Федеральное государственное автономное образовательное учреждение
высшего профессионального образования

Московский физико-технический институт
(государственный университет)

На правах рукописи

Рахуба Максим Владимирович

УДК 519.6

Тензорные методы решения многомерных частичных задач на собственные значения

01.01.07 — Вычислительная математика

ДИССЕРТАЦИЯ
На соискание учёной степени
кандидата физико-математических наук

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

Москва 2017

Оглавление

Введение

Актуальность и объект исследования. Настоящая диссертация посвящена решению многомерных задач на собственные значения. Основным объектом исследования являются *многомерные массивы*, которые естественным образом возникают, например, при дискретизации многомерных дифференциальных уравнений на прямоугольной сетке. Многомерный массив размера

$$\underbrace{n \times \cdots \times n}_{d}$$

задается с помощью n^d чисел. Это означает, что память и количество операций, необходимые для работы с таким массивом, растут экспоненциально с размерностью задачи d, что приводит к значительным затратам вычислительных ресурсов. Несложно оценить, что при $n = 2$ и $d = 300$ количество элементов такого массива равно 2^{300} и превышает оценку числа атомов во Вселенной 10^{80}.

Многомерные задачи на собственные значения возникают в ряде приложений, например, в задачах квантовой химии, базирующихся на решении уравнения Шредингера. Расчет больших молекул с помощью этого уравнения даже на современных суперкомпьютерах может занимать месяцы расчетного времени. Поэтому разработка новых быстрых методов решения многомерных задач на собственные значения является актуальной задачей.

Одним из способов преодоления экспоненциального роста числа параметров с размерностью задачи d является подход *тензорных разложений*, который получил активное развитие в последнее десятилетие. Тензорные разложения позволяют приближать с заданной точностью многомерные массивы с помощью небольшого числа параметров. Важной особенностью тензорного подхода является возможность строить тензорные разложения на основе стандартных алгоритмов вычислительной линейной алгебры. Это позволяет легко

адаптировать алгоритмы к задачам, возникающим в различных приложениях. Однако, даже если мы обладаем априорной информацией о том, что искомое решение может быть представлено с помощью тензорных разложений с небольшим числом параметров, поиск этого решения может оказаться нетривиальной задачей. В частности, существующие тензорные алгоритмы для поиска собственных значений и собственных векторов имеют сильную зависимость от числа параметров разложения, а также от числа собственных значений, которые требуется найти. Поэтому необходимо разрабатывать новые эффективные методы решения этой задачи.

Цель диссертационной работы. Целью настоящей диссертационной работы является разработка новых тензорных методов решения многомерных частичных задач на собственные значения. Под частичной задачей понимается поиск части спектра, например, нескольких минимальных собственных значений.

Научная новизна. Предложены новые методы решения многомерных частичных задач на собственные значения с использованием тензорных разложений. Для случая линейного оператора предложен новый метод нахождения целевого собственного значения, базирующийся на методе Якоби-Дэвидсона и оптимизации на нелинейных многообразиях. Получены результаты о сходимости метода. Также предложен метод ALS II, базирующийся на попеременной оптимизации и методе обратной итерации. Получены оценки локальной сходимости этого метода. Для попеременной минимизации функционалов предложена теория локальной сходимости, показывающая связь метода с мультипликативным методом Шварца. Предложен новый метод поиска нескольких собственных значений, базирующийся на итерационных методах и нелинейном предобуславливателе. На основе предложенных методов получено высокоточное решение уравнения Шредингера для расчета первых 84 колебательных уровней молекулы ацетонитрила.

На примере уравнений Хартри-Фока (ХФ) и Кона-Шэма (КШ) предложен новый метод решения задач на собственные значения с нелинейным оператором. Предложенный метод позволяет с заданной точностью решать уравнения

ХФ и КШ. Для быстрого вычисления возникающих при решении уравнений ХФ и КШ сверток предложен новый быстрый алгоритм.

Практическая значимость работы. Предложенные методы могут быть использованы как для решения многомерных частичных задач на собственные значения, возникающих при дискретизации дифференциальных уравнений, так и для изначально дискретных задач, например, для расчета спектра в модели Гейзенберга, если известно, что решение может быть представлено с помощью тензорных разложений с малым числом параметров. Более того, предложенные методы могут быть использованы для решения задач малой размерности с помощью подхода квантизации [70, 150]. Реализованный программный код может быть адаптирован под конкретную прикладную задачу. Для этого достаточно задать в требуемом формате многомерный оператор, собственные значения которого необходимо найти.

Теоретическая значимость работы заключается в обосновании сходимости предлагаемых методов решения многомерных задач на собственные значения. Также в рамках диссертации разработана теория локальной сходимости метода попеременных направлений (ALS) для минимизации функционалов.

Основные положения, выносимые на защиту. Основным результатом работы являются новые эффективные методы решения многомерных задач на собственные значения, их обоснование, а также применение к нескольким прикладным задачам. На защиту выносятся следующие результаты и положения.

1. Предложено обобщение метода Якоби-Дэвидсона с помощью подходов римановой оптимизации при ограничении на тензорные ранги решений. Получены результаты о сходимости метода.

2. Предложен ALS II метод, базирующийся на попеременной оптимизации и методе обратной итерации. Получены оценки локальной сходимости.

3. Предложена концепция предобуславливания на многообразиях с помощью подхода попеременной оптимизации. Концепция применена к ме-

тоду LOBPCG [76]. С помощью метода проведен высокоточный расчет колебательного спектра молекулы ацетонитрила.

4. Предложен новый тензорный метод решения уравнений Хартри-Фока и уравнений Кона-Шэма, являющихся задачами на собственные значения с нелинейным интегро-дифференциальным оператором.

5. Предложен быстрый алгоритм многомерной свертки в тензорных форматах на основе метода крестовой аппроксимации тензоров, который используется для быстрого вычисления оператора, возникающего в уравнениях Хартри-Фока и Кона-Шэма.

Апробация работы. Результаты диссертационной работы докладывались автором и обсуждались на следующих научных семинарах и конференциях:

- Zurich Colloquium in Applied and Computational Mathematics, Eidgenössische Technische Hochschule Zürich (ETH), 2017, Zürich
- International Conference on Scientific Computation and Differential Equations, University of Bath, 2017, Bath
- Научная конференция “Ломоносов”, 2017, Москва
- 88-th Annual Meeting of the International Association of Applied Mathematics and Mechanics (GAMM), 2017, Weimar
- 5-th workshop on “High dimensional quantum dynamics: challenges and opportunities” 2016, Rostock
- The Workshop “Quantum Dynamics: From Algorithms to Applications”, 2016, Greifswald
- 59-я научная конференция МФТИ, 2016, Москва
- 20-th Conference of the International Linear Algebra Society (ILAS), 2016, Leuven
- 4-th International Conference on Matrix Methods in Mathematics and Applications, Skolkovo Institute of Science and Technology, 2015, Moscow
- Workshop: Low-rank Optimization and Applications, Hausdorff Center for Mathematics, 2015, Bonn

- Workshop on Matrix Equations and Tensor Techniques, EPFL, 2013, Lausanne
- 56-я научная конференция МФТИ, 2013, Москва

Публикации. Основные результаты кандидатской диссертации опубликованы в следующих работах:

1. Работы, опубликованные в изданиях, входящих в перечень рецензируемых научных изданий, индексируемых Web of Science:

   (a) Rakhuba M. V., Oseledets I. V. Calculating vibrational spectra of molecules using tensor train decomposition // The Journal of Chemical Physics. — 2016. — Т. 145. — №. 12. – С. 124101.

   (b) Rakhuba M. V., Oseledets I. V. Fast multidimensional convolution in low-rank tensor formats via cross approximation // SIAM Journal on Scientific Computing. – 2015. — Т. 37. — №. 2. — С. A565-A582.

   (c) Rakhuba M. V., Oseledets I. V. Grid-based electronic structure calculations: The tensor decomposition approach // Journal of Computational Physics. — 2016. — Т. 312. — С. 19-30.

2. Работы, опубликованные в прочих изданиях:

   (a) Rakhuba M. V., Oseledets I. V. Jacobi-Davidson method on low-rank matrix manifolds // arXiv preprint arXiv:1703.09096. — 2017.

   (b) Oseledets I. V., Rakhuba M. V. and Uschmajew A. Alternating least squares as moving subspace correction // arXiv preprint: arXiv:1709.07286. — 2017.

   (c) Rakhuba M. V. Block eigensolvers on low-rank tensor manifolds // Proceedings of the 88-th Annual Meeting of the International Association of Applied Mathematics and Mechanics, Weimar, 2017.

   (d) Рахуба М. В. Малоранговые разложения многомерных массивов и их приложение в расчете колебательного спектра молекул // Сб. тез. конф. “Ломоносов 2017”, Москва, 2017.

   (e) Рахуба М. В., Оселедец И.В. Методы решения многомерных задач на собственные значения на малоранговых тензорных многообразиях

и их приложения в задачах квантовой химии // Тезисы 59-й научной конференции МФТИ, 2016.

(f) Rakhuba M. V. Making block eigensolvers really work in higher dimensions // Proceedings of the 20-th Conference of the International Linear Algebra Society (ILAS), Leuven, 2016.

(g) Рахуба М. В., Оселедец И.В. Быстрый алгоритм вычисления многомерной свертки на основе тензорных аппроксимаций и его применение для расчета электронной структуры молекул // Труды 56-й научной конференции МФТИ, 2013.

Личный вклад автора. Диссертационное исследование является самостоятельным законченным трудом автора. Лично автором была предложена идея малорангового метода Якоби-Дэвидсона [2a], а также идея метода предобуславливания на многообразиях [1a]. Исследования и разработка алгоритмов в работах [1a]–[3a], [2a] осуществлены совместно с И.В. Оселедцем, вклад авторов равнозначен. Теоретический результат в совместной работе [2b] принадлежит соавторам в равной степени. Реализация алгоритмов, а также подготовка численных экспериментов в работах [1a]–[1c], [2a], [2b] были выполнены автором самостоятельно. Постановка задачи в работах [1b], [1c] была выполнена И.В. Оселедцем. Автором совместно с Д.А. Колесниковым были получены оценки сходимости для предлагаемого метода ALS II, вклад авторов равнозначен.

Структура работы. Диссертация состоит из введения, четырех глав и заключения. Полный объем диссертации составляет 167 страниц с 19 рисунками и 12 таблицами. Список литературы содержит 152 наименования.

Содержание работы. В первой главе, “Тензорные вычисления”, вводятся основные понятия и теоремы, используемые в настоящей диссертационной работе. Вводится понятия тензора и тензорных разложений. Приводится математическая постановка задачи о поиске собственных значений и собственных векторов в тензорных форматах. Формулируются основные подходы к решению этой задачи. Эти подходы включают в себя классические итерационные методы с округлением по рангу, оптимизацию на римановых многообразиях, а

также попеременную оптимизацию, учитывающую полилинейную структуру тензорных разложений. Для попеременной оптимизации предложена теория локальной сходимости.

Вторая глава “Многомерные задачи на собственные значения с линейным оператором” посвящена решению частичной задачи на собственные значения с линейным оператором. В этой главе предлагается новый метод поиска целевого собственного значения и соответствующего собственного вектора на основе метода Якоби-Дэвидсона и римановой оптимизации. Предлагается метод ALS II, базирующийся на попеременной оптимизации и методе обратной итерации. Для обоих методов приводятся теоретические оценки сходимости. Далее в главе рассматривается блочная задача на собственные значения и предлагается нелинейный предобуславливатель для итерационных процессов, основанный на идеях попеременной оптимизации. Приводятся результаты расчета колебательного спектра молекулы ацетонитрила и проводится сравнение с известными результатами.

Третья глава “Задача на собственные значения для нелинейного оператора на примере уравнений Хартри-Фока и Кона-Шэма” посвящена разработке метода решения задач на собственные значения с нелинейным оператором. Задача рассматривается на примере уравнений Хартри-Фока и уравнений теории функционала плотности. Предлагается итерационный метод решения задачи и формулы пересчета матрицы Фока без вычисления производных. В качестве тензорного формата используется формат Таккера, приводится описание основных операций в итерационном процессе в этом формате. Приводится расчет для некоторых атомов, молекул, а также кластеров атомов. Проводится сравнение с известными квантовохимическими пакетами программ.

Наиболее вычислительно затратной операцией при решении уравнений Хартри-Фока и Кона-Шэма является вычисление трехмерных интегральных преобразований типа сверток. Четвертая глава “Вычисление многомерной свертки на основе метода крестовой аппроксимации в частотной области” посвящена быстрому алгоритму вычисления свертки в различных тензорных форматах. В этой главе приводится новый метод крестовой аппроксимации, имеющий меньшую сложность, чем существующие аналоги. На основе этого метода предлагается новый алгоритм вычисления многомерной свертки. При-

водятся оценки сложности алгоритма и его численное сравнение с существующими подходами.

Благодарности. Автор диссертации благодарит своего научного руководителя Оселедца Ивана Валерьевича за чуткое руководство с первого курса магистратуры Московского Физико-Технического Института. Творческая атмосфера, которую Ивану Валерьевичу удалось создать в своем научном коллективе, стимулировала автора активно заниматься научной деятельностью. Отдельная благодарность выражается Агошкову Валерию Ивановичу, под опекой которого началась научная деятельность автора. Также автор выражает признательность сотрудникам кафедры Института Вычислительной Математики им. Г. И. Марчука за формирование его как специалиста в области вычислительной математики.

Глава 1

Тензорные вычисления

Все алгоритмы, разработанные в настоящей диссертационной работе основываются на малопараметрических представлениях многомерных массивов, называемых тензорными форматами. В настоящей главе мы опишем используемые нами тензорные разложения и способы работы с ними. Будет также поставлена задача о поиске собственных значений в тензорных форматах и описаны основные подходы к ее решению. В частности, будет приведено описание итерационных методов с округлением по рангу, ALS подхода, а также методов римановой оптимизации, использующихся в рамках настоящей диссертации. Для ALS минимизации предлагается новая теория сходимости, показывающая взаимосвязь ALS подхода с мультипликативным методом Шварца и кривизной многообразия малоранговых тензоров.

1.1 Тензорные форматы

Под *тензором* в настоящей диссертации мы понимаем многомерный массив чисел. Тензоры мы будем обозначать большими курсивными буквами, например, $\mathcal{X}$ и будем рассматривать вещественные тензоры

$$\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}.$$

Элемент тензора $\mathcal{X}$ в позиции $(i_1, \ldots, i_d)$ будем обозначать как $\mathcal{X}(i_1, \ldots, i_d)$ или $\mathcal{X}_{i_1,\ldots,i_d}$. Число индексов d называется *размерностью* тензора. Индексы i_k изменяются от 1 и до n_k, если не сказано обратного (нумерация с нуля используется в Главе 4 для удобства работы со свертками), n_k называется *размером моды*.

Отметим, что количество элементов тензора при $n_1 = \cdots = n_d$ равно n^d и растет экспоненциально с размерностью задачи. Это приводит к тому, что уже в трехмерном случае проблематично использовать сетки с большим числом узлов по каждому из направлений. Для решения этой проблемы мы будем использовать *тензорные форматы*, которые базируются на идее *разделения переменных*. В дискретном случае двумерным аналогом идеи разделения переменных у матрицы $\mathcal{X}$ является скелетное разложение

$$\mathcal{X}_{ij} = \sum_{\alpha=1}^{r} u_{i\alpha} v_{j\alpha} \quad \text{или} \quad \mathcal{X} = UV^T,$$

где r – ранг матрицы $\mathcal{X}$, а столбцы матриц U и V составлены из $u_{:,\alpha}$ и $v_{:,\alpha}$ соответственно. На практике часто встречаются матрицы, которые имеют малый ранг только приближенно. Это означает, что существует представление

$$\|\mathcal{X} - UV^T\| \leq \varepsilon$$

для некоторого малого ε. Нахождение наиболее точного приближения заданного ранга зависит от выбора используемой нормы $\|\cdot\|$. В случае использования спектральной или Фробениусовой норм решение такой задачи дает классическая теорема Эккарта-Янга [34]. Согласно этой теореме минимум достигается на усеченном сингулярном разложении (SVD).

Отметим, что для хранения скелетного разложения необходимо только хранить матрицы U и V, содержащие $(n_1+n_2)r$ элементов, вместо $n_1 n_2$ элементов исходной матрицы. Однако для нахождения U, V с помощью SVD необходимо использовать все элементы матрицы, а сложность алгоритма равняется $\mathcal{O}(n_1 n_2 \min(n_1, n_2))$ [152]. То есть применение метода ограничивается небольшим размером матриц. Если матрица является разреженной или допускает быстрое умножение на вектор, то можно использовать итерационные методы поиска приближенного SVD разложения. Альтернативным подходом является метод крестовой аппроксимации [10, 135], “жадно” выбирающий небольшое количество элементов матрицы и строящий по ним малоранговое приближение.

В многомерном случае эффективное сжатие массивов является еще более актуальной задачей. Для сжатия многомерных массивов мы также бу-

дем использовать идею разделения переменных. Наиболее простым способом обобщить эту идею является каноническое разложение (CP формат или CANDECOMP/PARAFAC разложение), которое было предложено в работе [52] в 1927 году. Говорят, что тензор $\mathcal{X}$ представляется в каноническом формате, если он может быть записан в виде следующего разложения

$$\mathcal{X}(i_1,\ldots,i_d) = \sum_{\alpha=1}^{r} U_1(i_1,\alpha)\ldots U_d(i_d,\alpha),$$

где минимально возможное r при котором достигается равенство называется *каноническим рангом*. У CP-разложения есть серьезный недостаток: не существует устойчивых алгоритмов вычисления такого разложения для $d > 2$ [145]. Однако если известна аппроксимация тензора с небольшим значением ранга r, то существует большое число быстрых алгоритмов вычисления базовых операций линейной алгебры [15, 69, 47, 48, 73, 107, 16].

Формат Таккера [133, 144, 143, 73] является другим классическим примером тензорного разложения. Говорят, что тензор $\mathcal{X}$ представляется в формате Таккера [133], если он может быть записан в виде

$$\mathcal{X}(i_1,\ldots,i_d) = \sum_{\alpha_1,\ldots,\alpha_d} G(\alpha_1,\ldots,\alpha_d)\, U_1(i_1,\alpha_1)\ldots U_d(i_d,\alpha_d), \tag{1.1}$$

где α_k изменяется от 1 до r_k или в другой форме

$$\mathcal{X} = G \times_1 U_1 \cdots \times_d U_d,$$

где $\times_k$ означает произведение тензора на матрицу по k-й моде:

$$(G \times_k U_k)(\alpha_1,\ldots,i_k,\ldots,\alpha_d) = \sum_{i_k=1}^{r_k} G(\alpha_1,\ldots,\alpha_k,\ldots,\alpha_d) U_k(i_k,\alpha_k).$$

Минимальное r_k, необходимое для представления $\mathcal{X}$ в виде (1.1) называется рангом k-й моды. Тензор G называется ядром разложения, а U_k называются Таккеровскими факторами. Для разложения Таккера существуют устойчивые алгоритмы, основанные на сингулярном разложении, однако оно содержит $\mathcal{O}(r^d + dnr)$, $r = \max r_k$ элементов и, следовательно, число параметров все еще растет экспоненциально по d.

При больших d используются другие устойчивые тензорные форматы, такие как тензорный поезд (tensor train, TT) [108, 104] или иерархический формат Таккера (hierarchical Tucker, HT) [49, 43]. В отличии от формата Таккера, они не подвержены “проклятью размерности”. Рассмотрим TT-формат подробнее. Говорят, что тензор $\mathcal{X}$ записан в TT формате [149, 108, 104], если он представляется в форме

$$\mathcal{X}(i_1,\dots,i_d) = \sum_{\alpha_0,\dots,\alpha_d} X^{(1)}(\alpha_0,i_1,\alpha_1)\,X^{(2)}(\alpha_1,i_2,\alpha_2)\dots X^{(d)}(\alpha_{d-1},i_d,\alpha_d). \tag{1.2}$$

В (1.2) X_k имеют размеры $r_{k-1}\times n_k\times r_k$ и называются *TT-ядрами*, причем $r_0 = 1$ и $r_d = 1$. Минимально возможные числа r_k называются *TT-рангами*. Мы будем обозначать вектор с компонентами r_k за $\mathbf{r} = \{r_1,\dots,r_{d-1}\}$ и называть TT-рангом тензора. Также мы будем использовать обозначение TT-rank$(\mathcal{X}) = \mathbf{r}$. Разложение (1.2) может быть также записано в форме произведения матриц (Matrix Product State, MPS)

$$\mathcal{X}(i_1,\dots,i_d) = X^{(1)}(i_1)X^{(2)}(i_2)\dots X^{(d)}(i_d),$$

где $X^{(k)}(i_k)$ являются $r_{k-1}\times r_k$ матрицами, которые зависят от параметра i_k. Стоит упомянуть, что MPS представление является алгебраически эквивалентным TT-формату, и долгое время использовалось в квантовой теории информации и физике твердого тела для аппроксимации некоторых волновых функций [141, 110], детальное изложение можно найти в [125].

Аналогичным образом определяется TT разложение для операторов. Рассмотрим оператор $\mathcal{A}$:

$$\mathcal{A} : \mathbb{R}^{n_1\times\dots\times n_d} \to \mathbb{R}^{n_1\times\dots\times n_d},$$

действие которого на вектор $\mathcal{X}\in\mathbb{R}^{n_1\times\dots\times n_d}$ определяется следующим образом:

$$(\mathcal{A}\mathcal{X})(i_1,\dots,i_d) = \sum_{j_1,\dots,j_d} \mathcal{A}(i_1,\dots i_d,j_1,\dots,j_d)\mathcal{X}(j_1,\dots,j_d).$$

Определим TT разложение такого оператора следующим образом:

$$\mathcal{A}(i_1,\dots,i_d,j_1,\dots,j_d) = A^{(1)}(i_1,j_1)\dots A^{(d)}(i_d,j_d),$$

где TT-ядра $A^{(k)}(i_k,j_k)\in\mathbb{R}^{R_{k-1}\times R_k}$, $R_0 = R_d = 1$. Это представление содержит $\mathcal{O}(dn^2R^2)$ степеней свободы, где $R = \max_k R_k$.

Отметим, что в настоящей диссертации будут использованы TT разложение и разложение Таккера. Например, в Главе 2 для линейной задачи на собственные значения используется TT разложение. В Главе 3 задача на собственные значения рассматривается на примере трехмерного уравнения. Поэтому используется формат Таккера, который в трехмерном случае содержит меньше параметров, чем TT разложение. Для обоих разложений существуют программные пакеты, содержащие базовые операции над тензорами. Для реализации численных экспериментов в настоящей диссертации используются следующие программные пакеты:

- tucker3d, доступный по ссылке https://github.com/rakhuba/tucker3d. Этот комплекс программ полностью разработан автором настоящей диссертации. Он содержит основные арифметические операции с тензорами в формате Таккера, а также метод крестовой аппроксимации и алгоритм вычисления свертки, предложенные в настоящей диссертационной работе.

- ttpy, доступный по ссылке https://github.com/oseledets/ttpy. Этот комплекс разработан в группе И.В. Оселедца. Программный комплекс содержит основные арифметические операции в TT формате, а также метод крестовой аппроксимации, решение линейных систем, основной функционал Римановой оптимизации.

1.2 Тензорная арифметика

Если тензоры заданы в некотором малоранговом формате, тогда основные операции линейной алгебры, например, сложение двух тензоров может быть реализовано без вычисления всех элементов тензоров. Мы будем активно использовать это свойство в итерационных процессах, которые состоят из набора простых операций. Приведем описание вычисления тех операций, которые нам потребуются для формата тензорного поезда (TT) и формата Таккера.

Интерполяция тензора

Для начала рассмотрим вопрос о том, как получить малопараметрическое представление тензора. Для разложения Таккера, и для TT-разложения существуют алгоритмы, базирующиеся на последовательности SVD разложений, которые позволяют с заданной точностью получить тензорную аппроксимацию. Однако проблема заключается в том, что даже при небольших d тензор может не помещаться в память компьютера, а вычисление SVD разложения является затратной процедурой. Поэтому для избежания формирования всех элементов тензора мы предполагаем, что тензор задан в виде функции, которая вычисляет его любой наперед заданный элемент. Пусть также известно, что тензор имеет приближенно малый ранг. В случае $d = 2$ можно выбрать r столбцов и r строк так, что на их пересечении матрица имеет максимальный возможный объем (модуль детерминанта). В таком случае, по этим элементам можно построить интерполянт, приближающий исходную матрицу [42, 148].

Для многомерных разложений существуют аналогичные интерполяционные формулы, причем асимптотически количество требуемых элементов совпадает с числом параметров разложения. Метод крестовой аппроксимации для формата Таккера со сложностью $\mathcal{O}(nr^3)$ был предложен в работе [106]. Интерполяционная формула и оценки ошибки были также рассмотрены в [147]. В Главе 4 будет приведена новая версия метода крестовой аппроксимации, имеющая меньшую сложность $\mathcal{O}(nr^2 + r^4)$, благодаря которой удалось предложить быстрый алгоритм вычисления многомерной свертки. Метод крестовой аппроксимации для формата тензорного поезда был предложен в работе [109].

Важно отметить, что метод крестовой аппроксимации может быть также полезен для некоторых операций тензорной арифметики. Например, в ряде приложений возникает необходимость вычисления поэлементных функций от тензора. Так, в уравнении Кона-Шэма необходимо вычислять кубический корень и логарифм из электронной плотности. Пусть известно, что результат действия такой функции можно приблизить малоранговым тензором. В таком случае, найти малопараметрическое представление тензора, не вычисляя полностью всех его элементов, можно с помощью метода крестовой аппроксимации. Сначала по тензорному разложению вычисляются некоторые элементы

тензора. Затем от этих элементов берется рассматриваемая функция. На основе полученных значений метод крестовой аппроксимации выбирает, какие элементы тензора вычислять следующими. Процесс повторяется пока новые элементы, которые выбирает метод, не приближаются с достаточной точностью найденным к этому моменту разложением.

Округление

В ряде случаев можно уменьшать размер разложения с требуемой точностью или с требуемым значением ранга. Такая операция называется *округлением* тензорного разложения. Операция округления имеет смысл, например, при использовании итерационных процессов, когда с тензорами производится большое число арифметических операций. Действительно, после сложения двух тензоров результат может быть явным образом записан в тензорном формате, однако с рангами равными сумме рангов слагаемых. В случае поэлементного умножения ранг результата равен произведению рангов. Поэтому для избежания роста размера разложения необходимо использовать операцию округления. Приведем описание операции округления для форматов Таккера и тензорного поезда.

Для формата Таккера существует устойчивая операция округления, основанная на SVD разложении, которая позволяет уменьшить ранг с требуемой точностью. Рассмотрим тензор $\mathcal{X}$ с рангами равными r:

$$\mathcal{X} = G \times_1 U \times_2 V \times_3 W.$$

Для округления этого тензора сначала делается QR разложение каждого из факторов:

$$U = Q_u R_u, \quad V = Q_v R_v, \quad W = Q_w R_w.$$

В этом случае получаем

$$\mathcal{X} = R \times_1 Q_u \times_2 Q_v \times_3 Q_w, \quad R = G \times_1 R_u \times_2 R_v \times_3 R_w.$$

Затем вычисляем разложение Таккера от R с требуемой точностью

$$R \approx \hat{G} \times_1 \tilde{U} \times_2 \tilde{V} \times_3 \tilde{W}.$$

После этого шага получаем тензор $\hat{G}$ с рангами $\hat{r} \leqslant r$. Таким образом,

$$\mathcal{X} \approx \hat{G} \times_1 \hat{U} \times_2 \hat{V} \times_3 \hat{W}, \quad \hat{U} = U\tilde{U}, \quad \hat{V} = V\tilde{V}, \quad \hat{W} = W\tilde{W}.$$

В формате тензорного поезда процедура округления выглядит следующим образом. Рассмотрим *развертку* $\mathcal{X}_{(1)}$ тензора $\mathcal{X}$, имеющую размер $n_1 \times n_2 \dots n_d$, которая определяется как

$$\mathcal{X}_{(1)}(i_1, \overline{i_2 \dots i_d}) = \mathcal{X}(i_1, i_2, \dots, i_d).$$

Запишем ее с помощью скелетного разложения:

$$\mathcal{X}_{(1)} = UV^\top.$$

Далее необходимо вычислить QR разложение матриц U и V:

$$U = Q_u R_u, \quad V = Q_v R_v.$$

Следовательно,

$$\mathcal{X} = Q_u R_u R_v^\top Q_v^\top.$$

В итоге, для уменьшения ранга мы вычисляем сингулярное разложение $R_u R_v^\top$ и “обрезаем” сингулярные числа с заданной точностью. Отметим, что вычисление QR разложения матрицы $V \in \mathbb{R}^{n_2 \dots n_d \times r}$ может быть эффективно выполнено с помощью последовательности QR разложений для TT ядер исходного тензора. Сложность операции округления в TT формате равняется $\mathcal{O}(dnr^3)$ [104].

Сложение

Рассмотрим $\mathcal{Z} = \mathcal{X} + \mathcal{Y}$, где $\mathcal{X}$ и $\mathcal{Y}$ заданы в формате Таккера $\mathcal{X} = G^{(\mathcal{X})} \times_1 U^{(\mathcal{X})} \times_2 V^{(\mathcal{X})} \times_3 W^{(\mathcal{X})}$ и $\mathcal{Y} = G^{(\mathcal{Y})} \times_1 U^{(\mathcal{Y})} \times_2 V^{(\mathcal{Y})} \times_3 W^{(\mathcal{Y})}$. Результат $\mathcal{Z}$ также записывается в формате Таккера, но с рангом, равным сумме рангов слагаемых:

$$\mathcal{Z} = G^{(\mathcal{Z})} \times_1 U^{(\mathcal{Z})} \times_2 V^{(\mathcal{Z})} \times_3 W^{(\mathcal{Z})}$$

где

$$U^{(\mathcal{Z})} = \left[U^{(\mathcal{X})} | \, U^{(\mathcal{Y})}\right], \qquad V^{(\mathcal{Z})} = \left[V^{(\mathcal{X})} | \, V^{(\mathcal{Y})}\right], \qquad W^{(\mathcal{Z})} = \left[W^{(\mathcal{X})} | \, W^{(\mathcal{Y})}\right]$$

и

$$g^{(\mathcal{Z})}_{\alpha\beta\gamma} = \begin{cases} g^{(\mathcal{X})}_{\alpha\beta\gamma} & \text{для} \quad 1 \leqslant \alpha, \beta, \gamma \leqslant r, \\ g^{(\mathcal{Y})}_{\alpha\beta\gamma} & \text{для} \quad r < \alpha, \beta, \gamma \leqslant 2r, \\ 0 & \text{иначе.} \end{cases}$$

Для избежания роста ранга необходимо использовать округление.

Приведем теперь алгоритм сложения тензоров в ТТ формате. Пусть даны 2 тензора $\mathcal{X}$ и $\mathcal{Y}$ с ТТ-рангами $\mathbf{r}_1$, $\mathbf{r}_2$

$$\mathcal{X}(i_1, \ldots, i_d) = X^{(1)}(i_1) \ldots X^{(d)}(i_d),$$
$$\mathcal{Y}(i_1, \ldots, i_d) = Y^{(1)}(i_1) \ldots Y^{(d)}(i_d),$$

ядра суммы $\mathcal{Z} = \mathcal{X} + \mathcal{Y}$ могут быть записаны как [104]

$$Z^{(k)}(i_k) = \begin{bmatrix} X^{(k)}(i_k) & 0 \\ 0 & Y^{(k)}(i_k) \end{bmatrix}, \quad k = \overline{2, d-1}$$

и

$$Z^{(1)}(i_1) = \begin{bmatrix} X^{(1)}(i_1) & Y^{(1)}(i_1) \end{bmatrix}, \quad Z^{(d)}(i_d) = \begin{bmatrix} X^{(d)}(i_d) \\ Y^{(d)}(i_d) \end{bmatrix},$$

Таким образом, тензор $\mathcal{Z}$ может быть явно записан в виде ТТ-тензора с рангами $\mathbf{r}_1 + \mathbf{r}_2$.

Поэлементное (адамарово) произведение

Рассмотрим вычисление поэлементного (адамарова) произведения $\mathcal{Z} = \mathcal{X} \circ \mathcal{Y}$. Используем обозначения $U = \{\mathbf{u}_\alpha\}_{\alpha=1}^r$, $V = \{\mathbf{v}_\alpha\}_{\alpha=1}^r$, $W = \{\mathbf{w}_\alpha\}_{\alpha=1}^r$, тогда

$$\mathcal{Z} = G^{(\mathcal{Z})} \times_1 U^{(\mathcal{Z})} \times_2 V^{(\mathcal{Z})} \times_3 W^{(\mathcal{Z})},$$

где

$$G^{(\mathcal{Z})} = G^{(\mathcal{X})} \otimes G^{(\mathcal{Y})},$$

$$U^{(\mathcal{Z})} = \{\mathbf{u}^{(\mathcal{X})}_\alpha \circ \mathbf{u}^{(\mathcal{Y})}_\beta\}^r_{\alpha,\beta=1}, \; V^{(\mathcal{Z})} = \{\mathbf{v}^{(\mathcal{X})}_\alpha \circ \mathbf{v}^{(\mathcal{Y})}_\beta\}^r_{\alpha,\beta=1}, \; W^{(\mathcal{Z})} = \{\mathbf{w}^{(\mathcal{X})}_\alpha \circ \mathbf{w}^{(\mathcal{Y})}_\beta\}^r_{\alpha,\beta=1}.$$

Адамарово произведение двух тензоров $\mathcal{X}$ и $\mathcal{Y}$, заданных в ТТ-формате может быть записано через кронекерово произведения ядер

$$(\mathcal{X} \circ \mathcal{Y})_{i_1 \ldots i_d} = X^{(1)}(i_1) \ldots X^{(d)}(i_d) Y^{(1)}(i_1) \ldots Y^{(d)}(i_d) = \left(X^{(1)}(i_1) \otimes Y^{(1)}(i_1)\right) \cdot \ldots \cdot \left(X^{(d)}(i_d) \otimes Y^{(d)}(i_d)\right).$$

Отметим, что и для ТТ разложения и для разложения Таккера представление Адамарова произведения в тензорных форматах возможно только с рангами, равными произведению рангов входных тензоров. В результате, в случае использования операции округления с помощью сингулярного разложения получается полиномиальная зависимость от ранга с высокой степенью полинома. Для ТТ формата – $\mathcal{O}(dnr^6)$ операций. Поэтому на практике даже для необольших значений рангов необходимо использование альтернативных подходов. В настоящей работе используется подход метода крестовой аппроксимации.

Скалярное произведение и норма

Определим скалярное произведение двух тензоров $\mathcal{X}$ и $\mathcal{Y}$ следующим образом:

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1,\dots,i_d} \mathcal{X}(i_1,\dots,i_d)\mathcal{Y}(i_1,\dots,i_d).$$

Норма тензора естественным образом определяется через скалярное произведение:

$$\|\mathcal{X}\| = \langle \mathcal{X}, \mathcal{X} \rangle^{1/2}.$$

Норма в тензорных форматах вычисляется через Адамарово произведение тензоров и суммирования по всем элементам результата. Важно отметить, что для ускорения вычислений можно использовать специальную структуру факторов разложения Адамарова произведения [104].

Матрично-векторное умножение

Приведем описание матрично-векторного умножения только для ТТ-формата, так как в случае разложения Таккера оно не используется в диссертации. Рассмотрим умножение ТТ-матрицы $\mathcal{A}$ на ТТ-тензор $\mathcal{X}$: $\mathcal{Y} = \mathcal{A}\mathcal{X}$. Можно показать, что ядра $Y^{(k)}$ ТТ-разложения $\mathcal{Y}$ могут быть представлены в виде [104]:

$$Y^{(k)}(i_k) = \sum_{j_k} \left(A^{(k)}(i_k, j_k) \otimes X^{(k)}(j_k) \right),$$

а значит, получено представление $\mathcal{Y}$ с рангами равными произведению рангов $\mathbf{r} \circ \mathbf{R}$. Для уменьшения ранга можно использовать операцию округления,

сложность которой оценивается как $\mathcal{O}(dnr^3R^3)$. Для уменьшения сложности вычислений можно минимизировать норму разности тензора заданного ранга и $\mathcal{Y}$, и использовать специальную структуру ядер $Y^{(k)}$.

1.3 Задача на собственные значения в тензорных форматах

Предположим, что необходимо найти минимальное собственное значение λ_1 и соответствующий собственный вектор x_1 матрицы A размера $N \times N$:

$$Ax_1 = \lambda_1 x_1.$$

Мы рассматриваем специальный случай, когда $N = n_1 \cdot \cdots \cdot n_d$, и когда многомерный массив $\mathcal{X} = \text{reshape}(x, [n_1, \ldots, n_d]) \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ с некоторой заданной точностью имеет малый тензорный ранг $\mathbf{r}$. Здесь reshape обозначает стандартную MATLAB функцию, которая меняет размерность массива и размеры мод с сохранением общего количества элементов. Рассмотрим случай многомерной матрицы, полученной при дискретизации оператора Лапласа на прямоугольной сетке. Можно показать, что собственные векторы этой матрицы имеют тензорный ранг 1. Более интересными примерами являются, например, расчет колебательного спектра молекул или вычисление энергии спиновых систем в физике твердого тела.

Для наших целей удобно рассматривать матрицу A в виде многомерного линейного оператора:

$$\mathcal{A} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \times \cdots \times n_d},$$

то есть

$$\mathcal{A}(i_1 \ldots i_d, j_1 \ldots j_d) = A(\overline{i_1 \ldots i_d}, \overline{j_1 \ldots j_d}),$$

где

$$\overline{i_1 \ldots i_d} = i_1 + n_1(i_2 - 1) + n_1 n_2(i_3 - 1) + \cdots + n_1 \ldots n_d(i_d - 1),\ i_k = 1, \ldots n_k,\ k = 1, \ldots, d.$$

Мы также предполагаем, что оператор является *симметричным*, то есть для любых $i_k, j_k = 1, \ldots, n_k$

$$\mathcal{A}(i_1 \ldots i_d, j_1 \ldots j_d) = \mathcal{A}(j_1 \ldots j_d, i_1 \ldots i_d).$$

Предположим, что симметричный положительно определенный оператор $\mathcal{A}$ имеет собственные значения

$$\lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots$$

и соответствующие собственные векторы $\mathcal{X}_i$, $i = 1, 2, \dots$:

$$\mathcal{A}\mathcal{X}_i = \lambda_i \mathcal{X}_i.$$

Пусть $\mathcal{M}_{\mathbf{r}}$ обозначает множество тензоров с фиксированным рангом $\mathbf{r}$ в TT формате. Как мы увидим далее, $\mathcal{M}_{\mathbf{r}}$ образует гладкое подмногообразие в $\mathbb{R}^{n_1 \times \dots \times n_d}$. Задача ставится следующим образом:

$$\begin{aligned} \mathfrak{R}(\mathcal{X}) \equiv & \frac{\langle \mathcal{A}\mathcal{X}, \mathcal{X} \rangle}{\langle \mathcal{X}, \mathcal{X} \rangle} \to \min, \\ & \mathcal{X} \in \mathcal{M}_{\mathbf{r}}, \end{aligned} \tag{1.3}$$

где $\mathfrak{R}(\mathcal{X})$ обозначает отношение Рэлея. Если точный собственный вектор приближается с высокой точностью тензорным разложением, то и решение оптимизационной задачи (1.3) является приближением собственного вектора. Отметим, что в Главе 3 оператор $\mathcal{A}$ является нелинейным, то есть $\mathcal{A} \equiv \mathcal{A}(\mathcal{X})$. В этом случае вместо задачи (1.3) необходимо рассматривать задачу

$$\begin{aligned} \langle \mathcal{A}(\mathcal{X})\mathcal{X}, \mathcal{X} \rangle \to \min, \\ \mathcal{X} \in \mathcal{M}_{\mathbf{r}}, \quad \|\mathcal{X}\| = 1. \end{aligned}$$

Также в настоящей диссертации рассматривается поиск нескольких минимальных собственных значений. Без ограничения по рангу эта задача также может быть поставлена в виде следующей задачи оптимизации [119]

$$\begin{aligned} \operatorname{trace}(\mathbf{X}^\top A \mathbf{X}) \to \min, \\ \mathbf{X}^\top \mathbf{X} = I_B, \end{aligned} \tag{1.4}$$

где $\mathbf{X} = [\operatorname{vec}(\mathcal{X}_1), \dots, \operatorname{vec}(\mathcal{X}_B)] \in \mathbb{R}^{n_1 \dots n_d \times B}$ и vec обозначает векторизацию многомерного массива:

$$\operatorname{vec}(\mathcal{X})_{\overline{i_1 \dots i_d}} = \mathcal{X}(i_1, \dots, i_d), \quad i_k = 1, \dots, n_k, \quad k = 1, \dots, d.$$

Обычно ограничение на ранг для задачи (1.4) накладывается для блочного TT формата [24], однако для больших B ранг такого представления быстро растет. Поэтому в диссертации рассматривается ограничение независимо на ранг

каждого из векторов $\mathcal{X}_i$, $k = 1,\ldots,B$, и используется обобщение классических итерационных методов. Простейшие итерационные методы строятся на основе подхода “дефляции”. То есть сначала находится минимальное собственное значение и соответствующий собственный вектор, а затем на ортогональном подпространстве к уже найденым собственным векторам ищется следующее минимальное собственное значение и соответствующий собственный вектор. Однако, если собственные векторы находятся неточно, то ошибка при использовании дефляции может накапливаться. Более того, итерационные методы для поиска одного собственного значения могут сходиться медленно, если собственные значения расположены близко друг к другу. Поэтому более эффективным подходом является использование итерационных методов, в которых одновременно итерируются несколько векторов [4].

Рассмотрим теперь способы решения оптимизационных задач. Пусть необходимо решить (1.3). Если вектор $\mathcal{X}_1$, соответствующий минимальному собственному значению λ_1 имеет приближенно тензорный ранг $\mathbf{r}$, то можно ожидать, что решение $\mathcal{X}_*$ задачи (1.3) приближает $\mathcal{X}_1$ (с точностью до множителя), и $\lambda_1 \approx \mathcal{R}(\mathcal{X}_*)$. Существует несколько основных направлений для решения задач оптимизации с ограничением по рангу. Среди них обобщение стандартных итерационных методов с дополнительным округлением по рангу, риманова оптимизация и схема попеременных направлений (alternating linear scheme, ALS). В настоящей диссертации мы пользуемся идеями каждого из этих подходов для построения новых методов решения задач на собственные значения. Поэтому, для каждого из них приведем краткое описание и результаты, которые будут использоваться далее в диссертации. Для ALS подхода также предлагается теория сходимости.

1.4 Итерационные методы с округлением

Все основные операции, возникающие в итерационных процессах такие, как сложение, умножение на число, вычисление скалярных произведений и умножение матрицы на вектор могут быть эффективно сделаны без вычисления всего массива. Подробное описание арифметики в тензорных форматах дано в разделе 1.2. Благодаря наличию эффективной тензорной арифметики

можно обобщить классические итерационные процессы. Приведем в качестве примера метод градиентного спуска для отношения Рэлея

$$\mathcal{X}^{(k+1)} = \mathcal{X}^{(k)} - \tau_k \nabla \mathfrak{R}\left(\mathcal{X}^{(k)}\right), \tag{1.5}$$

где

$$\nabla \mathfrak{R}(\mathcal{X}) = \frac{2}{\langle \mathcal{X}, \mathcal{X} \rangle} \left(\mathcal{A}\mathcal{X} - \mathfrak{R}(\mathcal{X})\mathcal{X}\right)$$

и τ_k является параметром итерации, который в простейшем случае можно выбрать небольшой константой или выбирать адаптивно из условия оптимальности функционала на каждой итерации (линейный поиск). После каждой итерации требуется дополнительная нормализация

$$\mathcal{X}^{(k+1)} := \frac{\mathcal{X}^{(k+1)}}{\|\mathcal{X}^{(k+1)}\|}$$

для устойчивости вычислений. Также можно использовать предобусловленный метод градиентного спуска, известный как PINVIT (preconditioned inverse iteration) [96]:

$$\mathcal{X}^{(k+1)} = \mathcal{X}^{(k)} - \tau_k \mathcal{B}^{-1} \nabla \mathfrak{R}\left(\mathcal{X}^{(k)}\right),$$

где $\mathcal{B}$ обозначает предобуславливатель. Оценка сходимости PINVIT была найдена в серии работ Князевым и Неймейером [96, 77]. Сходимость метода определялась через константу γ

$$\|\mathcal{I} - \mathcal{B}^{-1}\mathcal{A}\|_{\mathcal{A}} \leq \gamma < 1.$$

Если предобуславливатель $\mathcal{B}^{-1}$ и оператор $\mathcal{A}$ заданы в ТТ формате, то имея $\mathcal{X}^{(k)}$ в ТТ формате, с использованием тензорной арифметики несложно получить $\mathcal{X}^{(k+1)}$ также в ТТ формате. Проблема заключается в том, что ранг у $\mathcal{X}^{(k+1)}$ может быть заметно больше, чем у $\mathcal{X}^{(k)}$. Обобщение итерационных методов на тензорные форматы заключается в том, что для избежания роста ранга необходимо после округлять тензорное представление с заданным рангом или с заданной точностью с помощью оператора округления $\mathcal{T}$ (раздел 1.2):

$$\begin{aligned} \mathcal{X}^{(k+1)} &= \mathcal{T}\left(\mathcal{X}^{(k)} - \tau_k \mathcal{B}^{-1} \nabla \mathfrak{R}\left(\mathcal{X}^{(k)}\right)\right), \\ \mathcal{X}^{(k+1)} &:= \frac{\mathcal{X}^{(k+1)}}{\|\mathcal{X}^{(k+1)}\|} \end{aligned} \tag{1.6}$$

В работах О. Лебедевой [84, 85] был получен следующий результат для сходимости итерационных методов для задач на собственные значения с округлением по точности

Теорема 1.1 ([85]). *Если для итерации* (1.5) *верно, что* $\|\mathcal{X}^{(k+1)} - \mathcal{X}\| \leq \gamma \|\mathcal{X}^{(k)} - \mathcal{X}\|$, *причем* $\gamma < 1/3$, *то для точности округления* $\epsilon < \sqrt{2}/3$ *метод* (1.6) *сходится так, что* $\|\mathcal{X}^{(k+1)} - \mathcal{X}\| \leq 3\gamma \|\mathcal{X}^{(k)} - \mathcal{X}\|$, *причем метод сходится вплоть до момента, когда решение попадает в* 3ϵ *окрестность* $\mathcal{X}$.

Округление по точности может привести к росту ранга. В работе [81] было замечено, что выбор хорошего предобуславливателя $\mathcal{B}$ может заметно уменьшить рост рангов в итерационном процессе. Аналогичным образом обобщаются более сложные итерационные методы такие, как LOBPCG. Подробное изложение LOBPCG в малоранговых форматах, его обобщение на блочный случай, а также построение нового предобуславливателя описаны в Главе 2 настоящей диссертации.

1.5 ALS оптимизация

Еще одним подходом является так называемая схема попеременных наименьших квадратов (alternating least squares, ALS). Несложно заметить, что подмножество TT тензоров со всеми фиксированными ядрами, кроме одного, образует линейное подпространство в пространстве всех тензоров. Это наблюдение может значительно упростить процесс оптимизации. Например, в случае решения линейных систем можно получить явные представления на неизвестное ядро через решения линейных систем с небольшим числом неизвестных ($n_i r_i r_{i-1}$ неизвестных для i-го ядра). Затем найденное ядро фиксируется и ищется следующее ядро. Эта процедура продолжается до сходимости.

Приведем формулировку метода ALS. Рассмотрим функционал $F(x)$, где $x = (x_1, \ldots, x_N)$ является набором векторов $x_i \in \mathbb{R}^{n_i}$. *Попеременная оптимизация* или *блочный покоординатный спуск* решает задачу

$$\min F(x) = \min F(x_1, \ldots, x_N)$$

с помощью попеременного обновления переменных x_i при фиксированных всех остальных переменных x_j, $j \neq i$:

$$x_i \leftarrow \arg\min_{\xi \in \mathbb{R}^{n_i}} F(x_1, \ldots, x_{i-1}, \xi, x_{i+1}, \ldots, x_N)$$

Этот подход используется в большом количестве приложений. В настоящей диссертации мы используем его для случая тензорных форматов. Для тензорных разложений этот метод принято называть ALS в силу того, что при фиксировании всех ядер, кроме одного, получается линейное пространство тензоров. Задача оптимизации выглядит следующим образом. Обновление ядра G_m, когда все ядра остальные ядра зафиксированы может быть найдено из

$$G_m \leftarrow \arg\min_{G} F(G_1, \ldots, G_{m-1}, G, G_{m+1} \ldots, G_d).$$

В этом случае ядра рассматриваются как векторы длиной $n_m r_m r_{m-1}$. Если F является квадратичным функционалом, то минимизация по каждому ядру является задачей линейных наименьших квадратов.

