения методов молекулярного моделирования для разработки новых лекарств. Достаточно высокая практическая предсказуемость достигается при погрешности расчёта энергии связывания белок-лиганд, не превосходящей 1 ккал/моль. Эта точность определяется многими факторами, среди которых важную роль играет описание взаимодействия молекул с растворителем – водой, поскольку все взаимодействия молекул в организме фактически происходят в водной среде. В настоящее время при молекулярном моделировании используются два метода описания взаимодействия молекул с растворителем. Первый – явная модель растворителя, в которой окружающий молекулу растворяемого вещества растворитель представляется в виде большого массива более или менее равномерно распределённых молекул воды, взаимодействующих друг с другом и с молекулой растворяемого вещества с помощью межмолекулярных, в том числе кулоновских, сил. В этой модели значительные вычислительные затраты требуются на описание взаимодействия огромного количества (десятки и сотни тысяч) молекул растворителя друг с другом и для усреднения их теплового движения – как правило методами молекулярной динамики. Из-за больших вычислительных затрат такие методы никогда не используются в задачах докинга, как, впрочем, и при квантово-химическом описании молекулярных систем. Второй метод описания растворителя – это неявная, континуальная модель растворителя [16, 43, 55], в которой весь растворитель, окружающий данную молекулу, представляется в виде, как правило, однородного диэлектрика с диэлектрической проницаемостью растворителя. В случае воды, имеющей высокую диэлектрическую проницаемость (78.5 при $T=300\text{K}$), растворитель заменяют иногда даже металлом ($\varepsilon = \infty$) – так называемая модель COSMO [36]. Такие модели позволяют сравнительно точно проводить расчёты кулоновских взаимодействий в системе белок-лиганд-растворитель, где растворитель (вода) вносит существенный вклад в экранировку кулоновских взаимодействий белок-лиганд. Неявные модели растворителя имеют в настоящее время сравнительно неплохие параметризацию (см., например, [10, 71]) и быстрое действие, однако последнего далеко недостаточно, чтобы применять эти модели в задачах докинга, в которых при позиционировании лиганда в белке приходится рассматривать миллионы мо-

лекулярных конфигураций. Поэтому на повестке дня рациональной разработки новых лекарств весьма остро стоит задача о существенном повышении скорости решения уравнений неявных моделей растворителя без потери точности.

В настоящей части главы предложен новый алгоритм решения матричных уравнений РСМ – Polarized Continuum Model [16, 55], позволяющий получить ускорение в сотни раз по сравнению с обычными итерационными методами [45, 71] без существенной потери точности. Этот новый алгоритм основан на использовании мультизарядового приближения больших матриц [39]. Большие матрицы в уравнении РСМ возникают при триангуляции поверхности, разделяющей молекулу и растворитель и дискретизации непрерывной плотности поляризованных зарядов, наводимых на поверхности диэлектрика атомными зарядами молекулы. Для получения высокой точности расчётов кулоновского взаимодействия белков с водой нужен достаточно малый шаг триангуляционной сетки ($\leq 0.1 \text{ \AA}$), что приводит к сотням тысяч поверхностных элементов и, соответственно, к большим размерам матриц, причем чем меньше шаг сетки и выше точность расчётов, тем больше получаем ускорение при применении мультизарядовой аппроксимации больших матриц для решения уравнений РСМ.

В настоящей работе приведено интегральное уравнение РСМ для вычисления поляризованных зарядов, его дискретизация, описан способ его решения обычным методом [45], реализованным в программе DISOLV [71], описан алгоритм и способ решения матричных уравнений РСМ с помощью мультизарядового приближения для больших матриц [39], реализованный в оригинальной программе MCBHSOLV, и проведено сравнение результатов расчётов обоими методами электростатической части энергии взаимодействия белков, лигандов и их комплексов с водой (далее для краткости – энергии сольватации), а также энергии десольватации при связывании этих белков и лигандов в комплексы, которая вычисляется как разность энергии сольватации комплекса и суммы энергий сольватации белка и лиганда, для 22 комплексов белок-лиганд, среди которых есть различные белки и лиганды разных размеров. Энергия десольватации непосредственно входит в свободную энергию связывания белок-лиганд, и точность её вычисления напрямую влияет на точность вычисления последней, а значит и на надёжность предсказания констант ингибирования молекул-кандидатов в ингибиторы. Особенностью этого сравнения является то, что в обеих программах (DISOLV

и MCVHSOLV) используются одни и те же молекулы, одни и те же молекулярные поверхности и их триангуляционные сетки и одни и те же заряды атомов соответствующих молекул.

4.1.1. Уравнение РСМ и его дискретизация на молекулярной поверхности

PCM (Polarized Continuum Model) [16, 55, 56, 71] является моделью для рассмотрения электростатических взаимодействий точечных зарядов с диэлектриком, причем заряды находятся в полости (полостях) диэлектрика, а диэлектрик вне полости (полостей) является однородным и бесконечным. Подобная модель может быть применена к описанию электростатического взаимодействия растворенной молекулы с растворителем, если считать растворитель диэлектриком, полость в котором образована атомами молекулы, каждый атом которой представлен в виде сферы определённого радиуса с точечным зарядом в центре. В рамках модели РСМ свободная энергия электростатического взаимодействия растворённой молекулы с растворителем ΔG_{pol} вычисляется по формуле (4.1) [71]:

$$\Delta G_{pol} = \frac{1}{2} \sum_i Q_i \int_{\Omega} \frac{\sigma(\mathbf{r})}{|\mathbf{r} - \mathbf{R}_i|} dS, \quad (4.1)$$

где Q_i – заряд i -ого атома молекулы,

\mathbf{R}_i – радиус-вектор i -ого атома молекулы,

Ω – поверхность, формирующая полость (полости) в растворителе,

σ – поверхностная плотность поляризационного заряда,

\mathbf{r} – радиус-вектор элемента поверхности.

В формуле (4.1) полагается, что относительная диэлектрическая проницаемость внутри полости равна 1.

В свою очередь, поверхностная плотность поляризационных зарядов $\sigma(\mathbf{r})$ является решением интегрального уравнения (4.2):

$$\sigma(\mathbf{r}) = \frac{1 - \varepsilon}{2\pi(1 + \varepsilon)} \left(\sum_i \frac{Q_i((\mathbf{r} - \mathbf{R}_i) \cdot \mathbf{n})}{|\mathbf{r} - \mathbf{R}_i|^3} + \int_{\Omega} \frac{\sigma(\mathbf{r}')((\mathbf{r} - \mathbf{r}') \cdot \mathbf{n})}{|\mathbf{r} - \mathbf{r}'|^3} dS' \right), \quad (4.2)$$

где ε – относительная диэлектрическая проницаемость растворителя,

\mathbf{n} – нормаль к поверхности полости, направленная в растворитель.

Для решения уравнения (4.2) поверхность полости (полостей) триангулируется и, таким образом, аппроксимируется связным набором дискретных элементов, где i -ый дискретный элемент характеризуется своим центром \mathbf{r}_i , нормалью \mathbf{n}_i , площадью S_i , и поляризационным зарядом q_i . После триангуляции интегральное уравнение (4.2) преобразуется к матричному уравнению (4.3):

$$Aq = BQ, \quad (4.3)$$

где q – вектор-столбец поляризационных зарядов дискретных элементов поверхности,

Q – вектор-столбец зарядов атомов молекулы.

Матрица A имеет размер $N \times N$, где N – количество дискретных элементов поверхности, матрица B имеет размер $N \times M$, где M – количество атомов в молекуле. Элементы матрицы A вычисляются по формуле (4.4), элементы матрицы B вычисляются по формуле (4.5):

$$A_{ij} = \begin{cases} \frac{(\varepsilon-1)}{4\pi(1+\varepsilon)} \frac{((\mathbf{r}_i-\mathbf{r}_j) \cdot \mathbf{n}_i) S_i}{|\mathbf{r}_i-\mathbf{r}_j|^3}, & i \neq j \\ \frac{\varepsilon}{1+\varepsilon} - \sum_{k \neq j} A_{kj}, & i = j \end{cases}, \quad (4.4)$$

$$B_{ij} = \frac{1-\varepsilon}{4\pi(1+\varepsilon)} \frac{((\mathbf{r}_i-\mathbf{R}_j) \cdot \mathbf{n}_i) S_i}{|\mathbf{r}_i-\mathbf{R}_j|^3}. \quad (4.5)$$

Энергия сольватации вычисляется по формуле (4.6):

$$\Delta G_{pol} = \frac{1}{2} Q^T D q, \quad (4.6)$$

где элементы матрицы D вычисляются из формулы (4.7):

$$D_{ij} = \frac{1}{|\mathbf{R}_i - \mathbf{r}_j|}. \quad (4.7)$$

Итак, чтобы решить электростатическую задачу определения поляризационных зарядов на поверхности диэлектрика, имитирующего растворитель и окружающего молекулу, необходимо построить эту поверхность, которая в неявных моделях растворителя называется поверхностью исключенного (из растворителя) объема или *SES* – Solvent Excluded Surface. В данной части главы для построения этой поверхности используется программа TAGSS [66–68, 71], которая строит и триангулирует *SES* следующим образом. Поверхность строится в две основные стадии: первичная и вторичная обкатка [15, 59], а затем генерирование сетки триангуляции с использованием параметров, полученных на первой стадии. Молекула

представляется совокупностью жёстких сфер, центры которых расположены на ядрах составляющих молекулу атомов; радиусы этих сфер различны для атомов разных типов и являются параметрами континуальной модели растворителя [71]. Первичная обкатка — это образование молекулярной поверхности путём перемещения по молекуле пробной сферы, имитирующей молекулу растворителя, так что в любой момент времени этого процесса должна быть как минимум одна точка касания пробной сферы и какой-либо сферы, соответствующей некоторому атому молекулы. Все возможные точки касания сферы первичной обкатки и сфер атомов определяют точки молекулярной поверхности *SES*. Такая модель построения хорошо воспроизводит форму молекулы с точки зрения растворителя, а детали поверхности, фактически, ограничиваются размером молекулы растворителя.