1.5.1 Общая теория сходимости ALS подхода

Приведем общую теорию сходимости ALS подхода, полученную автором диссертации в соавторстве с А. Ушмаевым и И. В. Оселедцем в работе [101]. В этой работе предложена интерпретация ALS подхода как последовательности оптимизационных задач на “двигающихся подпространствах”. В отличие от [117] в рамках предлагаемого подхода удается явно проиллюстрировать, что сходимость метода определяется через взаимосвязь классического мультипликативного метода Шварца и кривизны многообразия малоранговых матриц или тензоров.

Рассмотрим C^1 функционал $f : \mathbf{V} \to \mathbf{V}$ на Гильбертовом пространстве $\mathbf{V}$. Любому $x \in \mathbf{V}$ поставим в соответствие замкнутое подпространство $T(x)$ пространства $\mathbf{V}$. Затем предположим, что задано разложение $T(x)$ в сумму d замкнутых, вообще говоря, пересекающихся $T_i(x)$:

$$T(x) = T_1(x) + \cdots + T_d(x).$$

Затем определяется d отображений

$$\mathbf{P}_i : \mathbf{V} \to \mathcal{L}(\mathbf{V}), \quad i = 1, \ldots, d,$$

так, что для каждого $x \in \mathbf{V}$ линейный оператор $\mathbf{P}_i(x)$ является ортопроектором на подпространство $T_i(x)$. Соответственно, $\mathbf{P}(x)$ обозначает ортопроектор на $T(x)$.

Далее, пусть $\mathbf{S}_i$, $i = 1, \dots, d$, обозначают (нелинейные) операторы на $\mathbf{V}$ так, что $y = \mathbf{S}_i(x)$ удовлетворяет

$$y \in x + T_i(x), \qquad \mathbf{P}_i(x)\nabla f(y) = 0.$$

Это означает, что $\mathbf{S}_i$ отображает x в критическую точку f на гиперплоскости $x + T_i(x)$. Если, например, f является строго выпуклой и коэрцитивной, тогда оператор $\mathbf{S}_i$ определяется единственным образом и соответствует минимизации f на $x + T_i(x)$.

Попеременная оптимизация на "двигающихся" гиперплоскостях соответствует итерации следующего вида

$$x_{\ell+1} = \mathbf{S}(x_\ell) \coloneqq (\mathbf{S}_d \circ \cdots \circ \mathbf{S}_1)(x_\ell).$$

Нашей целью является исследование локальной сходимости такой итерации в соответствующих предположениях на гладкость. Для этого рассмотрим неподвижную точку

$$\bar{x} = \mathbf{S}_i(\bar{x})$$

для всех $\mathbf{S}_i$ в чьей окрестности все $\mathbf{P}_i$, $\mathbf{S}_i$ а также $\mathbf{P}$ являются непрерывно дифференцируемыми (по Фреше) отображениями. Тогда $\bar{x}$ также является неподвижной точкой $\mathbf{S}$.

Локальные свойства сжатия в окрестности $\bar{x}$ определяются свойствами производных $\mathbf{S}_i'(\bar{x})$ и рассматриваются в следующем параграфе. Некоторые свойства $\mathbf{S}_i'(\bar{x})$ могут быть получены путем дифференцирования уравнений

$$\mathbf{P}_i(x)(\mathbf{S}_i(x) - x) = \mathbf{S}_i(x) - x.$$

После дифференцирования получаем

$$\mathbf{P}_i'(x;h)(\mathbf{S}_i(x) - x) + \mathbf{P}_i(x)(\mathbf{S}_i'(x)h - h) = \mathbf{S}_i'(x)h - h \tag{1.7}$$

для всех $h \in \mathbf{V}$. Здесь оператор $\mathbf{P}_i'(x;h) \in \mathcal{L}(\mathbf{V})$ обозначает применение производной $\mathbf{P}_i(x)$ в x к h. Следовательно, в неподвижной точке $\bar{x} = \mathbf{S}_i(\bar{x})$ выполняется

$$(\mathbf{I} - \mathbf{P}_i(\bar{x}))\mathbf{S}_i'(\bar{x})h = (\mathbf{I} - \mathbf{P}_i(\bar{x}))h. \tag{1.8}$$

Из этого уравнения сразу следуют следующие свойства.

Утверждение 1.1. *Предположим, что* $\mathbf{P}_i$ *и* $\mathbf{S}_i$ *являются непрерывно дифференцируемыми в окрестности неподвижной точки* $\bar{x}$*. Тогда*

(i) *подпространство* $T_i(\bar{x})$ *является инвариантным подпространством* $\mathbf{S}_i'(\bar{x})$*,*

(ii) *ограничение* $\mathbf{S}_i'(\bar{x})$ *на ортогональное дополнение* $T_i(\bar{x})^\perp$ *имеет спектральный радиус не превосходящий единицы и равен одному, тогда и только тогда, когда* $T_i(\bar{x})^\perp$ *также является инвариантным подпространством* $\mathbf{S}_i'(\bar{x})$*.*

Вычисление производных

С помощью $\mathbf{A}(x) = \nabla^2 f(x)$ мы обозначаем гессиан f в x. Для краткости мы будем использовать следующие обозначения

$$\bar{\mathbf{P}}_i := \mathbf{P}_i(\bar{x}), \qquad \bar{\mathbf{P}} := \mathbf{P}(\bar{x}), \qquad \bar{\mathbf{A}} := \mathbf{A}(\bar{x}), \qquad \bar{\mathbf{B}}_i := (\bar{\mathbf{P}}_i \bar{\mathbf{A}} \bar{\mathbf{P}}_i)^{-1}.$$

Для получения формулы для $\mathbf{S}'(\bar{x})$ мы дифференцируем каждый $\mathbf{S}_i$ отдельно. Производные $\mathbf{S}_i'(\bar{x})$ вычисляются следующим образом.

Утверждение 1.2. *Предположим, что* $\mathbf{P}_i$ *и* $\mathbf{S}_i$ *непрерывно дифференцируемы в окрестности неподвижной точки* $\bar{x}$*, и что* f *является дважды непрерывно дифференцируемой в окрестности* $\bar{x}$*. Если линейный оператор* $\bar{\mathbf{P}}_i \bar{\mathbf{A}} \bar{\mathbf{P}}_i$ *обратим на* $T_i(\bar{x})$*, тогда*

$$\mathbf{S}_i'(\bar{x})h = h - \bar{\mathbf{B}}_i \bar{\mathbf{P}}_i \bar{\mathbf{A}} h - \bar{\mathbf{B}}_i \mathbf{P}_i'(\bar{x}; h) \nabla f(\bar{x}). \tag{1.9}$$

В частности,

$$\mathbf{S}_i'(\bar{x}) = \bar{\mathbf{B}}_i \mathbf{P}_i'(\bar{x}; h) \nabla f(\bar{x}) \quad \text{на} \quad T_i(\bar{x}).$$

Отметим, что из (1.7) и (1.8) следует, что для любого $h \in \mathbf{V}$ линейный оператор $\mathbf{P}_i'(\bar{x}; h)$ отображает в пространство $T_i(\bar{x})$. Следовательно, композиция $\bar{\mathbf{B}}_i$ с этим оператором определена.

Доказательство. Дифференцируя уравнение $\mathbf{P}_i(x) \nabla f(\mathbf{S}_i(x)) = 0$ получим

$$\mathbf{P}_i'(x; h) \cdot \nabla f(x) + \mathbf{P}_i(x) \mathbf{A}(x) \mathbf{S}_i'(x) h = 0 \tag{1.10}$$

для всех вариаций $h \in \mathbf{V}$. Разделяя $\mathbf{S}_i'(x)h$ в (1.10) на составные части на $T_i(\bar{x})$ и ортогональное дополнение получим

$$\mathbf{P}_i(x)\mathbf{A}(x)\mathbf{P}_i(x)\mathbf{S}_i'(x)h = -\mathbf{P}_i(x)\mathbf{A}(x)(\mathbf{I}-\mathbf{P}_i(x))\mathbf{S}_i'(x)h - \mathbf{P}_i'(x;h)\nabla f(x).$$

В неподвижной точке мы можем использовать (1.8). Следовательно,

$$\bar{\mathbf{P}}_i\bar{\mathbf{A}}\bar{\mathbf{P}}_i\mathbf{S}_i'(\bar{x})h = -\bar{\mathbf{P}}_i\bar{\mathbf{A}}(\mathbf{I}-\bar{\mathbf{P}}_i)h - \mathbf{P}_i'(\bar{x};h)\cdot\nabla f(\bar{x}).$$

Предполагая, что $\bar{\mathbf{P}}_i\bar{\mathbf{A}}\bar{\mathbf{P}}_i$ имеет обратный оператор $\bar{\mathbf{B}}_i$ на $T_i(\bar{x})$ получим

$$\bar{\mathbf{P}}_i\mathbf{S}_i'(\bar{x})h = \bar{\mathbf{P}}_i h - \bar{\mathbf{B}}_i\bar{\mathbf{P}}_i\bar{\mathbf{A}}h - \bar{\mathbf{B}}_i\mathbf{P}_i'(\bar{x};h)\nabla f(\bar{x}).$$

Используя (1.8) еще раз, мы получаем

$$\mathbf{S}_i'(\bar{x})h = (\mathbf{I}-\bar{\mathbf{P}}_i)\mathbf{S}_i'(\bar{x})h + \bar{\mathbf{P}}_i\mathbf{S}_i'(\bar{x})h = (\mathbf{I}-\bar{\mathbf{P}}_i)h + \bar{\mathbf{P}}_i h - \bar{\mathbf{B}}_i\bar{\mathbf{P}}_i\bar{\mathbf{A}}h - \bar{\mathbf{B}}_i\mathbf{P}_i'(\bar{x};h)\nabla f(\bar{x}),$$

что совпадает с (1.9). □

Упростим обозначения:

$$\bar{\mathbf{P}}_i^{\bar{\mathbf{A}}} := \bar{\mathbf{B}}_i\bar{\mathbf{P}}_i\bar{\mathbf{A}}.$$

Если $\bar{\mathbf{A}}$ является положительно определенным оператором, то $\bar{\mathbf{B}}_i$ всегда определен, и $\bar{\mathbf{P}}_i^{\bar{\mathbf{A}}}$ может быть интерпретирован как $\bar{\mathbf{A}}$-ортопроектор на подпространство $T_i(\bar{x})$. То есть он является ортопроектором по отношению к скалярному произведению $(x,y) \mapsto \langle x, \bar{\mathbf{A}}y\rangle$.[1]

Далее, определим линейный оператор $\bar{\mathbf{N}}_i$ на $\mathbf{V}$ так, что

$$\bar{\mathbf{N}}_i h := \mathbf{P}_i'(\bar{x};h)\nabla f(\bar{x}) \tag{1.11}$$

для всех h. Используя эти обозначения, и в предположениях Утверждения 1.2, $\mathbf{S}_i'(\bar{x})$ может быть записан как

$$\mathbf{S}_i'(\bar{x}) = (\mathbf{I}-\bar{\mathbf{P}}_i^{\bar{\mathbf{A}}}) - \bar{\mathbf{B}}_i\bar{\mathbf{N}}_i.$$

Формула для $\mathbf{S}'(\bar{x})$ получается как дифференцирование сложной функции. Сформулируем этот факт в виде теоремы.

[1] Заметим, что (для удобства опустим индекс) $(\bar{\mathbf{P}}^{\bar{\mathbf{A}}}x)^T\bar{\mathbf{A}}(\mathbf{I}-\bar{\mathbf{P}}^{\bar{\mathbf{A}}})x = x^T(\bar{\mathbf{A}}\bar{\mathbf{B}}\bar{\mathbf{P}}\bar{\mathbf{A}} - \bar{\mathbf{A}}\bar{\mathbf{B}}\bar{\mathbf{P}}\bar{\mathbf{A}}\bar{\mathbf{P}}\bar{\mathbf{P}}^{\bar{\mathbf{A}}})x = 0$ для всех $x \in \mathbf{V}$, так как $\bar{\mathbf{P}}\bar{\mathbf{A}}\bar{\mathbf{P}}\bar{\mathbf{P}}^{\bar{\mathbf{A}}} = \bar{\mathbf{B}}^{-1}\bar{\mathbf{B}}\bar{\mathbf{P}}\bar{\mathbf{A}} = \bar{\mathbf{P}}\bar{\mathbf{A}}$. Следовательно, $\bar{\mathbf{P}}^{\bar{\mathbf{A}}}x$ является $\bar{\mathbf{A}}$-ортогональным к $(\mathbf{I}-\bar{\mathbf{P}}^{\bar{\mathbf{A}}})x$.

Теорема 1.2. *Предположим, что все* $\mathbf{P}_i$ *и* $\mathbf{S}_i$ *являются непрерывно дифференцируемыми в окрестности неподвижной точки* $\bar{x}$*, и что* f *является дважды непрерывно дифференцируемой в окрестности* $\bar{x}$*. Предположим, что все* $\bar{\mathbf{B}}_i = (\bar{\mathbf{P}}_i \bar{\mathbf{A}} \bar{\mathbf{P}}_i)^{-1}$ *существуют на* $T_i(\bar{x})$*. Тогда*

$$\mathbf{S}'(\bar{x}) = \prod_{i=d}^{1} \mathbf{S}_i'(\bar{x}) = \prod_{i=d}^{1} [(\mathbf{I} - \bar{\mathbf{P}}_i^{\bar{\mathbf{A}}}) - \bar{\mathbf{B}}_i \bar{\mathbf{N}}_i]. \tag{1.12}$$

Случай отсутствия кривизны в неподвижной точке ($\bar{\mathbf{N}}_i = 0$)

Простым случаем для исследования является $\bar{\mathbf{N}}_i = 0$. В этом случае формула для $\mathbf{S}'(\bar{x})$ имеет вид

$$\mathbf{S}'(\bar{x}) = \prod_{i=d}^{1} (\mathbf{I} - \bar{\mathbf{P}}_i^{\bar{\mathbf{A}}}),$$

и известна, например, из *мультипликативного метода Шварца.* Следующее утверждение следует из стандартного результата для мультипликативного метода Шварца [35, Теорема 3.7] путем ограничения на подпространство $\bar{T}(x)$ и путем замены $\bar{\mathbf{A}}$ на $\bar{\mathbf{P}}\bar{\mathbf{A}}\bar{\mathbf{P}}$.

Теорема 1.3. *Предположим, что* $\bar{\mathbf{N}}_i = 0$ *и гессиан* $\bar{\mathbf{A}}$ *является положительно определенным на* $T(\bar{x})$*. Тогда* $\rho(\bar{\mathbf{P}}\mathbf{S}'(\bar{x})\bar{\mathbf{P}}) < 1$*. В частности,* $\|\bar{\mathbf{P}}\mathbf{S}'(\bar{x})h\|_{\bar{\mathbf{A}}} < \|h\|_{\bar{\mathbf{A}}}$ *для всех* $h \in T(\bar{x})$*, где* $\|x\|_{\bar{\mathbf{A}}} = (x^T \bar{\mathbf{A}} x)^{1/2}$ *является нормой на* $T(\bar{x})$*.*

Случай $\bar{\mathbf{N}}_i = 0$ возникает в двух важный сценариях.

Локально постоянные подпространства. Если подпространства $T_i(x)$ не меняются для всех x в окрестности $\bar{x}$, тогда $\mathbf{P}_i'(\bar{x}) = 0$. Этот случай возникает в классическом блочном покоординатном спуске, который также известен под названием *нелинейный блочный метод Гаусса-Зейделя*, и базируется на фиксированном непересекающемся разделении всего пространства $T(\bar{x}) = \mathbf{V}$. Следовательно, в этом случае мы получаем хорошо известный факт, что скорость локальной сходимости нелинейного метода Гаусса-Зейделя равна сходимости линейного Гаусса-Зейделя с гессианом в качестве матрицы системы, смотри, например, [99].

Нулевой градиент. Операторы $\bar{\mathbf{N}}_i$ также равны нулю в неподвижных точках, удовлетворяющих $\nabla f(\bar{x}) = 0$. Это происходит, в частности, когда рассматривается оптимизация в малоранговом случае, и глобальная критическая точка принадлежит рассматриваемому многообразию малоранговых матриц или тензоров. Мы будем пользоваться этим результатом при доказательстве сходимости ALS II метода в разделе 2.2.

1.5.2 ALS минимизация отношения Рэлея

Рассмотрим теперь частный случай — задачу минимизации (1.3) с помощью ALS подхода для $d = 2$. Используем скелетное разложение $\mathcal{X}$: $\mathcal{X} = UV^\top$, где $U \in \mathbb{R}^{n\times r}$, $V \in \mathbb{R}^{m\times r}$ — матрицы с полным столбцовым рангом. Это представление эквивалентно TT-формату в двумерном случае при $X^{(1)}_\alpha(i_1) = (U)_{i_1\alpha}$ и $X^{(2)}_\alpha(i_2) = (V)_{i_2\alpha}$.

Один шаг ALS минимизации отношения Рэлея с начального приближения $\mathcal{X} = \mathrm{vec}(UV^\top)$ может быть записан в виде последовательной оптимизации по U и V:

$$\begin{aligned} U &\leftarrow \operatorname*{arg\,min}_{U\in\mathbb{R}^{n\times r}} \mathfrak{R}(UV^\top), \\ V &\leftarrow \operatorname*{arg\,min}_{V\in\mathbb{R}^{m\times r}} \mathfrak{R}(UV^\top). \end{aligned} \tag{1.13}$$

Используя хорошо известное свойство кронекерова произведения

$$\mathrm{vec}(AXB) = (B^\top \otimes A)\,\mathrm{vec}(X),$$

не сложно убедиться, что эти шаги эквивалентны решению следующих обобщенных задач на собственные значения

$$\begin{aligned} (V\otimes I_n)^\top A(V\otimes I_n)\,\mathrm{vec}(U) &= \lambda^{\mathsf{R}}_{\min}\cdot(V^\top V\otimes I_n)\,\mathrm{vec}(U), \\ (I_m\otimes U)^\top A(I_m\otimes U)\,\mathrm{vec}(V^\top) &= \lambda^{\mathsf{L}}_{\min}\cdot(I_m\otimes U^\top U)\,\mathrm{vec}(V^\top). \end{aligned} \tag{1.14}$$

Действительно, для нахождения, например, U нам необходимо минимизировать

$$\begin{aligned} \min_{U\in\mathbb{R}^{n\times r}} \mathfrak{R}(UV^\top) &= \min_{U\in\mathbb{R}^{n\times r}} \frac{\langle \mathrm{vec}(UV^\top), A\,\mathrm{vec}(UV^\top)\rangle}{\langle \mathrm{vec}(UV^\top), \mathrm{vec}(UV^\top)\rangle} = \\ &= \min_{U\in\mathbb{R}^{n\times r}} \frac{\langle \mathrm{vec}(U), ((V\otimes I_n)^\top A(V\otimes I_n))\,\mathrm{vec}(U)\rangle}{\langle \mathrm{vec}(U), (V^\top V\otimes I_n)\,\mathrm{vec}(U)\rangle}, \end{aligned}$$

что приводит к первой обобщенной задаче на собственные значения из (1.14).

По аналогии с двумерным случаем в случае $d > 2$ мы получаем следующие линейные системы на векторизованное p-е ядро $x^{(p)} = \mathrm{vec}\,(X^{(p)})$, являющееся собственным вектором, отвечающим минимальному собственному числу $\lambda_{\min}$ в следующей задаче на собственные значения

$$(X^{\neq p\top} A X^{\neq p}) x^{(p)} = \lambda_{\min} X^{\neq p\top} X^{\neq p} x^{(p)},$$

где

$$X^{\neq p} = X^{<p} \otimes I_{n_p} \otimes X^{>p} \in \mathbb{R}^{n_1 \ldots n_d \times r_{p-1} n_p r_p}.$$

Матрица $X^{<p}$ имеет размер $n_1 \ldots n_{p-1} \times r_p$ и состоит из произведения первых $p-1$ ядер:

$$X^{<p}\left(\overline{i_1 i_2 \ldots i_{p-1}}, :\right) = X^{(1)}(i_1)\, X^{(2)}(i_2) \ldots X^{(p-1)}(i_{p-1}).$$

Матрица $X^{>p}$ определяется по аналогии. Обычно используется дополнительная ортогонализация $X^{<p}$ и $X^{>p}$ [54].

1.5.3 ALS минимизация для решения линейных систем

В настоящей диссертации мы также будем использовать решения линейных систем с помощью ALS итерации. Опишем один шаг ALS итерации в этом случае. Пусть $A \in \mathbb{R}^{nm \times nm}$ является симметричной положительно определенной матрицей. Пусть мы хотим решить систему $A\,\mathrm{vec}(\mathcal{X}) = \mathrm{vec}(\mathcal{F})$. Поставим ей в соответствие функционал энергии $\langle A\,\mathrm{vec}(\mathcal{X}), \mathrm{vec}(\mathcal{X})\rangle - 2\langle \mathrm{vec}(\mathcal{F}), \mathrm{vec}(\mathcal{X})\rangle$. Если целью является поиск решения заданного ранга, то функционал энергии минимизируется на многообразии матриц заданного ранга:

$$\begin{aligned} &\langle A\,\mathrm{vec}(\mathcal{X}), \mathrm{vec}(\mathcal{X})\rangle - 2\langle \mathrm{vec}(\mathcal{F}), \mathrm{vec}(\mathcal{X})\rangle \to \min \\ &\mathrm{rank}(\mathcal{X}) = r. \end{aligned}$$

Применим к этой задаче ALS минимизацию. Рассмотрим минимизацию функционала энергии сначала по U:

$$\min_{U \in \mathbb{R}^{n \times r}} \left\langle A\,\mathrm{vec}(UV^\top), \mathrm{vec}(UV^\top)\right\rangle - 2\left\langle \mathrm{vec}(\mathcal{F}), \mathrm{vec}(UV^\top)\right\rangle.$$

Используя свойство $(V \otimes I_n)\,\mathrm{vec}(U) = \mathrm{vec}(UV^\top)$, мы получаем линейную систему с неизвестным вектором $\mathrm{vec}(U) \in \mathbb{R}^{nr}$

$$(V^\top \otimes I_n) A (V \otimes I_n)\,\mathrm{vec}(U) = (V^\top \otimes I_n)\mathrm{vec}(\mathcal{F}),$$

где матрица $(V^\top \otimes I_n)A(V \otimes I_n)$ имеет размер $nr \times nr$. По аналогии с ALS минимизацией для отношения Рэлея в случае $d > 2$ мы получаем следующие линейные системы на векторизованное p-е ядро $x^{(p)} = \mathrm{vec}\,(X^{(p)})$

$$X^{\neq p\top} A X^{\neq p} x^{(p)} = X^{\neq p\top} \mathrm{vec}(\mathcal{F}),$$

где матрицы $X^{\neq p}$ определены ранее. Локальная матрица системы $X^{\neq p\top} A X^{\neq p} \in \mathbb{R}^{r_{p-1} n_p r_p \times r_{p-1} n_p r_p}$. Отметим, что даже если исходная матрица была разреженной, локальная матрица, вообще говоря, является плотной. Поэтому решение возникающих систем прямым методом становится затруднительным при $r_{p-1} n_p r_p > 10000$, например, при $r_{p-1} = r_p = 10$, $n_p = 100$. Однако локальная матрица может быть быстро умножена на вектор, так как имеет дополнительную структуру, если матрица A является суммой кронекеровых произведений (соответствующий тензор $\mathcal{A}$ имеет малый TT-ранг). Подробное описание быстрого умножения можно найти в [105].

1.6 Методы Римановой оптимизации

Альтернативным подходом к решению задачи (1.3) являются методы, базирующиеся на оптимизации на гладких многообразиях. Действительно, известно, что многообразие тензоров фиксированного TT-ранга

$$\mathcal{M}_{\mathbf{r}} \stackrel{\text{def}}{=} \{\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d} : \ \mathrm{TT\text{–}rank}(\mathcal{X}) = \mathbf{r}\} \tag{1.15}$$

является гладким [53], что позволяет использовать подход римановой оптимизации. В римановой оптимизации вместо всего пространства $\mathbb{R}^{n_1 \times \cdots \times n_d}$ задача рассматривается только на нелинейном многообразии, которому принадлежит решение. Перейдем к формальному описанию основных понятий римановой оптимизации, следуя [116, 1, 127]. Начнем с определения понятия гладкого многообразия.

Определение 1.1 ([116, Определение 2.1.3]). *Подмножество $\mathcal{M} \subset \mathbb{R}^n$ называется m-мерным гладким подмногообразием $\mathbb{R}^n$, если для любого $x \in \mathcal{M}$ найдется открытое множество $U \subset \mathbb{R}^n$ такое, что $U \cap \mathcal{M}$ является диффеоморфным некоторому открытому подмножеству $\Omega \subset \mathbb{R}^m$.*

Ключевым понятием, которое можно ввести на гладких многообразиях, является *касательная плоскость*. Она представляет линеаризацию многообразия в окрестности заданной точки. В итерационных методах на многообразиях вместо обычного евклидового градиента используется градиент (риманов градиент), дополнительно спроецированный на касательную плоскость, что позволяет строить следующее приближение достаточно "близко" к многообразию, в то же время сохраняя удобство работы на линейных пространствах. Более того, как мы увидим далее, касательное пространство тензоров фиксированного ранга имеет малую размерность, что позволяет строить эффективные алгоритмы. Приведем формальное определение касательной плоскости и касательного расслоения.

Определение 1.2 ([116, Определение 2.2.1]). *Пусть $\mathcal{M} \subset \mathbb{R}^n$ является m-мерным гладким подмногообразием. Зафиксируем точку $x \in \mathcal{M}$. Вектор $\eta \in \mathbb{R}^n$ называется касательным вектором $\mathcal{M}$ в точке $x \in \mathcal{M}$, если существует гладкая кривая $\gamma : \mathbb{R} \to \mathcal{M}$ такая, что*

$$\gamma(0) = x, \quad \gamma'(0) = \eta.$$

Множество

$$T_x\mathcal{M} \stackrel{\text{def}}{=} \{\gamma'(0) \mid \text{гладкая } \gamma : \mathbb{R} \to \mathcal{M} \ : \ \gamma(0) = x\}$$

всех касательных векторов $\mathcal{M}$ в точке x называется касательным пространством $\mathcal{M}$ в точке x.

Отметим, что размерность касательного пространства совпадает с размерностью самого многообразия.

Определение 1.3 ([116, Пример 2.6.5]). *Пусть $\mathcal{M} \subset \mathbb{R}^n$ является m-мерным гладким подмногообразием. Множество*

$$T\mathcal{M} \stackrel{\text{def}}{=} \{(x, \eta) \mid x \in \mathcal{M}, \eta \in T_x\mathcal{M}\}$$

называется касательным расслоением $\mathcal{M}$.

В дальнейшем для определения риманова градиента и гессиана нам понадобится обобщение на многообразия понятия производной отображения по

направлению. Пусть γ обозначает кривую, проходящую через x ($\gamma(0) = x$), причем $\gamma'(0) = \eta$. Тогда мы определяем обобщение производной гладкого отображения f по направлению как [1]

$$\mathrm{D}f(x)[\eta] \stackrel{\text{def}}{=} \left.\frac{df(\gamma(t))}{dt}\right|_{t=0}. \tag{1.16}$$

Для того, чтобы мерить расстояния и углы необходимо ввести аналог скалярного произведения на $T_x\mathcal{M} \times T_x\mathcal{M}$. Пусть ξ_x и η_x являются векторными полями на $\mathcal{M}$, которые в каждой точке многообразия x приписывают ей касательный вектор гладким образом. Если отображение $x \to g_x(\xi_x, \eta_x)$ является гладким для любых пар векторных полей ξ_x и η_x на $\mathcal{M}$, то g_x называется *римановой метрикой.* Гладкое многообразие, снабженное римановой метрикой называется *римановым многообразием.* Для подмногообразий $\mathbb{R}^n$ естественно определить риманову метрику через стандартное скалярное произведение на $\mathbb{R}^n$: $g_x(\xi, \eta) = \langle \xi, \eta \rangle$.

Риманов градиент f обозначается $\mathrm{grad} f(x)$ и определяется как единственный элемент из $T_x\mathcal{M}$, удовлетворяющий $g_x(\mathrm{grad} f(x), \eta) = \mathrm{D}f(x)[\eta]$ для любого $\eta \in T_x\mathcal{M}$. Если же отображение f является сужением некоторого гладкого отображения F, определенного на всем $\mathbb{R}^n$, то риманов градиент принимает следующую форму [1]

$$\mathrm{grad}\, f(x) = \mathrm{P}_{T_x\mathcal{M}} \nabla F(x),$$

где ∇ обозначает стандартный евклидовый градиент, а $\mathrm{P}_{T_x\mathcal{M}}$ – проектор на $T_x\mathcal{M}$.

В итерационных алгоритмах на многообразиях на каждой итерации необходимо делать сдвиг по геодезическим (*экспоненциальное отображение*), которые имеют смысл путей с наименьшей длиной. Локально геодезические однозначно задаются точкой на многообразии x и направлением ξ. Однако вычисление геодезических является сложной задачей с вычислительной точки зрения. Поэтому на практике используют *ретракцию*, которая приближает экспоненциальное отображение. Ретракция определяется следующим образом.

Определение 1.4 ([3, Секция 2.3])**.** *Ретракцией на многообразии $\mathcal{M}$ называется гладкое отображение R с касательного расслоения $T\mathcal{M}$ на $\mathcal{M}$ такое, что*

1. *R определено и является гладким в окрестности нулевого сечения $T\mathcal{M}$;*

2. $R(x, 0) = x$ *для любого* $x \in \mathcal{M}$;

3. $\frac{d}{dt} R(x, t\xi)\Big|_{t=0} = \xi$ *для любого* $x \in \mathcal{M}$ *и* $\xi \in T_x\mathcal{M}$.

Перейдем к итерационным процессам на многообразиях. Пусть на k-й итерации задано $x^{(k)}$ и некоторое направление поиска $\xi^{(k)} \in T_{x^{(k)}}\mathcal{M}$, тогда $x^{(k+1)}$ определяется из

$$x^{(k+1)} = R(x^{(k)}, \tau_k \xi^{(k)}). \tag{1.17}$$

Обсудим, из каких соображений выбирать шаг τ_k. Для конкретики рассмотрим функционал $\mathfrak{K}$. Параметры τ_k можно выбрать из соображения оптимальности, то есть

$$\tau_k = \arg\min_{\tau} \mathfrak{K}\Big(R(x^{(k)}, \tau\xi^{(k)})\Big),$$

однако обычно из-за сложного вида ретракции R поиск оптимального параметра является вычислительно затратным и используют *правило Армихо с возвратом*: необходимо найти наименьшее целое $l \geq 0$ такое, что

$$\mathfrak{K}(x^{(k)}) - \mathfrak{K}(R(x^{(k)}, 2^{-l}\tau_k\xi_k)) \geq -c2^{-l}\Big\langle \xi_k, \operatorname{grad} \mathfrak{K}(x^{(k)})\Big\rangle, \tag{1.18}$$

где c — маленькая константа (на практике можно выбрать $c = 10^{-4}$). Можно показать, что такой выбор шага удовлетворяет аналогу условия Армихо на многообразиях [1]. Для сходимости процесса требуется дополнительно наложить условие на вектор направления ξ_k. Последовательность $\xi_k \in T_{x^{(k)}}\mathcal{M}$ должна быть *градиентной*. Это означает, что любая ее подпоследовательность ξ_{n_k}, сходящаяся к нестационарной точке $\mathfrak{K}$, ограничена и удовлетворяет

$$\limsup_{k\to\infty} \Big\langle \operatorname{grad} \mathfrak{K}(x^{(k)}), \xi_{n_k}\Big\rangle < 0. \tag{1.19}$$

Сформулируем теорему сходимости итерационного метода на многообразиях.

Теорема 1.4 ([1, Теорема 4.3.1])**.** *Пусть в итерационном процессе* (1.17) *последовательность* ξ_k *является градиентной, а параметры* τ_k *выбираются из условия Армихо* (1.18)*. Тогда любая предельная точка* $\{x^{(k)}\}$*, принадлежащая* $\mathcal{M}$ *является стационарной точкой функционала* $\mathfrak{K}$*.*

Отметим, что если многообразие не компактно, то существование предельных точек на нем не является очевидным. Поэтому для некомпактных многообразий обычно ищется компактное множество, к которому принадлежит последовательность, порожденная итерационным процессом.

1.6.1 Риманова оптимизация на сфере

При выводе предлагаемого в настоящей диссертации в Главе 2 метода Якоби-Дэвидсона на малоранговом многообразии нам понадобятся формулы для ретракции и проекции на касательное пространство сферы $S^{n-1} = \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$. Размерность сферы равна

$$\dim(S^{n-1}) = n - 1.$$

Несложно показать, что ортопроектор на касательное пространство сферы может быть записан как [1]:

$$\mathrm{P}_{T_x S^{n-1}} z = (I - xx^\top)z. \tag{1.20}$$

Ретракция на сферу имеет следующий вид [1]:

$$R(x, \xi) = \frac{x + \xi}{\|x + \xi\|_2}. \tag{1.21}$$

1.6.2 Риманова оптимизация на тензорных многообразиях

Рассмотрим теперь многообразие тензоров фиксированного TT-ранга. Во-первых, тензоры $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ фиксированного TT-ранга $\mathbf{r}$ образуют гладкое многообразие $\mathcal{M}_\mathbf{r}$ размерности [53]

$$\dim(\mathcal{M}_\mathbf{r}) = \sum_{\alpha=1}^{d} r_{\alpha-1} n_\alpha r_\alpha - \sum_{\alpha=1}^{d-1} r_\alpha^2.$$

Рассмотрим точку $\mathcal{X} \in \mathcal{M}_\mathbf{r}$:

$$\mathcal{X}(i_1, \ldots, i_d) = X_1(i_1) \ldots X_d(i_d).$$

Выберем X_α левоортогональными ядрами, то есть

$$(X_\alpha^l)^\top X_\alpha^l = I_{r_\alpha},$$

где $X_\alpha^l \in \mathbb{R}^{r_{\alpha-1} n_\alpha \times r_\alpha}$ обозначает левую развертку трехмерного ядра X_α. Любой вектор $\Xi \in T_\mathcal{X} \mathcal{M}_\mathbf{r}$ может быть параметризован как [89]

$$\begin{aligned} \Xi(i_1, \ldots, i_d) = {} & \Delta_1(i_1) X_2(i_2) \ldots X_{d-1}(i_{d-1}) X_d(i_d) + \\ & + X_1(i_1) \Delta_2(i_2) \ldots X_{d-1}(i_{d-1}) X_d(i_d) + \cdots + \\ & + X_1(i_1) X_2(i_2) \ldots X_{d-1}(i_{d-1}) \Delta_d(i_d), \end{aligned} \tag{1.22}$$

где $\Delta_\alpha \in \mathbb{R}^{r_{\alpha-1} \times n_\alpha \times r_\alpha}$. Отметим, что (1.22) может быть также записано в виде

$$\Xi(i_1, \ldots, i_d) = \begin{bmatrix} \Delta_1(i_1) & X_1(i_1) \end{bmatrix} \begin{bmatrix} X_2(i_2) & 0 \\ \Delta_2(i_2) & X_2(i_2) \end{bmatrix} \ldots \begin{bmatrix} X_{d-1}(i_{d-1}) & 0 \\ \Delta_{d-1}(i_{d-1}) & X_{d-1}(i_{d-1}) \end{bmatrix} \begin{bmatrix} X_d(i_d) \\ \Delta_d(i_d) \end{bmatrix},$$

то есть тензор из касательного является тензором ранга не выше $2\mathbf{r}$. Это одно из ключевых свойств, определяющих эффективность оптимизации на тензорных многообразиях.

Заметим теперь, что количество параметров, содержащееся в матрицах Δ_i больше, чем размерность многообразия. Это означает, что представление с помощью такой параметризации не является единственным и требует наложения дополнительных условий — условий калибровки. Стандартными являются следующие условия калибровки: $(\Delta_\alpha^l)^\top X_\alpha^l = 0$, $\alpha = 1, \ldots, d-1$. Теперь запишем проекцию произвольного тензора $\mathcal{Z}$ на касательную плоскость. В работе [89] были получены следующие формулы

$$\Delta_\alpha^l = \mathrm{P}_\alpha^\perp \left(I_{n_\alpha} \otimes \mathcal{X}^{<\alpha} \right)^\top \mathcal{Z}^{(\alpha)} \mathcal{X}^{>\alpha} \left((\mathcal{X}^{>\alpha})^\top \mathcal{X}^{>\alpha} \right)^{-1},$$

где $\mathrm{P}_\alpha^\perp$ обозначает ортопроектор на пространство ортогональное столбцам X_α^l, $\mathcal{Z}^{(\alpha)}$ обозначает развертку тензора $\mathcal{Z}$, имеющую размер $\mathbb{R}^{n_1 \ldots n_\alpha \times n_{\alpha+1} \ldots n_d}$, матрица $\mathcal{X}^{<\alpha} \in \mathbb{R}^{n_1 \ldots n_{\alpha-1} \times r_\alpha}$ определяется как

$$\mathcal{X}^{<\alpha}(\overline{i_1 \ldots i_{\alpha-1}}, :) = X_1(i_1) \ldots X_{\alpha-1}(i_{\alpha-1}),$$

матрица $\mathcal{X}^{>\alpha} \in \mathbb{R}^{r_\alpha \times n_{\alpha+1} \ldots n_d}$ определяется аналогично. Чтобы избежать вычисления $\left((\mathcal{X}^{>\alpha})^\top \mathcal{X}^{>\alpha} \right)^{-1}$ требуется дополнительная ортогонализация ядер [80].

1.7 Выводы по главе

В настоящей главе рассмотрены ключевые понятия, используемые в диссертации. Дано определение основных тензорных разложений, и поставлена задача о поиске собственных значений в тензорных форматах. Также дан обзор основных способов решения задач в тензорных форматах: на основе классических итерационных методов, с помощью римановой оптимизации и с помощью попеременной оптимизации. Приведены основные результаты о сходимости этих методов. Для ALS подхода получена явная формула для спектрального радиуса производной ALS оператора, из которой следует связь с мультипликативным методом Шварца и локальная сходимость метода.

Глава 2

Многомерные задачи на собственные значения с линейным оператором

Настоящая глава посвящена решению задачи на собственные значения с линейным оператором. Целью является поиск некоторого числа минимальных (максимальных) собственных значений и соответствующих собственных векторов, то есть решение частичной задачи на собственные значения.

Сначала рассматривается поиск одного целевого (минимального, максимального или ближайшего к заданному числу) собственного значения. Для этой задачи предлагается обобщение метода Якоби-Дэвидсона, базирующееся на идеях римановой оптимизации и малоранговых разложениях тензоров. Приводится анализ сходимости метода. Затем рассматривается ALS II метод, являющийся комбинацией ALS подхода и метода обратной итерации. Для него приводятся оценки локальной сходимости в случае $d = 2$. Для поиска нескольких собственных значений и векторов рассматривается обобщение метода LOBPCG на тензорный случай и нелинейный предобуславливатель, базирующийся на процедуре ALS минимизации.

2.1 Метод Якоби-Дэвидсона на малоранговых тензорных многообразиях

В настоящем разделе предлагается обобщение метода Якоби-Дэвидсона (JD) [126] для решения задачи (1.3). По аналогии с оригинальным JD методом мы выводим малоранговый аналог уравнения Якоби, а также предлагаем ма-

лоранговую версию ускорения с использованием подпространств. Предлагаемый подход наследует преимущества оригинального JD метода. По сравнению с итерацией Рэлея и методом Дэвидсона, предлагаемый метод эффективен и когда возникающие линейные системы решаются точно, и при их неточном решении. Также рассматривается малоранговая версия обратной итерации с адаптивными сдвигами (итерация Рэлея), которая естественным образом выводится из уравнения Якоби в методе Якоби-Дэвидсона.

Известно, что JD метод является римановым методом Ньютона на единичной сфере $\{x : \|x\| = 1\}$ с дополнительным ускорением с использованием подпространств [1]. Мы используем эту интерпретацию и получаем новый метод, как неточный метод Ньютона на пересечении сферы и многообразия тензоров фиксированного ранга. При выводе метода мы предполагаем, что матрица A является действительной и симметричной, однако, мы проверяем наш подход также и на несимметричных матрицах. В предположении, что матрица A также имеет малый TT-ранг, мы получаем сложность линейную по размерности задачи d и квадратичную по размеру мод $n = \max_i n_i$.

2.1.1 Минимизация отношения Рэлея на сфере

Первым ингредиентом оригинального JD метода является уравнение Якоби. Оно может быть получено как шаг риманова метода Ньютона на сфере [1]. Мы также будем использовать эту интерпретацию при выводе его малоранговой версии, поэтому для удобства сначала приведем вывод для случая сферы.

Пусть матрица $A \in \mathbb{R}^{n \times n}$ является симметричной. Целью является решить следующую задачу оптимизации

$$\mathfrak{R}(x) = x^\top A x \to \min, \tag{2.1}$$

при условии $x \in S^{n-1}$, где S^{n-1} является единичной сферой, рассматриваемой в качестве подмногообразия $\mathbb{R}^n$ с соответсвующей метрикой $g_x(\xi, \eta) = \xi^\top \eta$. Подход римановой оптимизации подразумевает оптимизацию отношения Рэлея $\mathfrak{R}(x)$ не на всем пространстве, а сразу на S^{n-1}, то есть ограничения уже учтены с помощью пространства, на котором ищется решение. Ортогональная проекция z на касательное пространство (см. определение 1.2) $T_x S^{n-1}$ сферы S^{n-1} в

точке x может быть записана как (1.20):

$$\mathrm{P}_{T_xS^{n-1}} z = (I - xx^\top)z. \tag{2.2}$$

Следовательно, риманов градиент функционала (2.1) имеет следующий вид

$$\operatorname{grad} \mathfrak{R}(x) = \mathrm{P}_{T_xS^{n-1}} \nabla \mathfrak{R}(x) = (I - xx^\top)(2Ax), \tag{2.3}$$

где ∇ обозначает евклидов градиент. Гессиан $\operatorname{Hess}_x : T_xS^{n-1} \to T_xS^{n-1}$ может быть получен как [2]:

$$\begin{aligned} \operatorname{Hess}_x \mathfrak{R}(x)[\xi] =& \mathrm{P}_{T_xS^{n-1}} \left(\mathrm{D} \left(\operatorname{grad} \mathfrak{R}(x)\right)[\xi]\right) = \\ & 2\mathrm{P}_{T_xS^{n-1}} \left(\mathrm{D}\left(\mathrm{P}_{T_xS^{n-1}} Ax\right)[\xi]\right) = \\ & 2\mathrm{P}_{T_xS^{n-1}} (A\xi + \dot{\mathrm{P}}_{T_xS^{n-1}} Ax), \quad \xi \in T_xS^{n-1}, \end{aligned} \tag{2.4}$$

где D обозначает производную по направлению (формула (1.16)) и

$$\dot{\mathrm{P}}_{T_xS^{n-1}} Ax \equiv \mathrm{D}(\mathrm{P}_{T_xS^{n-1}})[\xi]Ax = -(x^\top Ax)\xi - (\xi^\top Ax)x.$$

Поскольку $\mathrm{P}_{T_xS^{n-1}} x = 0$ и $\mathrm{P}_{T_xS^{n-1}} \xi = \xi$, мы получаем, что

$$\operatorname{Hess}_x \mathfrak{R}(x)[\xi] = 2\mathrm{P}_{T_xS^{n-1}} (A - (x^\top Ax)I) \mathrm{P}_{T_xS^{n-1}} \xi. \tag{2.5}$$

Запишем k-й шаг риманова метода Ньютона

$$\operatorname{Hess}_{x_k} \mathfrak{R}(x_k)[\xi_k] = -\operatorname{grad} \mathfrak{R}(x_k), \quad \xi_k \in T_{x_k}S^{n-1}, \tag{2.6}$$

с применением ретракции (определение 1.4):

$$x_{k+1} = \frac{x_k + \xi_k}{\|x_k + \xi_k\|}, \tag{2.7}$$

которая возвращает $x_k + \xi_k$ обратно на многообразие S^{n-1}. Используя (2.2), (2.3) и (2.5) мы можем переписать (2.6) как

$$(I - x_kx_k^\top)(A - \mathfrak{R}(x_k)I)(I - x_kx_k^\top)\xi_k = -r_k, \quad x_k^\top \xi_k = 0, \tag{2.8}$$

где

$$\mathfrak{R}(x_k) = x_k^\top Ax_k, \quad r_k = (I - x_kx_k^\top)Ax_k = Ax_k - \mathfrak{R}(x_k)x_k.$$

Уравнение (2.8) называется *уравнением Якоби* [126]. Отметим, что без использования проектора $(I - x_k x_k^\top)$ мы бы получили аналог уравнения из метода Дэвидсона

$$(A - \mathfrak{R}(x_k)I)\,\xi_k = -r_k,$$

которое имеет решение $\xi_k = -x_k$ коллинеарное текущему приближению x_k. По этой причине, на практике, уравнение Дэвидсона решается неточно. В частности, в оригинальном методе Дэвидсона [25] матрица A заменяется на диагональную матрицу $\mathrm{diag}(A)$. В то же время, у уравнения Якоби (2.8) такой проблемы не возникает, так как ортогональность x_k поддерживается автоматически. Причем, даже если уравнение Якоби (2.8) решается неточно с использованием итерационных методов на подпространствах Крылова, его решение ξ_k будет автоматически ортогонально x_k, что является важной особенностью для вычислений. Более того, так как JD метод имеет интерпретацию метода Ньютона, скорость его сходимости является сверхлинейной.

2.1.2 Уравнение Якоби на многообразиях фиксированного ранга

Обобщим теперь полученное уравнение Якоби (2.8) на случай многообразия малоранговых тензоров. Пусть $x \in \mathbb{R}^{n^d}$ является собственным вектором матрицы A и пусть $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ является его тензоризацией так, что $x = \mathrm{vec}(\mathcal{X})$. Для простоты считаем, что $n_1 = \cdots = n_d = n$. Мы также делаем предположение, что тензоризованный собственный вектор $\mathrm{vec}(\mathcal{X})$ приближенно имеет малый тензорный ранг $\mathbf{r}$. В этом случае для приближения, например, младшего собственного значения необходимо решить следующую задачу оптимизации

$$\begin{aligned} \mathfrak{R}(x) = x^\top A x \to \min, \\ x \in S^{n^d - 1} \cap \mathcal{M}_{\mathbf{r}}, \end{aligned} \tag{2.9}$$

где вместо (1.15) для удобства мы рассматриваем многообразие с векторизованными тензорами фиксированного ранга

$$\mathcal{M}_{\mathbf{r}} = \{\mathrm{vec}(\mathcal{X}),\, \mathcal{X} \in \mathbb{R}^{n \times \cdots \times n} : \text{TT-rank}(\mathcal{X}) = \mathbf{r}\},$$

которое также образует гладкое подмногообразие многообразия $\mathbb{R}^{n^d}$. Отметим, что можно было использовать и отношение Рэлея из (1.3), однако при поиске

решения сразу на сфере минимизируемый функционал становится квадратичным, что значительно упрощает формулы.

По аналогии с выводом уравнения Якоби мы пересекаем многообразие $\mathcal{M}_{\mathbf{r}}$ со сферой S^{n^d-1}. Как мы увидим из следующей теоремы пересечение $S^{n^d-1} \cap \mathcal{M}_{\mathbf{r}}$ является гладким многообразием $\mathbb{R}^{n^d}$, а следовательно оптимизационная задача (2.9) может быть решена с использованием подходов римановой оптимизации.