Процедура первичной обкатки иногда, особенно в случае больших молекул, может приводить к нежелательным самопересечениям поверхности и изломам [71]; для их сглаживания применяется процедура вторичной обкатки [59]. Суть метода вторичной обкатки состоит в том, чтобы заменить фрагменты поверхности вблизи изломов и самопересечений на другие гладкие фрагменты, которые определяются путём обкатки небольшой сферой (называемой сферой вторичной обкатки) поверхности *SES* близи изломов. Диаметр пробной сферы вторичной обкатки выбирается так, чтобы он был меньше диаметра пробной сферы первичной обкатки, но больше определенного критического расстояния, зависящего от средней величины дискретного элемента поверхности. Обе стадии обкатки (первичная и вторичная) позволяют получить описание поверхности молекулы в виде наборов координат положения и ориентации сферических и тороидальных фрагментов, а также геометрической связанности этих фрагментов друг с другом. Исходными параметрами, на основе которых производятся все последующие вычисления, являются данные о координатах атомов, их радиусах (в данной работе они взяты из [71]), а также значения радиусов сфер первичной (1.4 Å) и вторичной обкаток (0.4 Å), а также критического расстояния (0.15 Å) — последние три параметра использовались в [71] при определении атомных радиусов. Критическое расстояние определяет некоторое минимальное расстояние, допустимое между двумя близкими, но поверхностно-удалёнными точками поверхности. Близкие точки считаются поверхностно-удалёнными, если минимальное расстояние между ни-

ми, вычисленное вдоль поверхности, велико по сравнению с прямым расстоянием.

Для построения сетки триангуляции поверхности в программе TAGSS используется метод последовательного добавления треугольников, в котором применяется процедура проецирования произвольной точки в объёме на ближайшую точку молекулярной поверхности. В свою очередь, процедура проецирования использует данные о фрагментах поверхности, которые вычисляются заранее. Для описания сетки триангуляции используются следующие основные структуры данных: наборы точек, рёбер и треугольников, где точка – вершина сетки триангуляции, содержит координаты точки, координаты нормали к поверхности, а также список номеров рёбер, выходящих из этой точки и список номеров треугольников, для которых точка является вершиной; ребро – сторона треугольника триангуляции, содержит номера точек-вершин на концах, а также номера треугольников, для которых это ребро является стороной; треугольник – треугольный плоский элемент, аппроксимирующий поверхность между тремя точками поверхности, содержит номера точек-вершин и номера рёбер-сторон.

Когда триангуляция поверхности полностью выполнена, то для каждого ребра существует по два треугольника, поскольку поверхность замкнута и не имеет самопересечений. Но при частично выполненной триангуляции можно говорить о множестве граничных рёбер, таких, что для них определён только один треугольник – с одной стороны, а второй не определён, ещё не построен. Самый простой пример такой ситуации – уединённый треугольник, построенный на трёх точках поверхности. Каждое его ребро может считаться граничным. В программе TAGSS используется такой треугольник в качестве «затравочного» треугольника. Для учёта граничных рёбер используется массив граничных рёбер, который называется граничным кольцом. При частично выполненной триангуляции возможны ситуации, когда граничных колец более чем одно, то есть имеет место множество граничных колец.

Основной алгоритм последовательного добавления треугольников представляет собой последовательность обработки граничных рёбер. Стартует алгоритм с затравочного треугольника, положение которого может быть выбрано произвольно относительно поверхности. На каждом шаге обработки текущего граничного ребра алгоритм использует один из нескольких возможных сценариев,

и, в общем случае, порождает одно или несколько новых граничных рёбер, относимых к новому граничному кольцу. Выбор того или иного сценария зависит от конкретной геометрической конфигурации окружающих текущее ребро соседних элементов триангуляции. В процессе обработки текущего граничного кольца генерируется одно или несколько новых граничных колец, которые будут обработаны на следующем цикле. Циклы обработки граничных колец продолжаются до тех пор, пока не останется ни одного граничного кольца и, следовательно, граничного ребра, что соответствует полностью построенной сетке триангуляции.

Выбор размеров треугольников осуществляется с помощью параметра «шаг сетки», который соответствует характерному размеру треугольника триангуляции и является входным параметром для алгоритма построения сетки. В случае правильного треугольника этот размер совпадает с радиусом описанной вокруг треугольника окружности. Поскольку треугольники всё же не равносторонние, то «шаг сетки» приближенно равен делённому на 1.5 среднему арифметическому от трёх высот треугольника. На основе значения «шага сетки» выстраивается затравочный треугольник, а также определяются новые треугольники в процессе построения сетки. После завершения построения сетки используется метод «утряски» сетки для получения более равномерной сетки, в результате чего получается триангуляционная сетка с высокой равномерностью, в которой большинство треугольников близки к равносторонним. На основании полученных треугольников строятся дискретные элементы, на которых решается матричное уравнение (4.3): каждая вершина треугольника становится центром дискретного элемента \mathbf{r}_i , а площадь S_i этого дискретного элемента равна трети суммы площадей всех треугольников вокруг этой вершины. То есть центры треугольников становятся вершинами дискретных элементов, а вершины треугольников – центрами дискретных элементов. Количество дискретных элементов равно количеству вершин триангуляционной сетки. Большинство дискретных элементов являются шестиугольниками. Площадь одного дискретного элемента в среднем равна $\frac{3\sqrt{3}}{2}$ квадрата «шага сетки». Нормаль \mathbf{n}_i i -ого дискретного элемента является нормалью к точной аналитической поверхности SES , проведенной в центре \mathbf{r}_i дискретного элемента, который также лежит на точной аналитической поверхности SES . В случае, когда треугольник лежит на сферическом элементе поверхности (что верно для большинства треугольников), его площадь считается по формуле площади

сферического треугольника. Для остальных треугольников их площади считаются по формуле площади плоского треугольника. Такие свойства поверхности важны для решения заданных на поверхности различных интегральных и дифференциальных задач, что нашло своё применение, например, в программе DISOLV [71], реализующей несколько неявных моделей растворителя.

4.1.2. Решение уравнения РСМ в программе DISOLV

Для решения СЛАУ (4.3) в программе DISOLV используется одношаговый итерационный метод [87] со специально подобранным начальным решением и набором параметров τ_{n+1} :

$$D_A \frac{q^{(n+1)} - q^{(n)}}{\tau_{n+1}} + Aq^{(n)} = BQ, \quad (4.8)$$

где матрица D_A – диагональная часть матрицы A . Начальное приближение для поляризационных зарядов рассчитывается по следующей формуле:

$$q^{(0)} = \frac{1 + \varepsilon}{\varepsilon} BQ. \quad (4.9)$$

Эта формула (4.9) является следствием уравнения (4.2) в предположении отсутствия взаимного влияния поляризационных зарядов друг на друга, за исключением того, что коэффициент перед суммой изменен, так как суммарный поляризационный заряд должен удовлетворять соотношению (4.10):

$$\sum_i q_i = \frac{1 - \varepsilon}{\varepsilon} \sum_k Q_k. \quad (4.10)$$

Уравнение (4.8) можно переписать в виде:

$$q^{(n+0.5)} = D_A^{-1}(BQ - (A - D_A)q^{(n)}), \quad (4.11)$$

$$q^{(n+1)} = (1 - \tau_{n+1})q^{(n)} + \tau_{n+1}q^{(n+0.5)}. \quad (4.12)$$

Для выбора τ_{n+1} используются следующие числовые значения, предложенные в работе [45]: если $\sum_i q_i^{(n)}$ и $\sum_i q_i^{(n+0.5)}$ находятся по разные стороны от равновесного значения суммарного поляризационного заряда $\frac{1-\varepsilon}{\varepsilon} \sum_k Q_k$, то

$$\tau_{n+1} = \frac{\sum_i q_i^{(n)} - \frac{1-\varepsilon}{\varepsilon} \sum_k Q_k}{\sum_i q_i^{(n)} - \sum_i q_i^{(n+0.5)}}, \quad (4.13)$$

иначе

$$\tau_{n+1} = 0.65. \quad (4.14)$$

Кроме этого, после 6-й итерации τ_{n+1} всегда вычисляется по формуле (4.14).

Итерации останавливаются тогда, когда изменение свободной энергии сольватации становится меньше заданного порога на протяжении трёх итераций подряд. Свободная энергия сольватации рассчитывается на каждой итерации по формуле (4.6).

4.1.3. Решение уравнения РСМ при помощи мультизарядового метода в программе MCBHSOLV

Входными параметрами для мультизарядового метода являются дискретизация поверхности молекулы и соответствующая этой дискретизации матрица A (4.4) (передается программе в виде функции от набора строк и столбцов, которая возвращает подматрицу на их пересечении). Программа MCBHSOLV сама строит иерархическое кластерное дерево для элементов дискретизации (по центрам этих элементов) и использует это дерево для построения \mathcal{H}^2 -разложения плотной матрицы A с требуемой точностью.

Построение \mathcal{H}^2 -разложения, как и матрично-векторное произведение в \mathcal{H}^2 -формате, требует $O(N)$ операций, где N — размер матрицы A (матрица A является квадратной и имеет размер $N \times N$). Именно быстрое построение разложения и еще более быстрые матрично-векторные операции позволяют успешно применять мультизарядовый метод для значительного ускорения вычисления энергий сольватации и десольватации различных молекул. В отличие от программы DISOLV, для решения уравнения (4.3) программа MCBHSOLV использует не одношаговый итерационный метод, а итерационный метод GMRES [51]. Достоинством метода GMRES является то, что решение ищется на Крыловских подпространствах [73], что значительно сокращает количество итераций, необходимых для решения уравнения с требуемой точностью.