Теорема 2.1. *Пусть $\mathcal{N}_{\mathbf{r}} = S^{n^d-1} \cap \mathcal{M}_{\mathbf{r}}$, тогда $\mathcal{N}_{\mathbf{r}}$ образует гладкое подмногообразие $\mathbb{R}^{n^d}$ размерности*

$$\dim(\mathcal{N}_{\mathbf{r}}) = \dim(\mathcal{M}_{\mathbf{r}}) - 1.$$

Доказательство. Гладкость пересечения многообразий следует из трансверсальности $\mathcal{M}_{\mathbf{r}}$ и S^{n^d-1} как подмногообразий в $\mathbb{R}^{n^d}$. Для проверки трансверсальности достаточно показать, что [87]

$$T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}} + T_{\mathcal{X}}S^{n^d-1} = \mathbb{R}^{n^d}.$$

Действительно, $\dim(S^{n^d-1}) = n^d - 1$, значит достаточно найти вектор ξ такой, что $\xi \in T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}$, но $\xi \notin T_{\mathcal{X}}S^{n^d-1}$. Таким вектором является, например, $\xi = \mathrm{vec}(\mathcal{X})$. Итак, благодаря трансверсальности $\mathcal{N}_{\mathbf{r}}$ образует гладкое подмногообразие $\mathbb{R}^{n^d}$ размерности

$$\dim(\mathcal{M}_{\mathbf{r}}) + \dim(S^{n^d-1}) - \dim(\mathbb{R}^{n^d}) = \dim(\mathcal{M}_{\mathbf{r}}) - 1.$$

□

Найдем теперь вид ортопроектора на касательное пространство $\mathcal{N}_{\mathbf{r}}$. Для этого сначала докажем следующую лемму. Утверждение леммы не является новым результатом и приводится для полноты изложения.

Лемма 2.1. *Пусть $\mathcal{L}$ и $\mathcal{K}$ являются линейными подпространствами в $\mathbb{R}^n$. Пусть также P является ортопроектором на $\mathcal{L}$, а Q является ортопроектором на $\mathcal{K}$. Тогда, если P и Q коммутируют, то их композиция PQ является ортопроектором на $\mathcal{L} \cap \mathcal{K}$.*

Доказательство. Покажем, что PQ является ортопроектором. Действительно,

$$(PQ)(PQ) = P(QP)Q = P(PQ)Q = P^2Q^2 = PQ,$$

и

$$(PQ)^\top = Q^\top P^\top = QP = PQ.$$

Покажем теперь, что $\xi \in \mathcal{L} \cap \mathcal{K}$ тогда и только тогда, когда $PQ\xi = \xi$. Пусть $\xi \in \mathcal{L} \cap \mathcal{K}$, следовательно

$$PQ\xi = P\xi = \xi.$$

В обратную сторону, пусть $PQ\xi = \xi$, тогда

$$\xi = PQ\xi = P(PQ\xi) = P\xi \in \mathcal{L},$$
$$\xi = QP\xi = Q(QP\xi) = Q\xi \in \mathcal{K},$$

и значит $\xi \in \mathcal{L} \cap \mathcal{K}$. □

Утверждение 2.1. *Ортопроектор* $\mathrm{P}_{T_\mathcal{X}\mathcal{N}_\mathbf{r}}$ *на* $T_\mathcal{X}\mathcal{N}_\mathbf{r}$ *в точке* $\mathrm{vec}(\mathcal{X}) \in \mathcal{N}_\mathbf{r}$ *может быть записан как*

$$\begin{aligned} \mathrm{P}_{T_\mathcal{X}\mathcal{N}_\mathbf{r}} &= \mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}\mathrm{P}_{T_\mathcal{X}S^{n^d-1}} = \mathrm{P}_{T_\mathcal{X}S^{n^d-1}}\mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}} \\ &= \mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}} - \mathrm{vec}(\mathcal{X})\mathrm{vec}(\mathcal{X})^\top, \end{aligned} \tag{2.10}$$

где $\mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}$ *является ортопроектором на касательное пространство* $\mathcal{M}_\mathbf{r}$.

Доказательство. Заметим, что

$$\mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}\mathrm{vec}(\mathcal{X}) = \mathrm{vec}(\mathcal{X}),$$

и, как следствие,

$$\mathrm{vec}(\mathcal{X})^\top = \mathrm{vec}(\mathcal{X})^\top\mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}.$$

Следовательно,

$$\begin{aligned} \mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}\mathrm{P}_{T_\mathcal{X}S^{n^d-1}} &= \mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}(I - \mathrm{vec}(X)\mathrm{vec}(X)^\top) \\ &= \mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}} - \mathrm{vec}(X)\mathrm{vec}(X)^\top = \mathrm{P}_{T_\mathcal{X}S^{n^d-1}}\mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}. \end{aligned}$$

Значит, ортопроекторы $\mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}$ и $\mathrm{P}_{T_\mathcal{X}S^{n^d-1}}$ коммутируют и согласно Лемме 2.1 их композиция является ортопроектором на пересечение $T_\mathcal{X}\mathcal{M}_\mathbf{r}$ и $T_\mathcal{X}S^{n^d-1}$. □

Вывод уравнения Якоби на многообразии $\mathcal{N}_{\mathbf{r}}$. Выведем обобщение оригинального уравнения Якоби, являющегося одним шагом метода Ньютона на $\mathcal{N}_{\mathbf{r}}$. Используя (2.10) и вводя обозначение $x = \mathrm{vec}(X)$ получим

$$\operatorname{grad} \mathfrak{R}(x) = \mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} \nabla \mathfrak{R}(x) = \mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} (I - xx^{\top}) \nabla \mathfrak{R}(x) = 2\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} (I - xx^{\top}) Ax. \tag{2.11}$$

По аналогии с (2.4) и используя (2.10) также получим

$$\begin{aligned} \operatorname{Hess}_{\mathcal{X}} \mathfrak{R}(x)[\xi] =& 2\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} (A\xi + \dot{\mathrm{P}}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} Ax) = \\ & 2\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} (A\xi - x\xi^{\top} Ax - \xi x^{\top} Ax + \dot{\mathrm{P}}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} Ax), \\ & \xi \in T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}. \end{aligned}$$

Согласно (2.10) $\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} x = \mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} \mathrm{P}_{T_{\mathcal{X}}S^{nm-1}} x = 0$, таким образом

$$\begin{aligned} \operatorname{Hess}_{\mathcal{X}} \mathfrak{R}(x)[\xi] = & 2\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} (A - (x^{\top}Ax)I)\xi + \mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} \dot{\mathrm{P}}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} Ax = \\ & 2\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} (I - xx^{\top})(A - (x^{\top}Ax)I)\xi + \mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} \dot{\mathrm{P}}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} Ax, \end{aligned}$$

где часть $\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} \dot{\mathrm{P}}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} Ax$ соответствует кривизне многообразия малоранговых тензоров. Этот член содержит обращения сингулярных чисел разверток. Сингулярные числа, в свою очередь, могут быть очень малы, если ранг для построения многообразия оказался больше, чем истинный ранг решения. Это, в свою очередь, ведет к неустойчивости при численных расчетах. По аналогии [80] мы пренебрегаем членом, ответственным за кривизну и получаем неточный метод Ньютона, который можно также интерпретировать как метод Гаусс-Ньютона для задач оптимизации с ограничениями. Действительно, отбрасывание члена связанного с кривизной эквивалентно линеаризации многообразия и, как следствие, ограничения. Итак, отбрасывая $\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}} \dot{\mathrm{P}}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} Ax$ получим

$$\operatorname{Hess}_{\mathcal{X}} \mathfrak{R}(x)[\xi] \approx 2\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} (I - xx^{\top})(A - \mathfrak{R}(x)I)\xi.$$

После симметризации

$$\operatorname{Hess}_{\mathcal{X}} \mathfrak{R}(x)[\xi] \approx 2\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} (I - xx^{\top})(A - \mathfrak{R}(x)I)(I - xx^{\top})\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} \xi. \tag{2.12}$$

Используя (2.11) и (2.12) мы можем переписать линейные системы, возникающие в неточном методе Ньютона как

$$\begin{gathered} (I - xx^{\top}) \left[\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} (A - \mathfrak{R}(x)I) \mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} \right] (I - xx^{\top})\xi = -\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}} (I - xx^{\top}) Ax, \\ \xi^{\top} x = 0, \quad \xi \in T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}. \end{gathered} \tag{2.13}$$

Уравнение (2.13) имеет вид схожий с оригинальным уравнением Якоби (2.8) с оператором $(A - \mathfrak{K}(x)I)$, спроецированным на $T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}$.

Ретракция. По аналогии с (2.7) после того, как мы нашли решение ξ из (2.27), нам необходимо отобразить вектор $x+\xi$ из касательного пространства на многообразие. Следующее утверждение дает явную формулу для ретракции на многообразие $\mathcal{N}_{\mathbf{r}}$.

Утверждение 2.2. *Пусть $R_{\mathbf{r}}$ является ретракцией с касательного расслоения $T\mathcal{M}_{\mathbf{r}}$ на $\mathcal{M}_{\mathbf{r}}$, тогда*

$$R(x,\xi) = \frac{R_{\mathbf{r}}(x,\xi)}{\|R_{\mathbf{r}}(x,\xi)\|}, \tag{2.14}$$

является ретракцией на $\mathcal{N}_{\mathbf{r}}$.

Доказательство. Убедимся, что R удовлетворяет определению ретракции 1.4:

1. Гладкость в окрестности нулевого элемента на $T\mathcal{N}_{\mathbf{r}}$;
2. $R(x,0) = x$ для всех $x \in \mathcal{N}_{\mathbf{r}}$;
3. $\frac{d}{dt}R(x,t\xi)\Big|_{t=0} = \xi$ для всех $x \in \mathcal{N}_{\mathbf{r}}$ и $\xi \in T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}$.

Первое свойство следует из гладкости ретракции $R_{\mathbf{r}}$. Второе свойство выполняется, так как $R_{\mathbf{r}}(x,0) = x$ и $\|x\| = 1$ для $x \in \mathcal{N}_{\mathbf{r}}$. Проверим третье свойство:

$$\begin{gathered}\frac{d}{dt}R(x,t\xi)\Big|_{t=0} = \frac{d}{dt}\left(\frac{R_{\mathbf{r}}(x,t\xi)}{\|R_{\mathbf{r}}(x,t\xi)\|}\right)\Bigg|_{t=0} = \\ \frac{\frac{d}{dt}R_{\mathbf{r}}(x,t\xi)\big|_{t=0}\,\|R_{\mathbf{r}}(x,t\xi)|_{t=0}\| - \frac{d}{dt}\,\|R_{\mathbf{r}}(x,t\xi)\|_{t=0}\,R_{\mathbf{r}}(x,t\xi)|_{t=0}}{\|\,R_{\mathbf{r}}(x,t\xi)|_{t=0}\,\|^2}.\end{gathered} \tag{2.15}$$

Поскольку $(x,\xi) = 0$ для $x \in \mathcal{N}_{\mathbf{r}}$, получаем

$$\begin{aligned}\frac{d}{dt}\|R_{\mathbf{r}}(x,t\xi)\|\Big|_{t=0} &= \frac{\left(\frac{d}{dt}R_{\mathbf{r}}(x,t\xi), R_{\mathbf{r}}(x,t\xi)\right)\Big|_{t=0} + \left(R_{\mathbf{r}}(x,t\xi), \frac{d}{dt}R_{\mathbf{r}}(x,t\xi)\right)\Big|_{t=0}}{2\,\|R_{\mathbf{r}}(x,t\xi)|_{t=0}\|} \\ &= \frac{(\xi,x)+(x,\xi)}{2\|x\|} = 0, \quad x \in \mathcal{N}_{\mathbf{r}}, \quad \xi \in T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}.\end{aligned}$$

Подставляя последние выражение в (2.15) и учитывая, что

$$\|R_{\mathbf{r}}(x,t\xi)|_{t=0}\| = \|R_{\mathbf{r}}(x,0)\| = \|x\| = 1,$$

получим $\frac{d}{dt}R(x,t\xi)\Big|_{t=0} = \xi$, что завершает доказательство. □

Замечание 2.1. *Ретракция* (2.14) *является композицией двух ретракций: сначала на малоранговое многообразие* $\mathcal{M}_{\mathbf{r}}$ *и затем на сферу* S^{n^d-1}. *Отметим, что композиция в обратном порядке ретракцией не является, так как не отображает вектор из касательного пространства на многообразие* $\mathcal{N}_{\mathbf{r}}$.

Стандартной ретракцией на $\mathcal{M}_{\mathbf{r}}$ является [3]

$$R_{\mathbf{r}}(x, \xi) \equiv R_{\mathbf{r}}(x + \xi) = \mathrm{P}_{\mathcal{M}_{\mathbf{r}}}(x + \xi),$$

где

$$\mathrm{P}_{\mathcal{M}_{\mathbf{r}}}(x + \xi) \equiv \arg\min_{y \in \mathcal{M}_{\mathbf{r}}} \|y - (x + \xi)\|.$$

Для достаточно малых поправок ξ ретракция может быть вычислена с использованием обычной операции округления тензоров, смотри раздел 1.2.

Свойства локальной системы. Отметим несколько важных свойств матрицы линейной системы (2.13). Предположим, что мы ищем наименьшее собственное число $\lambda_1 > 0$, и текущее $\mathfrak{R}(x)$ ближе к λ_1, чем к следующему собственному значению λ_2.

Легко заметить, что оператор $(I - xx^\top)\left[\mathrm{P}_{T_X\mathcal{M}_{\mathbf{r}}}(A - \mathfrak{R}(x)I)\mathrm{P}_{T_X\mathcal{M}_{\mathbf{r}}}\right](I - xx^\top)$ имеет ненулевое ядро. В то же время этот оператор является положительно определенным на подпространстве

$$\xi^\top x = 0, \quad \xi \in T_X\mathcal{M}_{\mathbf{r}}. \tag{2.16}$$

На самом деле,

$$\begin{aligned} &\min_{\substack{z \in T_X\mathcal{N}_{\mathbf{r}}, \\ \|z\|=1}} \left\langle z, (I - xx^\top)\left[\mathrm{P}_{T_X\mathcal{M}_{\mathbf{r}}}(A - \mathfrak{R}(x)I)\mathrm{P}_{T_X\mathcal{M}_{\mathbf{r}}}\right](I - xx^\top) z\right\rangle = \\ &\min_{\substack{z \in T_X\mathcal{N}_{\mathbf{r}}, \\ \|z\|=1}} \langle z, (A - \mathfrak{R}(x)I)\, z\rangle \geq \min_{\substack{z \perp x, \\ \|z\|=1}} \langle z, (A - \mathfrak{R}(x)I)\, z\rangle \geq \lambda_1 + \lambda_2 - 2\mathfrak{R}(x). \end{aligned} \tag{2.17}$$

Последнее неравенство следует из [97, Лемма 3.1]. Следовательно, матрица является положительно определенной на линейном подпространстве (2.16).

Покажем, что число обусловленности у оператора

$$(I - xx^\top)\left[\mathrm{P}_{T_X\mathcal{M}_{\mathbf{r}}}(A - \mathfrak{R}(x)I)\mathrm{P}_{T_X\mathcal{M}_{\mathbf{r}}}\right](I - xx^\top)$$

не ухудшается, когда $\mathfrak{R}(x)$ сходится к точному собственному значению. Число обусловленности определяется как

$$\kappa = \frac{\max\limits_{\substack{z \in T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}} \\ \|z\|=1}} q(z)}{\min\limits_{\substack{z \in T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}} \\ \|z\|=1}} q(z)}, \quad q(z) = \frac{\left\langle z, (I - xx^{\top})\left[\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}}(A - \mathfrak{R}(x)I)\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}}\right](I - xx^{\top})z\right\rangle}{\langle z, z\rangle}.$$

Аналогично с (2.17) можно показать, что

$$\kappa \leq \frac{\max\limits_{\substack{z: z \perp x, \\ z \neq 0}} q(z)}{\min\limits_{\substack{z: z \perp x, \\ z \neq 0}} q(z)}.$$

Последнее выражение может быть оценено с помощью результатов для оригинального уравнения Якоби [97], в случае которого число обусловленности не растет при сходимости $\mathfrak{R}(x)$ к точному собственному значению λ_1.

2.1.3 Ускорение с использованием подпространств

Поскольку предложенный метод Ньютона является неточным, необходимо дополнительно применить линейный поиск вдоль найденного направления

$$x_{\text{new}} = R(x + \tau_{\text{opt}}\xi), \tag{2.18}$$

где

$$\tau_{\text{opt}} = \arg\min_{\tau} \mathfrak{R}(R(x + \tau\xi)).$$

Коэффициент τ_{opt} может быть приближен с помощью правила Армихо [1]. Итерационный метод без ускорения с использованием подпространств сформулирован в Алгоритме 2.1.

Для ускорения сходимости можно использовать векторы, полученные на предыдущей итерации, как это делается в оригинальном методе Якоби-Дэвидсона. Однако, для избежания неустойчивости и для уменьшения вычислительной стоимости, мы используем понятие *векторного переноса* [1]. На каждой итерации мы проецируем базис, полученный на предыдущих итерациях, но на касательном пространстве текущего приближения. Рассмотрим описанную процедуру более детально.

Алгоритм 2.1 Метод Якоби-Дэвидсона на малоранговых многообразиях без ускорения на подпространствах

Require: Матрица A; начальное приближение x_0, ранг r

Ensure: λ и x – аппроксимация минимального собственного значения и соответствующего собственного вектора

1: **for** $k = 1, 2, \ldots$ до сходимости **do**

2: Решить (2.13): $\mathrm{P}_{T_{\mathcal{X}_k}\mathcal{N}_{\mathbf{r}}}(A - \mathfrak{K}(x_k))\mathrm{P}_{T_{\mathcal{X}_k}\mathcal{N}_{\mathbf{r}}}\xi_k = -\operatorname{grad}\mathfrak{K}(x_k)$

3: Найти τ_k: $\tau_k = \arg\min_\tau \mathfrak{K}(x_k + \tau\xi_k)$

4: Найти минимальный натуральный l:

$$\mathfrak{K}(x^{(k)}) - \mathfrak{K}(R(x^{(k)}, 2^{-l}\tau_k\xi_k)) \geq -c2^{-l}\left\langle \xi_k, \operatorname{grad}\mathfrak{K}(x^{(k)})\right\rangle$$

5: $x^{(k+1)} = \mathfrak{K}(R(x^{(k)}, 2^{-l}\tau_k\xi_k)$

6: **end for**

7: $\lambda \approx \mathfrak{K}(x^{(k+1)})$, $x \approx x^{(k+1)}$

После k итераций мы имеем базис $\mathcal{V}_{b-1} = [v_1, \ldots, v_{b-1}]$, $b \leq k$ и проецируем его на $T_{\mathcal{X}_k}\mathcal{M}_{\mathbf{r}}$:

$$\widetilde{\mathcal{V}}_{b-1} = [\mathrm{P}_{T_{\mathcal{X}_k}\mathcal{M}_{\mathbf{r}}}v_1, \ldots, \mathrm{P}_{T_{\mathcal{X}_k}\mathcal{M}_{\mathbf{r}}}v_{b-1}].$$

При необходимости можно дополнительно ортогонализовать столбцы $\widetilde{\mathcal{V}}_{b-1}$. Отметим, что ортогонализация в касательном пространстве является эффективной операцией, так как линейная комбинация любого числа векторов из касательного пространства максимум имеет ранг $2\mathbf{r}$. После того, как решение ξ_k уравнения (2.13) найдено, следующим шагом является дополнить базис $\widetilde{\mathcal{V}}_{b-1}$ вектором v_b, полученным с помощью ортогонализации ξ_k к $\widetilde{\mathcal{V}}_{b-1}$:

$$\mathcal{V}_b = [\widetilde{\mathcal{V}}_{b-1}, v_b] \tag{2.19}$$

Новое приближение к x вычисляется с помощью процедуры Рэлея-Ритца. А именно, мы вычисляем $\mathcal{V}_b^\top A\mathcal{V}_b$ и затем находим собственную пару (θ, c):

$$\mathcal{V}_b^\top A\mathcal{V}_b\, c = \theta c, \tag{2.20}$$

соответствующую искомому собственному значению. В итоге, вектор Ритца c дает нам новое приближение к x:

$$x_{k+1} = \mathcal{V}_b\, c.$$

Обратим внимание на то, что столбцы $\mathcal{V}_b$ принадлежат $T_{\mathcal{X}_k}\mathcal{M}_{\mathbf{r}}$, следовательно не возникает проблемы роста ранга. Если же нужно поддерживать фиксированный ранг $\mathbf{r}$, необходимо оптимизировать коэффициенты c:

$$x_{k+1} = R(\mathcal{V}_b c_{\text{opt}}), \quad c_{\text{opt}} = \underset{c_1,\ldots,c_b}{\arg\min}\, \mathfrak{K}(R(\mathcal{V}_b c)).$$

Задача оптимизации может быть решена, например, с использованием линейного поиска по каждому c_i последовательно, начиная с начального приближения, найденного в (2.20). Однако для уменьшения сложности, необходимо оптимизировать только коэффициент при v_b или просто использовать c вместо c_{opt}.

2.1.4 Сходимость метода

Для простоты рассмотрим случай $d = 2$. Для того, чтобы показать сходимость метода, воспользуемся общей теорией римановой оптимизации. В разделе 1.6 была сформулирована теорема 1.4 о глобальной сходимости метода в котором задана последовательность градиентных направлений, и в котором шаги выбираются по правилу Армихо. Покажем, что последовательность ξ_k, построенная с помощью последовательности решений (2.13), является градиентной. Для этого воспользуемся свойством (2.17) положительной определенности $\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}}(A - \mathfrak{K}(x))\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}}$ на подпространстве $\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}}\xi = \xi$:

$$\langle \xi, \operatorname{grad} \mathfrak{K}(x)\rangle = -\left\langle \xi, \mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}}(A - \mathfrak{K}(x))\mathrm{P}_{T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}}\xi \right\rangle < -(\lambda_1 + \lambda_2 - 2\mathfrak{K}(x))\|\xi\|^2.$$

Однако многообразие $\mathcal{N}_{\mathbf{r}}$ не является компактным в случае, когда ранг тензоров больше 1. Это означает, что мы, вообще говоря, не можем говорить о существовании предельных точек на многообразии. Действительно, не сложно построить последовательность точек многообразия $\mathcal{M}_{\mathbf{r}} \bigcap S^{n^d-1}$, сходящуюся к точке меньшего ранга за исключением случая $r = 1$. В последнем случае последовательность матриц с Фробениусовой нормой равной 1 не может сойтись к нулевой матрице. Из вышесказанного следует следующая теорема:

Теорема 2.2. *Пусть* $\mathbf{r} = \{1, 1, \ldots, 1\}$, *тогда любая предельная точка* x_* *последовательности* $\{x^{(k)}\}$ *из Алгоритма 2.1 принадлежит* $\mathcal{M}_r$ *и удовлетворяет* $\operatorname{grad} \mathfrak{K}(x_*) = 0$.

Для большего значения ранга многообразие не является замкнутым подмножеством в $\mathbb{R}^{n^d}$, поэтому мы не можем утверждать, что предельная точка ему принадлежит. В этом случае можно регуляризовать функционал с помощью нормы псевдообратной матрицы так, что последовательность остается на компактном множестве внутри многообразия [139]:

$$\mathfrak{R}_\mu(x) = \mathfrak{R}(x) + \mu^2 \|X^+\|_F^2, \quad x = \mathrm{vec}(X). \tag{2.21}$$

В случае $d > 2$ функционал регуляризируется с помощью суммы норм псевдообратных матриц развертки многомерного массива.

Теорема 2.3. *Любая предельная точка $x_*^{(\mu)}$ последовательности $\{x^{(k)}\}$ из Алгоритма 2.1 с регуляризированным функционалом $\mathfrak{R}_\mu$ из* (2.21) *вместо $\mathfrak{R}$ принадлежит $\mathcal{M}_r$ и удовлетворяет* $\mathrm{grad}\ \mathfrak{R}_\mu(x_*^{(\mu)}) = 0$.

Доказательство. Поскольку шаги выбираются из условия Армихо, мы имеем $\mathfrak{R}_\mu(x_k) \leq \mathfrak{R}_\mu(x_0)$:

$$\mathfrak{R}(x_k) + \mu^2 \|X_k^+\|_F^2 \leq \mathfrak{R}_\mu(x_0) \equiv C^2$$

Значит, $\|X_k^+\|_F^2 \leq C/\mu^2$. С другой стороны каждый X_k принадлежит единичной сфере $\|X_k\|_F^2 = 1$. В итоге имеем,

$$\sigma_r(X_k) \geq \frac{1}{\|X_k^+\|_F^2} \geq \frac{C}{\mu^2}$$

и

$$\sigma_1(X_k) \leq \|X_k\|_F^2 = 1.$$

То есть итерации остаются внутри компактного множества

$$\{X : \mathrm{rank}(X) = r,\ \sigma_r(X) \geq C/\mu^2,\ \sigma_1(X) \leq 1\}.$$

Из условия непрерывности градиента в предельной точке выполняется $\mathrm{grad}\ \mathfrak{R}_\mu(x_*) = 0$. □

На практике не удалось найти практического примера, когда у нерегуляризованного функционала были бы проблемы со сходимостью, даже если ранг решения был меньше ранга, с помощью которого строилось многообразие. Однако для надежности можно использовать регуляризированную версию, так как она не меняет асимптотическую сложность вычислений.

Отметим также, что если в регуляризированной версии минимальное сингулярное число ограничено снизу при $\mu \to 0$, то градиент регуляризационной поправки мал и мы нашли точку $x_*^{(\mu)}$, у которой $\operatorname{grad} \Re(x_*^{(\mu)}) \approx 0$. Аккуратное исследование зависимости минимального сингулярного числа от μ является нетривиальной задачей и не рассматривается в настоящей работе. Однако для иллюстрации рассмотрим пример для функционала более простого вида, чем отношение Рэлея:

$$\mathfrak{J}(X) = \|X\|_F^2,$$

и соответствующий регуляризированный функционал:

$$\mathfrak{J}_\mu(X) = \|X\|_F^2 + \mu^2 \|X^+\|_F^2$$

на многообразии матриц ранга фиксированного ранга r, $r \geq 1$. Легко показать, что единственная точка X_*, удовлетворяющая $\operatorname{grad} \mathfrak{J}(X_*) = 0$ является $X_* = 0$ и, значит, имеет ранг 0. Покажем, что точки $X_*^{(\mu)}$: $\operatorname{grad} \mathfrak{J}_\mu(X_*^{(\mu)}) = 0$ удовлетворяют

$$\lim_{\mu \to 0} \mu^2 \operatorname{grad} \|X_*^{(\mu)^+}\|_F^2 = 0.$$

Найдем риманов градиент регуляризированного члена $\|X^+\|_F^2$. Для этого сначала получим выражение для евклидова градиента с использованием формулы для производной псевдообратной из [128]:

$$\begin{aligned}
&\delta \|X^+\|_F^2 = 2 \left\langle X^+, \delta(X^+) \right\rangle = \\
&2 \left\langle X^+, -X^+ \delta(X) X^+ + X^+ X^{+\top} \delta(X^\top)(I - X X^+) + (I - X X^+) \delta(X^\top) X^{+\top} X^+ \right\rangle = \\
&2 \left\langle X^+, -X^+ \delta(X) X^+ \right\rangle = -2 \operatorname{trace}(X^{+\top} X^+ \delta(X) X^+) = \\
&-2 \operatorname{trace}((X^{+\top} X^+ X^{+\top})^\top \delta(X)) = -2 \left\langle X^{+\top} X^+ X^{+\top}, \delta X \right\rangle.
\end{aligned}$$

Значит,

$$\nabla \left(\mu^2 \|X^+\|_F^2 \right) = -2\mu^2 X^{+\top} X^+ X^{+\top} = -2\mu^2 U \Sigma^{-3} V^T,$$

где $X = U \Sigma V^T$ — сингулярное разложение X. В итоге,

$$0 = \operatorname{grad} \mathfrak{J}_\mu(X_*^{(\mu)}) = U \Sigma_*^{(\mu)} V^\top - 2\mu^2 U (\Sigma_*^{(\mu)})^{-3} V^T,$$

откуда

$$\Sigma_*^{(\mu)} = \sqrt{\mu} I_r$$

и в результате

$$\mu^2 \operatorname{grad} \|X^+\|_F^2 = -2\mu^2 U(\sqrt{\mu} I_r)^{-3} V^\top = -2\sqrt{\mu} U V^\top \to 0, \quad \mu \to 0.$$

Подробное изложение теории регуляризации для задач математической физики можно найти в [146].

2.1.5 Связь с обратной итерацией

Если линейная система (2.8) решается точно, то известно, что классический метод Якоби-Дэвидсона без ускорения с использованием подпространств эквивалентен [126] обратной итерации с адаптивным сдвигом (итерация Рэлея):

$$\begin{aligned} &(A - \mathfrak{R}(x_k) I)\tilde{x} = x_k, \\ &x_{k+1} = \frac{\tilde{x}}{\|\tilde{x}\|}. \end{aligned} \tag{2.22}$$

Выведем аналог итерации Рэлея для случая малоранговых многообразий. Для это рассмотрим случай, когда (2.13) решается точно. На k-й итерации уравнение (2.13) имеет вид

$$\begin{gathered} (I - x_k x_k^\top) \mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} (A - \mathfrak{R}(x_k) I) \mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} \xi_k = -\mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} (I - x_k x_k^\top) A x_k, \\ \mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} \xi_k = \xi_k, \quad x_k^\top \xi_k = 0. \end{gathered}$$

Следовательно,

$$\mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} (A - \mathfrak{R}(x_k) I) \mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} \xi_k - \alpha x_k = -\mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} (A - \mathfrak{R}(x_k) I) x_k,$$

где

$$\alpha = x_k^\top \left[\mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} (A - \mathfrak{R}(x_k) I) \mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} \right] \xi_k.$$

Вводя обозначение $\tilde{x} = x_k + \xi_k$, получим

$$\begin{aligned} &\left[\mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} (A - \mathfrak{R}(x_k) I) \mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} \right] \tilde{x} = x_k, \quad \mathrm{P}_{T_{\mathcal{X}_k} \mathcal{M}_\mathbf{r}} \tilde{x} = \tilde{x}, \\ &x_{k+1} = R(\tilde{x}). \end{aligned} \tag{2.23}$$

где мы опустили параметр α в силу того, что $R(\alpha \tilde{x}) = R(\tilde{x})$. Таким образом, (2.23) является обобщением обратной итерации с адаптивными сдвигами (итерации

Рэлея) (2.22) на случай малоранговых многообразий и, следовательно, является методом Гаусса-Ньютона.

Можно предположить, что метод Якоби-Дэвидсона сходится быстрее, чем итерация Рэлея (2.23), когда системы решаются неточно. Как мы показали в разделе 2.1.2 число обусловленности локальных систем не ухудшается при сходимости $\mathfrak{R}(x_k)$ к точному собственному значению. Это свойство положительно сказывается на сходимости, что было исследовано в случае оригинального метода Якоби-Дэвидсона [98]. Мы подкрепляем данный факт численными экспериментами в разделе 2.1.7.

2.1.6 Решение системы в касательном пространстве

Уравнение (2.13) может быть решено итерационным методом. Для $\xi \in T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}}$ умножение на матрицу

$$(I - xx^{\top})\mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}}(A - \mathfrak{R}(x)I)\xi$$

состоит из нескольких частей:

1. $\eta_1 = (A - \mathfrak{R}(x)I)\xi$ — умножение TT-матрицы на TT-тензор;
2. $\eta_2 = \mathrm{P}_{T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}}\eta_1$ — оротопроекция на $T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}$, TT-rank$(\eta_2) \leq 2\mathbf{r}$;
3. $\eta = \eta_2 - (x^{\top}\eta_2)x$ — ортопроекция на $T_{\mathcal{X}}S^{n^d-1}$, TT-rank$(\eta) \leq 2\mathbf{r}$.

Отметим, что так как ортопроектор на касательное пространство действует на все ядра разложения одновременно, возможна эффективная параллелизация матрично-векторного умножения по ядрам.

Если локальные системы плохо обусловлены, то для быстрой сходимости итерационного процесса необходимо использовать предобуславливатель. Систему можно записать в блочном виде с векторизованными неизвестными ядрами. Далее в качестве предобуславливателя можно использовать блочно-диагональный предобуславливатель Якоби. Отметим, что блочный предобуславливатель Якоби эквивалентен ALS итерации, примененной для каждого ядра независимо. Для решении системы с предобуславливателем мы фокусируемся на двумерном случае. Дело в том, что даже в двумерном случае необхо-

димо использовать специальную параметризацию, явно выделяющую матрицу сингулярных чисел. Разработка параметризации для многомерного случая является нетривиальной задачей и будет рассмотрена автором в будущих работах.

Локальная система для $d = 2$. Рассмотрим подробно построение локальной системы и блочного предобуславливателя в двумерном случае. Сначала обсудим вопрос параметризации. Верно следующее утверждение.

Утверждение 2.3. *Касательное пространство $\mathcal{N}_{\mathbf{r}}$ в точке $\mathrm{vec}(X) \in \mathcal{N}_{\mathbf{r}}$ с матрицей X, заданной в виде SVD разложения: $X = USV^\top$, $U^\top U = I$, $V^\top V = I$, $S = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$, $\sigma_1 \geq \cdots \geq \sigma_r > 0$, может быть параметризовано следующим образом*

$$T_{\mathcal{X}}\mathcal{N}_{\mathbf{r}} = \{\mathrm{vec}(U_\xi V^\top + UV_\xi^\top + US_\xi V^\top) : U_\xi \perp U,\, V_\xi \perp V,\, \mathrm{vec}(S_\xi) \perp \mathrm{vec}(S)\}.$$

Доказательство. Вектор $\xi \in T_{\mathcal{X}}\mathcal{M}_r$ может быть параметризован [139] как

$$\xi = \mathrm{vec}(U_\xi V^\top + UV_\xi^\top + US_\xi V^\top) \tag{2.24}$$

с условиями калибровки

$$U_\xi \perp U, \quad V_\xi \perp V. \tag{2.25}$$

Для получения параметризации $\xi \in T_X S^{n^2-1} \cap T_{\mathcal{X}}\mathcal{M}_r$ нам необходимо учесть, что $\xi \in T_X S^{n^2-1}$ и, следовательно, $\xi^\top x = 0$, что дает дополнительное условие калибровки

$$\mathrm{vec}(S_\xi) \perp \mathrm{vec}(S). \tag{2.26}$$

Что и требовалось доказать. □

Теперь получим явное выражение для локальной системы, для которой затем будет построен блочный предобуславливатель.

Утверждение 2.4. *Решение* (2.13)*, записанное в виде*

$$\xi = \mathrm{vec}(U_\xi V^\top + UV_\xi^\top + US_\xi V^\top),$$

может быть найдено из локальной линейной системы

$$(I - BB^\top)(A - \mathfrak{R}(x)I)_{\mathrm{loc}}(I - BB^\top)\tau_\xi = -(I - BB^\top)g, \quad B^\top \tau_\xi = 0, \tag{2.27}$$

где[1]

$$\tau_\xi = \begin{bmatrix} \operatorname{vec}(U_\xi) \\ \operatorname{vec}(V_\xi^\top) \\ \operatorname{vec}(S_\xi) \end{bmatrix}, \quad g = \begin{bmatrix} A_{v,v} \operatorname{vec}(US) \\ A_{u,u} \operatorname{vec}(SV^\top) \\ A_{vu,vu} \operatorname{vec}(S) \end{bmatrix}, \quad B = \begin{bmatrix} I_r \otimes U & 0 & 0 \\ 0 & V \otimes I_r & 0 \\ 0 & 0 & \operatorname{vec}(S) \end{bmatrix},$$

$$(A - \mathfrak{K}(x)I)_{\text{loc}} = \begin{bmatrix} (A - \mathfrak{K}(x)I)_{v,v} & (A - \mathfrak{K}(x)I)_{v,u} & (A - \mathfrak{K}(x)I)_{v,uv} \\ (A - \mathfrak{K}(x)I)_{u,v} & (A - \mathfrak{K}(x)I)_{u,u} & (A - \mathfrak{K}(x)I)_{u,vu} \\ (A - \mathfrak{K}(x)I)_{vu,v} & (A - \mathfrak{K}(x)I)_{vu,u} & (A - \mathfrak{K}(x)I)_{vu,vu} \end{bmatrix},$$

Доказательство. Заметим, что $\mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}}$ является суммой трех ортопроекторов

$$\mathrm{P}_{T_\mathcal{X}\mathcal{M}_\mathbf{r}} = \mathrm{P}_1 + \mathrm{P}_2 + \mathrm{P}_3,$$
$$\mathrm{P}_1 = VV^\top \otimes (I_n - UU^\top), \quad \mathrm{P}_2 = (I_m - VV^\top) \otimes UU^\top, \quad \mathrm{P}_3 = VV^\top \otimes UU^\top$$

Поскольку $\mathrm{P}_i\mathrm{P}_j = \mathrm{O}$, $i \neq j$ и $\mathrm{P}_i^2 = \mathrm{P}_i$ мы получаем

$$\begin{bmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{bmatrix} (I - xx^\top)(A - \mathfrak{K}(x)I)(I - xx^\top) \begin{bmatrix} \mathrm{P}_1 & \mathrm{P}_2 & \mathrm{P}_3 \end{bmatrix} \begin{bmatrix} \mathrm{P}_1\xi \\ \mathrm{P}_2\xi \\ \mathrm{P}_3\xi \end{bmatrix} = \begin{bmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{bmatrix} (I - xx^\top)Ax. \quad (2.28)$$

Легко убедиться, что

$$\mathrm{P}_1(I - xx^\top) = \mathrm{P}_1 = (V \otimes I_n)(V^\top \otimes (I_n - UU^\top)),$$
$$\mathrm{P}_2(I - xx^\top) = \mathrm{P}_2 = (I_m \otimes U)((I_m - VV^\top) \otimes U^\top),$$
$$\mathrm{P}_3(I - xx^\top) = (VV^\top \otimes UU^\top)(I - (V \otimes U)\operatorname{vec}(S)(\operatorname{vec}(S))^\top (V^\top \otimes U^\top)) =$$
$$(V \otimes U)(I_{r^2} - \operatorname{vec}(S)(\operatorname{vec}(S))^\top)(V^\top \otimes U^\top).$$

Найдем теперь способ параметриации. Вектор $\xi \in T_\mathcal{X}\mathcal{M}_\mathbf{r}$ может быть параметризован [139] как

$$\xi = \operatorname{vec}(U_\xi V^\top + UV_\xi^\top + US_\xi V^\top) \quad (2.29)$$

с дополнительными условиями калибровки

$$U_\xi \perp U, \quad V_\xi \perp V. \quad (2.30)$$

[1] Для $nm \times nm$ матрицы C мы ввели обозначения

$$C_{v,v} = (V_k^\top \otimes I_n)C(V_k \otimes I_n) \in \mathbb{R}^{nr \times nr},$$
$$C_{v,u} = (V_k^\top \otimes I_n)C(I_m \otimes U_k) \in \mathbb{R}^{nr \times mr},$$
$$C_{v,vu} = (V_k^\top \otimes I_n)C(V_k \otimes U_k) \in \mathbb{R}^{nr \times r^2}.$$

Матрицы $C_{u,v}, C_{u,u}, C_{u,vu}$ и $C_{vu,v}, C_{vu,u}, C_{vu,vu}$ определяются аналогичным образом.

Для получения параметризации $\xi \in T_{\mathcal{X}} S^{n^d-1} \cap T_{\mathcal{X}} \mathcal{M}_{\mathbf{r}}$ необходимо принять во внимание, что $\xi \in T_{\mathcal{X}} S^{n^d-1}$ и, следовательно, $\xi^\top x = 0$ дает еще одно условие калибровки

$$\operatorname{vec}(S_\xi) \perp \operatorname{vec}(S). \tag{2.31}$$

В итоге, мы получаем

$$\begin{aligned}
\mathrm{P}_1 \xi &= V \otimes (I_n - UU^\top) \operatorname{vec}(U_\xi), \\
\mathrm{P}_2 \xi &= (I_m - VV^\top) \otimes U \operatorname{vec}(V_\xi^\top), \\
\mathrm{P}_3 \xi &= V \otimes U \operatorname{vec}(S_\xi).
\end{aligned}$$

Таким образом, первую блочную строчку в (2.28) можно записать в качестве

$$\begin{aligned}
V \otimes (I_n - UU^\top)(& \underbrace{(V^\top \otimes I)(A - \mathfrak{R}(x)I)(V \otimes I_n)}_{(A-\mathfrak{R}(x)I)_{v,v}} (I_r \otimes (I_n - UU^\top)) \operatorname{vec}(U_\xi) + \\
& \underbrace{(V^\top \otimes I)(A - \mathfrak{R}(x)I)(I_m \otimes U)}_{(A-\mathfrak{R}(x)I)_{v,u}} ((I_m - VV^\top) \otimes I_r) \operatorname{vec}(V_\xi^\top) + \\
& \underbrace{(V^\top \otimes I)(A - \mathfrak{R}(x)I)(V \otimes U))}_{(A-\mathfrak{R}(x)I)_{v,uv}} (I_{r^2} - \operatorname{vec}(S)(\operatorname{vec}(S))^\top) \operatorname{vec}(S_\xi) = \\
V \otimes (I_n - UU^\top) & \underbrace{(V^\top \otimes I) A (V \otimes I_n)}_{A_{v,v}} \operatorname{vec}(US).
\end{aligned}$$

Поскольку V имеет полный столбцовый ранг, мы в точности получаем первую блочную строку в (2.27). Остальные блочные строки можно получить аналогичным способом. □

Блочный предобуславливатель Якоби для $d = 2$. В работе [126] было рассмотрено предобуславливание вида

$$M_d = (I - xx^\top) M (I - xx^\top),$$

где M является аппроксимацией матрицы $A - \mathfrak{R}(x)I$. Если линейная система с матрицей M может быть быстро решена, то для решения

$$M_d y = z,$$

можно использовать явную формулу

$$y = -\lambda M^{-1}x - M^{-1}z, \quad \lambda = -\frac{x^\top M^{-1} z}{x^\top M^{-1} x}. \tag{2.32}$$

Мы используем этот подход и рассматриваем предобуславливатель вида

$$M_d = (I - BB^\top) M_{\text{loc}} (I - BB^\top), \tag{2.33}$$

где M_{loc} является приближением матрицы $(A - \mathfrak{R}(x)I)_{\text{loc}}$. Даже если M легко обратима, это не значит, что то же самое выполняется для матрицы M_{loc}. Следовательно, мы используем блочный Якоби предобуславливатель.

$$\begin{aligned} M_d = (I - BB^\top) &\begin{bmatrix} A_{v,v} - \mathfrak{R}(x)I & 0 & \\ 0 & A_{u,u} - \mathfrak{R}(x)I & 0 \\ 0 & 0 & A_{vu,vu} - \mathfrak{R}(x)I \end{bmatrix} (I - BB^\top) = \\ &\begin{bmatrix} P_U^\perp (A_{v,v} - \mathfrak{R}(x)I) P_U^\perp & 0 & \\ 0 & P_V^\perp (A_{u,u} - \mathfrak{R}(x)I) P_V^\perp & 0 \\ 0 & 0 & P_S^\perp (A_{vu,vu} - \mathfrak{R}(x)I) P_S^\perp \end{bmatrix}, \end{aligned} \tag{2.34}$$

где спроецированные матрицы $P_U^\perp$, $P_V^\perp$ и $P_S^\perp$ определены следующим образом:

$$\begin{aligned} P_U^\perp &= I_r \otimes (I_n - UU^\top), \\ P_V^\perp &= (I_n - VV^\top) \otimes I_r, \\ P_S^\perp &= I_{r^2} - \text{vec}(S)\,(\text{vec}(S))^\top. \end{aligned}$$

Отметим, что линейная система с матрицей $A_{vu,vu} - \mathfrak{R}(x)I$ может быть решена с использованием прямого солвера благодаря малому размеру $r^2 \times r^2$. Таким образом, для решения

$$P_S^\perp (A_{vu,vu} - \mathfrak{R}(x)I) P_S^\perp y = P_S^\perp z, \quad y^\top \text{vec}(S) = 0,$$

мы можем использовать следующую формулу, которая следует из (2.32)

$$y = (A_{vu,vu} - \mathfrak{R}(x)I)^{-1} P_S^\perp z - \lambda_S (A_{vu,vu} - \mathfrak{R}(x)I)^{-1} \text{vec}(S),$$

где

$$\lambda_S = \frac{(\text{vec}(S))^\top (A_{vu,vu} - \mathfrak{R}(x)I)^{-1} P_S^\perp z}{(\text{vec}(S))^\top (A_{vu,vu} - \mathfrak{R}(x)I)^{-1} \text{vec}(S)}.$$

Выведем формулы для решения

$$P_U^{\perp}(A_{v,v} - \mathfrak{K}(x)I)P_U^{\perp}y = z, \quad P_U^{\perp}y = y$$

или эквивалентно

$$(I_r \otimes (I_n - UU^{\top}))(A_{v,v} - \mathfrak{K}(x)I)(I_r \otimes (I_n - UU^{\top}))y = z, \quad (I_r \otimes U^{\top})y = 0,$$

тогда

$$(A_{v,v} - \mathfrak{K}(x)I)y - (I_r \otimes U)\Lambda = z,$$

где матрица Λ выбрана из условия $(I_r \otimes U^{\top})y = 0$. Для предобуславливателя M_{vv}, аппроксимирующего $(A_{v,v} - \mathfrak{K}(x)I)$, имеем

$$y - M_{vv}^{-1}(I_r \otimes U)\Lambda = M_{vv}^{-1}z.$$

Умножая последнее уравнение на $(I_r \otimes U^{\top})$ получим

$$\Lambda = -\left[(I_r \otimes U^{\top})M_{vv}^{-1}(I_r \otimes U)\right]^{-1} M_{vv}^{-1}z,$$

и

$$y = M_{vv}^{-1}(I_r \otimes U)\Lambda + M_{vv}^{-1}z.$$

Аналогично для

$$P_V^{\perp}(A_{u,u} - \mathfrak{K}(x)I)P_V^{\perp}y = z, \quad P_V^{\perp}y = y$$

получаем формулу

$$y = M_{uu}^{-1}(V \otimes I_r)\Lambda + M_{uu}^{-1}z, \quad \Lambda = -\left[(V^{\top} \otimes I_r)M_{uu}^{-1}(V \otimes I_r)\right]^{-1} M_{uu}^{-1}z.$$

Матрицы $\left[(V^{\top} \otimes I_r)M_{uu}^{-1}(V \otimes I_r)\right]$ и $\left[(I_r \otimes U^{\top})M_{vv}^{-1}(I_r \otimes U)\right]$ имеют размеры $r^2 \times r^2$ и системы с ними могут быть решены с использованием прямых солверов. Основная сложность заключается в том, чтобы найти M_{uu}^{-1} и M_{vv}^{-1}. Их обращение зависит от конкретного приложения. Например, если $M = I \otimes F + G \otimes I$, то обратная матрица может быть приближена с использованием экспоненциальных сумм [70]

$$M^{-1} \approx \sum_{k=1}^{K} c_k\, e^{-t_k F} \otimes e^{-t_k G}, \tag{2.35}$$

которые мы также используем в вычислительных экспериментах. Можно также использовать итерационный метод для решения систем с диагональными блоками.

Отметим, что по аналогии с оригинальным методом Якоби-Дэвидсона, наш метод не является предобусловленным солвером. Предобуславливатель используется только для решения вспомогательных линейных систем.

Сложность метода. Если матрица A задана как оператор $\mathcal{A}$ в TT-формате с переупорядоченными индексами и TT-rank$(\mathcal{A}) = \mathbf{R}$, то сложность метода определяется следующим образом. Предположим, что система решается с помощью итерационного процесса. Стоимость умножения матрицы $\mathcal{A}$ на вектор ранга $2\mathbf{r}$ равна $\mathcal{O}(dn^2r^2R^2)$. Проекция результата на касательное пространство стоит $\mathcal{O}(dnr^3R^2)$ [127]. С точностью до числа итераций итоговая сложность алгоритма равна $\mathcal{O}(dnr^2R^2(r+n))$. Важно отметить, что при проекции на касательную плоскость задействуются одновременно все ядра разложения. Благодаря этому свойству метод может быть эффективно параллелизован по ядрам, например, с помощью TensorFlow.

Рассмотрим теперь случай, когда уравнение решается с помощью явной формулы для локальных систем (2.27) в случае $d = 2$. Пусть матрица A задана как (аналогично TT-формату матриц для тензоров)

$$A = \sum_{\alpha=1}^{R} F_\alpha \otimes G_\alpha, \tag{2.36}$$

где матрицы F_α и G_α имеют размеры $n \times n$. В оценках сложности мы дополнительно предполагаем, что F_α и G_α могут быть умножены на вектор со сложностью $\mathcal{O}(n)$, например, это выполняется для разреженных матриц F_α и G_α. В качестве примера, A может быть оператором Лапласа с малоранговым потенциалом.

Даже если исходная матрица A была разреженной, спроецированная матрица A_{loc} будет плотной. Однако A_{loc} допускает быстрое умножение на вектор.

Действительно, рассмотрим умножение на первую блочную строку A_{loc}:

$$\begin{aligned} u &= A_{v,v}\text{vec}(U) + A_{v,u}\text{vec}(V^\top) + A_{vu,vu}\text{vec}(S) \\ &= (V_k^\top \otimes I_n)A(\text{vec}(UV_k^\top + U_k V^\top + U_k S V_k^\top) \\ &= (V_k^\top \otimes I_n)A(\text{vec}(UV_k^\top + U_k(V^\top + SV_k^\top)), \end{aligned} \tag{2.37}$$

где мы уже учли, что вектор из касательного пространства $UV_k^\top + U_k V^\top + U_k S V_k$ имеет ранг $2r$ вместо $3r$, которое возникает при суммировании трех произвольных матриц ранга r. Это уменьшает сложность матрично-векторного умножения.

В итоге, подставляя (2.36) в (2.37), получим

$$u = (V_k^\top \otimes I_n)\left(\sum_{\alpha=1}^{R} F_\alpha \otimes G_\alpha\right)\Big((V_k \otimes I_n)\text{vec}(U) + (I_n \otimes U_k)\text{vec}(V^\top + SV_k^\top)\Big) =$$
$$\left(\sum_{\alpha=1}^{R}(V_k^\top F_\alpha V_k) \otimes G_\alpha\right)\text{vec}(U) + \left(\sum_{\alpha=1}^{R}(V_k^\top F_\alpha) \otimes (G_\alpha U_k)\right)\text{vec}(V^\top + SV_k^\top).$$

Вычисление $r \times r$ матрицы $V_k^\top F_\alpha V_k$ требует $\mathcal{O}(nr^2 + nr)$ операций. Умножение $V_k^\top F_\alpha V_k \otimes G_\alpha$ на вектор также стоит $\mathcal{O}(nr^2 + nr)$. Вычисление $V_k^\top F_\alpha$ и $G_\alpha U_k$ имеет сложность $\mathcal{O}(nr)$. Отметим, что вычисление $V_k^\top F_\alpha V_k$, $V_k^\top F_\alpha$ и $G_\alpha U_k$ делается только один раз на каждой внешней итерации. Следовательно, умножение матрицы на вектор имеет сложность $\mathcal{O}((n+m)Rr^2)$, так как мы умножаем матрицы $(V_k^\top F_\alpha V_k) \otimes G_\alpha$ и $(V_k^\top F_\alpha) \otimes (G_\alpha U_k)$ на вектор R раз. Система (2.27) может быть решена с помощью подходящего крыловского итерационного метода.

При ускорении с использованием подпространств мы проецируем векторы из V_b (2.19) на касательное пространство. Вычисление проекции каждого вектора стоит $\mathcal{O}(nr^2)$ операций. Таким образом, предполагая, что $r \ll n, m$, сложность одной внешей итерации алгоритма равняется $\mathcal{O}((n+m)r(R+r))$.

2.1.7 Вычислительный эксперимент

Собственные значения оператора конвекции-диффузии

Пусть необходимо найти наименьшее собственное значение оператора конвекции-диффузии

$$\mathcal{A}u \equiv -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + Vu, \quad (x, y) \in \Omega,$$
$$u|_{\partial\Omega} = 0,$$

где $\Omega = (-1/2, 1/2)^2$, и потенциал V выбирается так, чтобы решение имело малый ранг: $V \equiv V(x, y) = e^{-\sqrt{x^2+y^2}/10}$. Мы используем стандартную конечно-разностную дискретизацию на прямоугольной равномерной сетке $n \times n$. Для дискретизации первых производных используется конечная разность “назад”. Матрица потенциала V на сетке приближается с помощью сингулярного разложения с относительной точностью 10^{-10}. Дискретизация оператора A представляется в виде (2.36) с разреженными матрицами F_α, G_α и $R = 14$.

Малоранговая версия и оригинальный метод Якоби-Дэвидсона. Сравним поведение оригинального метода Якоби-Дэвидсона и предложенной малоранговой версии. На рисунке 2.1 изображена зависимость невязки от числа внешних итераций. Ранг многообразия $r = 5$, размер сетки $n = 150$. Можно наблюдать, что малоранговая версия стагнирует из-за ограничения на ранг.

Отметим, что стоимость каждой внутренней итерации различна: $\mathcal{O}(nrR)$ для предложенной версии и $\mathcal{O}(n^2)$ для оригинальной версии. Значит, предложенная версия является более эффективной для больших n. Тем не менее, Рисунок 2.1 показывает, что предложенный метод требует меньшего числа менее дорогих итераций для достижения заданной точности (до стагнации). Более того, чем менее точно мы решаем систему, тем более предложенный метод эффективен. Такой выигрыш может наблюдаться благодаря использованию дополнительной информации про решение, а именно, что оно имеет малый ранг.

Сравнение с малоранговым методом Дэвидсона и итерацией Рэлея. В этом эксперименте мы сравниваем поведение предложенного метода Якоби-

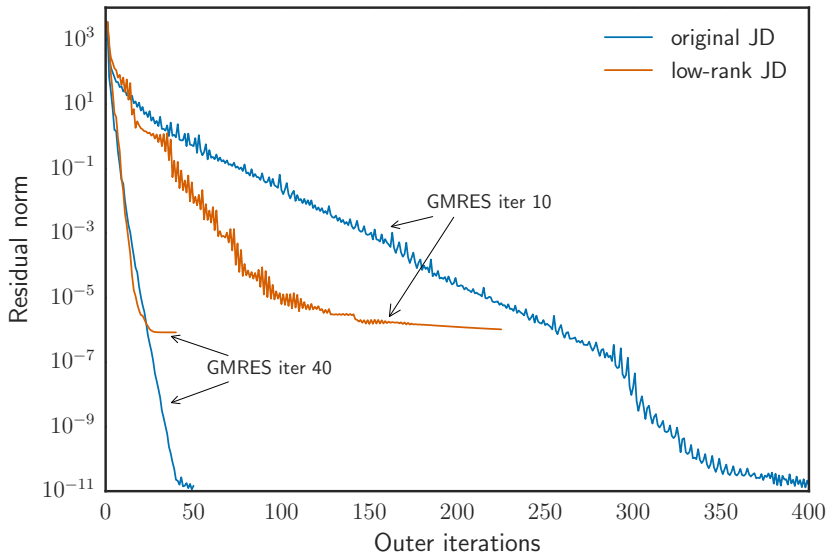


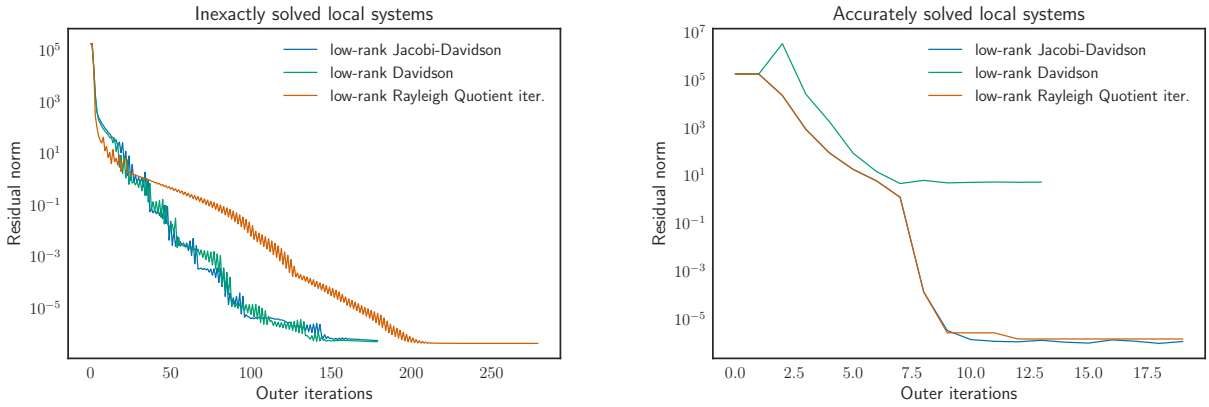
Рис. 2.1: Норма невязки в зависимости от числа внешних итераций для оригинального и предложенного методов. Графики сделаны для различного числа внутренних итераций GMRES для решения линейных систем. Параметры: $N = 150^2$, $r = 5$.

Дэвидсона и предложенной итерации Рэлея (2.23) (обратная итерация с адаптивным сдвигом). Мы также сравниваем их с подходом “Дэвидсона”, который не использует проекции $I - x_k x_k^\top$

$$\left[\mathrm{P}_{T_{\mathcal{X}_k}\mathcal{M}_{\mathbf{r}}}(A - \mathfrak{R}(x_k)I)\mathrm{P}_{T_{\mathcal{X}_k}\mathcal{M}_{\mathbf{r}}}\right]\xi_k = -\mathrm{P}_{T_{\mathcal{X}_k}\mathcal{M}_{\mathbf{r}}} r_k, \quad \mathrm{P}_{T_{\mathcal{X}_k}\mathcal{M}_{\mathbf{r}}}\xi_k = \xi_k. \tag{2.38}$$

Рисунок 2.2 иллюстрирует результаты сравнения. Как и ожидалось, когда линейные системы решаются точно, метод Дэвидсона стагнирует в силу того, что точным решением (2.38) является $-x_k$. То есть никакой дополнительной информации к предыдущему приближению x_k не добавляется. Этой проблемы не возникает, если локальные системы решаются неточно. Для итерации Рэлея мы наблюдаем противоположное поведение из-за ухудшения числа обусловленности локальных систем. Предложенный метод Якоби-Дэвидсона дает лучшую сходимость в обоих случаях.

Сравнение с ALS методом. Попеременный линейный метод (ALS) является стандартным подходом для малоранговой оптимизации. Идея заключается в следующем: имея $X = UV^\top$ мы минимизируем отношение Рэлея $\mathfrak{R}(x) \equiv$



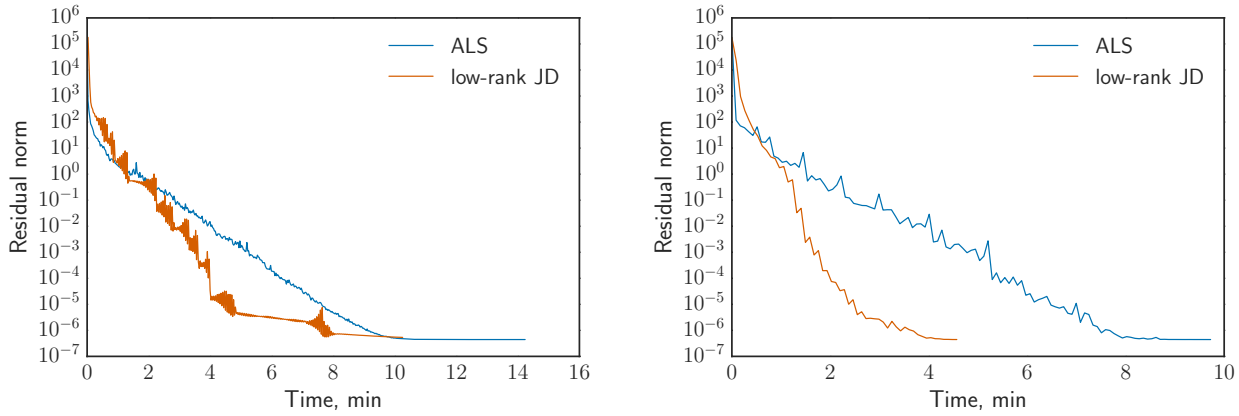
(a) (b)

Рис. 2.2: Невязка в зависимости от числа внешних итераций, $N = 2000^2$, $r = 3$. Локальные системы на Рисунке 2.2a были решены неточно с использованием 100 GMRES итераций, в то время как системы на Рисунке 2.2b были решены точно с предобуславливателем (2.35), $K = 20$ и 30 GMRES итераций.

$\widetilde{\mathfrak{K}}(U, V)$ попеременно по U и V. Минимизация по U приводит к задаче на собственные значения с матрицей $A_{v,v}$, в то время, как минимизация по V приводит к задаче на собственные значения с матрицей $A_{u,u}$.

Отметим, что в предложенном методе Якоби-Дэвидсона нам необходимо решать локальные системы, в то время как в ALS подходе мы решаем локальные задачи на собственные значения. Для честного сравнения мы запускаем оригинальный метод Якоби-Дэвидсона для решения локальных задач в ALS. Мы выбрали фиксированное число итераций, так как фиксированная точность для решения задач на собственные значения в ALS приводит к стагнации метода. Поскольку внутренний JD решатель имеет два типа итераций: итерация для решения локальной задачи и внешние итерации, нам необходимо настроить эти параметры для честного сравнения. Мы выбираем параметры так, что время каждой ALS итерации равно времени внешней итерации предложенного метода Якоби-Дэвидсона, и при этом метод сходится максимально быстро. Результаты расчетов представлены на Рисунке 2.3. В обоих случаях предложенный метод дает лучшую сходимость.

Ускорение с использованием подпространств. В настоящем параграфе мы исследуем поведение ускорения с использованием подпространств, опи-



(a) (b)

Рис. 2.3: Невязка в зависимости от времени для ALS и для предложенного метода. Рисунки 2.3a и 2.3b соответствуют 150 и 600 GMRES итераций для решения локальных задач в предложенном методе. Параметры локальных задач в ALS были выбраны для обеспечения такого же времени одной итерации, как и в JD методе, $N = 2000^2$.

санное в секции 2.1.3. Во-первых, на Рисунке 2.4 мы сравниваем оригинальное ускорение с использованием подпространств и версию с векторным переносом, когда подпространство проецируется на касательное пространство текущего приближения. Как и ожидалось, версия с дополнительной проекцией стагнирует когда точность приближения становилась равна точности малоранговой аппроксимации. В остальном методы дают схожую сходимость, при этом метод с проецированием подпространства дает преимущества при работе с малоранговой арифметикой. Рисунок 2.5 иллюстрирует преимущество спроецированной версии. Версия без проекций реализована с округлением по рангу (hard rank thresholding) линейных комбинаций (2.14). Коэффициенты в базисе находились с помощью метода Рэлея-Ритца без дополнительной оптимизации. Преимущество спроецированной версии можно объяснить тем, что коэффициенты оптимизируются точно на касательном пространстве, так как округление по рангу не требуется (в касательном пространстве тензоры имеют максимально возможный ранг равный $2\mathbf{r}$). В то же время при сложении векторов не из касательного пространства ранг быстро растет, и округление с фиксированным рангом может вносить значительную ошибку.

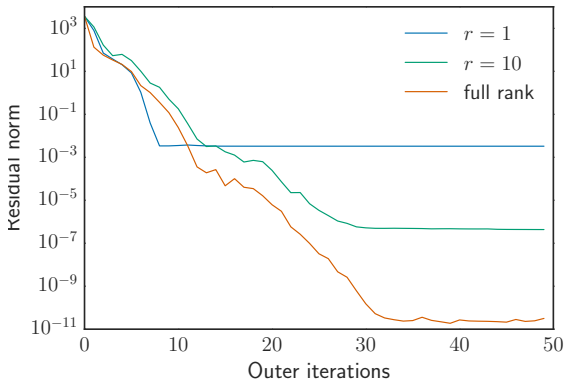


Рис. 2.4: Сравнение оригинального ускорения на подпространстве и версии с векторным переносом. В обоих случаях $N = 150^2$, локальные системы решаются с использованием 150 GMRES итераций.

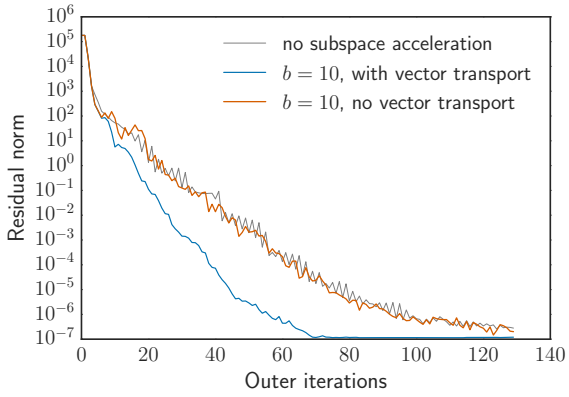


Рис. 2.5: Предложенный метод с ускорением на подпространстве в 2 случаях: когда базис спроецирован на касательное пространство и без проецирования. Параметры: $N = 2000^2$, $r = 5$, 150 GMRES итераций для решения локальных систем.

Сравнение с AMEn подходом. Проведем сравнение с алгоритмом EvAMEn, предложенным в [33] и базирующемся на AMEn (Alternating Minimal Energy) подходе [32]. Отличительной особенностью метода является возможность адаптировать ранг решения. На Рисунке 2.6 приведено сравнение с рассматриваемым оператором без конвективной части, так как код EvAMEn написан для симметричных систем. Из рисунка следует, что если локальные системы решаются неточно, то EvAMEn подход стагнирует, в то время, как предложенный метод сходится благодаря линейному поиску и ускорению с использованием подпространств. Более того, во время стагнации ранг в EvAMEn может заметно превышать ранг точного решения задачи оптимизации. Это приводит к падению скорости вычислений. В случае точного решения локальных систем EvAMEn сходится сопоставимо по количеству итераций с предложенным методом.

DMRG и адаптация ранга

Рассмотрим предложенный метод Якоби-Дэвидсона как вспомогательный шаг для алгоритма поиска наименьшего собственного значения в TT-

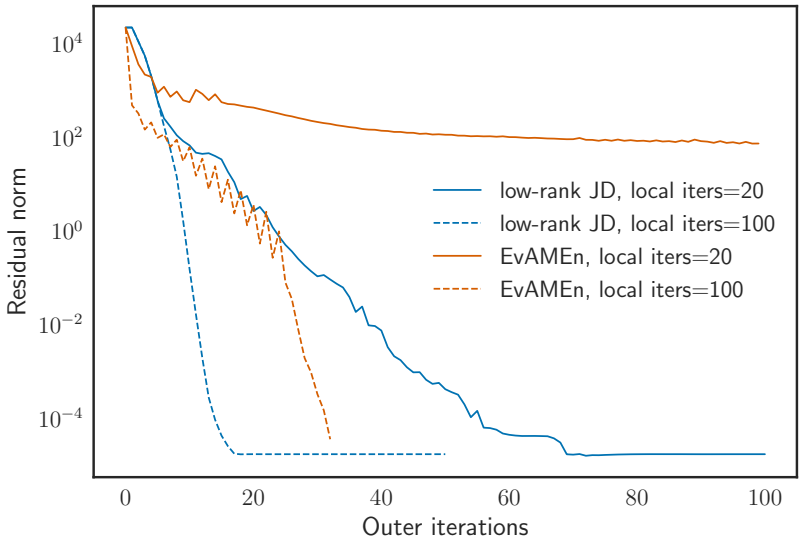


Рис. 2.6: Норма невязки в зависимости от числа внешних итераций для предложенного метода и EvAMEn алгоритма. Графики сделаны для различного числа внутренних итераций GMRES для решения линейных систем, $N = 500^2$.

формате. В частности, рассмотрим DMRG (density matrix renormalization group) алгоритм, который был предложен в [141].

DMRG алгоритм для минимизации отношения Релея в TT-формате похож на ALS оптимизацию. Отличие заключается лишь в том, что в DMRG алгоритме фиксируются все, кроме двух ядер $G_k(i_k), G_{k+1}(i_{k+1})$. Заметим, что такое представление является нелинейным по этим двум ядрам. Для получения линейной задачи на собственные значения в DMRG алгоритме вводится одно большое ядро

$$S(i_k, i_{k+1}) = G_k(i_k) G_{k+1}(i_{k+1}). \tag{2.39}$$

Минимизация по S является квадратичной задачей и может быть сведена к задаче на собственные значения с матрицей размера $r_k n^2 r_{k+2} \times r_k n^2 r_{k+2}$. Алгоритмические детали, как эффективно умножать эту матрицу на вектор можно найти в [105]. После того, как S найдено, ядра G_k и G_{k+1} в (2.39) могут быть получены с помощью SVD с рассматриваемой точностью. Таким образом, ранг в DMRG алгоритме подбирается адаптивным образом. Проблема заключается в том, что для больших n оптимизация по S требует $\mathcal{O}(n^2)$ операций вместо $\mathcal{O}(n)$ операций для стандартного ALS алгоритма.

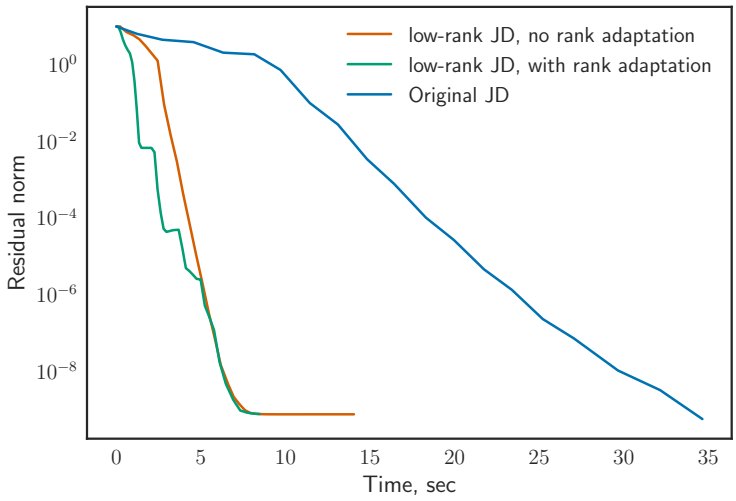


Рис. 2.7: Норма невязки по отношению к времени вычислений для решения локальных систем в DMRG алгоритме. Размер матрицы $820^2 \times 820^2$.

Мы предлагаем оптимизировать S сразу на малоранговом многообразии, имея априорное знание о том, что оно может быть представлено в виде (2.39) (S может быть рассмотрено как $nr_k \times nr_{k+2}$ матрица малого ранга). Отметим, что предложенный метод Якоби-Дэвидсона рассматривается на многообразии фиксированного ранга. Однако несложно модифицировать этот алгоритм, чтобы ранг можно было адаптировать. Для этого воспользуемся идеей из [136] и оптимизируем функционал на последовательности многообразий с увеличивающимися рангами. В частности, мы начинаем с $r = 2$, и когда норма спроецированного градиента $\|\mathrm{grad}\,\mathfrak{R}(x_k)\|$ мала, увеличиваем ранг $r := r + 4$.

В качестве примера рассмотрим систему с потенциалом Хенона-Хейлеса

$$\mathcal{A} = -\frac{1}{2}\Delta + \frac{1}{2}\sum_{k=1}^{d} x_k^2 + \lambda \sum_{k=1}^{d-1}\left(x_k^2 x_{k+1} - \frac{1}{3}x_{k+1}^3\right),$$

где $d = 10$, а параметр $\lambda = 0.14$ был выбран так, чтобы получить большие значения рангов основного состояния системы. Задача была дискретизована с использованием псевдоспектрального метода на Эрмитовой сетке [9] с 20 точками сетки по каждому направлению.

Реализация DMRG алгоритма была взята из программного комплекса TT-toolbox [129] с точностью 10^{-10} с матрицей из третьего прохода алгоритма при $k = 4$, $r_k = r_{k+2} = 41$. На Рисунке 2.7 представлена норма невязки по отношению

к времени вычислений для случая фиксированного ранга ($r = 30$), для оригинального метода Якоби-Дэвидсона со сложностью $\mathcal{O}(n^2)$ и для версии с адаптацией по рангу. Заметим, что времена расчета для случаев с и без адаптации по рангу сравнимы по времени, в то время как версия не использующая малоранговую структуру является более медленной. Эта разница возникает из-за дополнительного множителя n в оценке сложности.

2.2 ALS обратная итерация

В разделе 2.1.5 был получен метод обратной итерации на многообразии, имеющий интерпретацию метода Гаусса-Ньютона. Как было отмечено, возможна эффективная параллелизация этой итерации по ядрам, так как ортопроектор на касательное пространство действует на все ядра ТТ-разложения одновременно. В настоящем разделе рассматривается итерация, базирующаяся на ALS подходе, когда ядра ТТ-разложения оптимизируются последовательно.

Предлагаемая ALS обратная итерация (ALS inverse iteration, ALS II) имеет схожесть с обратной итерацией из раздела 2.1.5. Она также записывается через проекторы на многообразие, но вместо проекторов на все касательное пространство используется последовательное применение проекторов на подпространства касательного пространства.

2.2.1 Формулировка итерации

Сформулируем предлагаемый метод ALS II — обратная итерация на основе ALS подхода. Пусть $\mathcal{A}$ является симметричным положительно определенным оператором. Начнем описание с классической обратной итерации [57]

$$\begin{aligned} &(\mathcal{A} - \sigma\mathcal{I})\mathcal{X}_{k+1} = \mathcal{X}_k, \\ &\mathcal{X}_{k+1} := \mathcal{X}_{k+1} / \sqrt{\langle \mathcal{X}_{k+1}, \mathcal{X}_{k+1} \rangle}, \end{aligned} \tag{2.40}$$

где σ называется “сдвигом”. Известно, что чем ближе σ к искомому собственному значению λ_1, тем быстрее сходимость при условии, что возникающие линейные системы решаются точно. Из классической теории сходимости степен-

ного метода [152] следует, что скорость сходимости обратной итерации равна

$$\rho = \left| \frac{\lambda_1 - \sigma}{\lambda_2 - \sigma} \right|. \tag{2.41}$$

Также сдвиг σ можно выбрать адаптивно как текущее приближение к собственному значению $\lambda_k = (\mathcal{A}\mathcal{X}_k, \mathcal{X}_k)$. В этом случае хорошо известно, что метод сходится сверхлинейно (кубическая сходимость для симметричных матриц и квадратичная для несимметричных).

Кажется, что если σ близко к искомому собственному значению, то число обусловленности системы может быть крайне велико. Однако важно отметить, что правая часть в такой системе не является “случайной” и известно, что такие системы можно решать неточно [38, 13].

Перейдем к рассмотрению малорангового случая. В обратной итерации (2.40) нам необходимо найти TT представление $\mathcal{X}_{k+1}$ с помощью приближенного решения линейной системы

$$(\mathcal{A} - \sigma\mathcal{I})\mathcal{X}_{k+1} \approx \mathcal{X}_k. \tag{2.42}$$

Предположим, что и точный собственный вектор $\mathcal{X}_*$, и текущее приближение $\mathcal{X}_k$ лежит на многообразии $\mathcal{M}_{\mathbf{r}}$ тензоров TT-ранга $\mathbf{r}$. Решение (2.42) может иметь ранги больше, чем $\mathbf{r}$ и, следовательно, не принадлежать многообразию $\mathcal{M}_{\mathbf{r}}$. Предлагаемая ALS II итерация оставляет решение на $\mathcal{M}_{\mathbf{r}}$ и имеет следующий вид

$$\mathrm{P}^{\neq i}_{k+\frac{i-1}{d}} (\mathcal{A} - \sigma\mathcal{I}) \mathrm{P}^{\neq i}_{k+\frac{i-1}{d}} \mathcal{X}_{k+\frac{i}{d}} = \mathrm{P}^{\neq i}_{k+\frac{i-1}{d}} \mathcal{X}_k, \quad i = 1, \ldots, d, \tag{2.43}$$

где $\mathrm{P}^{\neq i}_{k+\frac{i-1}{d}}$ являются ортопроекторами на части касательного пространства в точке $\mathcal{X}_{k+\frac{i-1}{d}}$, где все ядра кроме i-го фиксированы. То, что пространство, где все ядра, кроме одного ядра TT-разложения, являются фиксированными, является частью касательного пространства следует из формулы (1.22). Для устойчивости вычислений после каждой итерации необходимо дополнительно нормировать $\mathcal{X}_{k+\frac{i}{d}}$. Отметим, что итерация (2.43) может быть реализована с использованием уже написанного кода решения линейных систем уравнений, так как она соответствует одному ALS проходу по всем ядрам для задачи оптимизации

$$\langle (\mathcal{A} - \sigma\mathcal{I})\mathcal{Y}, \mathcal{Y} \rangle - 2\langle \mathcal{X}_k, \mathcal{Y} \rangle \to \min,$$
$$\text{TT-rank}(\mathcal{Y}) = \mathbf{r}.$$

Решение этой задачи описано в разделе 1.5. Вычислительная сложность одного прохода равна $\mathcal{O}(dn^2r^2R^2)$, где $r = \max_i \mathbf{r}_i$, $R = \max_i \mathbf{R}_i$ и $\mathbf{R}$ является TT-рангом оператора $\mathcal{A}$.

На практике мы решаем возникающие локальные системы неточно с помощью фиксированного числа итераций MINRES метода. Отметим, что результаты численных расчетов показывают, что как и в случае со стандартной обратной итерацией (2.42), нет необходимости решать системы точно. Результаты о локальной сходимости итерации в случае точного решения локальных систем и $d = 2$ приведены в следующем параграфе.

2.2.2 Сходимость итерации

Приведем результаты сходимости предложенной ALS обратной итерации для случая $d = 2$. Для удобства будем рассматривать операторы в качестве двумерных матриц $A \in \mathbb{R}^{N\times N}$. Обозначим собственный вектор оператора A, отвечающий минимальному собственному значению λ_1 (простое), за x_*. Положим также, что $N = nm$ и мы обладаем априорным знанием о том, что x_*, представленный в виде $n\times m$ матрицы $\mathcal{X}_*$, имеет ранг r, то есть $\operatorname{rank}(\mathcal{X}_*) = r$, $x_* = \operatorname{vec}(\mathcal{X}_*)$.

Формула (2.43) не подходит для вычислений, так как возникающие линейные системы имеют размер $n^d \times n^d$. Для получения практического алгоритма необходимо использовать формулы из раздела 1.5.3. Однако для теоретического анализа будет удобно использовать формулы с ортопроекторами. В случае малоранговых матриц ортопроекторы на пространство строк и столбцов матрицы $\mathcal{X}$ могут быть записаны как $\mathcal{X}^+\mathcal{X}$, $\mathcal{X}\mathcal{X}^+$ соответственно, где $\mathcal{X}^+$ обо-

---

Алгоритм 2.2 ALS II при $d = 2$ (версия для анализа)

Require: $\mathcal{X}_0 \in \mathbb{R}^{n\times m}$, $\operatorname{rank}(\mathcal{X}_0) = r$ − начальное приближение.

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: Решить $A^{\mathsf{R}}[\mathcal{X}_k] \operatorname{vec}\left(\mathcal{X}^{\text{ii}}_{k+\frac{1}{2}}\right) = \operatorname{vec}(\mathcal{X}_k)$ при $\mathcal{X}^{\text{ii}}_{k+\frac{1}{2}}\mathcal{X}_k^+\mathcal{X}_k = \mathcal{X}^{\text{ii}}_{k+\frac{1}{2}}$

3: $\mathcal{X}^{\text{ii}}_{k+\frac{1}{2}} := \mathcal{X}^{\text{ii}}_{k+\frac{1}{2}} / \left\|\mathcal{X}^{\text{ii}}_{k+\frac{1}{2}}\right\|_F$

4: Решить $A^{\mathsf{L}}\left[\mathcal{X}^{\text{ii}}_{k+\frac{1}{2}}\right] \operatorname{vec}(\mathcal{X}^{\text{ii}}_{k+1}) = \operatorname{vec}\left(\mathcal{X}^{\text{ii}}_{k+\frac{1}{2}}\mathcal{X}^{\text{ii}^+}_{k+\frac{1}{2}}\mathcal{X}_k\right)$ при $\mathcal{X}^{\text{ii}}_{k+\frac{1}{2}}\mathcal{X}^{\text{ii}^+}_{k+\frac{1}{2}}\mathcal{X}^{\text{ii}}_{k+1} = \mathcal{X}^{\text{ii}}_{k+1}$,

5: $\mathcal{X}^{\text{ii}}_{k+1} := \mathcal{X}^{\text{ii}}_{k+1} / \left\|\mathcal{X}^{\text{ii}}_{k+1}\right\|_F$

6: **end for**

значает псевдообратную матрицу Мура-Пенроуза от $\mathcal{X}$. Поэтому ортопроекторы $\mathrm{P}_k^{\neq 1}$, $\mathrm{P}_{k+1/2}^{\neq 2}$ имеют вид $(\mathcal{X}_k^+\mathcal{X}_k \otimes I)$, $(I \otimes \mathcal{X}_{k+1/2}\mathcal{X}_{k+1/2}^+)$ соответственно. Введем обозначения

$$\begin{aligned} A^{\mathsf{R}}[\mathcal{X}] &= (\mathcal{X}^+\mathcal{X} \otimes I)\, A\, (\mathcal{X}^+\mathcal{X} \otimes I), \\ A^{\mathsf{L}}[\mathcal{X}] &= (I \otimes \mathcal{X}\mathcal{X}^+)\, A\, (I \otimes \mathcal{X}\mathcal{X}^+). \end{aligned} \tag{2.44}$$

ALS обратная итерация с использованием обозначений (2.44) подытожена в Алгоритме 2.2. Символ ii в $\mathcal{X}_k^{\mathsf{ii}}$ является сокращением от inverse iteration.

Как мы увидим далее, оценки локальной сходимости ALS II будут выражаться через сходимость ALS минимизации отношения Рэлея. Покажем, как ALS минимизация отношения Рэлея будет выглядеть с использованием $\mathcal{X}^+\mathcal{X}$, $\mathcal{X}\mathcal{X}^+$. Эквивалентность оригинальной ALS минимизации отношения Рэлея и версии с проекторами $\mathcal{X}^+\mathcal{X}$, $\mathcal{X}\mathcal{X}^+$ (Алгоритм 2.3) следует из следующего утверждения.

Утверждение 2.5. *Один шаг ALS минимизации отношения Рэлея* (1.14) *начиная с* $\mathcal{X}_0 = \mathrm{vec}(U_0 V_0^\top)$ *эквивалентен решению следующей задачи на собственные значения*

$$\begin{aligned} A^{\mathsf{R}}[\mathcal{X}_0]\,\mathrm{vec}(\mathcal{X}_{1/2}) &= \lambda_{\min}^{\mathsf{R}}\,\mathrm{vec}(\mathcal{X}_{1/2}), \\ A^{\mathsf{L}}[\mathcal{X}_{1/2}]\,\mathrm{vec}(\mathcal{X}_1) &= \lambda_{\min}^{\mathsf{L}}\,\mathrm{vec}(\mathcal{X}_1), \end{aligned} \tag{2.45}$$

с $\mathcal{X}_1 = \mathrm{vec}(U_1 V_1^\top)$ *и* $\lambda_{\min}^{\mathsf{R}}$, $\lambda_{\min}^{\mathsf{L}}$ *являющимися минимальными ненулевыми собственными значениями матриц* $A^{\mathsf{R}}[\mathcal{X}_0]$ *и* $A^{\mathsf{L}}[\mathcal{X}_{1/2}]$ *соответственно.*

Доказательство. Докажем утверждение для первого полушага (2.45). Доказательство для второго полушага является аналогичным. Имеем,

$$\min_{U \in \mathbb{R}^{n\times r}} \mathfrak{R}(U V_0^\top) = \min_{\mathcal{Y}:\mathcal{Y}=\mathcal{Y}\mathcal{X}_0^+\mathcal{X}_0} \mathfrak{R}(\mathcal{Y}\mathcal{X}_0^+\mathcal{X}_0) =$$

$$\min_{\mathcal{Y}:\mathcal{Y}=\mathcal{Y}\mathcal{X}_0^+\mathcal{X}_0} \mathfrak{R}((\mathcal{X}_0^+\mathcal{X}_0 \otimes I)\,\mathrm{vec}(\mathcal{Y})) = \min_{\mathcal{Y}:\mathcal{Y}=\mathcal{Y}\mathcal{X}_0^+\mathcal{X}_0} \frac{\left\langle \mathrm{vec}(\mathcal{Y}), A^{\mathsf{R}}[\mathcal{X}_0]\,\mathrm{vec}(\mathcal{Y}) \right\rangle}{\left\langle \mathrm{vec}(\mathcal{Y}), (\mathcal{X}_0^+\mathcal{X}_0 \otimes I)\,\mathrm{vec}(\mathcal{Y}) \right\rangle},$$

что приводит к эквивалентной обобщенной задаче на собственные значения с дополнительными ограничениями

$$A^{\mathsf{R}}[\mathcal{X}_0]\,\mathrm{vec}(\mathcal{X}_{1/2}) = \lambda_{\min}^{\mathsf{R}}\,(\mathcal{X}_0^+\mathcal{X}_0 \otimes I)\,\mathrm{vec}(\mathcal{X}_{1/2}), \quad \mathcal{X}_{1/2} = \mathcal{X}_{1/2}\mathcal{X}_0^+\mathcal{X}_0, \tag{2.46}$$

что благодаря свойству $\mathcal{X}_{1/2} = \mathcal{X}_{1/2}\mathcal{X}_0^+\mathcal{X}_0$ эквивалентно

$$A^{\mathsf{R}}[\mathcal{X}_0]\,\mathrm{vec}(\mathcal{X}_{1/2}) = \lambda^{\mathsf{R}}_{\min}\,\mathrm{vec}(\mathcal{X}_{1/2}), \quad \mathcal{X}_{1/2} = \mathcal{X}_{1/2}\mathcal{X}_0^+\mathcal{X}_0, \tag{2.47}$$

и также эквивалентно

$$A^{\mathsf{R}}[\mathcal{X}_0]\,\mathrm{vec}(\mathcal{X}_{1/2}) = \lambda^{\mathsf{R}}_{\min}\,\mathrm{vec}(\mathcal{X}_{1/2}), \quad \lambda^{\mathsf{R}}_{\min} \neq 0. \tag{2.48}$$

На самом деле, если $\mathrm{vec}(\mathcal{X}_{1/2})$ является решением (2.48), тогда оно также удовлетворяет $\mathcal{X}_{1/2} = \mathcal{X}_{1/2}\mathcal{X}_0^+\mathcal{X}_0$ так как

$$\begin{aligned}\lambda^{\mathsf{R}}_{\min}\,\mathrm{vec}(\mathcal{X}_{1/2}) = A^{\mathsf{R}}[\mathcal{X}_0]\,\mathrm{vec}(\mathcal{X}_{1/2}) = (\mathcal{X}_0^+\mathcal{X}_0 \otimes I)\,A^{\mathsf{R}}[\mathcal{X}_0]\,\mathrm{vec}(\mathcal{X}_{1/2}) = \\ = (\mathcal{X}_0^+\mathcal{X}_0 \otimes I)\,\lambda^{\mathsf{R}}_{\min}\,\mathrm{vec}(\mathcal{X}_{1/2}) = \lambda^{\mathsf{R}}_{\min}\,\mathrm{vec}(\mathcal{X}_{1/2}\mathcal{X}_0^+\mathcal{X}_0).\end{aligned}$$

И наоборот, если $\mathrm{vec}(\mathcal{X}_{1/2})$ является решением (2.47), тогда $\lambda^{\mathsf{R}}_{\min} \neq 0$:

$$\begin{aligned}\lambda^{\mathsf{R}}_{\min}\,\langle \mathrm{vec}(\mathcal{X}_{1/2}), \mathrm{vec}(\mathcal{X}_{1/2})\rangle = \left\langle \mathrm{vec}(\mathcal{X}_{1/2}), A^{\mathsf{R}}[\mathcal{X}_0]\,\mathrm{vec}(\mathcal{X}_{1/2})\right\rangle = \\ = \langle \mathrm{vec}(\mathcal{X}_{1/2}\mathcal{X}_0^+\mathcal{X}_0), A\,\mathrm{vec}(\mathcal{X}_{1/2}\mathcal{X}_0^+\mathcal{X}_0)\rangle = \langle \mathrm{vec}(\mathcal{X}_{1/2}), A\,\mathrm{vec}(\mathcal{X}_{1/2})\rangle > 0,\end{aligned}$$

благодаря положительной определенности A. Что и требовалось доказать. □

Алгоритм 2.3 ALS для минимизации отношения Рэлея (версия для анализа)

Require: $\mathcal{X}_0 \in \mathbb{R}^{n\times m}$, $\mathrm{rank}(\mathcal{X}_0) = r$ – начальное приближение.

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: Найти собственный вектор, соответствующий минимальному $\lambda^{\mathsf{R}}_{\min} \neq 0$:

$$A^{\mathsf{R}}[\mathcal{X}_k^{\mathsf{als}}]\,\mathrm{vec}(\mathcal{X}_{k+1/2}^{\mathsf{als}}) = \lambda^{\mathsf{R}}_{\min}\,\mathrm{vec}(\mathcal{X}_{k+1/2}^{\mathsf{als}}), \quad \|\mathcal{X}_{k+1/2}^{\mathsf{als}}\|_F = 1$$

3: Найти собственный вектор, соответствующий минимальному $\lambda^{\mathsf{L}}_{\min} \neq 0$:

$$A^{\mathsf{L}}[\mathcal{X}_{k+1/2}^{\mathsf{als}}]\,\mathrm{vec}(\mathcal{X}_k^{\mathsf{als}}) = \lambda^{\mathsf{L}}_{\min}\,\mathrm{vec}(\mathcal{X}_k^{\mathsf{als}}), \quad \|\mathcal{X}_{k+1}^{\mathsf{als}}\|_F = 1$$

4: **end for**

Основной результат

Напомним, что точный собственный вектор симметричной положительно определенной матрицы A, отвечающий минимальному собственному значению $\lambda_1(A)$ (простое), обозначается $\mathrm{vec}(\mathcal{X}_*)$. Обозначим также минимальное

ненулевое сингулярное число $\mathcal{X}_*$ за $s_{\min}(\mathcal{X}_*)$. Сформулируем результат о локальной сходимости ALS II для случая $d = 2$.

Теорема 2.4. *Пусть существует такая окрестность точки $\mathcal{X}_* \in \mathcal{M}_r$, $\|\mathcal{X}_*\| = 1$, что для любого начального приближения $\mathcal{X}_0^{\text{als}}$ из этой окрестности ALS итерация (Алгоритм 2.3) сходится:*

$$\|\mathcal{X}_{k+1}^{\text{als}} - \mathcal{X}_*\|_F \le \rho_{\text{als}} \|\mathcal{X}_k^{\text{als}} - \mathcal{X}_*\|_F, \quad \rho_{\text{als}} < 1,$$

тогда найдется $\sigma_0 < \lambda_1(A)$: для любого $\sigma \in (\sigma_0, \lambda_1(A))$ существует такая окрестность, что для любого $\mathcal{X}_0^{\text{ii}}$ из этой окрестности, ALS II (Алгоритм 2.2), примененный к $A - \sigma I$, также сходится:

$$\|\mathcal{X}_{k+1}^{\text{ii}} - \mathcal{X}_*\|_F \le \left(\rho_{\text{als}} + c\left|\frac{\lambda_1(A) - \sigma}{\lambda_2(A) - \sigma}\right|\right)\|\mathcal{X}_k^{\text{ii}} - \mathcal{X}_*\|_F,$$

где

$$c = \frac{C}{s_{\min}(\mathcal{X}_*)} \cdot \frac{\lambda_n(A)}{\lambda_2(A) - \lambda_1(A)},$$

где C является константой, не зависящей от $\sigma, A, \mathcal{X}_*$.

Замечание 2.2. *Отметим, что локальная сходимость и существование $\rho_{\text{als}} < 1$ для ALS минимизации отношения Рэлея следует из Теоремы 1.3.*

Вспомогательные утверждения

Для доказательства Теоремы 2.4 понадобится несколько вспомогательных утверждений, которые мы приводим в настоящей секции. Сформулируем сначала результат о сходимости собственного вектора в степенном методе без требования малоранговости решения. Отметим, что хотя сходимость степенного метода является классическим вопросом, нам не удалось найти требуемой оценки для нормы разности собственного вектора и вектора после шага обратной итерации. Поэтому утверждение приводится с доказательством.

Утверждение 2.6. *Пусть λ_1 и v_1 являются максимальным собственным значением и соответствующим собственным вектором симметричной положительно определенной матрицы $B \in \mathbb{R}^{n \times n}$: $Bv_1 = \lambda_1 v_1$ и при этом $\|v_1\|_2 = 1$. Пусть также $x_1 = Bx_0/\|Bx_0\|_2$ является приближением к собственному вектору, полученным*

после одного шага степенного метода. Тогда для любого $0 < \varepsilon < 1/2$ *и для любого* x_0*:* $\|x_0 - v_1\|_2 \leq \varepsilon$ *выполняется*

$$\|x_1 - v_1\|_2 \leq \frac{1 + \frac{\varepsilon}{2}}{\left(1 - \sqrt{2\varepsilon}\right)(1 - \varepsilon)} \frac{\lambda_2}{\lambda_1} \|x_0 - v_1\|_2.$$

Доказательство. Разложим x_0 по собственным векторам v_i матрицы B:

$$x_0 = c_1 v_1 + \cdots + c_n v_n, \quad \sum_{i=1}^{n} c_i^2 = \|x_0\|^2. \tag{2.49}$$

Поскольку $\|x_0 - v_1\| \leq \varepsilon$ имеем

$$(1 - c_1)^2 + \sum_{i=2}^{n} c_i^2 \leq \varepsilon^2.$$

Подставляя (2.49) в последнее выражение получим

$$(1 - c_1)^2 + \|x_0\|^2 - c_1^2 \leq \varepsilon^2,$$

и в результате,

$$c_1 \geq \frac{1 + \|x_0\|^2}{2} - \frac{\varepsilon^2}{2}. \tag{2.50}$$

Так как $c_1 \leq \|x_0\|$ и используя (2.50) несложно проверить, что

$$1 - \varepsilon \leq \|x_0\| \leq 1 + \varepsilon. \tag{2.51}$$

Подставляя последнее выражение в (2.50) имеем

$$c_1 \geq 1 - \varepsilon. \tag{2.52}$$

$$x_1 = \frac{Bx_0}{\|Bx_0\|} = \frac{\lambda_1 c_1 v_1 + \cdots + \lambda_n c_n v_n}{\|\lambda_1 c_1 v_1 + \cdots + \lambda_n c_n v_n\|} = \frac{c_1 v_1 + \frac{\lambda_2}{\lambda_1} c_2 v_2 + \cdots + \frac{\lambda_n}{\lambda_1} c_n v_n}{\left\|c_1 v_1 + \frac{\lambda_2}{\lambda_1} c_2 v_2 + \cdots + \frac{\lambda_n}{\lambda_1} c_n v_n\right\|},$$

Заметим, что

$$\|(I - v_1 v_1^\top) x_1\| = \frac{\left\|\frac{\lambda_2}{\lambda_1} c_2 v_2 + \frac{\lambda_3}{\lambda_1} c_3 v_3 + \cdots + \frac{\lambda_n}{\lambda_1} c_n v_n\right\|}{\left\|c_1 v_1 + \frac{\lambda_2}{\lambda_1} c_2 v_2 + \cdots + \frac{\lambda_n}{\lambda_1} c_n v_n\right\|} =$$

$$= \frac{\lambda_2}{\lambda_1} \frac{\sqrt{c_2^2 + \left(\frac{\lambda_3}{\lambda_2}\right)^2 c_3^2 + \cdots + \left(\frac{\lambda_n}{\lambda_2}\right)^2 c_n^2}}{\sqrt{c_1^2 + \left(\frac{\lambda_2}{\lambda_1}\right)^2 c_2^2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^2 c_n^2}} \leq \frac{\lambda_2}{\lambda_1} \frac{\sqrt{c_2^2 + c_3^2 + \cdots + c_n^2}}{\sqrt{c_1^2 + \left(\frac{\lambda_2}{\lambda_1}\right)^2 c_2^2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^2 c_n^2}}.$$

Также имеем

$$\|(I - v_1 v_1^\top)x_0\| = \|c_2 v_2 + \cdots + c_n v_n\| = \sqrt{c_2^2 + c_3^2 + \cdots + c_n^2}$$

и используя (2.52) получим

$$\sqrt{c_1^2 + \left(\frac{\lambda_2}{\lambda_1}\right)^2 c_2^2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^2 c_n^2} \geq c_1 \geq 1 - \varepsilon.$$

В результате, для $\varepsilon < 1/2$

$$\|(I - v_1 v_1^\top)x_1\| \leq \frac{1}{1-\varepsilon}\frac{\lambda_2}{\lambda_1}\|(I - v_1 v_1^\top)x_0\| \tag{2.53}$$

Оценим $\|(I - v_1 v_1^\top)x_0\|$ с помощью $\|x_0 - v_1\|$:

$$\begin{aligned}\|(I - v_1 v_1^\top)x_0\| = \|(x_0 - (v_1^\top x_0)v_1\| &\leq \|x_0 - v_1\| + \|v_1 - (v_1^\top x_0)v_1\| = \\ &= \|x_0 - v_1\| + |1 - v_1^\top x_0| = \|x_0 - v_1\| + \frac{1}{2}\|x_0 - v_1\|^2 \leq \\ &\leq \left(1 + \frac{\varepsilon}{2}\right)\|x_0 - v_1\|.\end{aligned}$$

Оценим $\|(I - v_1 v_1^\top)x_1\|$ с помощью $\|x_1 - v_1\|$, используя обратное неравенство треугольника и $\varepsilon < 1/4$

$$\begin{aligned}\|x_1 - (v_1^\top x_1)v_1\| = \|x_1 - v_1 + v_1 - (v_1^\top x_1)v_1\| &\geq \left|\|x_1 - v_1\| - \|v_1 - (v_1^\top x_1)v_1\|\right| = \\ &= \left|\|x_1 - v_1\| - |1 - (v_1^\top x_1)|\right| = \left|\|x_1 - v_1\| - \frac{1}{2}\|x_1 - v_1\|^2\right| = \\ &= \|x_1 - v_1\| \cdot \left|1 - \frac{1}{2}\|x_1 - v_1\|\right|.\end{aligned}$$

Последним шагом является поиск оценки сверху для $\|x_1 - v_1\|$ с помощью ε для того, чтобы оценить $|1 - \|x_1 - v_1\|/2|$:

$$\begin{aligned}\|x_1 - v_1\|^2 = 2|1 - v_1^\top x_1| &= 2\left|1 - \frac{c_1}{\sqrt{c_1^2 + \left(\frac{\lambda_2}{\lambda_1}\right)^2 c_2^2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^2 c_n^2}}\right| \leq \\ \leq 2\frac{\|x_0\| - c_1}{c_1} &\overset{(2.51),(2.52)}{\leq} \frac{4\varepsilon}{1-\varepsilon} \leq 8\varepsilon, \quad \varepsilon \leq \frac{1}{2}.\end{aligned}$$

Таким образом, для $\varepsilon < 1/2$

$$\|x_1 - (v_1^\top x_1)v_1\| \geq \left(1 - (2\varepsilon)^{1/2}\right)\|x_1 - v_1\|.$$

В итоге, подставляя полученное неравенство в (2.53) получим

$$\|x_1 - v_1\| \le \frac{1 + \frac{\varepsilon}{2}}{\left(1 - \sqrt{2\varepsilon}\right)(1 - \varepsilon)} \frac{\lambda_2}{\lambda_1} \|x_0 - v_1\|, \quad \varepsilon < 1/2.$$

Что и требовалось доказать. □

Теперь приведем аналог Утверждения 2.6 в применении к рассматриваемой ALS обратной итерации.