Итоговый алгоритм работы MCBHSOLV показан в Алгоритме 8. Для численных экспериментов использовалась последовательная версия программы.

Алгоритм 8 Алгоритм работы программы MCBHSOLV.

Дано: Файл с данными о дискретных элементах поверхности SES ($\mathbf{r}_i, \mathbf{n}_i, s_i$ для каждого элемента поверхности) и атомах (координаты \mathbf{R} и численные значения зарядов Q).

Вычислить: Энергия сольватации молекулы.

- 1: Считывание данных о дискретных элементах поверхности SES и атомах из файла при помощи пакета *h2tools*,
- 2: Построение иерархического кластерного дерева для дискретных элементов поверхности SES (по \mathbf{r}_i , встроено в пакет *h2tools*),
- 3: Вычисление очереди обработки кластерного дерева (встроено в пакет *h2tools*),
- 4: Вычисление правой части уравнения (4.3) умножением соответствующей матрицы B (4.5) на вектор зарядов атомов Q

$$f = BQ,$$

- 5: Вычисление выражения $Q^T D$, необходимого для вычисления полярной составляющей энергии сольватации (4.6)

$$d = Q^T D,$$

- 6: Построение мультизарядового разложения матрицы A (4.4) с 1 итерацией и параметром точности $\tau = 10^{-4}$ (по данному иерархическому кластерному дереву и очереди обработки кластерного дерева, встроено в пакет *h2tools*),
- 7: Решение уравнения (4.3) с вычисленной ранее правой частью при помощи метода GMRES

$$Aq = f,$$

- 8: Вычисление полярной составляющей энергии сольватации (4.6)

$$\Delta G_{pol} = \frac{1}{2}dq.$$

4.1.4. Численное решение уравнения РСМ

В данном разделе приведены результаты расчётов энергий сольватации и десольватации (только их электростатические составляющие) для 22 комплексов белок-лиганд, вычисленные тремя различными методами:

1. Решение интегрального уравнения (4.2) в рамках континуальной модели РСМ. Система линейных уравнений (4.3) решалась при помощи одношагового итерационного метода, описанного в главе 4.1.2, с помощью программы DISOLV, тщательная валидация которой представлена в работе [71]. В настоящей главе обозначается как РСМ.
2. Решение интегрального уравнения (4.2) в рамках континуальной модели РСМ. Матрица A из системы (4.3) аппроксимировалась мультитарядовым методом с параметром относительной точности 10^{-4} , при этом для хранения аппроксимации требовалась дополнительная оперативная память (до 4.5 Гигабайт). Решение строилось при помощи итерационного метода GMRES [51]. В настоящей главе обозначается как МСВН.
3. Вычисление электростатической составляющей взаимодействия молекул с растворителем с помощью эвристического метода Surface Generalised Born [53, 71], реализованного в программе DISOLV. В настоящей главе обозначается как SGB. Высокая скорость вычислений (для каждого атома молекулы вычисляются четыре поверхностных интеграла) при сравнительно высокой точности сделала подобные методы весьма популярными, особенно применительно к макромолекулам [43].

Для каждой молекулы построение поверхности и её триангуляция проводились с помощью программы TAGSS (подпрограмма DISOLV), а заряды атомов вычислялись в соответствии с силовым полем MMFF94 [32]. Трёхмерная структура комплексов бралась из базы данных Protein Data Base [46], идентификаторы комплексов приведены по горизонтальным осям соответствующих рисунков (см. ниже). Структуры белков были получены при помощи удаления из PDB-комплексов всех небелковых включений, водороды на белках были расставлены при помощи программы APLITE [2]. Лиганды также были взяты из PDB-комплексов, расстановка водородов на них осуществлялась с помощью программы Avogadro [3]. В

комплексах и в свободном состоянии положения лигандов получались локальной оптимизацией в вакууме в рамках силового поля MMFF94 (типизация атомов в этом поле проводилась программой APLITE) по всем атомам лигандов (в комплексах – при жестких белках) из начального нативного положения, т.е. из положения лиганда, в закристаллизованном комплексе непосредственно из базы PDB.

Для каждого белка и комплекса белок-лиганд поверхностные приближения были построены с тремя различными шагами сетки (0.3, 0.2 и 0.1 Å), а для каждого лиганда – с шагами 0.3, 0.2, 0.1 и 0.05 Å.

Оказалось, что различие в энергии сольватации между SGB и PCM почти не зависит от шага сетки. Это было учтено при построении графиков: для сравнения точности приводится только результат SGB на сетке с шагом 0.10 Å.

Для рассматриваемых молекул рассчитанные энергии сольватации находятся в диапазонах: для комплексов и белков $\approx (-4000)$ – (-990) ккал/моль, для лигандов $\approx (-160)$ – (-10) ккал/моль, а время счёта CPU (без учета времени построения сеток) при использовании метода PCM программы DISOLV составляет для шага сетки 0.1 Å: для комплексов и белков $\approx (27000)$ – (200000) с, для лигандов $\approx (17)$ – (150) с в зависимости от размера молекулы. Время счёта программы DISOLV обратно пропорционально шагу сетки в 4-ой степени. Так, переход от шага сетки 0.1 к 0.2 Å уменьшает время счёта примерно в 15 раз, а переход от шага сетки 0.2 к 0.3 Å уменьшает время счёта примерно в 5 раз.

На рис. 4.1 для каждого комплекса приведены значения модуля разности между энергиями сольватации комплекса, полученными с помощью метода PCM из программы DISOLV и с помощью метода PCM из программы MCBHSOLV (или с помощью метода SGB из программы DISOLV). Энергии сольватации сравниваются для каждого из трёх значений шага триангуляционных сеток (0.1, 0.2, 0.3 Å). Для разных методов расчёта каждой молекулы использовались одни и те же триангуляционные сетки данного шага. На горизонтальной оси отложены идентификаторы комплексов в базе данных Protein Data Base.

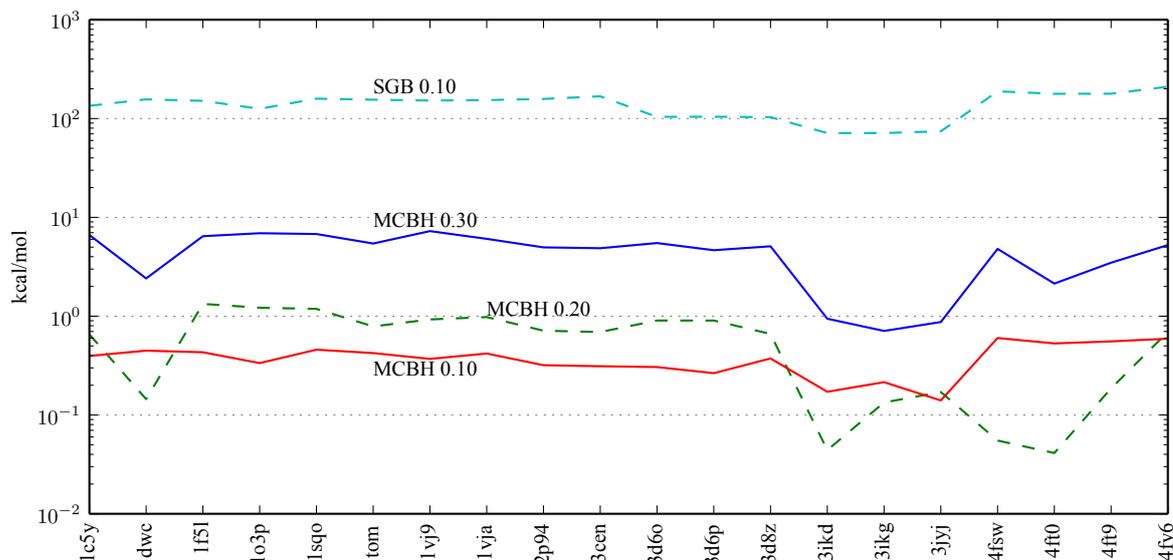


Рисунок 4.1: Модуль разности энергий сольватации, полученных для разных комплексов белок-лиганд с помощью метода РСМ из программы DISOLV и с помощью программы MCBHSOLV (или с помощью метода SGB из программы DISOLV).

Как можно видеть, результаты расчетов энергии сольватации быстрым (подробнее о быстродействии написано ниже) методом MCBH отличаются от результатов расчётов РСМ на величину меньшую 1 ккал/моль для триангуляционных сеток с шагами 0.1 и 0.2 Å (параметр относительной точности расчёта MCBH был задан равным 10^{-4}). Результаты расчётов быстрым эвристическим методом SGB отличаются от результатов РСМ значительно больше – на 100 – 200 ккал/моль. Для энергий сольватации белков была получена аналогичная картина.