Лемма 2.2. *Пусть* $\mathrm{vec}(\mathcal{X}_*)$ *является нормированным собственным вектором* $A^{\mathsf{R}}[\mathcal{X}]$ *(для* $A^{\mathsf{L}}[\mathcal{X}]$ *аналогично), соответствующим наименьшему ненулевому собственному значению. Пусть также* $\mathcal{X}_1$ *получено из*

$$A^{\mathsf{R}}[\mathcal{X}]\,\mathrm{vec}(\widetilde{\mathcal{X}}) = \mathrm{vec}(\mathcal{X}_0), \quad \widetilde{\mathcal{X}}\mathcal{X}^{+}\mathcal{X} = \widetilde{\mathcal{X}}$$
$$\mathcal{X}_1 = \frac{\widetilde{\mathcal{X}}}{\|\widetilde{\mathcal{X}}\|_F},$$

где $\mathcal{X}_0$ *удовлетворяет* $\|\mathcal{X}_0 - \mathcal{X}_*\|_F \le \varepsilon$ *для любого* $\varepsilon < 1/2$ *и* $\mathcal{X}_0\mathcal{X}^{+}\mathcal{X} = \mathcal{X}_0$*. Тогда,*

$$\|\mathcal{X}_1 - \mathcal{X}_*\|_F \le \frac{1 + \frac{\varepsilon}{2}}{\left(1 - \sqrt{2\varepsilon}\right)(1 - \varepsilon)} \frac{\lambda_1(A^{\mathsf{R}}[\mathcal{X}])}{\lambda_2(A^{\mathsf{R}}[\mathcal{X}])} \|\mathcal{X}_0 - \mathcal{X}_*\|_F.$$

Доказательство. Лемма является следствием утверждения 2.2, примененного к оператору $A^{\mathsf{R}}[\mathcal{X}]$, обращенному на подпространстве векторов $\mathcal{Y}$, удовлетворяющих $\mathcal{Y}\mathcal{X}^{+}\mathcal{X} = \mathcal{Y}$. Обозначим его за $B = (A^{\mathsf{R}}[\mathcal{X}])^{-1}$. Покажем, как B действует на $\mathrm{vec}(\mathcal{X}_0)$. Для этого разложим $\mathrm{vec}(\widetilde{\mathcal{X}})$, $\mathrm{vec}(\mathcal{X}_0)$ по собственным векторам v_i матрицы $A^{\mathsf{R}}[\mathcal{X}]$, соответствующим ненулевым собственным значениям

$$\mathrm{vec}(\widetilde{\mathcal{X}}) = \sum_i c_i v_i, \quad \mathrm{vec}(\mathcal{X}_0) = \sum_i b_i v_i.$$

Это можно сделать, так как $\widetilde{\mathcal{X}}\mathcal{X}^{+}\mathcal{X} = \widetilde{\mathcal{X}}$, $\mathcal{X}\mathcal{X}^{+}\mathcal{X}_0 = \mathcal{X}_0$ и, следовательно, $\mathrm{vec}(\widetilde{\mathcal{X}}), \mathrm{vec}(\mathcal{X}_0) \in \mathrm{Im}(A^{\mathsf{R}}[\mathcal{X}])$. Затем можно показать, что $b_i = c_i/\lambda_i(A^{\mathsf{R}}[\mathcal{X}])$ и, следовательно,

$$\mathrm{vec}(\widetilde{\mathcal{X}}) = B\,\mathrm{vec}(\mathcal{X}_0) = \sum_i \frac{c_i}{\lambda_i(A^{\mathsf{R}}[\mathcal{X}])} v_i,$$

где $\lambda_i(A^{\mathsf{R}}[\mathcal{X}])$ обозначает собственные значения, соответствующие собственным векторам v_i. Далее, имея действие оператора B, выраженное через собственные векторы v_i, можно дословно повторить доказательство Утверждения 2.2. □

Лемма 2.3. *Пусть* $\mathrm{vec}(\mathcal{X}^{\mathsf{L}})$, $\mathrm{vec}(\mathcal{Y}^{\mathsf{L}})$ *и* $\mathrm{vec}(\mathcal{X}_*)$ *являются нормированными собственными векторами, отвечающими наименьшим ненулевым собственным значениям матриц* $A^{\mathsf{L}}[\mathcal{X}]$, $A^{\mathsf{L}}[\mathcal{Y}]$ *и* A *соответственно, тогда*

1. *Для любого* $\mathcal{X}$, *удовлетворяющего* $\|\mathcal{X}^+\mathcal{X} - \mathcal{X}_*^+\mathcal{X}_*\| < 1$, *выполняется*

$$\|\mathcal{X}^{\mathsf{L}} - \mathcal{X}_*\|_F \leq 6\sqrt{2}\,\frac{\lambda_n(A)}{\lambda_2(A) - \lambda_1(A)}\|\mathcal{X}\mathcal{X}^+ - \mathcal{X}_*\mathcal{X}_*^+\|_F.$$

2. *Для любых* $\mathcal{X}$ *и* $\mathcal{Y}$, *удовлетворяющих* $\max\{\|\mathcal{X}^+\mathcal{X} - \mathcal{X}_*^+\mathcal{X}_*\|, \|\mathcal{Y}^+\mathcal{Y} - \mathcal{X}_*^+\mathcal{X}_*\|\} < (\lambda_2(A) - \lambda_1(A))/(8\lambda_n(A))$ *выполняется*

$$\|\mathcal{X}^{\mathsf{L}} - \mathcal{Y}^{\mathsf{L}}\|_F \leq 18\sqrt{2}\,\frac{\lambda_n(A)}{\lambda_2(A) - \lambda_1(A)}\|\mathcal{X}\mathcal{X}^+ - \mathcal{Y}\mathcal{Y}^+\|_F.$$

Доказательство. Для удобства введем следующие обозначения: $\mathrm{P} = I \otimes \mathcal{X}_*\mathcal{X}_*^+$, $\Delta\mathrm{P}_x = I \otimes (\mathcal{X}\mathcal{X}^+ - \mathcal{X}_*\mathcal{X}_*^+)$, $\Delta\mathrm{P}_y = I \otimes (\mathcal{Y}\mathcal{Y}^+ - \mathcal{X}_*\mathcal{X}_*^+)$ $v_1 = \mathrm{vec}(\mathcal{X}_*)$, $\Delta v_x = \mathrm{vec}(\mathcal{X}^{\mathsf{L}}) - \mathrm{vec}(\mathcal{X}_*)$, $\Delta v_y = \mathrm{vec}(\mathcal{Y}^{\mathsf{L}}) - \mathrm{vec}(\mathcal{X}_*)$, $\lambda_i = \lambda_i(A)$, $i = 1, \ldots, n$. Докажем оценку для $\|\Delta v_x\|_2$ через $\|\Delta\mathrm{P}_x\|_F$. Для x имеем (для y аналогично)

$$\begin{aligned}
&Av_1 = \lambda_1 v_1, \quad \mathrm{P}A\mathrm{P}v_1 = \lambda_1 v_1,\\
&(\mathrm{P} + \Delta\mathrm{P}_x)A(\mathrm{P} + \Delta\mathrm{P}_x)(v_1 + \Delta v_x) = (\lambda_1 + \Delta\lambda_x)(v_1 + \Delta v_x),\\
&\mathrm{P}v_1 = v_1, \quad (\mathrm{P} + \Delta\mathrm{P}_x)(v_1 + \Delta v_x) = (v_1 + \Delta v_x),\\
&\|v_1\| = 1, \quad \|v_1 + \Delta v_x\| = 1.
\end{aligned}$$

По теореме Бауэра-Файка и используя факт о том, что эрмитовы матрицы унитарно диагонализуемы, возмущение собственного значения $\Delta\lambda_x$ может быть оценено через норму матрицы возмущения:

$$\begin{aligned}
|\Delta\lambda_x| &\leq \|(\mathrm{P} + \Delta\mathrm{P}_x)A(\mathrm{P} + \Delta\mathrm{P}_x) - \mathrm{P}A\mathrm{P}\|_2 \leq\\
&\leq \lambda_n \cdot \left(2\|\Delta\mathrm{P}_x\|_2 + \|\Delta\mathrm{P}_x\|_2^2\right) \leq 3\lambda_n\|\Delta\mathrm{P}_x\|_2,
\end{aligned} \tag{2.54}$$

при $\|\Delta\mathrm{P}_x\|_2 \leq 1$. Также имеем

$$(\mathrm{P} + \Delta\mathrm{P}_x)(A - \lambda_1 I)(\mathrm{P} + \Delta\mathrm{P}_x)(v_1 + \Delta v_x) = \Delta\lambda_x(v_1 + \Delta v_x).$$

Представим $\Delta v_x = c_{1x}v_1 + \Delta v_x^{\perp}$, где $(\Delta v_x^{\perp}, v_1) = 0$. Тогда благодаря тому, что $(A - \lambda_1 I)v_1 = 0$

$$(\mathrm{P} + \Delta\mathrm{P}_x)(A - \lambda_1 I)(\mathrm{P} + \Delta\mathrm{P}_x)(v_1 + \Delta v_x) = (\mathrm{P} + \Delta\mathrm{P}_x)(A - \lambda_1 I)\Delta v_x^{\perp},$$

следовательно,

$$(\mathrm{P} + \Delta\mathrm{P}_x)(A - \lambda_1 I)\Delta v_x^{\perp} = \Delta\lambda_x(v_1 + \Delta v_x). \tag{2.55}$$

Умножая последнее выражение на $(\Delta v_x^{\perp})^{\top}$, получим

$$\begin{aligned}(\Delta v_x^{\perp}, (\mathrm{P} + \Delta\mathrm{P}_x)(A - \lambda_1 I)\Delta v_x^{\perp}) &= \Delta\lambda_x(\Delta v_x^{\perp}, v_1 + \Delta v_x),\\ ((\mathrm{P} + \Delta\mathrm{P}_x)\Delta v_x^{\perp}, (A - \lambda_1 I)\Delta v_x^{\perp}) &= \Delta\lambda_x(\Delta v_x^{\perp}, v_1 + \Delta v_x).\end{aligned} \tag{2.56}$$

Теперь перепишем $(\mathrm{P} + \Delta\mathrm{P}_x)\Delta v_x^{\perp}$. Используя $(\mathrm{P} + \Delta\mathrm{P}_x)(v_1 + \Delta v_x) = v_1 + \Delta v_x$ и $\mathrm{P}v_1 = v_1$ мы имеем

$$\begin{aligned}(\mathrm{P} + \Delta\mathrm{P}_x)\Delta v_x^{\perp} &= (\mathrm{P} + \Delta\mathrm{P}_x)(v_1 + \Delta v_x - v_1 - c_{1x}v_1) =\\ &= v_1 + \Delta v_x - v_1 - \Delta\mathrm{P}_x v_1 - c_{1x}v_1 - c_{1x}\Delta\mathrm{P}_x v_1 =\\ &= \Delta v_x - (c_{1x} + 1)\Delta\mathrm{P}_x v_1 - c_{1x}v_1.\end{aligned} \tag{2.57}$$

Подставляя последнее выражение в (2.56) и учитывая, что $(A - \lambda_1 I)v_1 = 0$, $(A - \lambda_1 I)\Delta v_x = \Delta v_x^{\perp}$, мы получим

$$\begin{aligned}(\Delta v_x^{\perp} - (c_{1x} + 1)\Delta\mathrm{P}_x v_1, (A - \lambda_1 I)\Delta v_x^{\perp}) &= \Delta\lambda_x(\Delta v_x^{\perp}, v_1 + \Delta v_x)\\ (\Delta v_x^{\perp}, (A - \lambda_1 I)\Delta v_x^{\perp}) &= \Delta\lambda_x(\Delta v_x^{\perp}, v_1 + \Delta v_x) + (c_{1x} + 1)\,(\Delta\mathrm{P}_x v_1, (A - \lambda_1 I)\Delta v_x^{\perp})\end{aligned}$$

Следовательно,

$$\begin{aligned}(\lambda_2 - \lambda_1)\|\Delta v_x^{\perp}\|^2 &\le (\Delta v_x^{\perp}, (A - \lambda_1 I)\Delta v_x^{\perp}) =\\ &= \Delta\lambda_x(\Delta v_x^{\perp}, v_1 + \Delta v_x) + (c_{1x} + 1)\,(\Delta\mathrm{P}_x v_1, (A - \lambda_1 I)\Delta v_x^{\perp}) \le\\ &\le |\Delta\lambda_x|\,\|\Delta v_x^{\perp}\| + (\lambda_n - \lambda_1)(|c_{1x}| + 1)\|\Delta\mathrm{P}_x\|_2\|\Delta v_x^{\perp}\|.\end{aligned}$$

Используя неравенство (2.54) для $|\Delta\lambda_x|$, учитывая $|c_{1x}| \le 2$ (из условий нормировки) и $\lambda_n - \lambda_1 \le \lambda_n$, мы получаем

$$\|\Delta v_x^{\perp}\| \le 6\,\frac{\lambda_n}{\lambda_2 - \lambda_1}\|\Delta\mathrm{P}_x\|_2 \tag{2.58}$$

Поскольку

$$\|\Delta v_x\|^2 = c_{1x}^2 + \|\Delta v_x^{\perp}\|^2,$$

нашей финальной целью является оценка c_{1x} через $\|\Delta v_x^{\perp}\|$. Мы получаем ее из условий нормировки:

$$\begin{aligned}\|v_1 + \Delta v_x\|^2 - \|v_1\|^2 &= 1 - 1 = 0,\\ 2(\Delta v_x, v_1) + \|\Delta v_x\|^2 &= 0,\\ 2c_{1x} + c_{1x}^2 + \|\Delta v_x^{\perp}\|^2 &= 0,\\ c_{1x} &= -1 \pm \sqrt{1 - \|\Delta v_x^{\perp}\|^2},\end{aligned}$$

$c_{1x} = -1-\sqrt{1-\|\Delta v_x^\perp\|^2}$ соответствует малому возмущению собственного вектора $-v_1$ и, следовательно, мы рассматриваем только

$$c_{1x} = -1 + \sqrt{1-\|\Delta v_x^\perp\|^2}. \tag{2.59}$$

Таким образом,

$$\begin{aligned}\|\Delta v_x\|^2 = c_{1x}^2 + \|\Delta v_x^\perp\|^2 = \left(-1+\sqrt{1-\|\Delta v_x^\perp\|^2}\right)^2 + \|\Delta v_x^\perp\|^2 = \\ = 2\left(1-\sqrt{1-\|\Delta v_x^\perp\|^2}\right) = \frac{2\|\Delta v_x^\perp\|^2}{1+\sqrt{1-\|\Delta v_x^\perp\|^2}} \le 2\|\Delta v_x^\perp\|^2.\end{aligned}$$

Подставляя (2.58) в последнее выражение,

$$\|\Delta v_x\| \le 6\sqrt{2}\,\frac{\lambda_n}{\lambda_2-\lambda_1}\|\Delta \mathrm{P}_x\|_2.$$

Теперь докажем вторую часть утверждения. Целью является оценка $\|\Delta v_x - \Delta v_y\|$ через $\|\Delta \mathrm{P}_x - \Delta \mathrm{P}_y\|$. Во-первых, из теоремы Бауэра-Файка,

$$\begin{aligned}|\Delta\lambda_x - \Delta\lambda_y| \le \|(\mathrm{P}+\Delta\mathrm{P}_x)A(\mathrm{P}+\Delta\mathrm{P}_x) - (\mathrm{P}+\Delta\mathrm{P}_y)A(\mathrm{P}+\Delta\mathrm{P}_y)\|_2 = \\ = \|(\Delta\mathrm{P}_x-\Delta\mathrm{P}_y)A\,\mathrm{P} + \mathrm{P}A(\Delta\mathrm{P}_x-\Delta\mathrm{P}_y) + (\Delta\mathrm{P}_x-\Delta\mathrm{P}_y)A\Delta\mathrm{P}_x + \\ + \Delta\mathrm{P}_y A(\Delta\mathrm{P}_x-\Delta\mathrm{P}_y)\|_2 \le 4\lambda_n\|\Delta\mathrm{P}_x-\Delta\mathrm{P}_y\|_2\end{aligned} \tag{2.60}$$

для $\|\Delta\mathrm{P}_x\| \le 1$, $\|\Delta\mathrm{P}_y\| \le 1$. Используя (2.55) мы имеем

$$\begin{aligned}(\mathrm{P}+\Delta\mathrm{P}_x)(A-\lambda_1 I)\Delta v_x^\perp = \Delta\lambda_x(v_1+\Delta v_x), \\ (\mathrm{P}+\Delta\mathrm{P}_y)(A-\lambda_1 I)\Delta v_y^\perp = \Delta\lambda_y(v_1+\Delta v_y),\end{aligned} \tag{2.61}$$

и, следовательно,

$$(\mathrm{P}+\Delta\mathrm{P}_x)(A-\lambda_1 I)\Delta v_x^\perp - (\mathrm{P}+\Delta\mathrm{P}_y)(A-\lambda_1 I)\Delta v_y^\perp = (\Delta\lambda_x-\Delta\lambda_y)v_1 + \Delta\lambda_x\Delta v_x - \Delta\lambda_y\Delta v_y.$$

или эквивалентно,

$$\begin{aligned}(\mathrm{P}+\Delta\mathrm{P}_x)(A-\lambda_1 I)(\Delta v_x^\perp - \Delta v_y^\perp) + (\Delta\mathrm{P}_x-\Delta\mathrm{P}_y)(A-\lambda_1 I)\Delta v_y^\perp = \\ = (\Delta\lambda_x-\Delta\lambda_y)(v_1+\Delta v_x) + \Delta\lambda_y(\Delta v_x-\Delta v_y).\end{aligned}$$

Следующим шагом является умножение последнего выражения на $(\Delta v_x^\perp - \Delta v_y^\perp)^\top$. Используя симметрию $\mathrm{P}+\Delta\mathrm{P}_x$:

$$\begin{aligned}\Big((\mathrm{P}+\Delta\mathrm{P}_x)(\Delta v_x^\perp-\Delta v_y^\perp), (A-\lambda_1 I)(\Delta v_x^\perp-\Delta v_y^\perp)\Big) = \\ \Big(\Delta v_x^\perp - \Delta v_y^\perp, -(\Delta\mathrm{P}_x-\Delta\mathrm{P}_y)(A-\lambda_1 I)\Delta v_y^\perp - (\Delta\lambda_x-\Delta\lambda_y)(v_1+\Delta v_x) + \Delta\lambda_y(\Delta v_x-\Delta v_y)\Big).\end{aligned} \tag{2.62}$$

Перепишем $(\mathrm{P}+\Delta\mathrm{P}_x)(\Delta v_x^\perp - \Delta v_y^\perp)$ и используем (2.57):

$$\begin{aligned}(\mathrm{P}+\Delta\mathrm{P}_x)(\Delta v_x^\perp - \Delta v_y^\perp) =& (\mathrm{P}+\Delta\mathrm{P}_x)\Delta v_x^\perp - (\mathrm{P}+\Delta\mathrm{P}_y)\Delta v_y^\perp + (\Delta\mathrm{P}_y - \Delta\mathrm{P}_x)\Delta v_y^\perp = \\ =& \Delta v_x - \Delta v_y - (1+c_{1x})(\Delta\mathrm{P}_x - \Delta\mathrm{P}_y)v_1 - (c_{1x}-c_{1y})\Delta\mathrm{P}_y v_1 + \\ & (c_{1y}-c_{1x})v_1 + (\Delta\mathrm{P}_y - \Delta\mathrm{P}_y)\Delta v_y^\perp.\end{aligned}$$

Подставляя последнее выражение в (2.62), мы получим

$$\begin{aligned}&(\lambda_2-\lambda_1)\|\Delta v_x^\perp - \Delta v_y^\perp\|^2 \le (\Delta v_x^\perp - \Delta v_y^\perp, (A-\lambda_1 I)(\Delta v_x^\perp - \Delta v_y^\perp)) = \\ &|(1+c_{1x})((\Delta\mathrm{P}_x - \Delta\mathrm{P}_y)v_1, (A-\lambda_1 I)(\Delta v_x^\perp - \Delta v_y^\perp)) + (c_{1x}-c_{1y})(\Delta\mathrm{P}_y v_1, (A-\lambda_1 I)(\Delta v_x^\perp - \Delta v_y^\perp)) + \\ &\Big((\Delta\mathrm{P}_y - \Delta\mathrm{P}_y)\Delta v_y^\perp, (A-\lambda_1 I)(\Delta v_x^\perp - \Delta v_y^\perp)\Big) + \\ &\Big(\Delta v_x^\perp - \Delta v_y^\perp, -(\Delta\mathrm{P}_x - \Delta\mathrm{P}_y)(A-\lambda_1 I)\Delta v_y^\perp - (\Delta\lambda_x - \Delta\lambda_y)(v_1+\Delta v_x) + \Delta\lambda_y(\Delta v_x - \Delta v_y)\Big)| \le \\ &5\lambda_n\|\Delta v_x^\perp - \Delta v_y^\perp\|\,\|\Delta\mathrm{P}_x - \Delta\mathrm{P}_y\|_2 + \lambda_n|c_{1x}-c_{1y}|\,\|\Delta\mathrm{P}_y\|_2\|\Delta v_x^\perp - \Delta v_y^\perp\| + |\Delta\lambda_x - \Delta\lambda_y|\,\|\Delta v_x^\perp - \Delta v_y^\perp\|_2 + \\ &3\lambda_n\|\Delta\mathrm{P}_y\|_2\|\Delta v_x^\perp - \Delta v_y^\perp\|^2\end{aligned}$$

Используя (2.60) и деля последнее неравенство на $\|\Delta v_x^\perp - \Delta v_y^\perp\|$, мы получаем

$$(\lambda_2-\lambda_1)\|\Delta v_x^\perp - \Delta v_y^\perp\| \le 9\lambda_n\|\Delta\mathrm{P}_x - \Delta\mathrm{P}_y\|_2 + \lambda_n\|\Delta\mathrm{P}_y\|_2|c_{1x}-c_{1y}| + 3\lambda_n\|\Delta\mathrm{P}_y\|_2\|\Delta v_x^\perp - \Delta v_y^\perp\| \tag{2.63}$$

Оценим $|c_{1x}-c_{1y}|$, используя (2.59):

$$\begin{aligned}|c_{1x}-c_{1y}| &= \left|\sqrt{1-\|\Delta v_x^\perp\|^2} - \sqrt{1-\|\Delta v_y^\perp\|^2}\right| = \frac{\left|\|\Delta v_x^\perp\|^2 - \|\Delta v_y^\perp\|^2\right|}{\sqrt{1-\|\Delta v_x^\perp\|^2} + \sqrt{1-\|\Delta v_y^\perp\|^2}} = \\ &= \frac{\|\Delta v_x^\perp\| + \|\Delta v_y^\perp\|}{\sqrt{1-\|\Delta v_x^\perp\|^2} + \sqrt{1-\|\Delta v_y^\perp\|^2}}\left|\|\Delta v_x^\perp\| - \|\Delta v_y^\perp\|\right|,\end{aligned}$$

и для $\|\Delta v_x^\perp\|, \|\Delta v_y^\perp\| \le 1/\sqrt{2}$ получим

$$|c_{1x}-c_{1y}| \le \left|\|\Delta v_x^\perp\| - \|\Delta v_y^\perp\|\right| \le \|\Delta v_x^\perp - \Delta v_y^\perp\|.$$

Подставляя последнее неравенство в (2.63) мы имеем

$$(\lambda_2-\lambda_1)\|\Delta v_x^\perp - \Delta v_y^\perp\| \le 9\lambda_n\|\Delta\mathrm{P}_x - \Delta\mathrm{P}_y\|_2 + 4\lambda_n\|\Delta\mathrm{P}_y\|_2\|\Delta v_x^\perp - \Delta v_y^\perp\|$$

и

$$\|\Delta v_x^\perp - \Delta v_y^\perp\| \le \frac{9\lambda_n}{\lambda_2-\lambda_1-4\lambda_n\|\Delta\mathrm{P}_y\|_2}\|\Delta\mathrm{P}_x - \Delta\mathrm{P}_y\|_2 \le 18\frac{\lambda_n}{\lambda_2-\lambda_1}\|\Delta\mathrm{P}_x - \Delta\mathrm{P}_y\|_2, \tag{2.64}$$

для $\|\Delta\mathrm{P}_y\|_2 \le (\lambda_2-\lambda_1)/(8\lambda_n)$. В итоге,

$$\|\Delta v_x - \Delta v_y\|^2 = (c_{1x}-c_{1y})^2 + \|\Delta v_x^\perp - \Delta v_y^\perp\|^2 \le 2\|\Delta v_x^\perp - \Delta v_y^\perp\|^2.$$

Используя (2.64), мы получаем требуемую оценку. Что и требовалось доказать. □

Замечание 2.3. *Отметим, что мы не могли использовать классические результаты из теории возмущения для задачи на собственные значения, так как эти оценки содержат минимальный зазор (относительно λ_1) между собственными значениями. Поскольку рассматриваемые проекторы имеют нулевое собственное значение, зазор равен λ_1 вместо $\lambda_2 - \lambda_1$, что мешает доказательству сходимости рассматриваемой итерации. Поэтому мы явным образом использовали дополнительную информацию о структуре возмущения.*

Лемма 2.4. *Верно следующее неравенство*

$$\|\mathcal{X}^+\mathcal{X} - \mathcal{Y}^+\mathcal{Y}\|_F \leq 2\max\{\|\mathcal{X}^+\|_2, \|\mathcal{Y}^+\|_2\}\,\|\mathcal{X} - \mathcal{Y}\|_F$$

Доказательство. По неравенству треугольника

$$\begin{aligned}\|\mathcal{X}^+\mathcal{X} - \mathcal{Y}^+\mathcal{Y}\|_F =& \|\mathcal{X}^+\mathcal{X}(I - \mathcal{Y}^+\mathcal{Y}) + (\mathcal{X}^+\mathcal{X} - I)\mathcal{Y}^+\mathcal{Y}\|_F \leq \\ \leq& \|\mathcal{X}^+\mathcal{X}(I - \mathcal{Y}^+\mathcal{Y})\|_F + \|(\mathcal{X}^+\mathcal{X} - I)\mathcal{Y}^+\mathcal{Y}\|_F\end{aligned} \tag{2.65}$$

Оценим сначала $\|\mathcal{X}^+\mathcal{X}(I - \mathcal{Y}^+\mathcal{Y})\|_F$. Поскольку $\|AB\|_F \leq \|A\|_2\|B\|_F$ и спектральная норма ортопроектора равна 1 имеем

$$\begin{aligned}\|\mathcal{X}^+\mathcal{X}(I - \mathcal{Y}^+\mathcal{Y})\|_F \leq& \|\mathcal{X}^+\|_2\|\mathcal{X}(I - \mathcal{Y}^+\mathcal{Y})\|_F = \|\mathcal{X}^+\|_2\|(\mathcal{X} - \mathcal{Y})(I - \mathcal{Y}^+\mathcal{Y})\|_F \leq \\ \leq& \|\mathcal{X}^+\|_2\,\|\mathcal{X} - \mathcal{Y}\|_F\,\|I - \mathcal{Y}^+\mathcal{Y}\|_2 = \|\mathcal{X}^+\|_2\|\mathcal{X} - \mathcal{Y}\|_F\end{aligned}$$

Используя симметричность A, B: $\|AB\|_F^2 = \|(AB)^\top\|_F^2 = \|BA\|_F^2$ для второго слагаемого в (2.65) мы получаем

$$\|(\mathcal{X}^+\mathcal{X} - I)\mathcal{Y}^+\mathcal{Y}\|_F = \|\mathcal{Y}^+\mathcal{Y}(\mathcal{X}^+\mathcal{X} - I)\|_F \leq \|\mathcal{Y}^+\|_2\|\mathcal{X} - \mathcal{Y}\|_F.$$

В итоге,

$$\begin{aligned}\|\mathcal{X}^+\mathcal{X} - \mathcal{Y}^+\mathcal{Y}\|_F \leq& (\|\mathcal{X}^+\|_2 + \|\mathcal{Y}^+\|_2)\,\|\mathcal{X} - \mathcal{Y}\|_F \leq \\ \leq& 2\max\{\|\mathcal{X}^+\|_2, \|\mathcal{Y}^+\|_2\}\,\|\mathcal{X} - \mathcal{Y}\|_F.\end{aligned}$$

Что и требовалось доказать. □

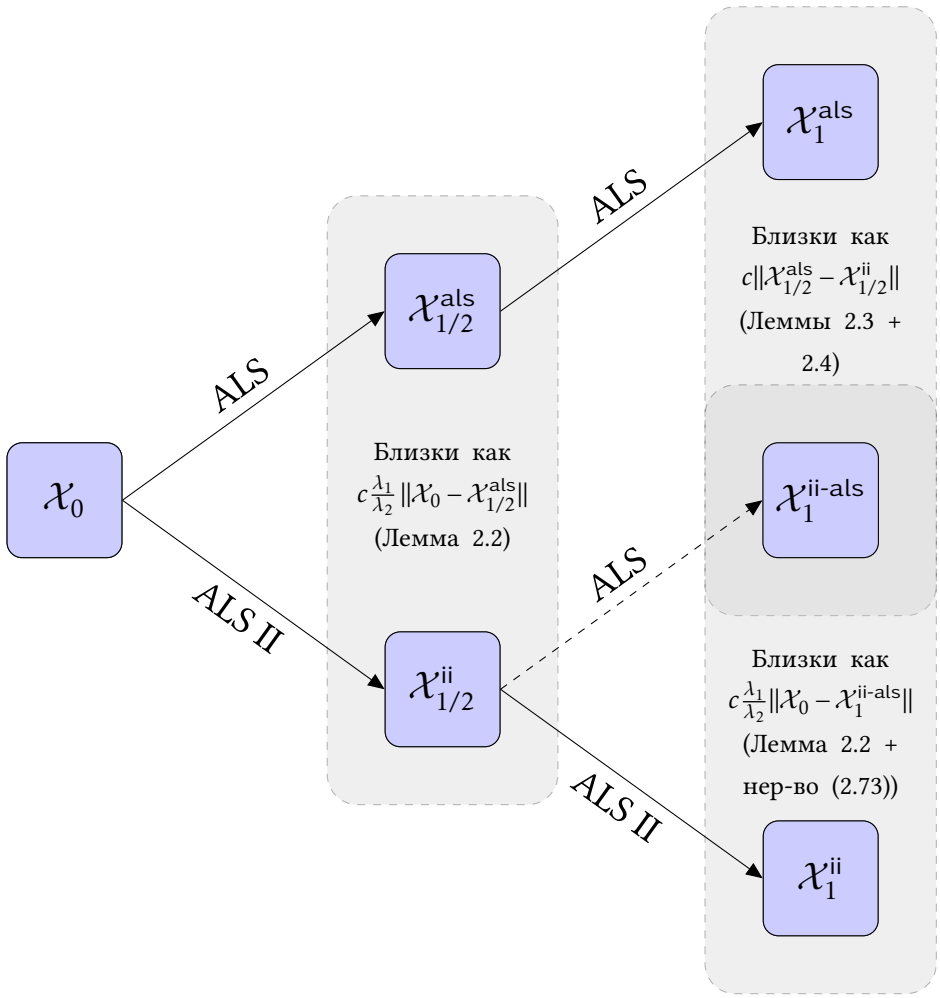


Рис. 2.8: Набросок доказательства Теоремы 2.4. Вспомогательный шаг $\mathcal{X}_1^{\text{ii–als}}$ является половиной шага ALS после $\mathcal{X}_{1/2}^{\text{ii}}$. Также $\|\mathcal{X}_1^{\text{als}} - \mathcal{X}_1^{\text{ii}}\| \le \|\mathcal{X}_1^{\text{als}} - \mathcal{X}_1^{\text{ii–als}}\| + \|\mathcal{X}_1^{\text{ii–als}} - \mathcal{X}_1^{\text{ii}}\|$. Для оценки $\|\mathcal{X}_0 - \mathcal{X}_1^{\text{ii–als}}\|$ (смотри рисунок) мы используем неравенство треугольника: $\|\mathcal{X}_0 - \mathcal{X}_1^{\text{ii–als}}\| \le \|\mathcal{X}_0 - \mathcal{X}_*\| + \|\mathcal{X}_* - \mathcal{X}_1^{\text{als}}\| + \|\mathcal{X}_1^{\text{als}} - \mathcal{X}_1^{\text{ii–als}}\|$. Оценка сверху для нормы $\|\mathcal{X}_1^{\text{als}} - \mathcal{X}_1^{\text{ii–als}}\|$ представлена на рисунке. Норма разности $\|\mathcal{X}_* - \mathcal{X}_1^{\text{als}}\|$ определяется через ρ_{als}.

Доказательство Теоремы 2.4

Идея доказательства изложена на Рисунке 2.8. Аккуратное доказательство приведено ниже. Для упрощения обозначений в доказательстве мы опускаем индекс k и считаем, что на k-ом шаге ALS II мы начинаем с $\mathcal{X}_0$ вместо $\mathcal{X}_k^{\text{ii}}$. Неравенства будут получены с точностью до числовых констант, которые будут обозначаться одним символом c.

По предположению теоремы ALS итерация для минимизации отношения Рэлея начинает сходиться со скоростью ρ_{als} для любого начального приближе-

ния из некоторой окрестности точного решения $U_\varepsilon(\mathcal{X}_*)$. Введем C_1 такую, что

$$C_1 = \sup_{k,\mathcal{X}_0^{\text{als}} \in U_\varepsilon(\mathcal{X}_*)} \frac{\|\mathcal{X}_{k+1/2}^{\text{als}} - X_*\|}{\|X_k^{\text{als}} - X_*\|}, \tag{2.66}$$

откуда следует, что

$$\|\mathcal{X}_{1/2}^{\text{als}} - X_*\| \le C_1 \|X_0 - X_*\|. \tag{2.67}$$

Обозначим также

$$C_2 = 14\sqrt{2}\frac{\lambda_n(A)}{\lambda_2(A) - \lambda_1(A)}.$$

Используя Леммы 2.3 и 2.4, несложно убедиться, что $C_1 \le cC_2 s_{\min}^{-1}(\mathcal{X}_*)$, хотя на практике эта константа заметно меньше. Также несложно проверить, что при $\|\mathcal{X}_{1/2}^{\text{als}}\mathcal{X}_{1/2}^{\text{als}^+} - \mathcal{X}_*\mathcal{X}_*^+\| \le \lambda_1/(9\lambda_n)$ выполняется

$$\frac{\lambda_1(A^{\mathsf{R}}[\mathcal{X}_{1/2}^{\text{als}}])}{\lambda_2(A^{\mathsf{R}}[\mathcal{X}_{1/2}^{\text{als}}])} \le 3\frac{\lambda_1(A)}{\lambda_2(A)}. \tag{2.68}$$

Действительно, используя (2.54) и Лемму 2.4,

$$|\lambda_1(A^{\mathsf{R}}[\mathcal{X}_{1/2}^{\text{als}}]) - \lambda_1(A)| \le 3\lambda_n(A)\|\mathcal{X}_{1/2}^{\text{als}}\mathcal{X}_{1/2}^{\text{als}^+} - \mathcal{X}_*\mathcal{X}_*^+\|.$$

Далее из вариационного принципа следует, что $\lambda_2(A) \le \lambda_2(A^{\mathsf{R}}[\mathcal{X}_*])$:

$$\lambda_2(A^{\mathsf{R}}[\mathcal{X}_*]) = \min_{\substack{\mathcal{X}_*\mathcal{X}_*^+\mathcal{X}=\mathcal{X},\\ \mathcal{X}\neq 0, \mathcal{X}\perp\mathcal{X}_*}} \frac{\left\langle \mathcal{X}, A^{\mathsf{R}}[\mathcal{X}_*]\mathcal{X}\right\rangle}{\langle \mathcal{X}, \mathcal{X}\rangle} \ge \min_{\substack{\mathcal{X}\neq 0,\\ \mathcal{X}\perp\mathcal{X}_*}} \frac{\langle \mathcal{X}, A\mathcal{X}\rangle}{\langle \mathcal{X}, \mathcal{X}\rangle} = \lambda_2(A).$$

Аналогичная оценка выполняется для $A^{\mathsf{L}}[\mathcal{X}_1^{\text{als}}]$.

Используя Лемму 2.2, неравенства (2.68) и (2.67), получим оценку сверху для $\|\mathcal{X}_{1/2}^{\text{als}} - \mathcal{X}_{1/2}^{\text{ii}}\|$:

$$\begin{aligned}\|\mathcal{X}_{1/2}^{\text{als}} - \mathcal{X}_{1/2}^{\text{ii}}\| \le c\frac{\lambda_1}{\lambda_2}\|\mathcal{X}_{1/2}^{\text{als}} - \mathcal{X}_0\| \le c\frac{\lambda_1}{\lambda_2}\left(\|\mathcal{X}_{1/2}^{\text{als}} - \mathcal{X}_*\| + \|\mathcal{X}_* - \mathcal{X}_0\|\right) \le \\ \le c(1 + C_1)\frac{\lambda_1}{\lambda_2}\|\mathcal{X}_* - \mathcal{X}_0\|.\end{aligned} \tag{2.69}$$

Для оценки $\|\mathcal{X}_1^{\text{ii}} - \mathcal{X}_*\|$ мы используем неравенство треугольника:

$$\|\mathcal{X}_1^{\text{ii}} - \mathcal{X}_*\| \le \|\mathcal{X}_1^{\text{als}} - \mathcal{X}_*\| + \|\mathcal{X}_1^{\text{als}} - \mathcal{X}_1^{\text{ii-als}}\| + \|\mathcal{X}_1^{\text{ii}} - \mathcal{X}_1^{\text{ii-als}}\|. \tag{2.70}$$

По предположению теоремы

$$\|\mathcal{X}_1^{\text{als}} - \mathcal{X}_*\| \le \rho_{\text{als}}\|\mathcal{X}_0 - \mathcal{X}_*\|. \tag{2.71}$$

Оценим оставшиеся части неравенства (2.70). Во-первых, используя Леммы 2.3 и 2.4, получаем

$$\begin{aligned}\|\mathcal{X}_1^{\text{als}} - \mathcal{X}_1^{\text{ii-als}}\| &\le C_2\|\mathcal{X}_{1/2}^{\text{als}^+}\mathcal{X}_{1/2}^{\text{als}} - \mathcal{X}_{1/2}^{\text{ii}^+}\mathcal{X}_{1/2}^{\text{ii}}\| \le C_2\frac{2}{s_{\min}}\|\mathcal{X}_{1/2}^{\text{als}} - \mathcal{X}_{1/2}^{\text{ii}}\| \\ &\le c(1+C_1)C_2\frac{2}{s_{\min}}\frac{\lambda_1}{\lambda_2}\|\mathcal{X}_* - \mathcal{X}_0\|,\end{aligned} \tag{2.72}$$

где также использовано, что

$$\|\mathcal{X}_{1/2}^{\text{als}^+}\| \le \frac{1}{s_{\min}(\mathcal{X}_*) - \|\mathcal{X}_{1/2}^{\text{als}} - \mathcal{X}_*\|} \le \frac{1}{s_{\min}(\mathcal{X}_*) - C_1\|\mathcal{X}_0 - \mathcal{X}_*\|} \le \frac{2}{s_{\min}(\mathcal{X}_*)},$$

при $\|X_0 - X_*\| \le s_{\min}(\mathcal{X}_*)/(2C_1)$. Аналогичная оценка верна для $\|\mathcal{X}_{1/2}^{\text{ii}^+}\|$. Далее, заметим, что

$$\begin{aligned}\|\mathcal{X}_{1/2}^{\text{ii}}\mathcal{X}_{1/2}^{\text{ii}^+}\mathcal{X}_0 - \mathcal{X}_1^{\text{ii-als}}\|_F &= \|\mathcal{X}_{1/2}^{\text{ii}}\mathcal{X}_{1/2}^{\text{ii}^+}\mathcal{X}_0 - \mathcal{X}_{1/2}^{\text{ii}}\mathcal{X}_{1/2}^{\text{ii}^+}\mathcal{X}_1^{\text{ii-als}}\|_F \\ &\le \|\mathcal{X}_{1/2}^{\text{ii}}\mathcal{X}_{1/2}^{\text{ii}^+}\|_2\|\mathcal{X}_0 - \mathcal{X}_1^{\text{ii-als}}\|_F = \|\mathcal{X}_0 - \mathcal{X}_1^{\text{ii-als}}\|_F.\end{aligned} \tag{2.73}$$

Используя Лемму 2.2 и неравенство (2.73), получим

$$\begin{aligned}\|\mathcal{X}_1^{\text{ii}} - \mathcal{X}_1^{\text{ii-als}}\| &\le c\frac{\lambda_1}{\lambda_2}\|\mathcal{X}_0 - \mathcal{X}_1^{\text{ii-als}}\| \le \\ &\le c\frac{\lambda_1}{\lambda_2}\left(\|\mathcal{X}_0 - \mathcal{X}_*\| + \|\mathcal{X}_* - \mathcal{X}_1^{\text{als}}\| + \|\mathcal{X}_1^{\text{als}} - \mathcal{X}_1^{\text{ii-als}}\|\right).\end{aligned} \tag{2.74}$$

Верхняя оценка для $\|\mathcal{X}_1^{\text{als}} - \mathcal{X}_1^{\text{ii-als}}\|$ получена в (2.72). Для $\|\mathcal{X}_* - \mathcal{X}_1^{\text{als}}\|$ справедлива оценка (2.71), поэтому

$$\|\mathcal{X}_1^{\text{ii}} - \mathcal{X}_1^{\text{ii-als}}\| \le c\left(2 + c(1+C_1)C_2\frac{2}{s_{\min}}\frac{\lambda_1}{\lambda_2}\right)C_2\frac{\lambda_1}{\lambda_2}\|\mathcal{X}_0 - \mathcal{X}_*\|. \tag{2.75}$$

Таким образом, подставляя (2.71), (2.72) и (2.75) в (2.70), мы получаем требуемую оценку. Осталось только применить доказательство к матрице $A - \sigma I$ и учесть, что $(\lambda_2 - \sigma) - (\lambda_1 - \sigma) = \lambda_2 - \lambda_1$ и $(\lambda_n - \sigma) < \lambda_n$. Что и требовалось доказать.

2.3 Блочный солвер с предобуславливанием на многообразии

В предыдущих частях Главы 2 мы предложили методы для поиска одного собственного значения. Для поиска нескольких собственных векторов можно использовать подход дефляции, когда собственные векторы находятся по

одному. Однако если некоторые собственные значения находятся близко друг к другу, то метод будет сходиться медленно. Для ускорения сходимости необходимо использовать блочные методы. Известным блочным солвером является LOBPCG, предложенный Князевым [76]. Обобщение LOBPCG на тензорные форматы было рассмотрено в работах [84, 82]. В [82] было численно показано, что для избежания роста рангов и/или неустойчивости вычислений необходимо использовать предобуславливатели. В качестве предобуславливателя в диссертации мы предлагаем решать систему в касательном пространстве. Мы называем такой подход предобуславливанием на многообразии [113] (manifold-preconditioning, MP). Как и в случае с обратной итерации можно рассмотреть решение в полном касательном пространстве или на последовательности подпространств с помощью ALS. В настоящем разделе мы фокусируемся на втором подходе.

Для поиска большого нескольких собственных значений передовым методом является eigb [24], который сравним (а на некоторых примерах превосходит) по скорости сходимости и стоимости вычислений аналог [112], разработанный в сообществе, занимающимся разработкой алгоритмов для MPS (matrix product states). Однако метод не подходит для вычисления большого числа собственных значений ($B \approx 100$), так как он использует блочный формат и, следовательно, имеет линейный по B рост рангов. Поэтому мы разработали новый метод, базирующийся на комбинации оптимизации на многообразиях и итерационных методов с округлением.

2.3.1 MP LOBPCG для одного собственного вектора

Начнем описание с поиска одного собственного вектора с помощью LOBPCG и предобуславливания на многообразиях (MP). Классический метод LOBPCG для поиска одного собственного значения выглядит следующим образом

$$\begin{aligned}
\mathcal{R}_k &:= \mathcal{B}(\mathcal{A}(\mathcal{X}_k) - \lambda_k \mathcal{X}_k), \\
\mathcal{P}_{k+1} &:= \alpha_2 \mathcal{R}_k + \alpha_3 \mathcal{P}_k, \\
\mathcal{X}_{k+1} &:= \alpha_1 \mathcal{X}_k + \mathcal{P}_{k+1}, \\
\mathcal{X}_{k+1} &:= \mathcal{X}_{k+1} / \sqrt{\langle \mathcal{X}_{k+1}, \mathcal{X}_{k+1} \rangle},
\end{aligned} \tag{2.76}$$

где $\mathcal{B}$ обозначает предобуславливатель, а вектор коэффициентов $\alpha = (\alpha_1, \alpha_2, \alpha_3)^\top$ выбирается из условия минимизации отношения Рэлея:

$$\mathfrak{R}(\mathcal{X}_{k+1}) = \frac{(\mathcal{A}\mathcal{X}_{k+1}, \mathcal{X}_{k+1})}{(\mathcal{X}_{k+1}, \mathcal{X}_{k+1})}.$$

Нахождение α эквивалентно решению следующей 3×3 задачи на собственные значения

$$\begin{bmatrix} \mathcal{X}_k \\ \mathcal{R}_k \\ \mathcal{P}_k \end{bmatrix} \mathcal{A}[\mathcal{X}_k, \mathcal{R}_k, \mathcal{P}_k]\,\alpha = \lambda \begin{bmatrix} \mathcal{X}_k \\ \mathcal{R}_k \\ \mathcal{P}_k \end{bmatrix} [\mathcal{X}_k, \mathcal{R}_k, \mathcal{P}_k]\,\alpha. \tag{2.77}$$

Здесь использованы обозначения $\mathcal{A}[\mathcal{X}_k, \mathcal{R}_k, \mathcal{P}_k] \equiv [\mathcal{A}\mathcal{X}_k, \mathcal{A}\mathcal{R}_k, \mathcal{A}\mathcal{P}_k]$ и

$$\begin{bmatrix} \mathcal{X}_k \\ \mathcal{R}_k \\ \mathcal{P}_k \end{bmatrix} \begin{bmatrix} \mathcal{X}_k & \mathcal{R}_k & \mathcal{P}_k \end{bmatrix} \equiv \begin{bmatrix} \langle \mathcal{X}_k, \mathcal{X}_k \rangle & \langle \mathcal{X}_k, \mathcal{R}_k \rangle & \langle \mathcal{X}_k, \mathcal{P}_k \rangle \\ \langle \mathcal{R}_k, \mathcal{X}_k \rangle & \langle \mathcal{R}_k, \mathcal{R}_k \rangle & \langle \mathcal{R}_k, \mathcal{P}_k \rangle \\ \langle \mathcal{P}_k, \mathcal{X}_k \rangle & \langle \mathcal{P}_k, \mathcal{R}_k \rangle & \langle \mathcal{P}_k, \mathcal{P}_k \rangle \end{bmatrix}.$$

Обсудим тензорную версию LOBPCG. Вычисление линейных комбинаций, умножение матрицы на вектор, а также вычисление скалярных произведений в тензорных форматах описаны в разделе 1.2. Для избежания роста ранга на шаге пересчета $\mathcal{X}_{k+1}$ применяется оператор округления с фиксированным значением ранга.

Ключевой частью LOBPCG метода является умножение вектора на предобуславливатель $\mathcal{B}$. В качестве предобуславливателя мы используем $\mathcal{B} \approx (\mathcal{A} - \sigma\mathcal{I})^{-1}$ (приближенное равенство записано в смысле ALS минимизации, о которой будет говориться дальше). Если системы решаются с высокой точностью, то имеет смысл дополнительно спроецировать матрицу $(\mathcal{A} - \sigma\mathcal{I})$ на ортогональное дополнение к текущему приближению решения как это было сделано в разделе 2.1 с обобщением метода Якоби-Дэвидсона.

По аналогии с ALS II (раздел 2.2) мы предлагаем решать следующую минимизационную задачу

$$\begin{gathered} \langle (\mathcal{A} - \sigma\mathcal{I})\mathcal{Y}, \mathcal{Y} \rangle - 2\langle (\mathcal{A}\mathcal{X}_k - \lambda_k \mathcal{X}_k), \mathcal{Y} \rangle \to \min, \\ \text{TT-rank}(\mathcal{Y}) = \mathbf{r} \end{gathered} \tag{2.78}$$

с одним или двумя проходами ALS минимизации. В качестве сдвига используется оценка снизу на наименьшее собственное значение. Мы называем конструкцию такого нелинейного предобуславливателя предобуславливателем на многообразии (*manifold preconditioner, MP*).

2.3.2 Блочный случай

В этом разделе мы обобщаем тензорную версию MP LOBPCG алгоритма на блочный случай. После того, как мы применяем MP LOBPCG итерацию, мы независимо корректируем каждый собственный вектор с помощью предложенной ALS II. Отметим, что некоторые собственные значения могут быть близки друг к другу или даже вырожденны. Поэтому нельзя применять ALS II метод к каждому собственному независимо. Для устранения этой проблемы предлагается блочная версия ALS SII (ALS Simultaneous II), которая применяется для сгруппированных в кластеры собственных значений. Описание этого разбиения на кластеры и блочного итерирования также будет приведено в настоящем разделе.

Блочный операции в TT-формате. Одной из важных операций является ортогонализация набора векторов в TT-формате. Ортогонализация выполняется с помощью QR разложения, полученного с использованием разложения Холецкого. Рассмотрим детально процедуру ортогонализации блочного вектора $\mathbf{X} = [\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(B)}]$. Во-первых, вычисляется матрица Грама

$$G = \mathbf{X}^\top \mathbf{X},$$

где мы использовали обозначение

$$\mathbf{X}^\top \mathbf{Y} \triangleq \begin{bmatrix} \left\langle \mathcal{X}^{(1)}, \mathcal{Y}^{(1)} \right\rangle & \ldots & \left\langle \mathcal{X}^{(1)}, \mathcal{Y}^{(B)} \right\rangle \\ \vdots & & \vdots \\ \left\langle \mathcal{X}^{(B)}, \mathcal{Y}^{(1)} \right\rangle & \ldots & \left\langle \mathcal{X}^{(B)}, \mathcal{Y}^{(B)} \right\rangle \end{bmatrix}.$$

Матрица Грама может быть вычислена за $\mathcal{O}(B^2 n d r^3)$ операций, так как вычисление одного скалярного произведения требует $\mathcal{O}(n d r^3)$ операций. Затем мы вычисляем разложение Холецкого $B \times B$ матрицы Грама $G = LL^\top$. Рассмотрим вычисление блочного матрично-векторного умножения, которое мы обозначаем как $\mathrm{BLOCK\_MATVEC}(\mathbf{X}, L)$. Для произвольной матрицы $M \in \mathbb{R}^{P \times B}$ функция $\mathrm{BLOCK\_MATVEC}(\mathbf{X}, M)$ возвращает блочный вектор $\mathbf{Y} = [\mathcal{Y}^{(1)}, \ldots, \mathcal{Y}^{(P)}]$ такой, что

$$\mathcal{Y}^{(i)} = \sum_{j=1}^{P} L_{ij} \mathcal{X}^{(j)}, \quad i = 1, \ldots, B. \tag{2.79}$$

Алгоритм 2.4 Вспомогательные функции

$\mathcal{Y} = \mathrm{MULTIFUNCRS}(\mathrm{func}, \mathbf{X})$: вычисляет $\mathrm{func}(\mathbf{X})$ с помощью метода крестовой аппроксимации.

$\mathbf{Y} = \mathrm{BLOCK\_MATVEC}(\mathbf{X}, M)$: блочное умножение блочного вектора $\mathbf{X}$, состоящего из TT тензоров на матрицу из действительных чисел M с использованием $\mathrm{MULTIFUNCRS}$ функции, см. (2.79).

$\mathbf{Y} = \mathrm{QR}(\mathbf{X})$: ортогонализует TT тензоры $\mathcal{X}_1, \ldots, \mathcal{X}_B$: $\mathbf{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_B)$ с использованием разложения Холецкого или с использованием модифицированного алгоритма Грама-Шмидта. Матрично-векторные умножения делаются с использованием $\mathrm{BLOCK\_MATVEC}$.

$\mathbf{X} = \mathrm{ALS}(\mathcal{A}, \mathbf{F}, n_{\mathrm{swp}})$: решает $\mathcal{A}(\mathcal{X}^{(i)}) = \mathcal{F}^{(i)}, i = 1, \ldots, \mathrm{length}(\mathbf{F})$ с использованием n_{swp} проходов ALS метода для минимизации функционала энергии с ограничением по рангу.

$\mathbf{Y} = \mathrm{ORTHO}(\mathbf{X}, \mathbf{Q})$: ортогонализует TT тензоры $\mathcal{X}_1, \ldots, \mathcal{X}_B$: $\mathbf{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_B)$ по отношению к $\mathbf{Q}$: $\mathcal{Y}^{(i)} = \mathcal{X}^{(i)} - \sum_{j=1}^{B} \left\langle \mathcal{X}^{(i)}, \mathcal{Q}^{(j)} \right\rangle \mathcal{Q}^{(j)}$. Для избежания роста ранга используется процедура округления если $\mathrm{length}(\mathbf{X})$ мало, и $\mathrm{MULTIFUNCRS}$ если $\mathrm{length}(\mathbf{X})$ велико.

$\mathbf{Y} = \mathcal{T}_r(\mathbf{X})$: округляет каждый тензор $\mathcal{X}_1, \ldots, \mathcal{X}_B$: $\mathbf{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_B)$ с рангом $\mathbf{r}$ с помощью процедуры округления.

Если P мало, скажем $P < 20$, тогда сложение может быть вычислено с увеличением ранга и дальнейшим округлением. Стандартное значение B в численных экспериментах $B = 80$ и $2B = 160$, так что для уменьшения сложности мы используем метод крестовой аппроксимации, который позволяет вычислять TT-разложение тензора по небольшому числу его элементов. А именно, тензор ранга r может быть восстановлен с использованием интерполяционной формулы [109] со сложностью $\mathcal{O}(dnr^3)$. Если тензор приближенно малого ранга, тогда он может быть приближен с заданной точностью с помощью того же подхода, и существуют оценки сходимости [121]. Для поиска TT-представления

$\mathcal{Y}^{(i)}$ мы вычисляем $\mathcal{O}(dnr^2)$ элементов тензора $\mathcal{Y}^{(i)}$, явно вычисляя элементы в $\mathcal{X}^{(j)}$ и суммируя их с коэффициентами L_{ij}. Этот подход позволяет вычислить произвольные функции блочного вектора $f(\mathbf{X})$ (при условии, что $f(\mathbf{X})$ также имеет малый ранг). Мы обозначаем этот подход как MULTIFUNCRS. Обычно MULTIFUNCRS используется в случае, если число тензоров на входе велико или если необходимо вычислить некоторую нелинейную функцию от входного тензора. Похожий подход использован в Главе 4 для быстрого вычисления свертки.

Результат решения блочной системы $\mathcal{A}(\mathbf{X}) = \mathbf{F}$ с использованием n_{swp} проходов для решения задачи оптимизации (2.78) с помощью ALS обозначается как $\mathbf{X} = \text{ALS}(\mathcal{A},\ \mathbf{F},\ n_{\text{swp}})$, где

$$\mathcal{A}(\mathbf{X}) \triangleq [\mathcal{A}(\mathcal{X}^{(1)}),\ldots,\mathcal{A}(\mathcal{X}^{(B)})].$$

Используемые вспомогательные функции представлены в Алгоритме 2.4.

Предобусловленный на многообразии LOBPCG метод. Мы используем MP LOBPCG метод для поиска начального приближения для ALS SII. Проблема заключается в том, что каждая итерация MP LOBPCG является вычислительно более затратной по сравнению с обратной итерацией для поиска большого числа собственных значений. Поэтому мы запускаем LOBPCG с малым значением рангов, и затем корректируем полученное решение с большими ранга с использованием предложенной обратной итерации.

LOBPCG алгоритм описан в Алгоритме 2.5. Вспомогательные функции описаны в Алгоритме 2.4. Мы также используем MATLAB обозначения для подматриц, например $S(B:3B, 1:B)$.

Блочное матрично-векторное умножение (2.79) возникает при умножении на $B \times 2B$ матрицу, где B число собственных векторов. Когда $2B$ является большим числом, мы используем MULTIFUNCRS для блочного матрично-векторного умножения вместо обычного округления по рангу от суммы. Это является наиболее затратной частью в алгоритме и сложность равняется $\mathcal{O}(B^2dnr^3)$ операций. Другой затратной частью является умножение TT-матрицы на TT-вектор: $\mathcal{O}(Bdn^2r^2R^2)$ арифметических операций. Таким образом, общая сложность метода равняется $\mathcal{O}(Bdnr^2(Br + nR^2))$.

Для ускорения вычислений, мы рассматриваем версию LOBPCG с дефляцией, чтобы исключить из вычислений сошедшиеся собственные векторы. В данном случае невязка должна быть ортогонализована по отношению к сошедшимся собственным векторам. Может быть также полезно увеличивать ранг несошедшихся собственных векторов после каждого шага дефляции.

Алгоритм 2.5 MP LOBPCG метод

Require: TT-матрица $\mathcal{A}$; начальное приближение $\mathbf{X}_0 = [\mathcal{X}_0^{(1)} \dots \mathcal{X}_0^{(B)}]$, где $\mathcal{X}_0^{(i)}$ являются TT тензорами; ранг r

Ensure: Λ и $\mathbf{X}$ – начальное приближение B наименьших собственных значений и собственных векторов $\mathcal{A}$

$\mathbf{X}_0 := \mathrm{QR}(\mathbf{X}_0)$

$(\mathbf{X}_0^\top \mathcal{A}(\mathbf{X}_0))\, S_0 = S_0 \Lambda_0$ ▷ Собственное разложение

$\mathbf{X}_0 := \mathrm{BLOCK\_MATVEC}(\mathbf{X}_0, S_0)$

$\mathbf{R}_0 := \mathcal{A}(\mathbf{X}_0) - \mathbf{X}_0 \Lambda_0$, $\mathbf{P}_0 := 0 \cdot \mathbf{X}_0$

for $k = 0, 1, \dots$ до сходимости **do**

  $\mathbf{R}_k := \mathrm{ORTHO}(\mathbf{R}_k, \mathbf{Q})$ ▷ $\mathbf{Q}$ – сошедшиеся векторы

  $\mathbf{R}_k := \mathrm{ALS}(\mathcal{A} - \sigma\mathcal{I},\ \mathbf{R}_k,\ n_{\mathrm{swp}})$

  $\widetilde{H} := [\mathbf{X}_k, \mathbf{R}_k, \mathbf{P}_k]^\top \mathcal{A}([\mathbf{X}_k, \mathbf{R}_k, \mathbf{P}_k])$

  $\widetilde{M} := [\mathbf{X}_k, \mathbf{R}_k, \mathbf{P}_k]^\top [\mathbf{X}_k, \mathbf{R}_k, \mathbf{P}_k]$

  $\widetilde{H}\widetilde{S}_k = \widetilde{M}\widetilde{S}_k\widetilde{\Lambda}_k, \widetilde{S}_k^\top \widetilde{M}\widetilde{S}_k = I_{3B}$ ▷ Собственное разложение

  $\mathbf{P}_{k+1} := \mathrm{BLOCK\_MATVEC}([\mathbf{R}_k, \mathbf{P}_k],\ \widetilde{S}_k(B{:}3B, 1{:}B))$

  $\mathbf{X}_{k+1} := \mathrm{BLOCK\_MATVEC}(\mathbf{X}_k,\ \widetilde{S}_k(1{:}B, 1{:}B)) + \mathbf{P}_{k+1}$

  $\mathbf{X}_{k+1} := \mathcal{T}_r(\mathbf{X}_{k+1})$, $\Lambda_{k+1} := \widetilde{\Lambda}_k(1{:}B, 1{:}B)$

  $\mathbf{R}_{k+1} := \mathcal{A}(\mathbf{X}_{k+1}) - \mathbf{X}_{k+1}\Lambda_{k+1}$

  if Некоторые векторы в $\mathbf{R}_{k+1}$ сошлись **then**

    Дополнить $\mathbf{Q}$ соответствующими векторами из $\mathbf{X}_{k+1}$

    Перезапустить алгоритм с новым $\mathbf{X}_0$

    (Опционально) увеличить r

  end if

end for

return $\Lambda_{k+1}, \mathbf{X}_{k+1}$

ALS блочная обратная итерация

В настоящем подразделе мы представляем блочную версию ALS обратной итерации. Предположим, что у нас уже есть начальное приближение для собственного вектора и собственного значения линейного оператора $\mathcal{A}$, представленного в TT-формате. Сначала мы разделяем собственные значения из начального приближения на кластеры собственных значений. Близость собственных значений определяется с помощью порогового параметра. Если кластер состоит из одного собственного значения, мы запускаем версию, описанную в разделе 2.2. В противоположном случае необходимо дополнительно ортогонализовывать собственные векторы на каждой итерации. Итоговый алгоритм описан в Алгоритме 2.6.

Алгоритм 2.6 Блочная ALS обратная итерация (ALS SII) со сдвигами

Require: TT-матрица $\mathcal{A}$; начальное приближение $\mathbf{X}_0 = [\mathcal{X}_0^{(1)} \dots \mathcal{X}_0^{(B)}]$ где $\mathcal{X}_0^{(i)}$ являются TT тензорами; ранг r

Ensure: Λ и $\mathbf{X}$ – аппроксимация первых B собственных значений, близких к σ и соответствующие собственные векторы

Сгруппировать близкие друг другу собственные значения в кластеры собственных значений
$X := [\,]$, $\Lambda := [\,]$
for каждого кластера с номером ν **do**
    Вычислить сдвиг σ_ν – среднее собственное значение в кластере ν
    Вычислить $\mathbf{X}_0^\nu$ – соответствующий подвектор в $\mathbf{X}_0$
    $\mathbf{X}_0^\nu := \mathrm{QR}(\mathbf{X}_0^\nu)$
    for $k = 0, 1, \dots$ до сходимости **do**
        $\mathbf{X}_{k+1}^\nu := \mathrm{ALS}(\mathcal{A} - \sigma_\nu \mathcal{I},\ \mathbf{X}_k^\nu,\ n_{\mathrm{swp}})$
        $\mathbf{X}_{k+1}^\nu := \mathrm{QR}(\mathbf{X}_{k+1}^\nu)$
    end for
    $\mathbf{X} := [\mathbf{X}, \mathbf{X}_{k+1}^\nu]$, $\Lambda := [\Lambda, \mathrm{diag}({\mathbf{X}_{k+1}^\nu}^\top \mathcal{A}(\mathbf{X}_{k+1}^\nu))]$
end for
return $\Lambda, \mathbf{X}$

Если размер кластера гораздо меньше, чем общее число требуемых собственных значений, то сложность поиска собственных значений внутри каждого кластера полностью определяется сложностью решения систем $\mathcal{O}(dn^2r^2R^2)$. Таким образом, общая сложность обратной итерации зависит линейно от B: $\mathcal{O}(Bdn^2r^2R^2)$. Отметим, что каждый кластер можно итерировать независимо.

2.3.3 Расчет колебательного спектра молекул

Прототип программы написан на языке Python с использованием библиотеки ttpy https://github.com/oseledets/ttpy. Код предложенного алгоритма доступен по ссылке https://bitbucket.org/rakhuba/ttvibr. Для базовых операций линейной алгебры была использована библиотека MKL. Python и MKL устанавливались с помощью Anaconda Python Distribution https://www.continuum.io. Версия Python 2.7.11. Версия MKL 11.1-1. Вычисления были сделаны на одном ядре процессора Intel Core i7 2.6 GHz с 8GB RAM. Однако был использован только 1 поток.

Дискретизация. По аналогии с [132] мы рассматриваем уравнение Шредингера без $\pi-\pi$ члена и члена в кинетической энергии, записанной в нормальных координатах, имеющего вид потенциальной энергии. Гамильтониан в этом случае имеет следующий вид

$$\mathcal{H} = -\frac{1}{2}\sum_{i=1}^{d}\omega_i\frac{\partial^2}{\partial q_i^2} + V(q_1,\dots,q_d), \tag{2.80}$$

где V обозначает поверхность потенциальной энергии (PES).

Мы дискретизуем задачу (2.80) с использованием псевдоспектрального метода (DVR) на прямоугольной эрмитовой сетке [9], так что каждая собственная функция представляется в виде

$$\Psi_k(q_1,\dots,q_d) \approx \sum_{i_1,\dots,i_d=1}^{n} \mathcal{X}^{(k)}_{i_1,\dots,i_d}\,\psi_{i_1}(q_1)\dots\psi_{i_d}(q_d), \tag{2.81}$$

где $\psi_i(q_i)$ обозначает одномерные DVR базисные функции.

Гамильтониан (2.80) состоит из двух частей: часть, имеющая вид Лапласиана и PES. Хорошо известно, что Лапласиан может быть записан с использо-

ванием тензорных произведений:

$$\mathcal{D} = D_1 \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes D_d,$$

где D_i является дискретизацией одномерного оператора по i-й моде.

Псевдоспектральная дискретизация PES записывается как $\mathcal{V}$. Оператор, соответствующий умножению на $\mathcal{V}$ является диагональным. В итоге, гамильтониан записывается как

$$\mathcal{A} = D_1 \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes D_d + \operatorname{diag}(\mathcal{V}). \tag{2.82}$$

64-D билинейный осциллятор. Во-первых, мы тестируем наш подход на модельном гамильтониане, когда решение известно аналитически. Следуя [132] мы выбираем 64-х мерный билинейный осциллятор

$$\mathcal{H} = \sum_{i=1}^{d} \frac{\omega_i}{2} \left(-\frac{\partial^2}{\partial q_i^2} + q_i^2 \right) + \sum_{j=1}^{d-1} \sum_{i>j} \alpha_{ij} q_i q_j,$$

с $\omega_j = \sqrt{j/2}$ и $\alpha_{ij} = 0.1$. TT-ранг такого осциллятора равен 3 и не зависит от d или n.

Для решения этой задачи мы сначала используем MP LOBCPG метод с рангом $r = 15$ и затем корректируем решение с помощью ALS SII итерации. Параметр δ для разбиения уровней энергии на кластеры в ALS SII равнялся 10^{-4} (E_i и E_{i+1} находятся в одном кластере, если $|E_i - E_{i+1}| < \delta \cdot |E_{i+1}|$). Размер мод $n = 15$ не зависит от номера моды. Как следует из Рисунка 2.9 ALS SII итерация значительно улучшает точность решения. LOBPCG метод запускался с 10 итерациями. Вычисления MP LOBPCG метода заняли порядка 3 часов CPU времени и коррекция решения с помощью ALS SII заняла дополнительно 30 минут.

Мы также тестируем тензорную версию предобусловленной обратной итерации (PINVIT) (1.6). Рисунок 2.10 иллюстрирует сходимость последних 10 собственных значений для различных методов. PINVIT итерация, которая также допускает явное предобуславливание, сходится к неправильным собственным значениям. LOBPCG метод без предобуславливания оказывается неустойчивым в силу округления по рангу, которое может вносить значительную

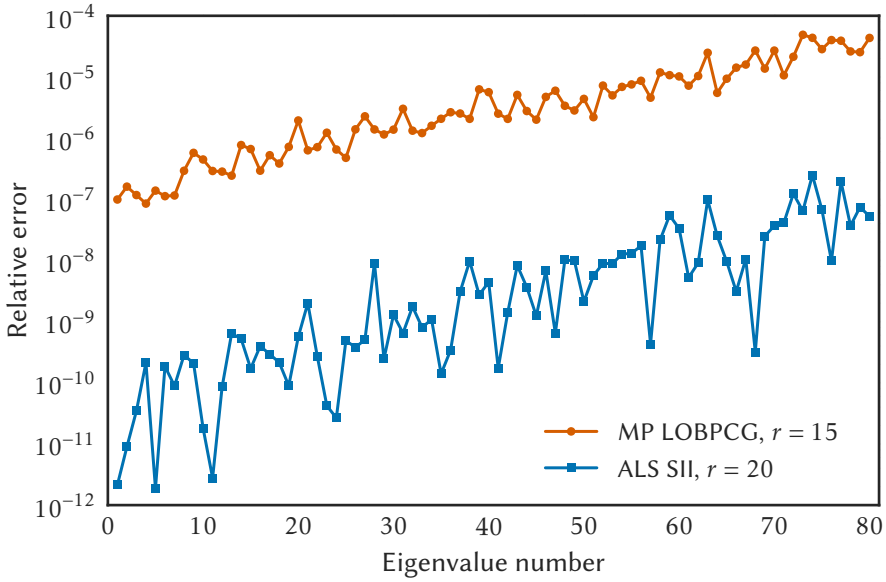


Рис. 2.9: Относительная ошибка в зависимости от номера собственного значения для 64-мерного билинейного осциллятора. ALS SII использует решение MP LOBPCG в качестве начального приближения.

ошибку. Отметим также, что все рассмотренные методы сошлись к правильным собственным значениям, когда число собственных значений было менее 30.

Молекула ацетонитрила (CH_3CN). В настоящем параграфе мы представляем вычисление колебательного спектра молекулы ацетонитрила. Используемый гамильтониан описан в [6] и может быть записан следующим образом

$$V(q_1,\dots,q_{12}) = \frac{1}{2}\sum_{i=1}^{12}\omega_i q_i^2 + \frac{1}{6}\sum_{i=1}^{12}\sum_{j=1}^{12}\sum_{k=1}^{12}\phi^{(3)}_{ijk}q_i q_j q_k + \frac{1}{24}\sum_{i=1}^{12}\sum_{j=1}^{12}\sum_{k=1}^{12}\sum_{l=1}^{12}\phi^{(4)}_{ijkl}q_i q_j q_k q_l.$$

Он состоит из 323 членов: 12 членов в кинетической энергии, 12 квадратичных членов, 108 кубических, и 191 членов четвертого порядка в потенциальной энергии. Выбрали такой же размер базиса, как и в работе [86], а именно размеры мод были равны $\{9,7,9,9,9,9,7,7,9,9,27,27\}$. Мы обнаружили, что ранги гамильтониана для этой молекулы не сильно зависят от перестановки индексов, в частности, наибольший наблюдаемый ранг гамильтониана при случайной

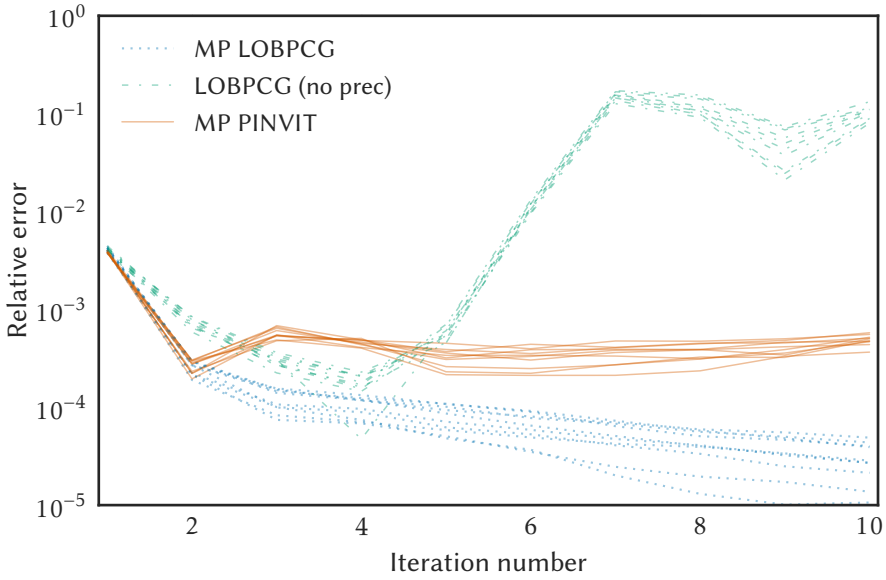


Рис. 2.10: Относительная ошибка по отношению к номеру итерации для 64-мерного билинейного осциллятора и различных итерационных методов. Для каждого метода представлена сходимость последних 10 собственных значений. MP обозначает предобуславливание на многообразии (manifold preconditioner).

перестановке индексов был равен 31, а наименьший 23. В вычислениях мы использовали такую перестановку индексов, при которой массив ω_i отсортирован по убыванию элементов. Таблица 2.1 содержит ранги частей гамильтониана в TT-формате. Отметим, что общий ранг суммы потенциалов представлен уже после округления, и, следовательно, не равен сумме рангов потенциалов.

Для сборки потенциала V требуется складывать члены ранга 1 третьего $q_i q_j q_k$ и четвертого $q_i q_j q_k q_l$ порядков. Каждый член может быть записан явно в TT-формате. Как отмечалось ранее, после каждого сложения ранг суммы увеличивается, поэтому необходимо использовать процедуру округления. Напомним, что округление требует $\mathcal{O}(dn^2r^3)$ операций. Таким образом, сложность сборки гамильтониана равна

$$\mathcal{O}\left(\left(\operatorname{nnz}\left(\phi^{(3)}_{ijk}\right) + \operatorname{nnz}\left(\phi^{(4)}_{ijkl}\right)\right) dn^2R^3\right),$$

где nnz обозначает число ненулей, n обозначает максимальный размер моды и R обозначает максимальный ранг $\mathcal{A}$. Сборка гамильтониана занимает порядка одной секунды.

Таблица 2.1: TT-ранги частей гамильтониана с точностью $\epsilon = 10^{-10}$.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}
Квадратичная	2	2	2	2	2	2	2	2	2	2	2
Кубическая	3	6	11	14	14	14	14	12	9	5	3
Четвертой степени	5	7	12	19	23	26	24	18	15	8	5
Итоговый	5	9	14	21	25	26	24	18	15	8	5

Мы запустили LOBPCG метод с TT-рангом равным 12 для каждой моды с использованием предобуславливания на многообразии. Начальное приближение выбирается из решения задачи на собственные значения с гамильтонианом, содержащим только гармоническую часть. Собственные векторы в этом случае имеют тензорный ранг 1 и являются произведением одномерных гармонических осцилляторов, а следовательно, могут быть явно записаны в TT-формате как TT-тензоры ранга 1. Сдвиг для предобуславливателя выбирался равным наименьшей энергии многомерного гармонического осциллятора. Сходимость каждого собственного значения представлена на Рисунке 2.11. Полученные собственные векторы используются в качестве начального приближения к ALS SII с рангом равным 25. Сдвиги были выбраны равным энергиям, полученным с помощью MP LOBPCG солвера. Пороговый параметр δ для разделения уровней энергии на кластеры для ALS SII был выбран равным 10^{-3}. Результаты расчетов были затем улучшены с помощью ALS SII с рангом $r = 40$. Как следует из Таблицы 2.2 и Рисунка 2.12 обе расчета являются более точными, чем H-RRBPM метод. Вычисление энергии с рангом $r = 40$ дает все уровни энергии меньше, чем в случае сеток Смоляка [6], что означает, что они вычислены более точно. Отметим, что в тоже время решение с $r = 40$ требует заметно меньше памяти, чем метод Смоляка (180 MB по сравнению с 1.5 GB).

Времена расчета и память, требуемые для вычисления спектра молекулы ацетонитрила с помощью H-RRBPM метода были взяты из [132]. На молекуле ацетонитрила был также протестирован недавно предложенный eigb [24] метод. Расчет с помощью eigb метода занял несколько дней. Проблема заключается в том, что в этом методе все собственные векторы рассматриваются в одном

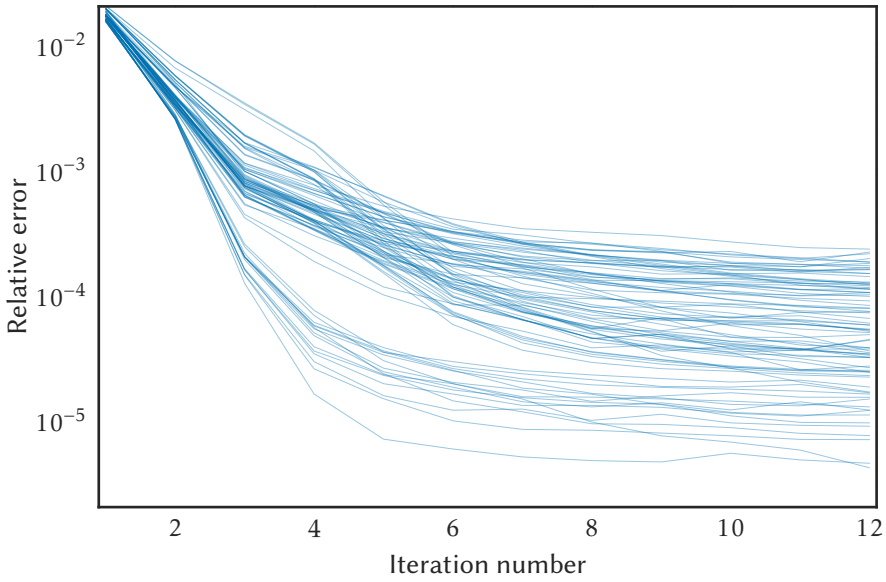


Рис. 2.11: Относительная ошибка каждого из 84 собственных значений для молекулы ацетонитрила по отношению к числу итераций для MP LOBPCG метода. Относительная ошибка вычисляется с использованием квадратур Смоляка [6] в качестве референсного значения.

базисе (блочный TT-формат), что ведет к сильному завышению рангов. Тем не менее, eigb является эффективным методом, когда требуется найти небольшое количество собственных значений.

Альтернативные подходы. Простейшим базисным набором для приближения собственных функций является прямое произведение одномерных базисных функций. В этом случае обычно доступно быстрое умножение гамильтониана на вектор и можно использовать стандартные крыловские методы решения частичной задачи на собственные значения, например, метод Ланцоша. Однако проблема заключается в том, что необходимо хранить массивы, содержащие n^d элементов. В качестве альтернативы можно обрезать (prune) прямое произведение одномерных базисных функций [5, 12, 26] или же использовать произведение базисных функций, являющихся произведением функций с более чем одной переменной [19, 7].

Мы же используем прямое произведение одномерных базисных функций и затем уменьшаем n^d с помощью приближения многомерного массива

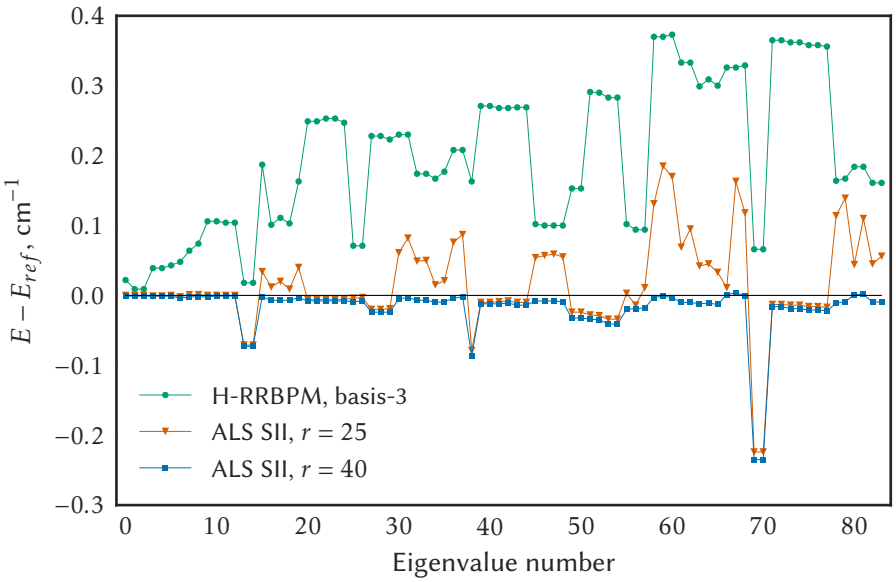


Рис. 2.12: Ошибка $E - E_{\text{ref}}$ собственных значений молекулы ацетонитрила по отношению к номеру собственного значения. Референсные энергии E_{ref} получены с помощью сеток Смоляка [6]. Отрицательные значения ошибки соответствуют более точным значениям по сравнению с сетками Смоляка. Черная прямая обозначает нулевой уровень ошибки.

коэффициентов в этом базисе в TT-формате. Отметим, что можно использовать и другие тензорные разложения, смотри обзоры [79, 44].

Каноническое разложение (также называется CP-разложение или CANDECOMP/PARAFAC) собственных векторов в задаче расчета колебательного спектра было рассмотрено в работе Лекрека и Каррингтона [86]. В этой работе авторы используют RRBPM (rank-reduced block power method) солвер. Каждая итерация этого метода включает умножение матрицы на вектор, что может быть эффективно реализовано в тензорной арифметике. Проблема заключается в том, что метод медленно сходится. Более того, известно, что задача приближения тензора в каноническом формате плохо обусловлена, что ведет к проблемам при вычислениях [145].

Иерархический RRBPM (H-RRBPM) метод был предложен в работе [132] Томасом и Каррингтоном, и значительно улучшал исходный RRBPM метод. В этом методе также используется базис, состоящий из суммы произведений, од-

нако сильно связанные координаты рассматриваются вместе, а затем иерархически разлагаются с помощью тензорных произведений.

MCTDH (Multi Configuration Time Dependent Hartree) подход [92] также использует тензорные форматы, а именно разложение Таккера. Этот подход позволяет уменьшить сложность, но все еще подвержен проклятью размерности. Проклятье размерности преодолевается в многослойном MCDTH методе (ML MCDTH) [140], который имеет сходство с иерархическим разложением Таккера [43].

Рассмотрим также тензорные алгоритмы решения задачи на собственные значения в математическом сообществе. Существуют два основных направления. Во-первых, это обобщение итерационных методов на тензорную арифметику с округлением по рангу после каждой итерации. Степенной метод со сдвигами для канонического разложения был обобщен в работах [15, 14] и использован в RRBPM методе. Предобусловленная обратная итерация (PINVIT) для тензорных форматов была рассмотрена в [90, 115, 114]. Отметим, что обратная итерация отличается от PINVIT, которая по сути является предобусловленным градиентным спуском. Тензорная версия обратной итерации, базирующаяся на итерационном решении возникающих линейных систем была рассмотрена в [137].

PINVIT итерация позволяет явно использовать предобуславливание, то есть не требует точных обращений матрицы. Построение предобуславливателей в тензорных форматах для задачи на собственные значения было рассмотрено в [137, 70, 82, 90]. Также существует подход к построению предобуславливателя для оператора общего вида [90], базирующийся на итерации Ньютона-Шульца. Однако из-за большого числа матричных умножений, этот подход является крайне вычислительно затратным. Можно также использовать предобуславливатель, построенный на приближенном решении системы линейных уравнений, например, в работах [32, 54, 17, 28] предложены различные способы решения линейных систем в тензорных форматах.

Более эффективный LOBPCG метод в тензорных форматах был рассмотрен в [85, 82]. Мы используем этот метод и строим предобуславливатель, базирующийся на оптимизационной процедуре. Сравнение PINVIT и LOBPCG

в тензорных форматах приведено в результатах численных расчетов в разделе 2.3.3.

В качестве альтернативы итерационным методам можно рассмотреть задачу оптимизации – минимизация отношения Рэлея с ограничениями на ранг. Этот подход был рассмотрен физиками в [112] и независимо в [24]. Единственным недостатком является то, что задача решается в блочном формате, что приводит к росту рангов и заметно замедляет метод при поиске большого числа собственных векторов (более 50). Тем не менее, этот подход становится эффективным и точным, если требуется найти небольшое число собственных значений.

2.4 Выводы по главе

Таким образом, в настоящей главе был предложен метод решения задач на собственные значения в тензорных форматах с линейным оператором. А именно, предложен новый метод поиска одного собственного вектора и собственного значения с помощью метода Якоби-Дэвидсона. Было отмечено, что предложенный метод позволяет избежать роста рангов, устойчив к неточному решению локальных систем, а также может быть эффективно параллелизован по ядрам разложения. Также обобщен метод обратной итерации с помощью подхода Римановой оптимизации и с помощью подхода попеременных направлений. Для последнего получены оценки локальной сходимости.

Для поиска нескольких собственных значений предложена концепция предобуславливания на многообразии, использующаяся в методе LOBPCG. Был произведен расчет колебательного спектра молекулы ацетонитрила и произведено сравнение с несколькими современными методами расчета колебательного спектра молекул. Результаты расчетов показывают, что предложенный метод дает более точное значение колебательных уровней при меньших затратах памяти.

Таблица 2.2: Энергетические уровни и значение ошибки (cm^{-1}) для молекулы ацетонитрила для предложенных методов MP LOBPCG, ALS SII и для H-RRBPM метода. Абсолютная ошибка приведена относительно решения, полученного на сетках Смоляка. Времена расчета H-RRBPM были взяты из [132]. Серый фон у чисел обозначает отрицательное значение ошибки.

		H-RRBPM [132]			**MP LOBPCG**	**ALS SII**		Reference
		Basis-1	Basis-2	Basis-3	$r = 12$	$r = 25$	$r = 40$	Smolyak [6]
		6.7 Mb	29 Mb	139 Mb	9.5 Mb	41 Mb	104 Mb	1.5 Gb
		44 сек	11 мин	3.2 ч	17 мин	+9 мин	+12 мин	
Level	Sym.	$E - E_{\text{ref}}$	$E - E_{\text{ref}}$	$E - E_{\text{ref}}$	$E - E_{\text{ref}}$	$E - E_{\text{ref}}$	$E - E_{\text{ref}}$	E	E_{ref}
ZPVE		0.485	0.118	0.022	0.056	0.001	-0.001	9837.4063	9837.4073
ν_{11}	E	0.25	0.04	0.01	0.03	0.000	-0.001	360.990	360.991
		0.25	0.07	0.01	0.04	0.000	-0.001	360.990	360.991
$2\nu_{11}$	E	0.42	0.23	0.04	0.09	-0.001	-0.001	723.180	723.181
		0.42	0.23	0.04	0.09	-0.001	-0.001	723.180	723.181
$2\nu_{11}$	A_1	0.44	0.23	0.04	0.09	0.000	-0.001	723.826	723.827
ν_4	A_1	0.59	0.16	0.05	0.06	-0.002	-0.004	900.658	900.662
ν_9	E	1.21	0.10	0.07	0.08	0.001	-0.002	1034.124	1034.126
		1.21	0.10	0.07	0.11	0.001	-0.002	1034.124	1034.126
$3\nu_{11}$	A_2	0.67	0.31	0.11	0.17	0.000	-0.002	1086.552	1086.554
$3\nu_{11}$	A_1	0.67	0.31	0.11	0.18	0.000	-0.001	1086.553	1086.554
$3\nu_{11}$	E	0.74	0.31	0.11	0.18	0.000	-0.001	1087.775	1087.776
		0.75	0.31	0.11	0.23	0.000	-0.001	1087.775	1087.776
$\nu_4 + \nu_{11}$	E	0.82	0.16	0.02	0.07	-0.071	-0.073	1259.809	1259.882
		0.82	0.26	0.02	0.15	-0.071	-0.073	1259.809	1259.882
ν_3	A_1	1.34	0.47	0.18	1.03	0.034	-0.002	1388.971	1388.973
$\nu_9 + \nu_{11}$	E	1.71	0.21	0.10	0.66	0.012	-0.007	1394.682	1394.689
		1.72	0.21	0.11	0.83	0.020	-0.007	1394.682	1394.689
$\nu_9 + \nu_{11}$	A_2	1.65	0.19	0.10	0.84	0.009	-0.007	1394.900	1394.907
$\nu_9 + \nu_{11}$	A_1	1.70	0.38	0.16	0.88	0.040	-0.003	1397.684	1397.687
$4\nu_{11}$	E	1.17	0.55	0.25	0.31	-0.006	-0.008	1451.093	1451.101
		1.17	0.55	0.25	0.40	-0.006	-0.008	1451.093	1451.101
$4\nu_{11}$	E	1.26	0.55	0.25	0.35	-0.006	-0.008	1452.819	1452.827
		1.41	0.55	0.25	0.40	-0.006	-0.008	1452.819	1452.827
$4\nu_{11}$	A_1	1.45	0.55	0.25	0.34	-0.006	-0.008	1453.395	1453.403
ν_7	E	1.30	0.10	0.07	0.23	-0.004	-0.009	1483.220	1483.229

		1.31	0.11	0.07	0.32	-0.003	-0.008	1483.221	1483.229
$\nu_4 + 2\nu_{11}$	E	1.73	0.64	0.22	0.32	-0.020	-0.024	1620.198	1620.222
		1.74	0.64	0.22	0.51	-0.020	-0.024	1620.198	1620.222
$\nu_4 + 2\nu_{11}$	A_1	1.79	0.64	0.22	0.51	-0.019	-0.024	1620.743	1620.767
$\nu_3 + \nu_{11}$	E	1.62	0.65	0.23	1.39	0.061	-0.005	1749.525	1749.53
		1.62	0.90	0.23	1.70	0.082	-0.003	1749.527	1749.53
$\nu_9 + 2\nu_{11}$	A_1	2.22	0.42	0.18	1.31	0.049	-0.007	1756.419	1756.426
$\nu_9 + 2\nu_{11}$	A_2	2.22	0.42	0.18	1.33	0.050	-0.007	1756.419	1756.426
$\nu_9 + 2\nu_{11}$	E	2.13	0.39	0.17	0.89	0.015	-0.010	1757.123	1757.133
		2.13	0.40	0.18	1.13	0.021	-0.009	1757.124	1757.133
$\nu_9 + 2\nu_{11}$	E	2.43	0.59	0.21	1.34	0.076	-0.004	1759.768	1759.772
		2.43	0.78	0.21	1.61	0.087	-0.002	1759.770	1759.772
$2\nu_4$	A_1	2.04	0.31	0.16	0.13	-0.079	-0.087	1785.120	1785.207
$5\nu_{11}$	E	1.70	1.02	0.27	0.41	-0.010	-0.012	1816.787	1816.799
		1.70	1.02	0.27	0.42	-0.010	-0.012	1816.787	1816.799
$5\nu_{11}$	A_1	1.83	1.03	0.27	0.68	-0.009	-0.012	1818.940	1818.952
$5\nu_{11}$	A_2	1.83	1.03	0.27	0.69	-0.008	-0.011	1818.941	1818.952
$5\nu_{11}$	E	1.89	1.04	0.27	0.45	-0.010	-0.014	1820.017	1820.031
		1.89	1.04	0.27	0.54	-0.010	-0.014	1820.017	1820.031
$\nu_7 + \nu_{11}$	A_2	1.68	0.19	0.10	0.90	0.054	-0.008	1844.250	1844.258
$\nu_7 + \nu_{11}$	E	1.70	0.20	0.10	0.98	0.057	-0.008	1844.322	1844.33
		1.71	0.20	0.10	1.28	0.059	-0.008	1844.322	1844.33
$\nu_7 + \nu_{11}$	A_1	1.72	0.20	0.10	1.30	0.055	-0.009	1844.681	1844.69
$\nu_4 + \nu_9$	E	3.01	0.29	0.15	0.30	-0.024	-0.033	1931.514	1931.547
		3.03	0.29	0.16	0.35	-0.024	-0.032	1931.515	1931.547
$\nu_4 + 3\nu_{11}$	A_1	2.20	1.66	0.29	0.73	-0.028	-0.034	1981.815	1981.849
$\nu_4 + 3\nu_{11}$	A_2	2.20	1.66	0.29	0.93	-0.029	-0.035	1981.815	1981.85
$\nu_4 + 3\nu_{11}$	E	2.48	1.65	0.28	0.63	-0.034	-0.041	1982.816	1982.857
		2.48	1.66	0.29	0.72	-0.034	-0.041	1982.816	1982.857
$2\nu_9$	A_1	5.58	1.59	0.10	0.93	0.003	-0.020	2057.048	2057.068
$2\nu_9$	E	4.67	1.49	0.10	0.53	-0.014	-0.019	2065.267	2065.286
		4.92	1.51	0.10	1.07	0.011	-0.018	2065.268	2065.286
$\nu_3 + 2\nu_{11}$	E	7.27	1.12	0.37	2.41	0.131	-0.003	2111.377	2111.38
		7.29	1.12	0.37	2.55	0.185	-0.001	2111.379	2111.38
$\nu_3 + 2\nu_{11}$	A_1	7.71	1.05	0.38	1.85	0.170	-0.003	2112.294	2112.297
$\nu_9 + 3\nu_{11}$	E	1.61	0.63	0.33	2.07	0.069	-0.010	2119.317	2119.327
		1.81	0.63	0.33	2.73	0.095	-0.010	2119.317	2119.327
$\nu_9 + 3\nu_{11}$	E	1.00	0.56	0.30	1.59	0.042	-0.012	2120.529	2120.541
		1.53	0.56	0.31	1.94	0.045	-0.011	2120.530	2120.541
$\nu_9 + 3\nu_{11}$	A_2	1.16	0.55	0.30	2.53	0.033	-0.013	2120.897	2120.91

$\nu_9 + 3\nu_{11}$	E	0.40	1.34	0.32	1.71	0.011	0.001	2122.835	2122.834
		0.49	1.34	0.33	2.37	0.163	0.004	2122.838	2122.834
$\nu_9 + 3\nu_{11}$	A_1	0.29	1.42	0.33	3.58	0.118	-0.001	2123.300	2123.301
$2\nu_4 + \nu_{11}$	E	2.70	1.85	0.06	-0.03	-0.224	-0.235	2142.379	2142.614
		2.70	1.99	0.06	0.97	-0.224	-0.235	2142.379	2142.614
$6\nu_{11}$	E	3.09	1.70	0.37	0.53	-0.013	-0.016	2183.619	2183.635
		3.09	1.70	0.37	0.62	-0.013	-0.016	2183.619	2183.635
$6\nu_{11}$	E	3.86	1.74	0.36	0.98	-0.014	-0.019	2186.119	2186.138
		3.86	1.74	0.36	1.09	-0.014	-0.019	2186.119	2186.138
$6\nu_{11}$	E	4.24	1.77	0.36	1.07	-0.016	-0.021	2187.621	2187.642
		4.38	1.78	0.36	1.18	-0.016	-0.021	2187.621	2187.642
$6\nu_{11}$	A_1	4.46	1.78	0.36	0.97	-0.017	-0.022	2188.122	2188.144
$\nu_7 + 2\nu_{11}$	A_1	1.98	0.45	0.17	1.65	0.114	-0.011	2206.615	2206.626
$\nu_7 + 2\nu_{11}$	A_2	1.98	0.45	0.17	2.31	0.139	-0.009	2206.624	2206.633
$\nu_7 + 2\nu_{11}$	E	1.97	0.46	0.18	2.18	0.044	0.001	2206.767	2206.766
		1.97	0.46	0.18	2.68	0.110	0.002	2206.768	2206.766
$\nu_7 + 2\nu_{11}$	E	2.01	0.45	0.16	1.93	0.045	-0.010	2207.549	2207.559
		2.03	0.46	0.16	2.18	0.056	-0.009	2207.550	2207.559

Глава 3

Задача на собственные значения для нелинейного оператора на примере уравнений Хартри-Фока и Кона-Шэма

Расчет электронной структуры атомов, молекул, кластеров молекул и твердых тел представляет важную задачу для широкого круга приложений. В настоящей главе мы предлагаем новый метод для решения уравнений Хартри-Фока (ХФ) [50] и Кона-Шэма (КШ) [78]. Эти уравнения являются задачей на собственные значения с трехмерным нелинейным интегро-дифференциальным оператором. Стандартным способом решения этих уравнений является приближение решения на подпространстве базисных функций с глобальным носителем, например, базисные функции могут быть гауссианами. Такой подход является классическим, и для него существует большое число программных пакетов. Выбор базисных функций диктуется сложностью итерационного процесса и требуемой точностью. Также использование глобального набора полуэмпирических базисных функций вводит *ошибку набора базисных функций*, контролирование которой является нетривиальной задачей. В то же время, методы, базирующиеся на последовательности вложенных пространств, позволяют контролировать ошибку аппроксимации. Среди этих подходов следует отметить представление функций с помощью вейвлетов и с помощью разделения переменных [93, 18, 36], а также конечно-элементные методы на неструктурированных сетках [51], конечно-разностный метод [21] и GPAW (projector-augmented wave) метод [91]. Стандартные конечно-элементные или конечно-разностные методы на равномерных сетках для трехмерных задач имеют слож-

ность $\mathcal{O}(n^3 \log n)$ (с использование БПФ для вычисления интегральных операторов). На неравномерных сетках БПФ не может быть использовано, поэтому сложность возрастает до $\mathcal{O}(n^6)$, где n является размером сетки по каждому направлению.