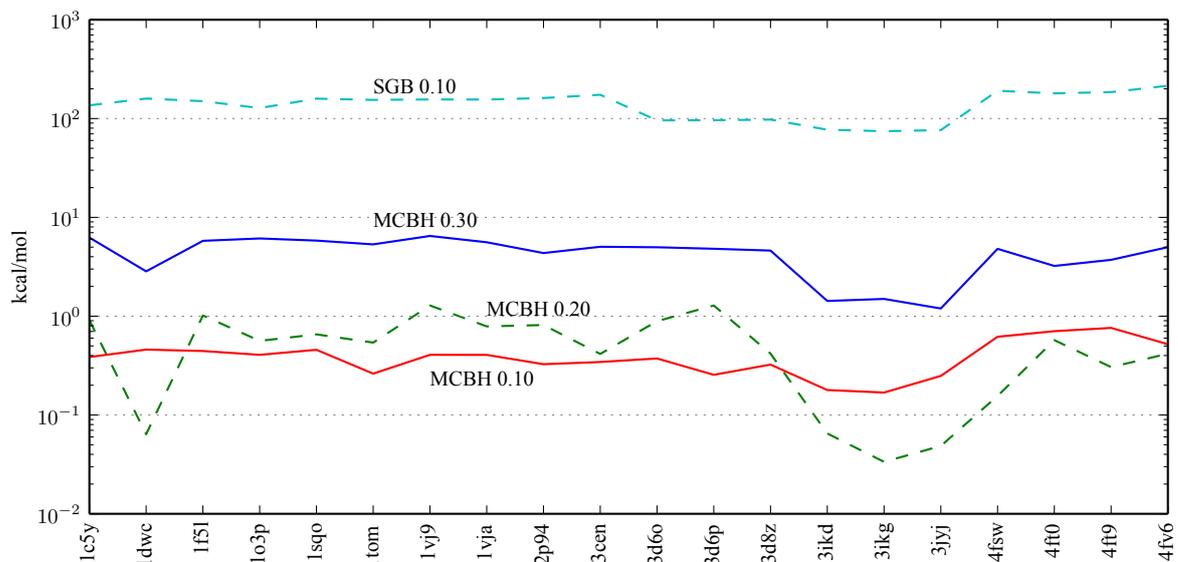


Рисунок 4.2: Модуль разности энергий сольватации, полученных для разных белков с помощью метода PCM из программы DISOLV и с помощью программы MCBHSOLV (или с помощью метода SGB из программы DISOLV).

Для лигандов расхождение значений энергии сольватации для PCM и MCBH (или SGB) значительно меньше: между результатами, полученными с помощью PCM и MCBH для шага сетки 0.05 \AA модуль разности энергий сольватации для лигандов разных комплексов составляет величину $10^{-4} - 10^{-2}$ ккал/моль и даже для шага 0.3 \AA не превосходит 1 ккал/моль. Результаты расчётов методом SGB отличаются от результатов расчетов методом PCM не более, чем на 5 ккал/моль.

На рис. 4.3 для каждого комплекса приведены значения модуля разности между энергиями десольватации комплекса, полученными с помощью метода PCM из программы DISOLV и с помощью метода PCM из программы MCBHSOLV (или с помощью метода SGB из программы DISOLV). Энергии десольватации сравниваются для каждого из трёх значений шага триангуляционных сеток ($0.1, 0.2, 0.3 \text{ \AA}$). Для разных методов расчёта каждой молекулы использовались одни и те же триангуляционные сетки данного шага. На горизонтальной оси отложены идентификаторы комплексов в базе данных Protein Data Base. Как было отмечено ранее, энергия десольватации вычисляется как разность энергии сольватации комплекса и суммы энергий сольватации белка и лиганда.

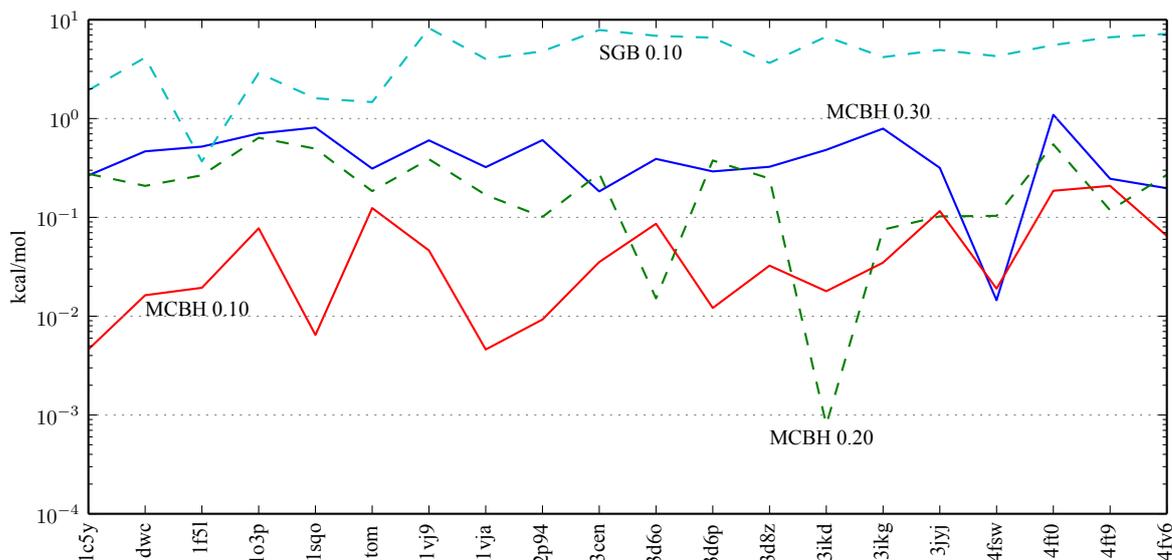


Рисунок 4.3: Модуль разности энергий десольватации, полученных для разных комплексов белок-лиганд с помощью метода PCM из программы DISOLV и с помощью метода PCM из программы MCBHSOLV (или с помощью метода SGB из программы DISOLV).

Как можно видеть, при вычислении энергии десольватации происходит заметная компенсация различий значений энергий сольватации белка, лиганда и их комплекса, полученных разными методами. Так, результаты расчётов методом MCBH отличаются от соответствующих результатов PCM менее, чем на 1 ккал/моль для всех рассмотренных шагов триангуляционных сеток, а для шага 0.1 \AA различие составляет всего $10^{-2} - 10^{-1}$ ккал/моль (параметр относительной точности расчета MCBH был задан равным 10^{-4}). Результаты расчётов энергии десольватации методом SGB отличаются от результатов PCM не более, чем на 10 ккал/моль.

На рис. 4.4 показано ускорение вычисления энергии десольватации при помощи метода MCBH (или SGB) относительно метода PCM на соответствующих сетках (не учитывается время построения поверхности *SES* вокруг молекул).

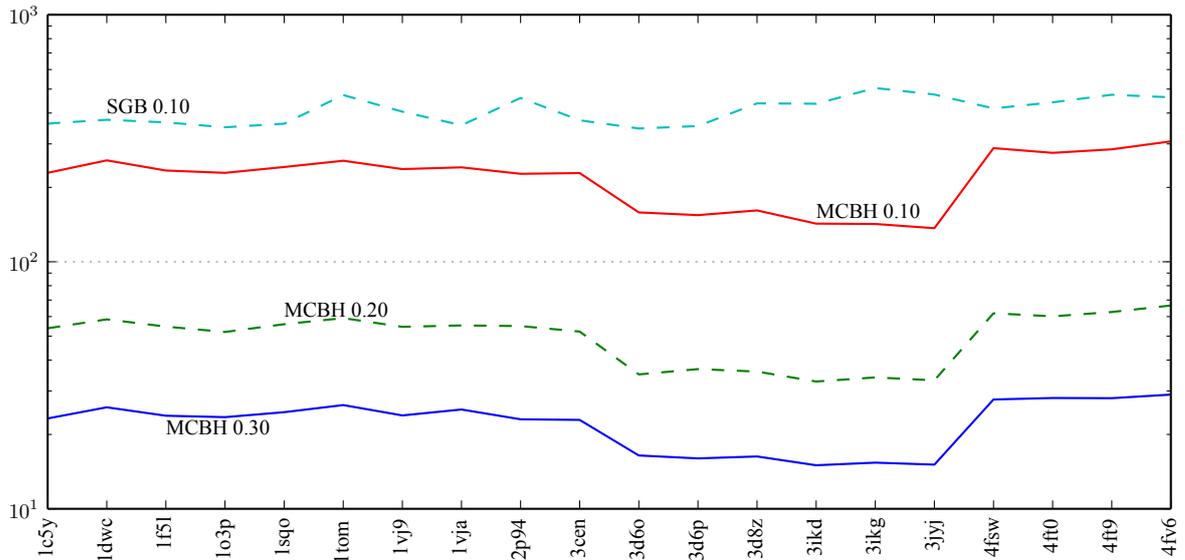


Рисунок 4.4: Ускорение расчетов при помощи метода МСВН относительно метода РСМ на соответствующих сетках.

Из этого рисунка видно, что в методе МСВН получается существенное ускорение по сравнению с методом РСМ на тех же сетках, причем это ускорение тем больше, чем меньше шаг сетки, а значит и существенно больше массив триангуляционных треугольников и размер матриц A и B в уравнении для вычисления поляризационных зарядов. Это ускорение слабо меняется от комплекса к комплексу и для шага сетки 0.1 \AA для отдельных комплексов достигает 300 раз! Столь большое ускорение обусловлено двумя причинами: (i) переходом от одношагового итерационного метода [87] решения матричного уравнения (4.3) в программе DISOLV к гораздо более эффективному итерационному методу GMRES [51] в программе MCBHSOLV и (ii) использование быстрого построения \mathcal{H}^2 -разложения плотной матрицы A и еще более быстрых матрично-векторных произведений в \mathcal{H}^2 -формате. При этом оперативная память, необходимая для хранения \mathcal{H}^2 -разложения матрицы A из уравнения (4.3) для сетки с шагом 0.1 \AA , составляла от 1.5 до 4.5 Гигабайт, в зависимости от размера и формы комплекса. В связи с тем, что базисные подматрицы мультизарядового метода (также называемые матрицы перехода между деревьями источников и приёмников) есть подматрицы исходной матрицы A , их можно не хранить, а вычислять по мере необходимости, что позволяет сократить максимальное требование по оперативной памяти с 4.5 Гигабайт до 0.9 Гигабайт, но увеличивает время, необходимое для

каждого матрично-векторного произведения в \mathcal{H}^2 -формате (в худшем случае это увеличивает полное время работы программы MCBHSOLV в 2 раза). Однако, для численных экспериментов использовалась та версия программы, которая хранит все базисные подматрицы.