Одним из перспективных способов уменьшить сложность, предложенным и изученным в серии работ Хоромского и Хоромской [64, 130, 65, 75, 68, 66, 74, 67], является использование *тензорного разложения* массивов коэффициентов разложения по базису. Было показано, что тензор коэффициентов может быть приближен с помощью разложения Таккера. Используя это разложение, можно уменьшить сложность вычисления отдельных операций вплоть до $\mathcal{O}(n \log n)$, где n является одномерным размером сетки. Это позволяет использовать очень мелкие сетки. Однако, для того, чтобы найти решение, в этих работах используется промежуточный базисный набор глобальных базисных функций, вычисляется матрица Фока и обновляются коэффициенты в этом базисе.

Основная сложность при решении КШ/ХФ полностью в тензорном формате заключается в том, что все операции необходимо выполнять в рамках этих тензорных форматов, не формируя всех элементов возникающих массивов. Это требует разработки новых итерационных методов.

В рамках настоящей диссертации мы предлагаем эффективный солвер со сложностью $\mathcal{O}(n \log n)$, базирующийся на малоранговых тензорных разложениях, который не требует дополнительного базисного набора. Разработан новый итерационный метод, который решает исходную задачу, при этом поддерживая все возникающие промежуточные массивы в формате Таккера. Предлагаемый метод имеет несколько важных особенностей: используется блочная интегральная итерация, быстрый алгоритм вычисления свертки в малоранговых форматах, описанный в Главе 4, а также формула для вычисления матрицы Фока без производных. Более того, мы дополнительно делаем экстраполяцию на последовательности сеток, которая дает $\mathcal{O}(h^4)$ сходимость метода по сетке, где h является шагом сетки.

Глава имеет следующую структуру. В разделе 3.1 мы формулируем уравнения Хартри-Фока и Кона-Шэма. Раздел 3.2 содержит описание блочной итерации Грина и вывод формул для вычисления матрицы Фока без производных.

В разделах 3.3 и 3.4 приводится дискретизация задачи и описание операций в тензорном формате. В разделе 3.5 обсуждается сложность предложенного алгоритма. В разделе 3.6 мы приводим результаты численных расчетов для атомов с замкнутой оболочкой вплоть до аргона, для некоторых молекул и кластеров из атомов водорода.

3.1 Формулировка уравнений Хартри-Фока и Кона-Шэма

Рассмотрим систему с замкнутыми оболочками с N_e электронами и $N = N_e/2$ орбиталями, пусть также N_{nuc} обозначает число ядер с зарядами Z_α, расположенными в $\mathbf{R}_\alpha \in \mathbb{R}^3, \alpha = 1, \ldots, N_{nuc}$. Тогда уравнения ХФ/КШ может быть записано как [60]

$$H(\Phi)\phi_i \equiv \left(-\frac{1}{2}\Delta + V(\Phi)\right)\phi_i = \lambda_i \phi_i, \quad i = \overline{1,N} \tag{3.1}$$

где ϕ_i обозначает неизвестные орбитали, которые дополнительно удовлетворяют условию ортогональности

$$\int_{\mathbb{R}^3} \phi_i(\mathbf{r})\phi_j(\mathbf{r})\, d\mathbf{r} = \delta_{ij},$$

и λ_i обозначает энергии орбиталей. В уравнениях Кона-Шэма

$$V(\Phi) \equiv \tilde{V}(\rho) = V_{ext} + V_{coul}(\rho) + V_{xc}(\rho), \tag{3.2}$$

где

$$\rho(\mathbf{r}) = 2\sum_{i=1}^{N} |\phi_i(\mathbf{r})|^2,$$

является электронной плотностью. Внешний потенциал V_{ext} описывает взаимодействие между электронами и ядрами системы:

$$V_{ext}(\mathbf{r}) = -\sum_{\alpha=1}^{N_{nuc}} \frac{Z_\alpha}{|\mathbf{r} - \mathbf{R}_\alpha|}. \tag{3.3}$$

Потенциал $V_{coul}(\mathbf{r})$ задан в виде

$$V_{coul}(\mathbf{r}) = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}\, d\mathbf{r}',$$

а V_{xc} зависит только от электронной плотности и отвечает за обменную и корреляционную части оператора. Мы рассматриваем приближение локальной плотности (LDA) и функционал, предложенный Пердью и Цунгером [111]. Отметим, что предлагаемая концепция может явным образом быть обобщена на LDA функционалы, учитывающие спин (LSDA) и более точные функционалы, содержащие производные плотности, например, B3LYP. Однако для последнего, в силу наличия градиента плотности, контроль ошибки приближения будет отличаться, и мы оставляем этот вопрос для дальнейших исследований.

В случае уравнения ХФ потенциал $V(\Phi)$ имеет вид

$$V(\Phi) = V_{ext} + V_{coul}(\rho) - \hat{K}(\Phi),$$

где

$$\hat{K}(\Phi)\ \phi_i = \int_{\mathbb{R}^3} \frac{\tau(\mathbf{r},\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,\phi_i(\mathbf{r}')\,d\mathbf{r}', \quad \tau(\mathbf{r},\mathbf{r}') = \sum_{i=1}^{N} \phi_i(\mathbf{r})\phi_i(\mathbf{r}').$$

Поскольку $\hat{K}$ зависит не только от плотности $\rho(\mathbf{r})$, но и явным образом от всех орбиталей, решение уравнения ХФ является более затратным с вычислительной точки зрения по сравнению с решением уравнения КШ.

Мы обозначаем матрицы с множителями Лагранжа $\mathbf{F} = \mathbf{F}(\Phi)$

$$F_{\alpha\beta} = \int_{\mathbb{R}^3} \phi_\alpha H(\Phi)\phi_\beta\,d\mathbf{r}, \quad \alpha,\beta = \overline{1,N},$$

и называем ее матрицей Фока, в силу ее сходства с обычной матрицей Фока в фиксированном базисе: на каждой итерации мы вычисляем галеркинскую матрицу оператора Фока в текущем представлении орбиталей. Она является диагональной, когда Ф является точным решением (3.1).

Полная энергия может быть вычислена как

$$E = 2\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\iint_{\mathbb{R}^6}\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,d\mathbf{r}d\mathbf{r}' + \iint_{\mathbb{R}^6}\frac{|\tau(\mathbf{r},\mathbf{r}')|^2}{|\mathbf{r}-\mathbf{r}'|}\,d\mathbf{r}d\mathbf{r}' + E_{nn},$$

для уравнения ХФ и как

$$E = 2\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\iint_{\mathbb{R}^6}\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,d\mathbf{r}d\mathbf{r}' + E_{xc}(\rho) + E_{nn},$$

для уравнений КШ, где

$$E_{nn} = \sum_{i=1}^{N_{nuc}} \sum_{j>i}^{N_{nuc}} \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|},$$

описывает отталкивание между ядрами.

3.2 Итерационный метод

Стандартным итерационным методом решения уравнений КШ/ХФ является итерация самосогласованного поля (self-consistent field, SCF) [60]:

$$H^{(k)}\, \phi_i^{(k+1)} = \lambda_i^{(k+1)} \phi_i^{(k+1)}, \quad i = \overline{1, N}.$$

Для наших целей реализация SCF итерации является такой же сложной, как и решение исходной задачи в силу нелинейности представления Таккера. Следовательно, мы используем более подходящую для наших целей блочную итерацию Грина для уравнения в форме Липпмана-Швингера [93, 62] для (3.1), которая является более удобной для имплементации алгоритма в тензорных форматах. Отметим, что эта итерация может быть рассмотрена как предобусловленный градиентный спуск [27, 59]. В следующем подразделе приводится описание итерации и представляется итерационный процесс, который не содержит вычисления производных.

3.2.1 Блочная итерация Грина

Перепишем (3.1) в следующем виде

$$\phi_i = -2(-\Delta - 2\lambda_i)^{-1} V\phi_i.$$

Действие оператора $(-\Delta - 2\lambda_i)^{-1}$ может быть записано в виде свертки с потенциалом Юкавы

$$(-\Delta - 2\lambda_i)^{-1} V\phi(\mathbf{r}) \equiv \int_{\mathbb{R}^3} \frac{e^{-\sqrt{-2\lambda_i}\,|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}\, V\phi_i(\mathbf{r}')\, d\mathbf{r}'. \tag{3.4}$$

Мы также заметили, что прямое применение (3.4) ведет к проблемам с устойчивостью в численной реализации, что будет обсуждаться в разделе 3.3. В

этом случае более эффективным подходом является решение экранированного уравнения Пуассона с использованием конечно-разностного метода.

Для простоты мы начинаем описание итерационного процесса для системы с одной орбиталью $\phi \equiv \phi_1$. В этом случае k-й шаг итерации Грина имеет следующий вид

$$\tilde{\phi} = 2\left(-\Delta - 2\lambda^{(k)}\right)^{-1} V^{(k)}\phi^{(k)}, \quad \phi^{(k+1)} = \tilde{\phi}/\|\tilde{\phi}\|.$$

Энергия вычисляется как

$$\lambda^{(k+1)} = \left(H^{(k+1)}\phi^{(k+1)},\ \phi^{(k+1)}\right). \tag{3.5}$$

Гамильтониан $H^{(k)} \equiv H\left(\phi^{(k)}\right)$ содержит оператор Лапласа Δ. Ошибка аппроксимации в формате Таккера контролируется только в L_2 норме, поэтому дискретное дифференцирование может усилить ошибку. Для избежания этой проблемы (3.5) может быть переписано как

$$\lambda^{(k+1)} = \lambda^{(k)} + \frac{\left(V^{(k+1)}\tilde{\phi} - V^{(k)}\phi^{(k)},\ \tilde{\phi}\right)}{\left(\tilde{\phi},\ \tilde{\phi}\right)}. \tag{3.6}$$

Обобщение на случай молекул или атомов более, чем с одной орбиталью выглядит следующим образом. Мы начинаем с модификации каждой из орбиталей отдельно:

$$\hat{\phi}_i = 2\left(-\Delta - 2\lambda_i^{(k)}\right)^{-1} V^{(k)}\phi_i^{(k)}, \tag{3.7}$$

и затем ортогонализуем $\widehat{\Phi} = \left(\hat{\phi}_1, \ldots, \hat{\phi}_N\right)$ с помощью вычисления разложения Холецкого матрицы Грама

$$\int_{\mathbb{R}^3} \widehat{\Phi}^T\widehat{\Phi}\, d\mathbf{r} = LL^T, \quad \widetilde{\Phi} = \widehat{\Phi}L^{-T}. \tag{3.8}$$

Затем мы вычисляем $N \times N$ матрицу Фока

$$F = \int_{\mathbb{R}^3} \widetilde{\Phi}^T H^{(k+1)}\, \widetilde{\Phi}\, d\mathbf{r}, \tag{3.9}$$

диагонализуем F

$$\Lambda^{(k+1)} = S^{-1}FS, \tag{3.10}$$

и делаем преобразование орбиталей с помощью матрицы S:

$$\Phi^{(k+1)} = \widetilde{\Phi}S. \tag{3.11}$$

Новые значения энергии орбиталей являются диагональной частью $\Lambda^{(k+1)}$. Шаги итерационного процесса объединены в Алгоритме 3.1.

Алгоритм 3.1 Блочная итерация Грина

1: Вычислить $\widehat{\Phi} = (\hat{\phi}_1, \ldots, \hat{\phi}_N)$: $\hat{\phi}_i = 2\left(-\Delta - 2\lambda_i^{(k)}\right)^{-1} V^{(k)} \phi_i^{(k)}$.
2: Ортогонализовать $\widehat{\Phi}$: $\widetilde{\Phi} = \widehat{\Phi} L^{-T}$, где L: $LL^T = \int_{\mathbb{R}^3} \widehat{\Phi}^T \widehat{\Phi}\, d\mathbf{r}$.
3: Вычислить матрицу Фока $F = \int_{\mathbb{R}^3} \widetilde{\Phi}^T H^{(k+1)}\, \widetilde{\Phi}\, d\mathbf{r}$ с помощью формулы без производных из Утверждения 3.1.
4: Найти новые энергии с помощью диагонализации F: $\Lambda^{(k+1)} = S^{-1} F S$
5: Найти новые орбитали: $\Phi^{(k+1)} = \widetilde{\Phi} S$.

3.2.2 Вычисление матрицы Фока без производных

Для вычисления матрицы Фока (3.9) необходимо приблизить действие оператора Лапласа на орбитали. Поскольку все вычисления в тензорных форматах делаются приближенно, вычисление производных ведет к потере точности. Следовательно, мы представляем следующую формулу без производных для вычисления матрицы Фока.

Утверждение 3.1. *В Алгоритме 3.1 матрица Фока* (3.9) *может быть записана в следующем виде*

$$F = \int_{\mathbb{R}^3} \left(\widetilde{\Phi}^T V^{(k+1)} \widetilde{\Phi} - \widetilde{\Phi}^T V^{(k)} \Phi^{(k)} L^{-T} + L^{-1} \widehat{\Phi}^T \widehat{\Phi} \Lambda^{(k)} L^{-T} \right) d\mathbf{r}, \tag{3.12}$$

где $V^{(k)} \equiv V\left(\Phi^{(k)}\right)$.

Доказательство. Из (3.9) и (3.8) получаем

$$F = L^{-1} \left(\int_{\mathbb{R}^3} \widehat{\Phi}^T H^{(k+1)}\, \widehat{\Phi}\, d\mathbf{r} \right) L^{-T}. \tag{3.13}$$

Матрица

$$\int_{\mathbb{R}^3} \widehat{\Phi}^T H^{(k+1)}\, \widehat{\Phi}\, d\mathbf{r}$$

может быть записана поэлементно в следующем виде

$$\int_{\mathbb{R}^3} \hat{\phi}_\alpha H^{(k+1)} \hat{\phi}_\beta \, dx = \int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2}\Delta + V^{(k+1)} \right) \hat{\phi}_\beta \, dx =$$
$$\int_{\mathbb{R}^3} \hat{\phi}_\alpha V^{(k+1)} \hat{\phi}_\beta \, dx + \int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2}\Delta \pm \lambda_\beta^{(k)} \right) \hat{\phi}_\beta \, dx =$$
$$\int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(V^{(k+1)} + \lambda_\beta^{(k)} \right) \hat{\phi}_\beta \, dx + \int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2}\Delta - \lambda_\beta^{(k)} \right) \hat{\phi}_\beta \, dx.$$

Учитывая, что

$$\hat{\phi}_\beta = \left(-\frac{1}{2}\Delta - \lambda_\beta^{(k)} \right)^{-1} V^{(k)} \phi_\beta^{(k)},$$

имеем

$$\int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2}\Delta - \lambda_\beta^{(k)} \right) \widetilde{\phi}_\beta \, dx =$$
$$\int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2}\Delta - \lambda_\beta^{(k)} \right) \left(-\frac{1}{2}\Delta - \lambda_\beta^{(k)} \right)^{-1} V^{(k)} \phi_\beta^{(k)} \, dx = \int_{\mathbb{R}^3} \hat{\phi}_\alpha V^{(k)} \phi_\beta^{(k)} \, dx. \tag{3.14}$$

В итоге, подставляя полученное выражение в (3.13), мы получаем (3.12). □

3.2.3 DIIS ускорение сходимости

Для ускорения сходимости мы используем метод ускорения итераций DIIS (Direct Inversion in the Iterative Subspace), также известный как ускорение Андерсона. Этот метод используется для решения уравнений ТФП [60], а также встречается в других приложениях, например, для решения уравнений конвекции-диффузии [88]. Для полноты изложения приведем короткое описание схемы. Используемый итерационный процесс может быть записан в виде $\rho^{(k)} = G(\rho^{(k-1)})$. В DIIS подходе плотность на $(k+1)$-й итерации $\rho^{(k+1)}$ представляется в виде линейной комбинации плотностей, полученных на предыдущих итерациях

$$\rho^{(k+1)} = (1 - \beta_k) \sum_{j=1}^{m} \alpha_j \rho^{(k-m+j)} + \beta_k \sum_{j=1}^{m} \alpha_j G\left(\rho^{(k-m+j)} \right),$$

где коэффициенты $\alpha = (\alpha_1, \ldots, \alpha_m)$ являются решением следующей задачи минимизации

$$\alpha = \underset{\tilde{\alpha}_1, \ldots, \tilde{\alpha}_m}{\arg\min} \left\| \sum_{j=0}^{m} \tilde{\alpha}_j \left[\rho^{(k-m+j)} - G\left(\rho^{(k-m+j)} \right) \right] \right\|$$

с дополнительным ограничением

$$\sum_{j=0}^{m} \tilde{\alpha}_j = 1.$$

3.3 Дискретизация

Известно, что орбитали затухают экспоненциально на бесконечности [58]:

$$\phi_i(\mathbf{r}) = \mathcal{O}\left(e^{-\sqrt{-2\lambda_i}|\mathbf{r}|}\right), \quad |\mathbf{r}| \to +\infty. \tag{3.15}$$

В результате, умножение и линейные комбинации орбиталей также затухают экспоненциально при $|\mathbf{r}| \to +\infty$. Поэтому, мы заменяем все пространство $\mathbb{R}^3$ конечной областью $\Omega = [-L, L]^3$, где L зависит от выбранной точности и оценки значения энергии орбитали с номером λ_N.

В Ω вводится равномерная прямоугольная сетка $\omega^h = \omega_1^{h_1} \times \omega_2^{h_2} \times \omega_3^{h_3}$ с $h = 2L_i/n_i$, где $\omega_i^h = \{-L_i + kh_i : k = 0, ..., n\}$, $i = 1, 2, 3$. Мы используем равномерные сетки для получения структурированных операторов и, следовательно, для уменьшения вычислительной сложности. Низкий порядок аппроксимации на равномерных сетках будет увеличен с помощью использования экстраполяции. Для простоты используем сетки с $h_1 = h_2 = h_3$ и $L_1 = L_2 = L_3$. Алгоритм (3.7)–(3.11) требует вычисления трехмерных сверток

$$w(\mathbf{r}) \equiv \int_{\mathbb{R}^3} \frac{f(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}', \quad \mathbf{r} \in \mathbb{R}^3. \tag{3.16}$$

Замечание 3.1. *Можно использовать тот факт, что задача поиска $w(\mathbf{r})$ эквивалентна решению уравнения $-\Delta w = 4\pi f$ и кажется, что лучше решать менее вычислительно затратное уравнение Пуассона вместо вычисления свертки. Тем не менее, $w(\mathbf{r}) = \mathcal{O}(1/|\mathbf{r}|)$ при $|\mathbf{r}| \to \infty$. Граничные условия неизвестны и, следовательно, для получения точности ϵ необходимо выбирать $L = \mathcal{O}(1/\epsilon)$.*

С другой стороны, в силу экспоненциального затухания орбиталей на бесконечности, можно вычислять свертки с потенциалом Юкавы (оператор $(-\Delta - 2\lambda)^{-1}$ в (3.7)) с помощью явного обращения экранированного оператора Лапласа. Отметим, что вычисление свертки с потенциалом Юкавы приводит к

численно несимметричной матрице Фока и негативно влияет на сходимость. Дело в том, что численный аналог формулы вычисления матрицы Фока без производных (3.12) является верными только если действие $(-\Delta - 2\lambda)^{-1}$ получено с помощью решения экранированного уравнения Пуассона.

Таким образом, для дискретизации $(-\Delta - 2\lambda)^{-1}$ мы используем стандартную 7-ми точечную конечно-разностную схему. Для дискретизации сверток (3.16) мы используем галеркинскую схему и кусочно постоянные базисные функции $\chi_{\mathbf{i}}$ с носителем на $\Omega_{\mathbf{i}}$, где $\mathbf{i} \in \mathcal{I} \equiv \{0, \ldots, n-1\}^3$ и $\Omega_{\mathbf{i}}$ являются кубами размера $h \times h \times h$, отцентрированными в точках $r_{\mathbf{i}} = (x_i, y_j, z_k) \in \omega^h$. Поэтому,

$$w(\mathbf{r}_i) \approx \mathcal{W}_{\mathbf{i}} \equiv \sum_{\mathbf{j} \in \mathcal{I}} \mathcal{F}_{\mathbf{j}} \mathcal{Q}_{\mathbf{i}-\mathbf{j}}, \tag{3.17}$$

где

$$\mathcal{Q}_{\mathbf{i}-\mathbf{j}} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\chi_{\mathbf{i}}(\mathbf{r}) \chi_{\mathbf{j}}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r} d\mathbf{r}', \quad \mathcal{F}_{\mathbf{i}} = \int_{\mathbb{R}^3} f(\mathbf{r}) \chi_{\mathbf{i}}(\mathbf{r}) d\mathbf{r}. \tag{3.18}$$

Все члены при дискретизации рассматриваемых уравнений имеют ошибку аппроксимации $\mathcal{O}(h^2)$, поэтому мы ожидаем второй порядок сходимости схемы по сетке. Для получения дискретизации более высокого порядка можно использовать трансляционно инвариантные базисные функции $\chi_{\mathbf{i}}(\mathbf{r}) = \chi(\mathbf{r} - \mathbf{r}_{\mathbf{i}})$ с кусочно-полиномиальной функцией $\chi(\mathbf{r})$.

3.4 Операции в малоранговом формате

Для того чтобы реализовать итерационный алгоритм в формате Таккера, нам необходим ряд алгоритмов тензорной арифметики. Линейные операции такие, как сложение или умножение на число, как и вычисление скалярных произведений могут быть реализованы с помощью аналитических формул, смотри Главу 1. Вычисление многомерной свертки, нелинейных функций плотности и решение уравнения Пуассона описывется в настоящей секции.

3.4.1 Обменно-корреляционный функционал

Метод крестовой аппроксимации, описанный в Главе 4, является ключевым элементом приближения нелинейных функций от плотности, возникаю-

щих при вычислении обменно-корреляционного потенциала $V_{xc}(\rho)$. Идея заключается в том, что вычисляются некоторые элементы трехмерного массива плотности, заданного в виде его разложения Таккера. Затем от этих элементов вычисляется требуемая нелинейная функция. Полученные значения передаются в метод крестовой аппроксимации, на каждой итерации которого выбираются элементы, которые необходимо вычислить следующими. Отметим, что вообще говоря, дискретизованный $V_{xc}(\rho)$ может не иметь малоранговой структуры, однако $V_{xc}(\rho)\,\phi_i$ уже является малоранговым. В качестве алгоритма крестовой аппроксимации используется Schur-Cross3D из Главы 4, который имеет сложность $\mathcal{O}(r^4 + nr^2)$.

3.4.2 Многомерная свертка

Вычисление свертки является наиболее затратной частью при вычислении матрицы Фока на каждой итерации. Мы вычисляем свертку с помощью *cross-conv* алгоритма, описанного в Главе 4. Напомним, что алгоритм имеет сложность $\mathcal{O}(r^4 + nr^2)$ и является наиболее быстрым алгоритмом свертки для интересных на практике размеров мод n вплоть до $n \sim 2^{15}$. Отметим, что свертка вычисляется с массивом $\mathcal{Q}$ из (3.18), который может быть приближен в формате Таккера с точностью ϵ рангом $r = \mathcal{O}(\log n \log^{-1} \epsilon)$. Для представления тензора

$$\mathcal{Q}_{\mathbf{i}} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\chi_{\mathbf{i}}(x)\chi_{\mathbf{0}}(y)}{\|x-y\|}\, dxdy$$

в формате Таккера с линейной по n сложностью мы используем следующий прием из [94]. Для начала мы представляем функцию $1/\|x\|$ как

$$\frac{1}{\|x\|} = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-\|x\|^2 e^{2t} + t} dt. \tag{3.19}$$

Зафиксируем точность ϵ. Аппроксимируя интеграл (3.19) с помощью правила трапеций, на меньших интервалах с помощью формулы трапеций (a_ϵ, b_ϵ) получаем

$$\frac{1}{\|x\|} \approx \sum_{\alpha=1}^{K_\epsilon} \omega_\alpha e^{-x_1^2 t_\alpha} e^{-x_2^2 t_\alpha} e^{-x_3^2 t_\alpha},$$

где

$$\omega_\alpha = \begin{cases} \dfrac{\Delta_\epsilon}{\sqrt{\pi}} e^{t_\alpha}, & \alpha = 1, K_\epsilon, \\ \dfrac{2\Delta_\epsilon}{\sqrt{\pi}} e^{t_\alpha}, & \alpha = \overline{2, K_\epsilon - 1}, \end{cases}$$

и $\Delta_\epsilon = (b_\epsilon - a_\epsilon)/K_\epsilon$ есть шаг дискретизации. Таким образом,

$$\frac{1}{\|x-y\|} \approx \sum_{\alpha=1}^{K_\epsilon} \omega_\alpha e^{-(x_1-y_1)^2 t_\alpha} e^{-(x_2-y_2)^2 t_\alpha} e^{-(x_3-y_3)^2 t_\alpha}. \tag{3.20}$$

Поэтому

$$\begin{aligned} \mathcal{Q}_{ijk} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\chi_{ijk}(x)\chi_{\mathbf{0}}(y)}{\|x-y\|} dxdy \approx \\ \sum_{\alpha=1}^{K_\epsilon} \omega_\alpha \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} e^{-(x_1-y_1)^2 t_\alpha} e^{-(x_2-y_2)^2 t_\alpha} e^{-(x_3-y_3)^2 t_\alpha} \chi_{ijk}(x)\chi_{\mathbf{0}}(y) dxdy = \\ \sum_{\alpha=1}^{K_\epsilon} \omega_\alpha \int_{x_i-h/2}^{x_i+h/2} \int_{x_0-h/2}^{x_0+h/2} e^{-(x_1-y_1)^2 t_\alpha} dx_1 dy_1 \\ \int_{x_j-h/2}^{x_j+h/2} \int_{x_0-h/2}^{x_0+h/2} e^{-(x_2-y_2)^2 t_\alpha} dx_2 dy_2 \int_{x_k-h/2}^{x_k+h/2} \int_{x_0-h/2}^{x_0+h/2} e^{-(x_3-y_3)^2 t_\alpha} dx_3 dy_3. \end{aligned}$$

В результате получаем тензор $\mathcal{Q}$ в каноническом тензорном формате:

$$\mathcal{Q}_{ijk} = \sum_{\alpha=1}^{K_\epsilon} \omega_\alpha U^{(1)}_{i\alpha} U^{(2)}_{j\alpha} U^{(3)}_{k\alpha},$$

где

$$U^{(p)}_{l\alpha} = \int_{x_l-h/2}^{x_l+h/2} \int_{x_0-h/2}^{x_0+h/2} e^{-\left(x_p-y_p\right)^2 t_\alpha} dx_p dy_p, \quad p = 1,2,3, \quad l = 1,\ldots,n.$$

После округления с точностью ϵ мы получаем представление в формате Таккера со значительно меньшими рангами. Таблица 3.1 со значениями K_ϵ и интервалов (a_ϵ, b_ϵ), зависящих от требуемой точности ϵ, была любезно предоставлена Ольгой Лебедевой.

3.4.3 Уравнение Пуассона

Опишем, как решать уравнение Пуассона в формате Таккера. Как уже было отмечено, дискретный аналог формулы (3.12) дает симметричную матрицу

Таблица 3.1: Параметры (3.19) дискретизации для получения относительной точности ϵ.

ϵ	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
K_ϵ	65	80	125	145	200	220
a_ϵ	−15	−15	−20	−20	−25	−25
b_ϵ	10	10	15	15	20	20

только, если $(-\Delta + \mu I)^{-1}$ вычисляется не как свертка с потенциалом Юкавы, а как обращение оператора Лапласа со сдвигом. В противном случае матрица получается симметричной только приближенно, и это негативно сказывается на сходимости интегральной итерации.

Для быстрого обращения оператора Лапласа со сдвигом в малоранговом формате мы используем идею из [100], которая базируется на методе Фурье решения уравнения Пуассона и методе крестовой аппроксимации. Известно, что дискретизация оператора Лапласа со сдвигом $-\Delta + \mu I$ на равномерной сетке с нулевыми граничными условиями выглядит следующим образом:

$$(-\Delta + \mu I)_h = I \otimes I \otimes A_h + I \otimes A_h \otimes I + A_h \otimes I \otimes I + \mu\, I \otimes I \otimes I,$$

где матрица $A_h = 1/h^2 \operatorname{tridiag}(-1, 2, -1)$ — матрица одномерного оператора Лапласа задачи Дирихле. Известно, что $A_h = SDS^{-1}$, где S есть матрица дискретного синус преобразования (DST), на которую можно умножить вектор за $\mathcal{O}(n \log n)$ операций, а

$$D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n), \quad \lambda_k = \frac{4}{h^2} \sin^2 \frac{\pi k}{2(n+1)}.$$

В силу сказанного и свойств тензорного произведения $\otimes$ получаем, что действие оператора $(-\Delta + \mu I)_h^{-1}$ на некоторый тензор $\mathcal{F}_h$ записывается как

$$(-\Delta + \mu I)_h^{-1} \mathcal{F}_h = S \otimes S \otimes S \left(S^{-1} \otimes S^{-1} \otimes S^{-1} \left(\mathcal{F}_h ./ \Lambda \right) \right),$$

где $\Lambda_{ijk} = \lambda_i + \lambda_j + \lambda_k + \mu$ и ./ обозначает поэлементное деление. Несложно показать, что $S \otimes S \otimes S$ и $S^{-1} \otimes S^{-1} \otimes S^{-1}$ не меняют ранга тензора $\mathcal{F}_h$. Поэлементное деление на тензор Λ выполняется с помощью метода крестовой аппроксимации.

При этом, так как каждый элемент Λ может быть рассмотрен как сумма трех слагаемых, каждый из которых зависит только от одного индекса, канонический ранг Λ не превышает 3. Таким образом, обращение оператора $(-\Delta + \mu I)_h$ имеет асимптотически такую же вычислительную сложность, как и вычисление свертки. Однако в силу того, что размер тензоров не нужно увеличивать в два раза, и в силу малости ранга тензора Λ, этот шаг имеет меньшую константу в оценке сложности.

3.5 Сложность метода

Приведем оценку сложности метода в терминах n, r, где r является наибольшим рангом среди всех орбиталей. Мы также предполагаем, что вычисления выполняются на $n \times n \times n$ сетке. Предварительные вычисления до итерационного процесса включают вычисление внешнего потенциала (3.3) и галеркинского тензора сверточного ядра (3.18). Вычисление внешнего потенциала требует $N_{nuc} \cdot \mathcal{O}(r^3 + 3nr)$ операций. Вычисление галеркинского тензора — $\mathcal{O}(r^3 + 3nr)$ операций.

Оценим стоимость одной итерации. Обозначим за $K_{cross} = \mathcal{O}(r^4 + nr^2)$ сложность метода крестовой аппроксимации, вычисленной с помощью метода крестовой аппроксимации Schur-Cross3D, и за $K_{conv} = \mathcal{O}(r^4 + nr^2 + rn\log n)$ обозначим сложность вычисления свертки [114]. Вычисление $V^{(k)}\phi_i^{(k)}$ в (3.7) выполняется следующим образом. Сначала мы вычисляем V_{coul}, которое состоит из одной свертки, имеющей сложность K_{conv}. Потенциал Кулона вычисляется один раз на каждой итерации. В случае уравнения КШ мы запускаем метод крестовой аппроксимации для вычисления $V^{(k)}\phi_i^{(k)}$, так как V_{xc} не имеет малорановой структуры. Для уравнения ХФ нам дополнительно необходимо вычислить обменный потенциал, который требует N^2 вычислений свертки и N^2 поэлементных произведений. Таким образом, следующий шаг имеет сложность NK_{cross} для КШ и N^2K_{cross} для ХФ. Применение оператора $(-\Delta - \mu I)^{-1}$ также требует NK_{conv} операций, но с приблизительно в два раза меньшей константой в K_{conv}, так как один из входных тензоров имеет фиксированный ранг 3.

Шаг (3.8) требует N^2 вычислений скалярного произведения, и, следовательно, имеет сложность N^2K_{cross}, так как поэлементные произведения счи-

таются с помощью метода крестовой аппроксимации. Сложность $N^2 K_{cross}$ доминирует по сравнению с разложением Холецкого, так как мы предполагаем, что $N \ll r^4$. Вычисление матрицы Фока (3.12) требует $V^{(k)}\widetilde{\Phi}$ операций. Отметим, что $V^{(k)}\Phi^{(k)}$ имеет такую же сложность, как и скалярные произведения. Таким образом, итоговая оценка сложности каждой итерации равна $N^2 \cdot \mathcal{O}(r^4 + nr^2 + rn\log n)$ операций.

Отметим, что алгоритм может быть сделан параллельным. Одним из способов является параллелизация по орбиталям. В этом случае для вычисления матрицы Фока или для ортогонализации орбиталей необходимо собрать все орбитали на одном вычислительном узле. Однако мы ожидаем, что эта операция требует мало ресурсов, благодаря малым требованиям к памяти, необходимой для хранения обриталей. Другим способом является параллелизация по одномерному размеру сетки n. Единственной нелокальной операцией в этом случае является преобразование Фурье, для которого также существуют алгоритмы параллелизации [23].

3.6 Численный эксперимент

Программный код написан на языке Python и доступен по ссылке https://github.com/rakhuba/tensorchem. Для написания программного кода был использован пакет tucker3d, доступный по ссылке https://github.com/rakhuba/tucker3d. Этот пакет был также написан автором настоящей диссертации. Отметим, что код написан на языке Python и может быть заметно ускорен с помощью переписывания наиболее затратных частей алгоритма, например, на языках C или Fortran.

Размер области. Проиллюстрируем, как точность расчетов зависит от размера области вычислений. Напомним, что орбитали затухают экспоненциально как $\exp(-\sqrt{-2\lambda_{\text{HOMO}}}\|x\|)$, $\|x\| \to \infty$. Следовательно, можно ожидать, что ошибка вносимая конечным размером расчетной области имеет экспоненциальное затухание в зависимости от размера области. Рисунок 3.1 иллюстрирует это предположение. На этом рисунке представлена зависимость ошибки $E_h(a)$ по отношению к $E_h(\infty)$ как функции от размера области $a = 2L$. Все вычисления в этом

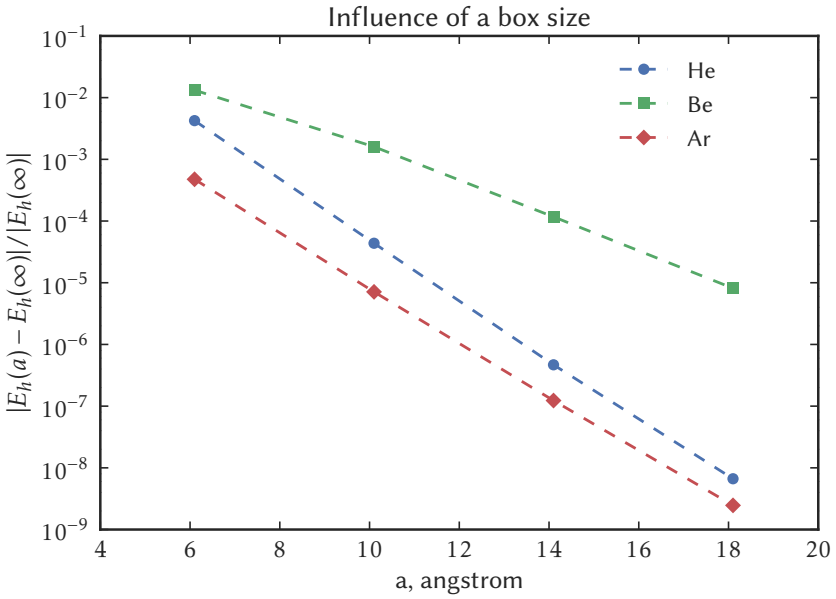


Рис. 3.1: Зависимость относительной точности $E_h(a)$ от размера области a для фиксированного $h = 0.1$ Å, $\epsilon = 10^{-9}$. Для вычисления $E_h(\infty)$ размер области выбирался равным $a = 50$ Å.

эксперименте сделаны с фиксированным шагом сетки $h = 0.1$ Å и точностью округления тензорных разложений $\epsilon = 10^{-9}$. $E_h(\infty)$ было оценено в $a = 50$ Å.

Экстраполяция. Поскольку предложенный подход полностью базируется на сеточном подходе, можно сначала рассчитать орбитали на грубой сетке, например, при $n = 128$, и затем использовать начальное приближение для расчета на более мелких сетках (в экспериментах мы использовали последовательность сеток с 2^k узлов). В силу линейной сложности предложенного алгоритма общее время расчета на последовательности сеток не более чем в 2 раза больше времени расчета на самой мелкой сетке:

$$t_{\text{extrapolated}} = t_n + t_{n/2} + \cdots + t_{n/2^l} = Cn + Cn/2 + \cdots + Cn/2^l < 2Cn = 2t_{\text{n}}, \qquad (3.21)$$

где постоянная C не зависит от n. Это делает процедуру экстраполяции эффективной частью алгоритма.

Для экстраполяции мы использовали процесс Эйткена [61], что эквивалентно экстраполяции Ричардсона для точного второго порядка. Процесс Эйткена ускоряет сходимость последовательности $\{E_m\}$ так, что новая последова-

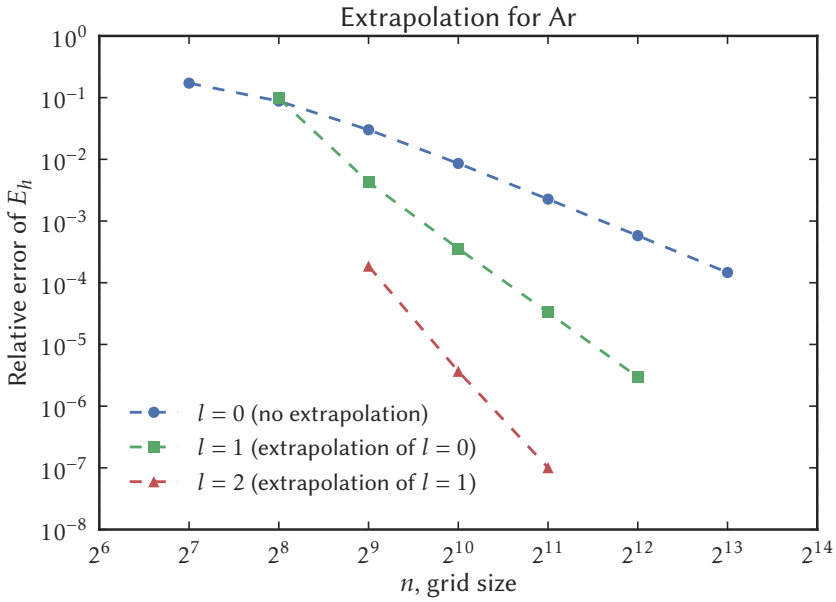


Рис. 3.2: Экстраполяция полной энергии как функции размера сетки для Аргона. Ошибка вычисляется по отношению к высокоточным результатам из [56].

тельность $\{E'_m\}$

$$E'_m = E_{m+2} - \frac{(E_{m+2} - E_{m+1})^2}{E_{m+2} - 2E_{m+1} + E_m}, \tag{3.22}$$

сходится быстрее, чем E_m, когда m стремится к бесконечности. Аналогичный трюк может быть применен к E'_m. Результаты экстраполяции представлены на Рисунке 3.2.

Поведение факторов Таккера. Рисунок 3.3 показывает поведение факторов Таккера при измельчении сетки. Он иллюстрирует, что увеличение точности происходит благодаря лучшей аппроксимации в точках расположения ядер. А именно, в этих точках находятся сильно осциллирующие узкие пики, которые медленно сходятся по отношению к размеру сетки.

Вычисление производных энергии. Отметим, что важно уметь также вычислять и другие величины помимо полной энергии, например, силы (градиент энергии). Сравнение только энергий не является единственным критерием качества дискретизации метода. Следовательно, мы также представляем результаты расчета сил для молекулы водорода в Таблице 3.2. Для вычисления

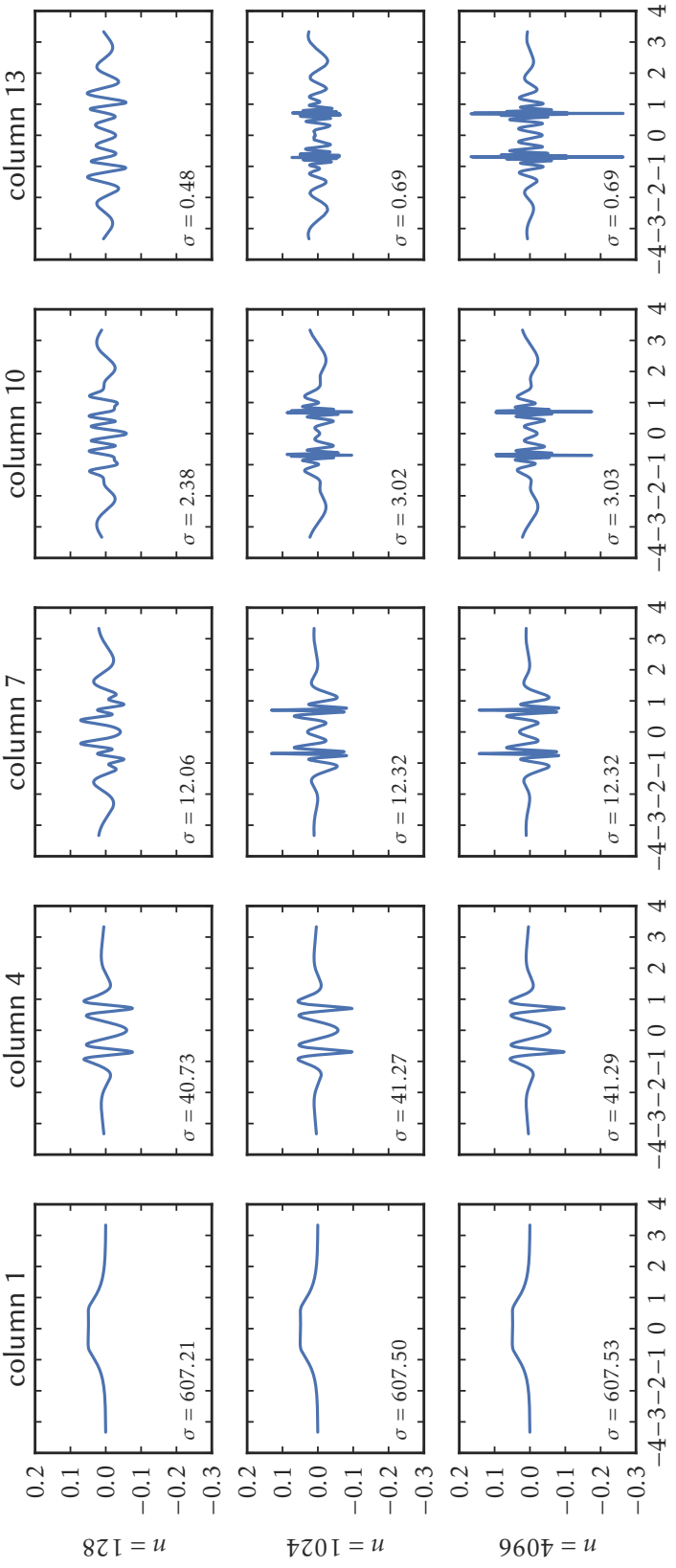


Рис. 3.3: Столбцы фактора Таккера U для плотности молекулы H_2. Размер области $a = 20$, $\epsilon = 10^{-7}$.

Метод	$N = 4096$	$N = 8192$	aug-cc-pVQZ	aug-cc-pV5Z
Норма градиента	0.00539108	0.00539092	0.005269	0.005377

Таблица 3.2: Норма градиента полной энергии в зависимости от расстояния между атомами водорода. Вычисления приведены для ХФ, $\epsilon = 10^{-9}$, размер области $a = 20$, параметр сглаживания $c = 0.01$ [95].

сил мы используем теорему Гельмана-Фейнмана и сглаженный внешний потенциал [95].

Расчет энергий атомов с помощью метода Хартри-Фока. Для удобства представления результатов введем понятие *эффективного ранга* (erank). Для Таккеровских рангов (r_1, r_2, r_3) и размера моды n эффективный ранг r определяется как действительное решение следующего полиномиального уравнения:

$$r^3 + 3nr = r_1 r_2 r_3 + n(r_1 + r_2 + r_3).$$

Как несложно заметить из определения, представление с рангами (r, r, r) дает такой же объем памяти, как и представление с рангами (r_1, r_2, r_3).

Сначала мы презентуем вычисления для уравнения ХФ для атомов с замкнутой оболочкой He, Be, Ne, Ar. Таблица 3.4 представляет полную энергию и энергию высшей заполненной молекулярной орбитали (HOMO). Результаты сравниваются с высокоточными результатами для атомов [56]. Мы запускали наш метод с относительной точностью $\epsilon = 10^{-7}$. Отметим, что тензорные представления округлялись с относительной точностью вместо абсолютной для точного вычисления HOMO энергий и в то же время для более быстрого вычисления нижних орбиталей. Экстраполяция выполнена на последовательности сеток: с 128^3 до 8192^3.

Результаты в Таблице 3.4 показывают систематическую сходимость полной и HOMO энергий. Для всех рассмотренных атомов с размером сетки вплоть до $n^3 = 8192^3$ точек, размера сетки было достаточно для получения полной энергии с $\epsilon = 10^{-7}$. Без использования экстраполяции необходимо использовать более мелкие сетки. Отметим, что по сравнению с базисным подходом, предлагаемый метод дает более точные результаты, так как с помощью

ϵ	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-11}
Энергия	-2.8615	-2.861678	-2.8616801	-2.861680000	-2.86167999593

Таблица 3.3: Полная энергия He для различных значений относительной точности округления ϵ. Энергия была получена с использованием экстраполяции на последовательности сеток: от 128^3 до 8196^3. Полная энергия в работе [56] равна -2.861 679 996.

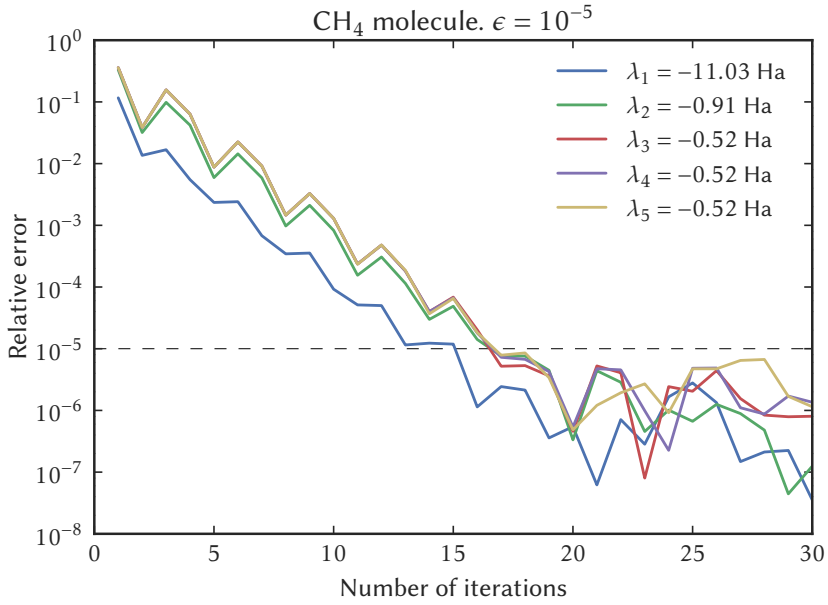


Рис. 3.4: Относительная ошибка энергии каждой орбитали в зависимости от номера итерации ($\epsilon = 10^{-5}$).