При увеличении требуемой точности, возрастет и требуемая память, которую можно сократить за счёт вышеописанного действия (что повлечет за собой увеличение времени расчётов). В результате полученного ускорения для наиболее аккуратных вычислений (шаг сетки 0.1 \AA) вместо 4.7 суток (для комплекса 4fv6), которые необходимы для расчёта энергии десольватации с помощью программы DISOLV методом PCM, программе MCBHSOLV требуется всего около 20 минут для получения практически того же значения энергии десольватации. Ускорение расчета энергии десольватации методом SGB по сравнению с PCM программы DISOLV для шага сетки 0.1 \AA близко к ускорению MCBH для шага сетки 0.1 \AA .

Пользуясь существенным сокращением времени счёта с помощью метода PCM, реализованного в программе MCBHSOLV, исследуем теперь вопрос о величине ошибки решения интегрального уравнения PCM на дискретной поверхности, связанной с произволом расположения на поверхности SES начального элемента триангуляции. Очевидно, при уменьшении шага сетки триангуляции величина энергии сольватации данной молекулы должна стремиться к точному решению уравнения PCM на гладкой поверхности SES и не зависеть от положения триангуляционной сетки на этой поверхности. Для заданного шага сетки величина этой ошибки определяется величиной вариаций значения энергии сольватации, точнее электростатической части энергии взаимодействия молекулы с растворителем, при случайном движении сетки триангуляции по поверхности [71]. Вариации положения сеток на поверхности задавались значениями двух параметров, определяющих ориентации затравочного треугольника триангуляции (см. конец раздела 4.1.1), каждый из которых равномерно пробегал диапазон своих значений – по 10 точек для каждого из этих двух параметров. Таким способом всего для каждого шага сетки для каждой молекулы было получено около 100 сеток триангуляции, на которых и вычислялись энергии сольватации каждой молекулы. Естественно, для заданного шага и одной и той же молекулы при использовании разных сеток получались несколько различные значения энергии сольватации. Проводя усреднение по всем сеткам данного шага, получались средние зна-

чения энергии сольватации для каждой молекулы, и с их помощью вычислялись средние значения энергии десольватации каждого комплекса, представленные на следующем рис. 4.5.

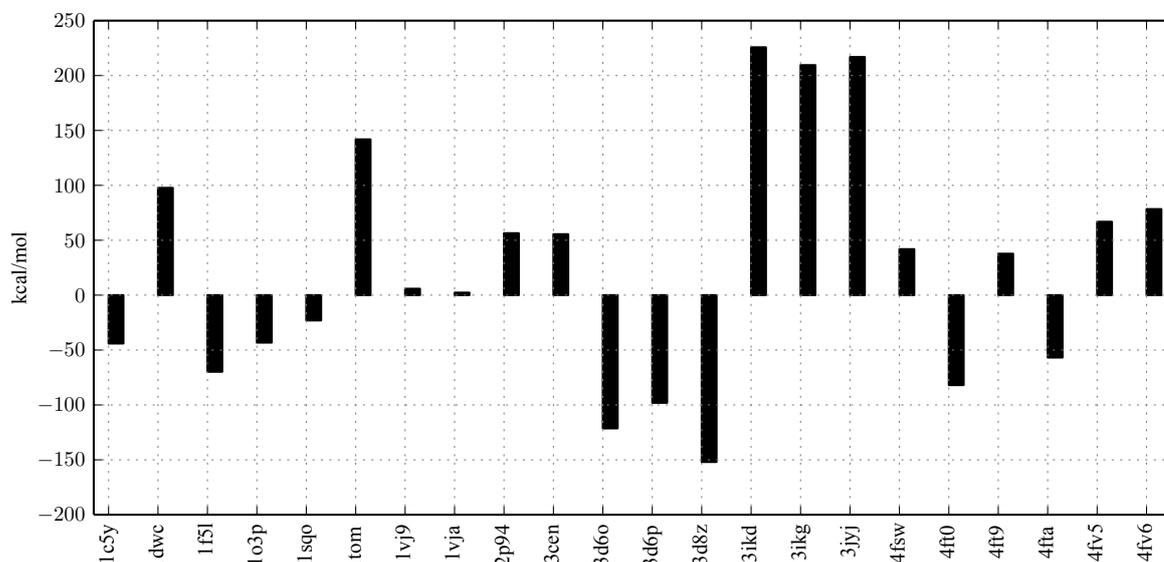


Рисунок 4.5: Средние значения энергии десольватации для каждого из тестовых комплексов на шагах сетки 0.1 \AA . Различие между средними значениями энергии десольватации для разных шагов триангуляционной сетки для каждого комплекса не превосходит 2 ккал/моль.

Как видим, энергии десольватации различных комплексов могут различаться как по знаку, так и по величине от единиц до сотен ккал/моль. Величина ошибки энергии сольватации из-за вариации положения триангуляционной сетки на поверхности *SES* видна из рис. 4.6, на котором представлены среднеквадратичные отклонения энергии сольватации каждого комплекса, полученные для триангуляционных сеток различного шага.

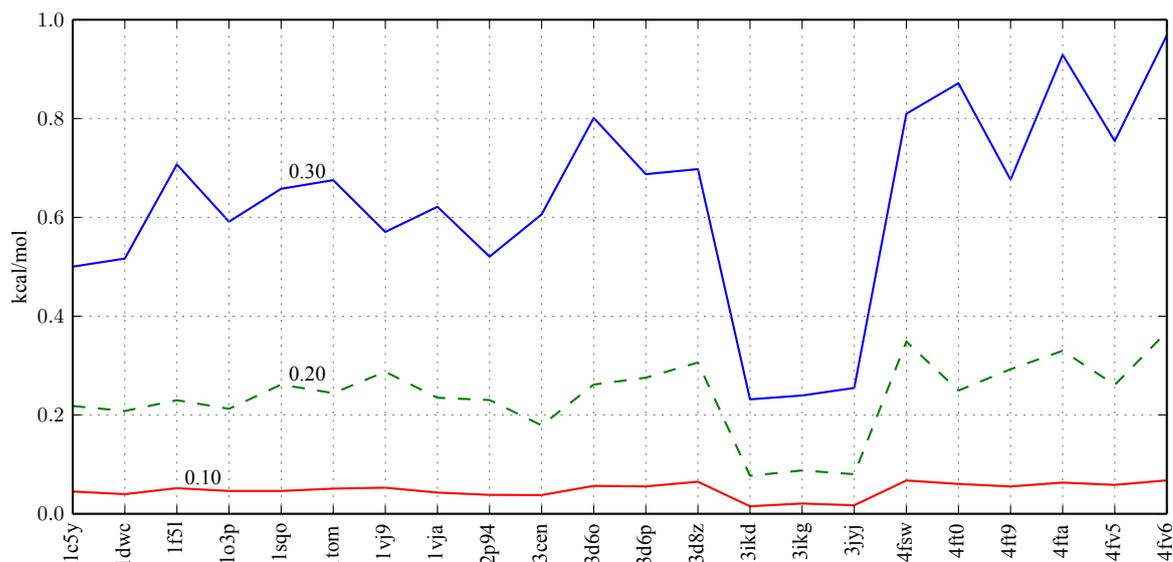


Рисунок 4.6: Среднеквадратичное отклонение энергии сольватации каждого из 22 комплексов для шагов сетки 0.3, 0.2 и 0.1 Å.

Для белков получаются практически такие же величины, что и для комплексов, а для всех лигандов для всех исследованных шагов сеток (0.05, 0.1, 0.2 и 0.3 Å) среднеквадратичное отклонение энергии сольватации не превышает 0.08 ккал/моль.

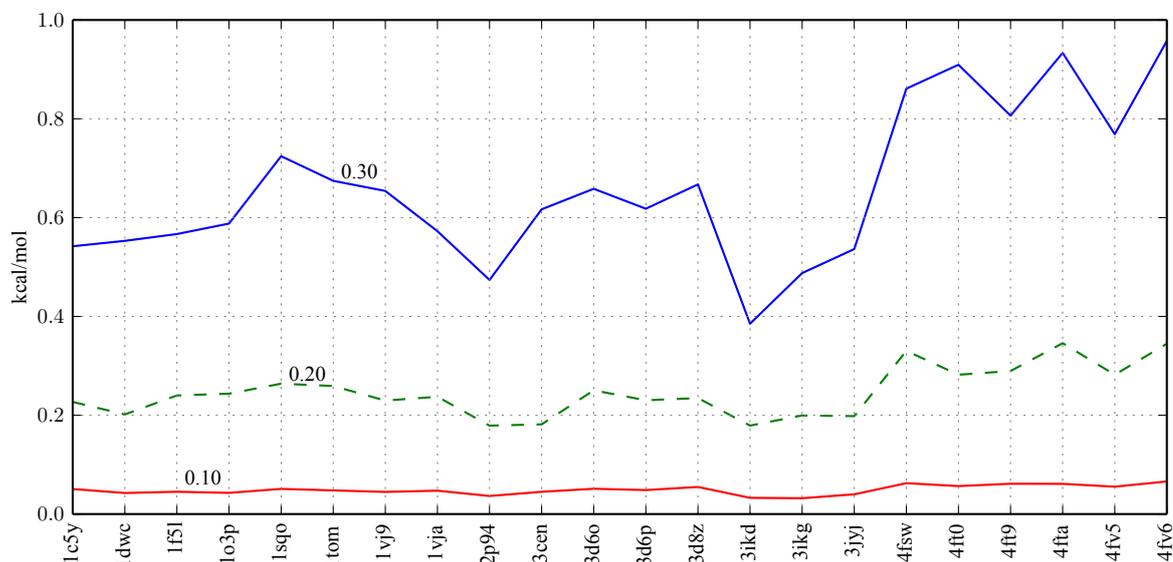


Рисунок 4.7: Среднеквадратичное отклонение энергии сольватации каждого из 22 белков для шагов сетки 0.3, 0.2 и 0.1 Å.

На рис. 4.8 представлены среднеквадратичные отклонения энергии десольватации для каждого комплекса.

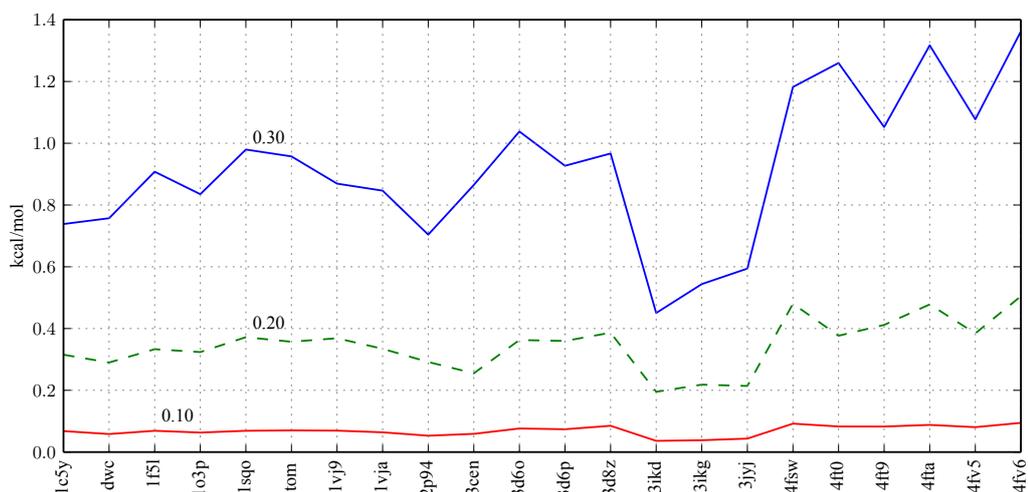


Рисунок 4.8: Среднеквадратичные отклонения энергии десольватации каждого из тестовых комплексов для шагов триангуляционной сетки 0.1, 0.2 и 0.3 Å. Среднеквадратичное отклонение энергии десольватации считалось как корень квадратный из суммы квадратов среднеквадратичных отклонений энергий сольватации для комплекса, белка и лиганда, так как случайные величины (энергии лиганда, белка и комплекса) являются независимыми.

Как видим, неопределенность в вычислении энергии десольватации, обусловленная произволом расположения начального элемента дискретизации, для шага сеток 0.1, 0.2 и 0.3 Å не превосходит 0.1, 0.5 и 1.4 ккал/моль соответственно. Таким образом, программа MCBHSOLV на поверхности с шагом триангуляции 0.2 Å затратит на вычисление энергии десольватации несколько минут, а описанная выше неопределенность получающейся энергии десольватации не превысит 1 ккал/моль. Кроме того, отличие этой энергии десольватации от энергии десольватации из программы DISOLV также не превысит 1 ккал/моль.

Средние значения (по каждой молекуле) среднеквадратичного отклонения энергии (по всем сеткам для каждой молекулы) приведены в табл.4.1:

Таблица 4.1: Усредненные по всем комплексам среднеквадратичные отклонения энергий сольватации лигандов, белков и комплексов, а также энергии десольватации.

Шаг сетки, Å	Энергия сольватации, ккал/моль			Энергия десольватации, ккал/моль
	лиганда	белка	комплекса	
0.30	0.0371	0.6616	0.6313	0.9197
0.20	0.0140	0.2468	0.2384	0.3461
0.10	0.0028	0.0489	0.0480	0.0690
0.05	0.0006	n/a	n/a	n/a

Исследование полученных энергий на нормальность распределения показало, что среднее значение p -value для всех лигандов, белков и комплексов на всех шагах сетки для теста Д'Агостино и Пирсона [17, 63] составляет 0.476. Однако, для того, чтобы отрицать нормальность распределения полученных энергий, вычисленное p -value должно не превосходить 0.05 по значению. Таким образом, с высокой вероятностью распределение энергий подчиняется нормальному закону.

Следует отметить, что изменение среднего значения энергии сольватации с изменением шага сетки заметно превосходит стандартные ошибки этих средних из-за вариаций положения сетки; в данном случае стандартная ошибка среднего составляет 1/10 от среднеквадратичного отклонения энергии сольватации (число сеток было около 100). Порядок этих величин следующий. Различия средних величин энергии сольватации комплексов для шагов сетки 0.3 и 0.1 Å составляет ≈ 10 ккал/моль, а различия на шагах 0.2 и 0.1 Å ≈ 2 ккал/моль, в то время, как стандартные ошибки среднего значения энергии сольватации для комплексов по порядку величины равны 0.14, 0.05 и 0.01 ккал/моль для шагов сетки 0.3, 0.2 и 0.1 Å соответственно. Различия же между средними энергиями десольватации для разных шагов сетки гораздо меньше соответствующих различий для энергий сольватаций. Это связано с тем, что различия между средними энергиями сольватации для комплекса по большей части компенсируются различиями между средними энергиями сольватации свободного белка. Описанный факт свидетельствует о том, что дискретизация интегральных уравнений вносит дополнительную систематическую ошибку в решение интегральных уравнений РСМ, помимо неопределенности, связанной с произвольным расположением начального

элемента триангуляции. Эта систематическая ошибка явно зависит от величины шага триангуляции и, предположительно, связана с отличием плоских элементов дискретизированной поверхности от первоначально гладкой поверхности и с описанием заряда поверхностного элемента как точечного вместо непрерывно распределённого. Выявление закономерностей в поведении этой систематической ошибки требует более детального изучения.

4.1.5. Сравнение с мозаично-скелетным методом

Для сравнения мультизарядового метода с мозаично-скелетным, последний был применён для построения аппроксимации матрицы A из уравнения (4.4). В качестве тестового примера была выбрана молекула, состоящая из 203444 дискретных поверхностных элементов. Значение параметра относительной точности τ было установлено значение 10^{-4} , как и в случае с мультизарядовой аппроксимацией. Время измерено в секундах, память в мегабайтах. Сравнительные результаты приведены в следующей таблице:

Таблица 4.2: Сравнение мультизарядового и мозаично-скелетного методов для построения аппроксимации матрицы A из уравнения (4.4) при параметре относительной точности $\tau = 10^{-4}$.

	MSKEL	MSKEL	МСВН	МСВН	МСВН
N	time	mem	time	fullmem	lowmem
203444	2245с	6882МВ	370с	1488МВ	362МВ

Столбец “MSKEL time” отображает время построения аппроксимации мозаично-скелетным методом,

столбец “MSKEL mem” показывает память, необходимую для хранения мозаично-скелетной аппроксимации,

столбец “МСВН time” отвечает за время построения аппроксимации мультизарядовым методом,

в столбце “МСВН fullmem” указана полная память для хранения мультизарядового аппроксиманта,

в столбце “МСВН lowmem” записана память для хранения мультизарядового аппроксиманта без базисных подматриц (которые можно вычислять по ходу счёта матрично-векторных произведений).

4.1.6. Выводы по разделу «Континуальная модель растворителя»

В настоящем разделе рассматриваются расчёты полярной части энергии взаимодействия молекул с водой в континуальной модели растворителя РСМ – электростатическая составляющая энергии сольватации больших и малых молекул с водой. Рассмотрен способ существенного ускорения решения уравнения для поляризационных зарядов на поверхности SES раздела молекула-растворитель в рамках континуальной модели растворителя. Разработаны новые алгоритмы, основанные на мультизарядовом методе, их программная реализация MCBHSOLV и проведено сравнение результатов и времён счёта с соответствующими величинами, полученными с помощью программы DISOLV [71]. Сравнение проведено на тестовом наборе комплексов белок-лиганд, содержащем различные белки и лиганды. Особенностью этого сравнения является то, что в обеих программах используются одни и те же молекулы, одни и те же молекулярные поверхности и их триангуляционные сетки и одни и те же заряды соответствующих атомов рассмотренных молекул. Сравнение показало, что программа MCBHSOLV работает быстрее программы DISOLV в 100 и более раз практически без потери точности вычислений, причем, чем меньше шаг триангуляции сольватационной поверхности, тем больше ускорение. Ускорение получено применением двух методов. Во-первых, с помощью недавно разработанного мультизарядового метода аппроксимации матриц больших размеров [39], которые хорошо приближаются \mathcal{H}^2 -матрицами [11, 31]. Быстрое построение \mathcal{H}^2 -разложения и еще более быстрые матрично-векторные операции в \mathcal{H}^2 -формате позволяют успешно применить мультизарядовый метод для существенного ускорения решения матричного уравнения РСМ для поляризационных зарядов. Во-вторых, путём перехода от одношагового итерационного метода решения уравнения для поляризационных зарядов в континуальной модели растворителя, реализованного в программе DISOLV [71], к итерационному методу GMRES [51], основным достоинством которого являет-

ся то, что решение ищется на Крыловских подпространствах [73], и это приводит к значительному сокращению количества итераций, необходимых для решения уравнения для поляризованных зарядов с требуемой точностью.

Столь большое ускорение расчетов позволило достаточно подробно исследовать зависимость энергии взаимодействия зарядов молекул с поляризованными зарядами на поверхности раздела молекула-растворитель от шага триангуляционной сетки и неопределенность этой энергии из-за произвола расположения начального элемента триангуляции для тестового набора комплексов белок-лиганд. Было найдено, что подобная неопределенность энергии десольватации – величины, через которую учёт растворителя непосредственно входит в энергию связывания белок-лиганд, составляет, в среднем, 0.92, 0.35 и 0.07 ккал/моль для шагов сетки 0.3, 0.2 и 0.1 Å соответственно. Отличие в результатах по энергии десольватации в модели РСМ на идентичных сетках между программами MCBHSOLV и DISOLV не превосходят ≈ 1.0 , ≈ 0.6 и ≈ 0.2 ккал/моль для шагов сетки 0.3, 0.2 и 0.1 Å соответственно. Таким образом, для вычисления энергии десольватации с приемлемой точностью можно использовать программу MCBHSOLV с шагом сетки 0.2 Å и тратить при этом всего несколько минут на комплекс.

Заключение

Диссертация посвящена блочно-малоранговым матрицам, возникающим при решении различных физических задач. Особое внимание уделено специальным блочно-малоранговым структурам: \mathcal{H} - и \mathcal{H}^2 -матрицам. Алгебраические методы построения приближений матриц \mathcal{H} -матрицами основаны на малоранговых разложениях матриц. Например, в мозаично-скелетном методе для этого используется скелетная или крестовая аппроксимации. Структура \mathcal{H}^2 -матриц позволяет использовать фактор малоранговости подматриц на порядок эффективнее, однако до настоящего времени не было предложено ни одной эффективной реализации алгебраического метода приближения матриц \mathcal{H}^2 -матрицами, обладающей линейной или квазилинейной сложностью от размера матрицы (количество строк или столбцов).

«Мультизарядовый» метод, являющийся основным результатом диссертации, позволяет строить \mathcal{H}^2 -приближение матрицы в «мультизарядовом» формате. Физический смысл этого формата опирается на следующую гипотезу: для любой достаточно удалённой точки пространства потенциал, созданный кластером заряженных частиц в эту точку пространства, можно приблизить потенциалом, создаваемым относительно небольшим количеством заряженных тел. На матричном уровне это эквивалентно построению приближения матрицы по некоторым из её строк или столбцов. Выбор таких базисных строк и столбцов может быть осуществлен без использования всей матрицы: достаточно выбрать «хорошую» подматрицу. «Мультизарядовый» метод выбирает такие подматрицы итерационно, независимо от поставленной физической задачи, из-за чего его можно отнести к классу алгебраических, матричных методов. Количество итераций, необходимых для приближения матрицы с требуемой точностью, зависит лишь от поставленной задачи. Для представленных примеров показано, что это количество составляет 1-2 итерации. Неявным, но значимым, преимуществом «мультизарядо-

вого» метода является количество элементов матрицы, необходимых для построения \mathcal{H}^2 -приближения. Количество необходимых вычислений элементов матрицы составляет $O(N)$, тогда как мозаично-скелетному методу требуется $(N \log N)$ элементов матрицы. Предполагая, что вычисление одного элемента матрицы является тяжеловесной операцией (как, например, вычисление интеграла по дискретному элементу поверхности) получаем дополнительное ускорение относительно мозаично-скелетного метода.

Литература

1. Anderson C.R. An implementation of the fast multipole method without multipoles // SIAM Journal on Scientific and Statistical Computing. — 1992. — Vol. 13, no. 4. — P. 923–947.
2. Application of the docking program SOL for CSAR benchmark / A.V. Sulimov, D.C. Kutov, I.V. Oferkin et al. // *J. Chem. Inf. Model.* — 2013. — Vol. 53, no. 8. — P. 1946–1956.
3. Avogadro – Free cross-platform molecule editor. — http://avogadro.openmolecules.net/wiki/Main_Page (accessed June 29, 2013).
4. Banerjee P.K., Butterfield R. Boundary element methods in engineering science. — McGraw-Hill London, 1981. — Vol. 17.
5. Barnes J., Hut P. A hierarchical $O(N \log N)$ force-calculation algorithm // *Nature*. — 1986. — Vol. 324. — P. 446–449.
6. Bebendorf M., Venn R. Constructing nested bases approximations from the entries of non-local operators // *Numer. Math.* — 2012. — Vol. 121, no. 4. — P. 609–635.
7. Benzi M. Preconditioning techniques for large linear systems: a survey // *Journal of Computational Physics*. — 2002. — Vol. 182, no. 2. — P. 418–477.
8. Benzi M., Meyer C.D., Tuma M. A sparse approximate inverse preconditioner for the conjugate gradient method // *SIAM Journal on Scientific Computing*. — 1996. — Vol. 17, no. 5. — P. 1135–1149.
9. Benzi M., Tuma M. A sparse approximate inverse preconditioner for nonsymmetric linear systems // *SIAM Journal on Scientific Computing*. — 1998. — Vol. 19, no. 3. — P. 968–994.

10. Bordner A.J., Cavasotto C.N., Abagyan R.A. Accurate transferable model for water, n-Octanol, and n-Hexadecane solvation free energies // *J. Phys. Chem. B.* — 2002. — Vol. 106, no. 42. — P. 11009–11015.
11. Börm S. *Efficient numerical methods for non-local operators: \mathcal{H}^2 -matrix compression, algorithms and analysis.* — Eur. Math. Soc., 2010.
12. Brandt A. Multilevel computations of integral transforms and particle interactions with oscillatory kernels // *Computer Physics Communications.* — 1991. — Vol. 65, no. 1. — P. 24–38.
13. Cheng H., Greengard L., Rokhlin V. A fast adaptive multipole algorithm in three dimensions // *Journal of Computational Physics.* — 1999. — Vol. 155, no. 2. — P. 468–498.
14. Coifman R., Rokhlin V., Wandzura S. The fast multipole method for the wave equation: A pedestrian prescription // *Antennas and Propagation Magazine, IEEE.* — 1993. — Vol. 35, no. 3. — P. 7–12.
15. Connolly M.L. Solvent-accessible surfaces of proteins and nucleic acids // *Science.* — 1983. — Vol. 221, no. 4612. — P. 709–713.
16. Cramer C., Truhlar D. Implicit solvation models: equilibria, structure, spectra, and dynamics // *Chem. Rev.* — 1999. — Vol. 99, no. 8. — P. 2161–2200.
17. D'Agostino R., Pearson E.S. Tests for departure from normality. Empirical results for the distributions of b^2 and $\sqrt{b^1}$ // *Biometrika.* — 1973. — Vol. 60, no. 3. — P. 613–622.
18. Darve E. The fast multipole method: numerical implementation // *Journal of Computational Physics.* — 2000. — Vol. 160, no. 1. — P. 195–240.
19. De Hoog F.R., Mattheij R.M.M. Subset selection for matrices // *Linear Algebra and its Applications.* — 2007. — Vol. 422, no. 2. — P. 349–359.
20. Ding H., Karasawa N., Goddard III W.A. Atomic level simulations on a million particles: The cell multipole method for Coulomb and London nonbond interactions // *The Journal of chemical physics.* — 1992. — Vol. 97, no. 6. — P. 4309–4315.

21. Gantmacher F.R. The theory of matrices. — Taylor & Francis, 1960. — Vol. 2.
22. Golub G.H., Van Loan C.F. Matrix computations. — JHU Press, 2012. — Vol. 3.
23. Goreinov S. A., Tyrtyshnikov E. E. The maximal-volume concept in approximation by low-rank matrices // Contemporary Mathematics. — 2001. — Vol. 208. — P. 47–51.
24. Goreinov S. A., Tyrtyshnikov E. E., Zamarashkin N. L. A theory of pseudo-skeleton approximations // [Linear Algebra Appl.](#) — 1997. — Vol. 261. — P. 1–21.
25. Greengard L., Rokhlin V. A fast algorithm for particle simulations // [J. Comput. Phys.](#) — 1987. — Vol. 73, no. 2. — P. 325–348.
26. Greengard L., Rokhlin V. A new version of the fast multipole method for the Laplace equation in three dimensions // *Acta numerica.* — 1997. — Vol. 6. — P. 229–269.
27. Grote M.J., Huckle T. Parallel preconditioning with sparse approximate inverses // *SIAM Journal on Scientific Computing.* — 1997. — Vol. 18, no. 3. — P. 838–853.
28. Hackbusch W. A Sparse Matrix Arithmetic Based on \mathcal{H} -Matrices. Part I: Introduction to \mathcal{H} -Matrices // [Computing.](#) — 1999. — Vol. 62, no. 2. — P. 89–108.
29. Hackbusch W., Börm S. H_2 -matrix approximation of integral operators by interpolation // *Applied Numerical Mathematics.* — 2002. — Vol. 43, no. 1. — P. 129–143.
30. Hackbusch W., Khoromskij B.N. A Sparse \mathcal{H} -Matrix Arithmetic. Part II: Application to Multi-Dimensional Problems // [Computing.](#) — 2000. — Vol. 64, no. 1. — P. 21–47.
31. Hackbusch W., Khoromskij B., Sauter S.A. [On \$\mathcal{H}^2\$ -matrices.](#) — Springer, 2000.
32. Halgren T.A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94 // [J. Comput. Chem.](#) — 1996. — Vol. 17, no. 5-6. — P. 490–519.
33. Ho K.L., Greengard L. A fast direct solver for structured linear systems by recursive skeletonization // [SIAM J. Sci. Comput.](#) — 2012. — Vol. 34, no. 5. — P. A2507–A2532.

34. How to find a good submatrix : Research Report : 08-10 / ICM HKBU ; Executor: S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov et al. — Kowloon Tong, Hong Kong : 2008. — URL: <http://www.math.hkbu.edu.hk/ICM/pdf/08-10.pdf>.
35. How to find a good submatrix / S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov et al. // Matrix Methods: Theory, Algorithms, Applications / Ed. by V. Olshevsky, E. Tyrtyshnikov. — World Scientific, Hackensack, NY, 2010. — P. 247–256.
36. Klamt A., Schuurmann G. COSMO: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient // *J. Chem. Soc., Perkin Trans. 2*. — 1993. — P. 799–805.
37. Linear scaling density functional calculations via the continuous fast multipole method / C.A. White, B.G. Johnson, P.M.W. Gill, M. Head-Gordon // *Chemical Physics Letters*. — 1996. — Vol. 253, no. 3. — P. 268–278.
38. Makino J. Yet another fast multipole method without multipoles—pseudoparticle multipole method // *Journal of Computational Physics*. — 1999. — Vol. 151, no. 2. — P. 910–920.
39. Adaptive nested cross approximation of non-local operators : arXiv preprint : 1309.1773 ; Executor: A. Yu. Mikhalev, I. V. Oseledets : 2013. — URL: <http://arxiv.org/abs/1309.1773>.
40. Multipole method for microstructured optical fibers. I. Formulation / T.P. White, B.T. Kuhlmeij, R.C. McPhedran et al. // *JOSA B*. — 2002. — Vol. 19, no. 10. — P. 2322–2330.
41. Multipole method for microstructured optical fibers. II. Implementation and results / B.T. Kuhlmeij, T.P. White, G. Renversez et al. // *JOSA B*. — 2002. — Vol. 19, no. 10. — P. 2331–2340.
42. Nabors K., White J. FastCap: A multipole accelerated 3-D capacitance extraction program // *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*. — 1991. — Vol. 10, no. 11. — P. 1447–1459.

43. Onufriev A. Continuum electrostatics solvent modeling with the generalized Born model // *Modeling Solvent Environments: Applications to Simulations of Biomolecules*. — WILEY-VCH, 2010. — P. 127–166.
44. Pissanetzky S. *Sparse matrix technology*. — Academic Press, 1984.
45. Pomelli C.S., Tomasi J. A new formulation of the PCM solvation method: PCM-QINTn // *Theor. Chem. Acc.* — 1997. — Vol. 96, no. 1. — P. 39–43.
46. Protein Data Bank. — <http://www.pdb.org/pdb/home/home.do> (accessed June 29, 2013).
47. Rose D.J., Willoughby R.A. et al. *Sparse matrices and their applications*. — Springer, 1972.
48. Saad Y. ILUT: A dual threshold incomplete LU factorization // *Numerical linear algebra with applications*. — 1994. — Vol. 1, no. 4. — P. 387–402.
49. Saad Y. SPARSKIT: a basic tool kit for sparse matrix computations. — 1994.
50. Saad Y. ILUM: a multi-elimination ILU preconditioner for general sparse matrices // *SIAM Journal on Scientific Computing*. — 1996. — Vol. 17, no. 4. — P. 830–847.
51. Saad Y., Schultz M.H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems // *SIAM J. Sci. and Stat. Comput.* — 1986. — Vol. 7, no. 3. — P. 856–869.
52. Sherman J., Morrison W.J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix // *The Annals of Mathematical Statistics*. — 1950. — P. 124–127.
53. Surface generalized Born method: A simple, fast, and precise implicit solvent model beyond the Coulomb approximation / A.N. Romanov, S.N. Jabin, Y.B. Martynov et al. // *J. Phys. Chem. A*. — 2004. — Vol. 108, no. 43. — P. 9323–9327.
54. Tewarson R.P. *Sparse matrices*. — Academic Press New York, 1973. — Vol. 69.

55. Tomasi J., Persico M. Molecular interactions in solution: an overview of method based on continuous distributions of the solvent // [Chem. Rev.](#) — 1994. — Vol. 94, no. 7. — P. 2027–2094.
56. Totrov M., Abagyan R. Rapid boundary element solvation electrostatics calculations in folding simulations: Successful folding of a 23-residue peptide // [Peptide Science](#). — 2001. — Vol. 60, no. 2. — P. 124–133.
57. Tyrtysnikov E. E. Mosaic-skeleton approximations // [Calcolo](#). — 1996. — Vol. 33, no. 1. — P. 47–57.
58. Tyrtysnikov E. E. Incomplete cross approximation in the mosaic-skeleton method // [Computing](#). — 2000. — Vol. 64, no. 4. — P. 367–380.
59. Vorobjev Y.N., Hermans J. SIMS: computation of a smooth invariant molecular surface // [Biophys. J.](#) — 1997. — Vol. 73. — P. 722–732.
60. Woodbury M.A. Inverting modified matrices // Memorandum report. — 1950. — Vol. 42. — P. 106.
61. Ying L., Biros G., Zorin D. A kernel-independent adaptive fast multipole algorithm in two and three dimensions // [J. Comput. Phys.](#) — 2004. — Vol. 196, no. 2. — P. 591–626.
62. The continuous fast multipole method / C.A. White, B.G. Johnson, P.M.W. Gill, M. Head-Gordon // [Chemical physics letters](#). — 1994. — Vol. 230, no. 1. — P. 8–16.
63. d'Agostino R.B. An omnibus test of normality for moderate and large size samples // [Biometrika](#). — 1971. — Vol. 58, no. 2. — P. 341–348.
64. The fast multipole method (FMM) for electromagnetic scattering problems / N. Engheta, W.D. Murphy, V. Rokhlin, M.S. Vassiliou // [Antennas and Propagation, IEEE Transactions on](#). — 1992. — Vol. 40, no. 6. — P. 634–641.
65. Василевский Ю.В., Ольшанский М.А. Краткий курс по многосеточным методам и методам декомпозиции области. — М.: МАКС ПРЕСС, 2007.

66. Жабин С.Н., Сулимов В.Б. Свидетельство № 2006613753 о государственной регистрации программ для ЭВМ. Зарегистрировано в реестре программ для ЭВМ Федеральной службы по интеллектуальной собственности, патентам и товарным знакам 27 октября 2006.
67. Жабин С.Н., Сулимов В.Б. Программа построения доступной растворителю поверхности для произвольных органических молекул и интерактивный просмотр положений лигандов в активных центрах белков // Сборник материалов XIII российского национального конгресса «Человек и лекарство», 3-5 апреля 2006. — 2006. — С. 15.
68. Жабин С.Н., Сулимов В.Б. Построение гладких молекулярных поверхностей с адаптивной триангуляцией: программа TAGSS // Научная визуализация (электронный журнал, Национальный исследовательский ядерный университет «МИФИ», ISSN 2079-3537). — 2011. — Т. 3, № 2. — С. 27–53.
69. Икрамов Х.Д. Разреженные матрицы // Итоги науки и техники. Серия «Математический анализ». — 1982. — Т. 20. — С. 179–260.
70. Компьютерный дизайн лекарственных средств: программа докинга SOL / А.Н. Романов, О.А. Кондакова, Ф.В. Григорьев и др. // Вычислительные методы и программирование. — 2008. — Т. 9, № 2. — С. 64–84.
71. Континуальная модель растворителя: программа DISOLV – алгоритмы, реализация и валидация / О.Ю. Купервассер, С.Н. Жабин, Я.Б. Мартынов и др. // Вычислительные методы и программирование. — 2011. — Т. 12. — С. 246–261.
72. Костомаров Д.П., Фаворский А.П. Вводные лекции по численным методам. — Логос М., 2004.
73. Крылов А.Н. О численном решении уравнения, которым в технических вопросах определяются частоты малых колебаний материальных систем // Известия Российской академии наук. Серия математическая. — 1931. — С. 491–539.

74. Лебедев В.И. Функциональный анализ и вычислительная математика. — М.: ФИЗМАТЛИТ, 2005.
75. Лифанов И.К. Метод сингулярных интегральных уравнений и численный эксперимент в математической физике, аэродинамике, теории упругости и дифракции волн. — ТОО "Янус" М., 1995.
76. Лифанов И.К. Особые интегральные уравнения и методы их численного решения// Учебное пособие. — М.: Макс Пресс, 2006.
77. Лифанов И.К., Тыртышников Е.Е. Теплицевы матрицы и сингулярные интегральные уравнения // Вычислительные процессы и системы. — Т. 72. — Москва : Москва, 1990. — С. 94–278.
78. Марчук Г.И. Методы вычислительной математики. — Наука, 1980.
79. Марчук Г.И., Агошков В.И. Введение в проекционно-сеточные методы: Учебное пособие. — Наука, 1981.
80. Марчук Г.И., Дымников В.П., Залесный В.Б. Математические модели в геофизической гидродинамике и численные методы их реализации. — Гидрометеиздат, 1987.
81. Марчук Г.И., Шайдуров В.В. Повышение точности решений разностных схем. — Наука, 1979.
82. Матричные методы и вычисления: Сборник научных трудов / Под ред. Тыртышникова Е.Е. — М.: Институт Вычислительной Математики РАН, 1999.
83. Методы решения задач математической физики / В.И. Агошков, П.Б. Дубовский, В.П. Шутяев, Г.И. Марчук. — М.:Физматлит, 2002.
84. Писсанецки С., Икрамов Х.Д., Капорин И.Е. Технология разреженных матриц: Пер. с англ. — Мир, 1988.
85. Реализация поддержки параллельных вычислений в программах докинга SOLGRID и SOL / И.В. Офёркин, А.В. Сулимов, О.А. Кондакова, В.Б. Сулимов // Вычислительные методы и программирование. — 2011. — Т. 12, № 1. — С. 205–219.

86. Садовничий В.А., Сулимов В.Б. Суперкомпьютерные технологии в медицине // Суперкомпьютерные технологии в науке, образовании и промышленности / Под ред. Вл.В. Воеводина. В.А. Садовничего, Г.И. Савина. — М: Изд-во Моск. ун-та, 2009. — С. 16–23.
87. Самарский А.А. Введение в численные методы. — СПб.: Лань, 2005. — ISBN: [5-8114-0602-9](#).
88. Самарский А.А., Гулин А.В. Численные методы. — Москва: Наука, 1989.
89. Тихонов А.Н., Самарский А.А. Уравнения математической физики. — Изд-во Моск. ун-та М., 1999.
90. Тыртышников Е.Е. Теплицевы матрицы, некоторые их аналоги и приложения. — Отд. вычисл. математики АН СССР, 1989.
91. Тыртышников Е.Е. Методы численного анализа. — ИЦ Академия М., 2007.
92. Тьюарсон Р., Пейсахович Э.М., Икрамов Х.Д. Разреженные матрицы: Пер. с англ. — Мир, 1977.