предлагаемого подхода получаются меньшие значения энергии (вариационный принцип), близкие по значению к высокоточным результатам из [56]. Однако вычисление с помощью базисного подхода для атомов требует меньшего времени расчета. Как будет видно в следующих подразделах, время расчета предлагаемого подхода становится сравнимо с базисным подходом при расчете энергии молекул, а при расчета кластеров предлагаемые подход является более эффективным и по точности, и по времени.

Расчет энергий молекул с помощью уравнения Кона-Шэма. Приведем результаты расчета молекул с помощью КШ с LDA функционалом. Геомет-

Атом	Метод	Полная эн.	HOMO эн.	erank(ρ)	Время
He	$n = 8192$	−2.861 670	−0.917 950	17	1.1 мин
	Экстраполяция	−2.861 680	−0.917 955		2.0 мин
	[56]	−2.861 680	−0.917 956		
	aug-cc-pVQZ	−2.861 521	−0.917 915		0.4 сек
	aug-cc-pV5Z	−2.861 625	−0.917 935		0.8 сек
Be	$n = 8192$	−14.572 256	−0.309 263	19	3.6 мин
	Экстраполяция	−14.573 023	−0.309 269		6.9 мин
	[56]	−14.573 023	−0.309 270		
	aug-cc-pVQZ	−14.572 976	−0.309 269		0.6 сек
	aug-cc-pV5Z	−14.573 011	−0.309 270		4.4 сек
Ne	$n = 8192$	−128.518 74	−0.850 523	20	14.5 мин
	Экстраполяция	−128.547 08	−0.850 410		30.0 мин
	[56]	−128.547 09	−0.850 410		
	aug-cc-pVQZ	−128.544 69	−0.850 210		1.9 сек
	aug-cc-pV5Z	−128.546 87	−0.850 391		8.4 сек
Ar	$n = 8192$	−526.740 5	−0.591 024	22	85 мин
	Экстраполяция	−526.817 4	−0.590 017		150 мин
	[56]	−526.817 5	−0.591 017		
	aug-cc-pVQZ	−526.816 9	−0.591 013		3.4 сек
	aug-cc-pV5Z	−526.817 3	−0.591 011		18.5 сек

Таблица 3.4: Полная энергия, HOMO энергия, ранг плотности и память, необходимая для хранения всех орбиталей атомов. Экстраполяция выполнялась на последовательности сеток: с сетки 128^3 до 8196^3. Вычисления с базисами aug-cc-pVQZ и aug-cc-pV5Z были выполнены с использованием GAMESS [37]. Серым выделены минимальные значения энергии.

рии молекул были взяты из базы данных NIST [118]. В Таблице 3.5 приведены результаты расчета полной и HOMO энергий с базисными наборами aug-cc-pVXZ (X=Q, 5). Размер области выбирался адаптивно в зависимости от HOMO энергии. Вычисления с базисными наборами были проведены с помощью программного код GAMESS [37].

Отметим, что энергия, полученная с экстраполяцией меньше, чем для базисного набора aug-cc-pV5Z примерно на 10^{-4} Хартри и для полной и для HOMO энергий. В силу вариационного принципа предложенный метод является более точным. Времена расчета сравнимы с GAMESS. Как будет видно в следующем подразделе, предлагаемый подход является одновременно более точным и быстрым в случае кластеров атомов. Рисунок 3.4 иллюстрирует быструю сходимость всех орбиталей для $\epsilon = 10^{-5}$. Как и ожидалось, итерационный процесс перестает сходится когда точность достигает ϵ. Отметим, что в остальных экспериментах использовалось $\epsilon = 10^{-7}$, и орбитали сходились до этой же точности примерно за 30 итераций.

Приведем результаты расчетов системы из атомов водорода, расчеты для которой также были приведены в работе [66] в качестве модельного примера. Система состоит из конечного набора атомов водорода, расположенных в узлах примитивной кубической структуры. Как и во всех предыдущих экспериментах мы наблюдали систематическую сходимость полной энергии до точности округления тензоров ϵ. Таблица 3.7 иллюстрирует, что ранги орбиталей растут сублинейно от размера кластера. Таким образом, мы ожидаем, что предлагаемый алгоритм является эффективным для систем с регулярным расположением атомов. Время вычисления одной итерации на сетке $n = 1024$ и при $\epsilon = 10^{-5}$ равняется 9 секундам для $H_{3\times2\times2}$ и 68 секундам для $H_{8\times2\times2}$. Время расчета ожидаемо растет квадратично с размером системы: $(8/3)^2 \approx 7.1$ и $68/9 \approx 7.5$.

Молекула	Метод	Полная эн.	HOMO эн.	erank(ρ)	Время, мин
H_2	$n = 2^{10}$, $\epsilon = 10^{-4}$	−1.137 293 4	−0.378 661	8	0.11
	$n = 2^{10}$, $\epsilon = 10^{-6}$	−1.137 350 3	−0.378 667	16	0.25
	Экстраполяция	−1.137 392 2	−0.378 668		0.50
	aug-cc-pVQZ	−1.137 249 9	−0.378 649		0.06
	aug-cc-pV5Z	−1.137 374 8	−0.378 665		0.32
CH_4	$n = 2^{12}$, $\epsilon = 10^{-5}$	−40.110 271	−0.348 990	28	12.1
	$n = 2^{12}$, $\epsilon = 10^{-7}$	−40.115 911	−0.348 989	48	29.5
	Экстраполяция	−40.119 813	−0.348 984		55.0
	aug-cc-pVQZ	−40.118 644	−0.348 964		8.5
	aug-cc-pV5Z	−40.119 299	−0.348 982		51.8
C_2H_6	$n = 2^{12}$, $\epsilon = 10^{-4}$	−79.069 926	−0.299 745	25.3	34.9
	$n = 2^{12}$, $\epsilon = 10^{-7}$	−79.069 964	−0.299 763	49.4	57.6
	Экстраполяция	−79.075 142	−0.299 761		94.1
	aug-cc-pVQZ	−79.070 784	−0.299 724		44.9
	aug-cc-pV5Z	−79.072 763	−0.299 762		

Таблица 3.5: Полная энергия, HOMO энергия, эффективные ранги плотности и время расчета. Экстраполяция выполнялась на последовательности сеток: с сетки 128^3 до указанного в таблице n^3. Вычисления с базисами aug-cc-pVQZ и aug-cc-pV5Z были выполнены с использованием GAMESS [37]. В случае aug-cc-pV5Z базиса для C_2H_6 не хватило памяти для расчета, и вычисления проводились на кластере с дополнительной параллелизацией. Серым выделены минимальные значения энергии.

Кластер	Метод	Полная эн.	erank(ρ)	Время, мин
$H_{3\times2\times2}$	$n = 2^9$, $\epsilon = 10^{-4}$	−6.585 490	12.3	0.62
	aug-cc-pVDZ	−6.577 212		0.48
$H_{8\times2\times2}$	$n = 2^9$, $\epsilon = 10^{-4}$	−17.599 497	10.6	6.0
	aug-cc-pVDZ	−17.043 676		6.2

Таблица 3.6: Полная энергия, HOMO энергия, эффективные ранги плотности и время расчета. Вычисления с базисом aug-cc-pVDZ были выполнены с использованием GAMESS [37]. Серым выделены минимальные значения энергии.

Кластер	Мин. ранги орбиталей	Макс. ранги орбиталей
$H_{1\times2\times2}$	$16\times16\times14$	$17\times17\times15$
$H_{3\times2\times2}$	$28\times28\times21$	$35\times35\times20$
$H_{8\times2\times2}$	$25\times25\times22$	$36\times36\times26$
$H_{1\times2\times1}$	$13\times12\times12$	$13\times12\times12$
$H_{2\times2\times1}$	$16\times16\times14$	$17\times17\times15$
$H_{5\times2\times1}$	$16\times18\times11$	$16\times20\times13$
$H_{9\times2\times1}$	$14\times18\times11$	$17\times22\times13$
$H_{16\times2\times1}$	$16\times19\times11$	$21\times27\times14$
$H_{1\times4\times1}$	$13\times12\times13$	$13\times13\times13$
$H_{3\times4\times1}$	$19\times20\times13$	$23\times23\times15$
$H_{8\times4\times1}$	$23\times20\times12$	$32\times27\times15$

Таблица 3.7: Ранги орбиталей с минимальными и максимальными рангами для различных кластеров атомов водорода.

Расчет энергий кластеров молекул с помощью уравнения Кона-Шэма. В Таблице 3.6 приведено сравнение с базисным подходом. Из нее следует, что предлагаемый подход является одновременно более быстрым и более точным по сравнению с базисным подходом. Это объясняется регулярным расположением атомов и, следовательно, маленькими рангами орбиталей. Так же отметим, что использовался базис aug-cc-pVDZ, так как на нем получаются сравни-

мые с предлагаемым методом времена расчета. Для остальных кластеров из Таблицы 3.7 наблюдается аналогичное поведение для энергий.

3.7 Выводы по главе

Тензорный подход к решению трехмерных задач расчета электронной структуры позволяет достигать требуемой точности решения с невысокой вычислительной сложностью. Метод превосходит по точности базисный подход, а для кластеров с регулярным расположением атомов превосходит базисный подход также по скорости вычислений (сравнение проводилось с программным комплексом GAMESS [37]). Предложенный подход также можно использовать, например, для верификации других методов и для построения высокоточных глобальных базисных наборов [131]. Эффективность метода может быть повышена путем уменьшения зависимости сложности от числа электронов системы: например, для вычисления поэлементных произведений $\phi_i \phi_j$ можно использовать локализацию орбиталей [138].

Глава 4

Вычисление многомерной свертки на основе метода крестовой аппроксимации в частотной области

Настоящая глава посвящена cross-conv алгоритму [114] вычисления многомерной свертки, которая является вычислительно наиболее трудоемкой операцией при решении уравнений КШ/ХФ из Главы 3. Отметим, что cross-conv алгоритм представляет интерес не только при решении этих уравнений. Многомерная свертка также находит применение в различных приложениях, таких как моделирование баланса популяций [20], обработка сигналов и изображений [142], а также финансовая математика [83]. Поэтому описание алгоритма ведется для задач произвольной размерности, а не только для трехмерного случая. Также алгоритм рассматривается не только для используемого в решении уравнений КШ/ХФ формата Таккера, но для широкого круга тензорных форматов.

Приведем краткое описание идеи алгоритма. В начале используется классическая идея представления дискретной свертки в виде нескольких преобразований Фурье и одного поэлементного произведение в “частотной области”. Можно показать, что преобразование Фурье может быть эффективно выполнено в тензорных форматах. Однако вычисление поэлементного произведения имеет сильную зависимость от ранга. В настоящей работе предлагается интерполировать поэлементное произведение с помощью *метода крестовой аппрок-*

симации, новая эффективная версия которого в трехмерном случае также будет описана в настоящей главе.

4.1 Известные подходы

Классический подход к вычислению дискретной свертки базируется на использовании Быстрого Преобразования Фурье (БПФ). Сложность этого подхода составляет $\mathcal{O}(n^d \log n)$ операций для сетки с n^d точками. Это намного быстрее, чем наивное суммирование со сложностью $\mathcal{O}(n^{2d})$, однако неприменимо для больших значений d и/или n.

На основе тензорных разложений несложно получить линейную по d или n сложность, однако может возникнуть сильная зависимость от ранга. Можно показать, что ранг результата (свертки) равен произведению рангов входных тензоров, после чего необходимо уменьшить ранг результата с требуемой точностью. Этот подход был рассмотрен в работах [123, 71]. Он приводит к сильной зависимости от ранга и становится неприменим уже при $r \sim 100$. В работе [63] был предложен алгоритм вычисления свертки в так называемом квантизованом TT (Quantized TT, QTT) [72, 102] формате. Этот алгоритм имеет сложность $\mathcal{O}(d \log^\alpha n)$ и асимптотически превосходит другие методы. Однако для практически интересных значений n алгоритм, предложенный в настоящей работе, является более быстрым. Это происходит из-за большой константы в оценке сложности QTT алгоритма.

4.2 Многомерная свертка и ее дискретизация

Свертка двух функций $f, g : \mathbb{R}^d \to \mathbb{R}$ определяется как многомерное интегральное преобразование

$$(f * g)(x) \equiv \int_{\mathbb{R}^d} f(y)\, g(x-y)\, dy, \quad x \in \mathbb{R}^d. \tag{4.1}$$

С помощью подходящей дискретизации на равномерной сетке (4.1) сводится к вычислению дискретной свертки

$$(f * g)(x_{\mathbf{i}}) \approx \sum_{\mathbf{j}} \mathcal{F}_{\mathbf{j}}\, \mathcal{G}_{\mathbf{i}-\mathbf{j}}, \tag{4.2}$$

где $\mathbf{i}, \mathbf{j} \in \{0, \ldots, n-1\}^d$ являются мульти-индексами. Для получения структурированных матриц использование равномерных сеток является стандартным, однако не обязательным. Могут быть также использованы неравномерные сетки [45, 46]. В настоящей главе мы рассматриваем только случай равномерных сеток, и вычисление дискретной свертки (4.2) является основной целью.

Для удобства опишем хорошо известные факты о дискретизации свертки (4.1), смотри, например [71]. Мы предполагаем, что f достаточно мала вне круга $\Omega = [-L, L]^d$, так что при интегрировании мы можем заменить $\mathbb{R}^d$ на Ω. Размер этого гиперкуба зависит от конкретной задачи, однако во многих задачах, таких как вычисление электронной структуры, функции затухают экспоненциально при $\|x\| \to \infty$. Существует три стандартных метода дискретизации свертки: галеркинский метод, метод коллокации и схема Нистрема.

Для начала введем в области Ω равномерную тензорную сетку $\omega^h = \omega_1^h \times \cdots \times \omega_d^h$ с $h = 2L/n$, где $\omega_i^h = \{-L + kh : k = 0, ..., n\}$, $i = 1, \ldots, d$. Для простоты, рассмотрим кусочно-постоянные функции $\phi_{\mathbf{i}}$ с носителем на $\Omega_{\mathbf{i}}$, где $\mathbf{i} \in \mathcal{I} \equiv \{0, \ldots, n-1\}^d$ и $\Omega_{\mathbf{i}}$ являются h^d кубами с центрами в $y_{\mathbf{i}}$. Таким образом, имеем

$$(f * g)(x) \approx \sum_{\mathbf{i} \in \mathcal{I}} \mathcal{F}_{\mathbf{i}} \int_{\Omega_{\mathbf{i}}} \phi_{\mathbf{i}}(y) g(x - y)\, dy, \tag{4.3}$$

где $\mathcal{F}_{\mathbf{i}}$ являются коэффициентами в разложении $f(y) \approx \sum_{\mathbf{i} \in \mathcal{I}} \mathcal{F}_{\mathbf{i}} \phi_{\mathbf{i}}(y)$. В результате, точки коллокации $x_{\mathbf{j}}$ дают дискретную свертку:

$$\mathcal{W}_{\mathbf{j}} \equiv (f * g)(x_{\mathbf{j}}) \approx \sum_{\mathbf{i}} \mathcal{F}_{\mathbf{i}} \mathcal{G}_{\mathbf{i}-\mathbf{j}}, \quad \mathbf{j} \in \mathcal{I}, \tag{4.4}$$

где

$$\mathcal{G}_{\mathbf{i}-\mathbf{j}} = \int_{\Omega_{\mathbf{i}}} \phi_{\mathbf{i}}(y) g(x_{\mathbf{j}} - y)\, dy \tag{4.5}$$

является многоуровневой Теплицевой матрицей. Проблема метода коллокации заключается в том, что он приводит к несимметричным Теплицевым матрицам, даже если исходная свертка была симметричной. Естественным выбором является метод галеркина, который тоже приводит к дискретной свертке с

$$\mathcal{G}_{\mathbf{i}-\mathbf{j}} = \int_{\mathbb{R}^d} \phi_{\mathbf{i}}(x) \phi_{\mathbf{j}}(y) g(x - y)\, dx dy, \quad \mathcal{F}_{\mathbf{i}} = \int_{\mathbb{R}^d} f(x) \phi_{\mathbf{i}}(x)\, dx. \tag{4.6}$$

Для получения схем высокого порядка необходимо использовать трансляционно-инвариантный базис функций более высокого порядка

$\phi_{\mathbf{i}}(y) = \psi(y - y_{\mathbf{i}})$, где $\psi(y)$ – подходящая кусочно-полиномиальная функция. Вычисление матричных элементов в (4.5) или (4.6) может оказаться трудоемким даже для кусочно-постоянных функций. Простой альтернативой является схема Нистрема при использовании сдвинутых сеток

$$(f * g)(x_{\mathbf{j}}) \approx h^d \sum_{\mathbf{i} \in \mathcal{I}} f(y_{\mathbf{i}}) g(x_{\mathbf{j}} - y_{\mathbf{i}}), \quad \mathbf{j} \in \mathcal{I}, \tag{4.7}$$

где $x_{\mathbf{j}}$ являются сдвинутыми на $h/2$ относительно $y_{\mathbf{i}}$. Для определенного класса функций эта схема дает второй порядок аппроксимации с точностью до логарифмического множителя [29].

4.3 Метод крестовой аппроксимации

Как было отмечено, cross-conv алгоритм, предлагаемый в настоящей главе, базируется на использовании метода крестовой аппроксимации. Метод крестовой аппроксимации позволяет получить малопараметрическое представление тензора (матрицы), если тензор (матрица) не помещается в память компьютера или его вычисление на основе сингулярного разложения является слишком дорогой операцией. Также этот алгоритм может быть полезен при вычислении, например, нелинейных функций от многомерных массивов, заданных в тензорном формате. В cross-conv алгоритме он используется для вычисления поэлементного произведения тензоров.

Впервые метод крестовой аппроксимации в трехмерном случае был предложен в работе [106]. Сложность полученного алгоритма составляла $\mathcal{O}(nr^3)$ при количестве используемых элементов тензора $\mathcal{O}(nr + r^3)$. В настоящем разделе приведена обновленная версия метода крестовой аппроксимации, имеющая меньшую сложность $\mathcal{O}(nr^2 + r^4)$, благодаря которой удалось предложить быстрый алгоритм вычисления многомерной свертки с более слабой зависимостью от ранга по сравнению с предыдущими подходами. Для начала приведем новую версию метода крестовой аппроксимации в двумерном случае, а затем обобщим ее на трехмерный.

4.3.1 Метод крестовой аппроксимации с дополнением по Шуру

Случай $d = 2$**.** Пусть матрица $\mathcal{X} \in \mathbb{R}^{n \times m}$ имеет ранг r с точностью ϵ. В таком случае эта матрица может быть представлена с точностью $(r+1)\epsilon$ следующим образом

$$\mathcal{X} \approx U \hat{\mathcal{X}}^{-1} V^T, \tag{4.8}$$

где $\hat{\mathcal{X}} \in \mathbb{R}^{r \times r}$ есть подматрица максимального объема, а $U \in \mathbb{R}^{n \times r}$, $V \in \mathbb{R}^{m \times r}$ – столбцы и строки, содержащие $\hat{\mathcal{X}}$ как пересечение. Разложение (4.8) называется псевдоскелетным [42]. Это разложение показывает, что матрицу, которая может быть приближена малоранговой матрицей с некоторой точностью, можно приблизить по небольшому числу ее элементов. Однако поиск подматрицы наибольшего объема является NP-сложной задачей. В работе [55] предложен алгоритм maxvol быстрого поиска подматрицы, которая является не “сильно хуже” подматрицы наибольшего объема. На основе этого алгоритма возможно построение жадного алгоритма, который на каждом шаге квазиоптимальным образом добавляет новые “хорошие” столбцы и строки в разложение (4.8). В результате получится матрица с рангом cr, где $c \gtrsim 1$. Описанная концепция порождает семейство методов, в котором можно в разной последовательности и в разном количестве вычислять столбцы матриц U и V в (4.8).

Идея предлагаемого алгоритма заключается в следующем. Во-первых, столбцы и строки добавляются в матрицы U и V симметричным образом. То есть на каждой итерации с помощью матриц U и V независимым образом выбираются “хорошие” строки и столбцы, которые еще не были выбраны. После этого эти строки и столбцы добавляются в матрицы $V^\top$ и U соответственно. Во-вторых, вместо записи $\mathcal{X} \approx U \hat{\mathcal{X}}^{-1} V^T$ мы используем эквивалентную $\mathcal{X} \approx \mathcal{U} \hat{\mathcal{X}} \mathcal{V}^T$, где $\mathcal{U} = U \hat{\mathcal{X}}^{-1}$ и $\mathcal{V} = V \hat{\mathcal{X}}^{-T}$. Для пересчета $\mathcal{U}$ и $\mathcal{V}$ при добавлении новых столбцов предложены *формулы быстрого пересчета.*

Утверждение 4.1. *Пусть* $U^{new} = [U \,|u]$, $V^{new} = [V \,|v]$, *где* u, v *– новые столбцы, которые необходимо добавить в базис. Пусть* $S_u = u - U \hat{\mathcal{X}}^{-1} \hat{u}$, $S_v = v - V \hat{\mathcal{X}}^{-1} \hat{v}$ *– соответствующие дополнения по Шуру, где символом* $\wedge$ *обозначены maxvol под-*

матрицы. Тогда

$$\mathcal{U}^{new} = \left[\mathcal{U} - S_u \hat{S_u}^{-1} \hat{\mathcal{X}} \mid S_u \hat{S_u}^{-1}\right]$$
$$\mathcal{V}^{new} = \left[\mathcal{V} - S_v \hat{S_v}^{-1} \hat{\mathcal{X}} \mid S_v \hat{S_v}^{-1}\right]$$

Выбор новых u, v осуществляется с помощью maxvol процедуры в дополнениях по Шуру S_u, S_v, которые согласно утверждению 4.1 вычисляются при обновлении $\mathcal{U}, \mathcal{V}$. Таким образом, мы выбираем каждую строку *только один раз*. Критерий остановки метода может выбираться различным образом. В нашей реализации остановка производится по ошибке интерполирования новых столбцов u, v на следующей итерации, по сравнению с тем, что дает интерполяция на предыдущей итерации. В качестве начального приближения для повышения устойчивости можно использовать U и V, являющиеся случайными матрицами.

Случай $d = 3$**.** Трехмерный случай базируется на идее описанного двумерного случая. Рассмотрим его более подробно. На каждом шаге метода необходимо найти r_0 "хороших" распорок по каждой из мод трехмерного массива. Отметим, что r_0 является параметром алгоритма и может влиять на сходимость процесса. В экспериментах мы выбирали $r_0 \sim 1 - 4$.

Предположим, что после $K-1$ й итерации нам дан подтензор: $\hat{\mathcal{X}}^{(K-1)}$ размера $(K-1)r_0 \times (K-1)r_0 \times (K-1)r_0$ и $U_i^{(K)}$, $i = 1, 2, 3$ размеров $n \times Kr_0$, где K является номером итерации. Для того, чтобы найти $\hat{\mathcal{X}}^{(K)}$ нам нужно найти подматрицы максимального объема в матрицах $U_i^{(K)}$. Однако нет гарантии, что maxvol подматрица в $U_i^{(K)}$ содержит хоть одну строку maxvol подматрицы в $U_i^{(K-1)}$. Это ведет к тому, что необходимо пересчитывать подтензор $\hat{\mathcal{X}}^{(K-1)}$ и к тому, что нужно добавлять Kr_0 новых столбцов в $U_i^{(K)}$ вместо r_0. Для избежания этого можно найти r_0 наиболее линейно независимых к $U_i^{(K-1)}$ строк. Мы предлагаем делать это с помощью дополнения по Шуру по аналогии с двумерным случаем. Таким образом, мы находим

$$i_1 = \mathrm{maxvol}(S_1), \quad i_2 = \mathrm{maxvol}(S_2), \quad i_3 = \mathrm{maxvol}(S_3)$$

являющихся индексами новых строк и затем вычисляем

$$\hat{\mathcal{X}}^{(K)} \equiv \mathcal{X}(\mathcal{I}_1^{(K)}, \mathcal{I}_2^{(K)}, \mathcal{I}_3^{(K)}),$$

где

$$\mathcal{I}_1^{(K)} = \mathcal{I}_1^{(K-1)} \cup i_1, \quad \mathcal{I}_2^{(K)} = \mathcal{I}_2^{(K-1)} \cup i_2, \quad \mathcal{I}_3^{(K)} = \mathcal{I}_3^{(K-1)} \cup i_3.$$

Следующим шагом является поиск “хороших” распорок $\hat{\mathcal{X}}^{(K)}$ по каждой моде для добавления их в матрицы $U_i^{(K)}$. Для этого мы вычисляем развертки $X_{(1)}$, $X_{(2)}$, $X_{(3)}$ и находим maxvol строки в дополнении по Шуру разверток.

В итоге мы добавляем maxvol распорок из соответствующих разверток в $U_i^{(K)}$, $i = 1, 2, 3$ и вычисляем

$$U_i^{(K+1)} \left(\hat{U}_i^{(K+1)} \right)^{-1}$$

с помощью Алгоритма 4.1. Этот алгоритм позволяет находить “хорошие” строки, не меняя “хорошие” строки с предыдущих итераций. Предполагая, что $r_0 \ll r \ll n$, общая сложность Алгоритма 4.1 равна $(2r_0 + 1)nr$. Назовем описанный алгоритм Schur-Cross3D (Алгоритм 4.2).

Алгоритм 4.1 Обновление факторов с помощью дополнения по Шуру

Require: $U \in \mathbb{C}^{n \times r}$, $u \in \mathbb{C}^{n \times r_0}$, $r_0 \leqslant r$ и $\mathcal{U} = U\,[U(\mathcal{I}, :)]^{-1}$, где $\mathcal{I}$ является мультииндексом длинной r

Ensure: $\mathcal{U}^{new} = U^{\mathrm{new}}\,[U^{\mathrm{new}}(\mathcal{I}^{new}, :)]^{-1}$, где $U^{\mathrm{new}} = [U\,|u]$, $\mathcal{I}^{new} = \mathcal{I} \cup i_0$ и i_0 является мульиндексом длины r_0

1: $S = u - \mathcal{U}u(\mathcal{I}, :)$ — $nr_0 + nr_0 r$

2: $i_0 = \mathrm{maxvol}(S)$ — nr_0^2

3: $U_2 = S\,[S(i_0, :)]^{-1}$ — $nr_0^2 + r_0^3$

4: $U_1 = \mathcal{U} - U_2 \mathcal{U}(i_0, :)$ — $nr + r_0 rn$

5: $\mathcal{U}^{new} = [U_1\,|U_2]$

Алгоритм 4.2 Schur-Cross3D

Require: Функция $\mathcal{X}(i,j,k)$, вычисляющая определенный элемент тензора $\mathcal{X}$, точность ϵ и r_0 – число распорок, которые необходимо добавлять на каждой итерации

Ensure: Разложение Таккера $\mathcal{X}$: $\mathcal{X} \approx \hat{\mathcal{X}} \times_1 \mathcal{U}_1 \times_2 \mathcal{U}_2 \times_3 \mathcal{U}_3 + \mathcal{E}$, $\|\mathcal{E}\| \leqslant \epsilon$

1: Выбрать индексы $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$ из начального приближения или случайным образом

2: **while** ошибка $> \epsilon$ **do**

3: Обновить $\hat{\mathcal{X}} = \mathcal{X}(\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3)$

4: Вычислить $r \times r^2$ развертки $\hat{\mathcal{X}}$: $\hat{X}_{(1)}, \hat{X}_{(2)}, \hat{X}_{(3)}$

5: Используя дополнение по Шуру найти номера новых распорок u_1, u_2, u_3 в развертках $\hat{X}_{(1)}, \hat{X}_{(2)}, \hat{X}_{(3)}$ $3r_0r^3$

6: Ошибка может быть оценена как норма разности между u_1, u_2, u_3 и их приближением, найденным через $\hat{\mathcal{X}}, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3$ на текущей итерации $3r_0(nr + r^3)$

7: Добавить u_1, u_2, u_3 в $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3$ и найти i_0^1, i_0^2, i_0^3 – номера новых “хороших” строк используя Алгоритм 4.1 $3(2r_0 + 1)nr$

8: $\mathcal{I}_1 := \mathcal{I}_1 \cup i_0^1$, $\mathcal{I}_2 := \mathcal{I}_2 \cup i_0^2$, $\mathcal{I}_3 := \mathcal{I}_3 \cup i_0^3$

9: $r := r + r_0$

10: **end while**

Таким образом, общая сложность Schur-Cross3D равна

$$3(3r_0 + 1)n \sum_{k=1}^{r} k + 6r_0 \sum_{k=1}^{r} k^3 = \mathcal{O}(nr^2 + r^4)$$

Отметим, что алгоритм дополнительно требует $nr + r^3$ вычисления функции. Код алгоритма можно найти на https://github.com/rakhuba/tucker3d (функция multifun).

4.3.2 Известные теоретические оценки

Отметим, что метод крестовой аппроксимации не использует всех элементов матрицы (или тензора), поэтому можно придумать контрпример когда алгоритм не находит приближения с требуемоей точностью для любой страте-

гии выбора элемента. Например, если алгоритм завершил работу, можно выбрать любой элемент массива, который не был использован в методе крестовой аппроксимации, и заменить его на другой сколь угодно большой. Такое возмущение имеет ранг 1. Однако в практически интересных случаях проблем с методом крестовой аппроксимации не наблюдается. Это означает, что массивы, встречающиеся на практике принадлежат некоторому “хорошему” подклассу. Конструктивное описание такого класса является нерешенной полностью задачей. Однако существуют некоторые важные результаты, которые необходимо упомянуть.

Если матрица (тензор) точно имеет малый ранг, тогда скелетное разложение является точным, и если во время сэмплирования мы не столкнулись с нулевыми строчкой или столбцом, процедура гарантированно сойдется. В случае, если матрица является малоранговой только приближенно, ошибка умножается на некоторую константу. Принцип максимального объема [41] утверждает, что если выбранные строки и столбцы содержат матрицу *максимального объема*, то ошибка может быть оценена как

$$\|\mathcal{X} - \mathcal{X}_{\text{skel}}\|_C \leq (r+1)\sigma_{r+1}.$$

Этот результат был обобщен на трехмерный и многомерный случаи в [106, 39, 121, 147]. На практике используются жадные алгоритмы. Для метода крестовой аппроксимации матриц, являющихся дискретизацией функций на стеках, сходимость метода была получена в работах [10, 135]. Недавно был получен важный результат в [22] для класса матриц вида

$$A = U\Phi V^\top,$$

где $U \in \mathbb{R}^{n\times r}$, $V \in \mathbb{R}^{m\times r}$ являются матрицами с ортонормированными столбцами, Φ – $r \times r$ матрица, а U и V являются μ-когерентными (то есть, $\max_{ij}|U_{ij}| \leq \mu/\sqrt{n}$), тогда достаточно насчитать $l = \mathcal{O}(r \log n)$ столбцов для получения малой ошибки с большой вероятностью. Эти результаты могут быть обобщены на другие форматы, базирующиеся на сингулярном разложени (Таккер, TT и HT форматы), так как каждый из этих форматов может быть рассмотрен как последовательность сингулярных разложений для некоторых вспомогательных матриц.

4.4 Cross-conv алгоритм

4.4.1 Описание алгоритма

Рассмотрим d-мерную дискретную свертку двух тензоров $f_{\mathbf{i}}$ и $g_{\mathbf{j}}$

$$\mathcal{W}_{\mathbf{j}} = \sum_{\mathbf{i} \in \mathcal{I}} \mathcal{F}_{\mathbf{i}} \mathcal{G}_{\mathbf{i}-\mathbf{j}}, \quad \mathbf{j} \in \mathcal{I}. \tag{4.9}$$

Это выражение можно рассмотреть в смысле умножения вектора на *многоуровневую Теплицеву матрицу* с элементами $\mathcal{G}_{\mathbf{i}-\mathbf{j}}$ (свойств многоуровневых Теплицевых матриц смотри в [134]). Вычисление (4.9) как прямой суммы требует $\mathcal{O}(n^{2d})$ операций. С помощью БПФ можно уменьшить эту сложность до $\mathcal{O}(n^d \log n)$. Классический БПФ алгоритм будет являться отправной точкой для эффективного малорангового алгоритма свертки.

Идея метода БПФ заключается в замене умножения Теплицевой матрицы на вектор на умножение на вектор *циркулянта* большего размера. Например, одноуровневая $n \times n$ Теплицева матрица $\{\mathcal{G}_{i-j}\}_{i,j=0}^{n-1}$ может быть вложена в $(2n-1) \times (2n-1)$ циркулянт, который полностью определяется своим первым столбцом

$$\mathcal{G}_c \equiv \{\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_{n-1}, \mathcal{G}_{1-n}, \mathcal{G}_{2-n}, \ldots, \mathcal{G}_{-1}\}.$$

В d-мерном случае “первый столбец” многоуровнего циркулянта определяется как тензор $\mathcal{G}_c$:

$$\mathcal{G}_c(i_1, \ldots, i_d) = \mathcal{G}_{\tau(i_1), \ldots, \tau(i_d)}, \quad i_1, \ldots, i_d \in \overline{0,\, 2n-2},$$

где

$$\tau(i) = \begin{cases} i, & i \in \overline{0,\, n-1}, \\ i - 2n + 1, & i \in \overline{n,\, 2n-2}. \end{cases}$$

Для начала мы вкладываем $\mathcal{F}$ в тензор большего размера $\mathcal{F}_q$ с размерами мод $(2n_1 - 1, \ldots, 2n_d - 1)$, заполняя оставшуюся часть нулями:

$$\mathcal{F}_q(i_1, \ldots, i_d) = \begin{cases} \mathcal{F}_{i_1, \ldots, i_d}, & i_1, \ldots, i_d \in \overline{0,\, n-1}, \\ 0, & \text{иначе.} \end{cases}$$

Согласно дискретной теореме о свертке, многоуровневые циркулянты диагонализуются с помощью тензорного произведения одномерных матриц Фурье, а

собственные значения могут быть вычислены как дискретное преобразование Фурье первых столбцов. Так как умножение на диагональную матрицу может быть записано в виде поэлементного произведения, получаем

$$\widetilde{\mathcal{W}} = \mathsf{F}^{-1}\Big(\mathsf{F}(\mathcal{G}_c) \circ \mathsf{F}(\mathcal{F}_q)\Big), \tag{4.10}$$

где $\widetilde{\mathcal{W}}$ является расширенным тензором свертки с размером $(2n-1)$ по каждой моде. В этом тензоре нас интересует только его подтензор $\mathcal{W}$:

$$\mathcal{W}(i_1, \ldots, i_d) = \widetilde{\mathcal{W}}(i_1, \ldots, i_d), \quad i_1, \ldots, i_d \in \overline{0, n-1}.$$

Оператор F обозначает многомерное преобразование Фурье:

$$\mathsf{F}(\mathcal{X})(i_1, \ldots, i_d) = \sum_{j_1, \ldots, j_d = 0}^{n-1} e^{-2\pi i \left[\frac{i_1 j_1}{n} + \ldots + \frac{i_d j_d}{n}\right]} \mathcal{X}(j_1, \ldots, j_d).$$

Рассмотрим вопрос о том, как использовать эту формулы, если ее элементы заданы в некотором малоранговом формате. Для простоты мы рассмотрим случай, когда $\mathcal{G}_c$ и $\mathcal{F}_q$ заданы в ТТ-формате.

$$\begin{aligned} \mathcal{G}_c(i_1, \ldots, i_d) &= G_1^{(\mathcal{G}_c)}(i_1) \ldots G_d^{(\mathcal{G}_c)}(i_d), \\ \mathcal{F}_q(i_1, \ldots, i_d) &= G_1^{(\mathcal{F}_q)}(i_1) \ldots G_d^{(\mathcal{F}_q)}(i_d), \end{aligned} \tag{4.11}$$

однако описанная идея будет применима также и к остальным форматам, основанным на сингулярном разложении, а именно к формату Таккера, HT формату и скелетному разложению.

Преобразование Фурье тензора $\mathcal{X}$ размера $n \times \cdots \times n$ в ТТ-формате может быть записано как

$$\begin{aligned} \mathsf{F}(\mathcal{X})(i_1, \ldots, i_d) &= \sum_{j_1, \ldots, j_d} e^{-2\pi i \left[\frac{i_1 j_1}{n} + \ldots + \frac{i_d j_d}{n}\right]} G_1^{(\mathcal{X})}(j_1) \ldots G_d^{(\mathcal{X})}(j_d) = \\ &= \sum_{j_1} e^{-2\pi i \frac{i_1 j_1}{n}} G_1^{(\mathcal{X})}(j_1) \ldots \sum_{j_d} e^{-2\pi i \frac{i_d j_d}{n}} G_d^{(\mathcal{X})}(j_d) = \\ &= \mathsf{F}_{1D}\left(G_1^{(\mathcal{X})}\right)(i_1) \ldots \mathsf{F}_{1D}\left(G_d^{(\mathcal{X})}\right)(i_d), \end{aligned} \tag{4.12}$$

где в качестве F_{1D} мы обозначили одномерное преобразование Фурье. Из (4.12) следует, что преобразование Фурье не меняет ранга тензора, на которое оно действует.

Приведем пошаговое описание алгоритма.

Шаг 1. Вычислить тензоры $\mathsf{F}(\mathcal{G}_c)$ и $\mathsf{F}(\mathcal{F}_q)$ в рассматриваемом формате. Как следует из (4.12), преобразование Фурье тензора не меняет его рангов и эквивалентно применению одномерных преобразований Фурье каждого фактора в разложении Таккера или каждого ядра в ТТ случае. Следовательно, в ТТ формате

$$\begin{aligned}\mathsf{F}(\mathcal{G}_c)(i_1,\ldots,i_d) &= \mathsf{F}_{1D}\left(G_1^{(\mathcal{G}_c)}\right)(i_1)\ldots\mathsf{F}_{1D}\left(G_d^{(\mathcal{G}_c)}\right)(i_d),\\ \mathsf{F}(\mathcal{F}_q)(i_1,\ldots,i_d) &= \mathsf{F}_{1D}\left(G_1^{(\mathcal{F}_q)}\right)(i_1)\ldots\mathsf{F}_{1D}\left(G_d^{(\mathcal{F}_q)}\right)(i_d).\end{aligned} \tag{4.13}$$

Шаг 2. На этом шаге мы вычисляем $\Theta = \mathsf{F}(\mathcal{G}_c) \circ \mathsf{F}(\mathcal{F}_q)$ и этот шаг является ключевым в нашем алгоритме. Наивный подход заключается в прямом вычислении r^2 одномерных сверток. Округление необходимо практически всегда и в случае ТТ-формата ведет к алгоритму со сложностью $\mathcal{O}(dnR^3)$, где $R = r^2$. Такой алгоритм может быть применим к системам только с рангами $r_k \sim 100$. Существуют более сложные алгоритмы, которые базируются на итерационных схемах, например, для формата Таккера [40, 124] и для ТТ-формата [103, 30]. Мы предлагаем вычислять поэлементное произведение следующим образом. Заметим, что вычисление любого заранее заданного элемента произведения является дешевой операцией. Эта ситуация идеально подходит для использования *метода крестовой аппроксимации*. Мы приведем сравнение требуемого числа операций для каждого из форматов в следующем разделе. Итак, мы вычисляем необходимые элементы тензоров $\mathsf{F}(\mathcal{G}_c)$ и $\mathsf{F}(\mathcal{F}_q)$, перемножаем их и строим тензор Θ в рассматриваемом формате согласно выбранным с помощью метода крестовой аппроксимации элементам. Это единственный шаг, где делается аппроксимация. Предположим, что ошибка аппроксимации равна δ, тогда

$$\Theta = \widetilde{\Theta} + \Delta\Theta,$$

где

$$\widetilde{\Theta}(i_1,\ldots,i_d) = G_1^{(\Theta)}(i_1)\ldots G_d^{(\Theta)}(i_d),$$

является аппроксимацией Θ вычисленная с помощью метода крестовой аппроксимации с относительной точностью
$\|\Delta\Theta\|/\|\Theta\| = \delta$.

Шаг 3. Вычислить F_{1D}^{-1} каждого ядра $\widetilde{\Theta}$. Следовательно, аппроксимация $\widetilde{\mathcal{W}}$ имеет вид

$$\widetilde{\mathcal{W}} = \mathsf{F}^{-1}(\Theta)(i_1, \ldots, i_d) = \mathsf{F}_{1D}^{-1}\left(G_1^{(\Theta)}\right)(i_1) \ldots \mathsf{F}_{1D}^{-1}\left(G_d^{(\Theta)}\right)(i_d) + \mathsf{F}^{-1}(\Delta\Theta).$$

Несложно оценить пороговое значение δ необходимое для поддержания заданной точности всей свертки. Предположим, что $\epsilon = \|\Delta\widetilde{\mathcal{W}}\|/\|\widetilde{\mathcal{W}}\|$ – это необходимая точность, где $\Delta\widetilde{\mathcal{W}} = \mathsf{F}^{-1}(\Delta\Theta)$. В силу унитарной инвариантности Фробениусовой нормы, имеем

$$\epsilon = \frac{\|\Delta\widetilde{\mathcal{W}}\|}{\|\widetilde{\mathcal{W}}\|} = \frac{\|\mathsf{F}^{-1}(\Delta\Theta)\|}{\|\mathsf{F}^{-1}(\Theta)\|} = \frac{\|\Delta\Theta\|}{\|\Theta\|} = \delta.$$

Таким образом, чтобы поддерживать заданную точность ϵ необходимо использовать метод крестовой аппроксимации с такой же точностью $\delta(\epsilon) = \epsilon$.

4.4.2 Сложность алгоритма в различных форматах

Оценим сложность описанного алгоритма в различных форматах. Для простоты в оценках сложности мы предполагаем $\mathcal{G}_c$, $\mathcal{F}_q$ и $\widetilde{\mathcal{W}}$ имеют $n_k = n$ и $r_k = r$. Наше дополнительное предположение в оценке сложности заключается в том, что свертка может быть приближена рангом $R_k \ll r^2$. Это предположение необходимо проверять в каждом отдельном случае, однако это стандартно для такого типа алгоритмов.

Скелетное разложение. Сначала рассмотрим двумерный случай. В двумерном случае единственным способом разделить переменные является приближение матрицы $\mathcal{X} \in \mathbb{C}^{n\times m}$ с помощью скелетного разложения:

$$\mathcal{X} \approx UV^T,$$

где $U \in \mathbb{C}^{n\times r}$, $V \in \mathbb{C}^{m\times r}$ и r является рангом матрицы $\mathcal{X}$ с заданной точностью. Крестовый метод вычисления скелетного разложения требует вычисления r столбцов и r строк. Вычисление одного столбца или одной строки матрицы, заданной ее скелетным разложением, стоит $\mathcal{O}(nr)$ операций. В самом деле, рассмотрим вычисление j-го столбца:

$$\mathcal{X}(:, j) = UV(j, :)^T.$$

Вычисление произведения $UV(j,:)^T$ требует nr операций. В результате, вычисление r строк и столбцов матрицы $\Theta = \mathsf{F}(\mathcal{G}_c) \circ \mathsf{F}(\mathcal{F}_q)$ из шага 2 требует $\mathcal{O}(nr^2)$ операций. Дополнительные операции, связанные с расходами метода крестовой аппроксимации также имеют сложность $\mathcal{O}(nr^2)$ [135, 10]. Отметим, что БПФ операции из шагов 1 и 3 требуют дополнительно $\mathcal{O}(rn\log n)$ операций. Таким образом, итоговая сложность алгоритма в двумерном случае равна $\mathcal{O}(nr^2 + rn\log n)$.

Формат Таккера. Формат Таккера содержит экспоненциальное по d число параметров $\mathcal{O}(r^d + nrd)$, но он может быть эффективен для задач с небольшим значением d, особенно для случая $d = 3$. Оценим сложность алгоритма трехмерной свертки в формате Таккера. В работе [106] был впервые предложен метод крестовой аппроксимации для формата Таккера (смотри другие подходы в [74, 11]).

Напомним, что в методе крестовой аппроксимации требуется вычисление r распорок по каждому направлению. Определим сложность вычисления распорок. Пусть $\mathcal{X}$ это трехмерный тензор, заданный в формате Таккера

$$\mathcal{X}(i_1, i_2, i_3) = \sum_{\alpha_1,\alpha_2,\alpha_3} G^{(\mathcal{X})}(\alpha_1, \alpha_2, \alpha_3)\, U_1^{(\mathcal{X})}(i_1, \alpha_1)\, U_2^{(\mathcal{X})}(i_2, \alpha_2)\, U_3^{(\mathcal{X})}(i_3, \alpha_3),$$

Одна распорка определяется двумя индексами. Для определенности зафиксируем индексы i_2, i_3. Нам необходимо посчитать результат для всех $i_1 = 0, \ldots, n_1 - 1$. Для этого мы сначала вычисляем

$$B_{i_2 i_3}(\alpha_1) = \sum_{\alpha_2,\alpha_3} G^{(\mathcal{X})}(\alpha_1, \alpha_2, \alpha_3)\, U_2^{(\mathcal{X})}(i_2, \alpha_2)\, U_3^{(\mathcal{X})}(i_3, \alpha_3),$$

что требует $\mathcal{O}(r^3)$ операций. Отсюда

$$\mathcal{X}(:, i_2, i_3) = \sum_{\alpha_1} U_1^{(\mathcal{X})}(:, \alpha_1) B_{i_2 i_3}(\alpha_1),$$

на что требуется $\mathcal{O}(nr)$ операций. Таким образом, вычисление одной распорки $\Theta = \mathsf{F}(\mathcal{G}_c) \circ \mathsf{F}(\mathcal{F}_q)$, если $\mathcal{G}_c$ и $\mathcal{F}_q$ стоит $\mathcal{O}(nr + r^3)$. Так как метод крестовой аппроксимации использует r распорок в каждом из направлений, сложность поэлементного произведения будет $\mathcal{O}(nr^2 + r^4)$. Как и в двумерном случае для

тензора $\mathcal{X}$, в формате Таккера БПФ $\mathcal{F}(\mathcal{X})$ не меняет таккеровских рангов и эквивалентно трем одномерным преобразованиям Фурье от каждого из факторов. Таким образом, сложность шагов 1 и 3 будет $\mathcal{O}(nr\log n)$. Итоговая сложность вычисления свертки в формате Таккера – $\mathcal{O}(nr^2 + rn\log n + r^4)$.

TT формат. Для больших размерностей формат Таккера становится неприменим на практике и поэтому необходимо использовать TT или HT форматы, которые имеют линейные по d размеры. Метод крестовой аппроксимации был предложен изначально в работе [109] и позднее значительно улучшен в [122] и [120]. Можно показать, что асимптотическая сложность этих алгоритмов в нашем случае будет $\mathcal{O}(dnr^3)$. Алгоритм состоит из d матрично-матричных умножений матриц $r \times r$ на матрицы $r \times nr$. БПФ стоимость для TT формата – $\mathcal{O}(r^2 n\log n)$. Итак, итоговая сложность алгоритма $\mathcal{O}(dnr^3 + r^2 n\log n)$, что позволяет использовать его в случае больших n и d.

HT и расширенный TT форматы. Если n очень велико, то дополнительное снижение сложности может быть достигнуто за счет использования HT или расширенного TT форматов. Вариант метода крестовой аппроксимации для HT-формата можно найти в [8]. Можно показать, что в этом случае сложность алгоритма будет $\mathcal{O}(dnr^2 + dr^4 + rn\log n)$. Сложность алгоритма для разных форматов представлена в Таблице 4.1.

Таблица 4.1: Сложность cross-conv алгоритма для малоранговых форматов

Формат	Сложность
Скелетное разложение	$\mathcal{O}(nr^2 + rn\log n)$
Таккер 3D	$\mathcal{O}(nr^2 + r^4 + rn\log n)$
TT	$\mathcal{O}(dnr^3 + r^2 n\log n)$
HT/расширенный TT	$\mathcal{O}(dnr^2 + dr^4 + rn\log n)$

4.5 Численный эксперимент

В настоящем разделе в качестве численного эксперимента мы приводим вычисление трехмерной свертки с ядром Ньютона

$$V(x) = \left(f * \frac{1}{\|\cdot\|}\right)(x) \equiv \int_{\mathbb{R}^3} \frac{f(y)}{\|x-y\|}\, dy, \tag{4.14}$$

которая возникает при решении уравнения КШ/ХФ. Здесь $x = (x_1, x_2, x_3) \in \mathbb{R}^3$ и $\|x\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$.

Schur-Cross3D и cross-conv алгоритмы реализованы на языке Python. Их реализация и программный пакет базовых тензорных операций можно найти по ссылке https://github.com/rakhuba/tucker3d. Версия численного эксперимента, описанного в настоящем разделе, находится по ссылке https://bitbucket.org/rakhuba/crossconv-experiment. Там же можно найти данные плотностей молекул. Для основных операций линейной алгебры использована библиотека MKL. Python и MKL взяты из Enthought Python Distribution (EPD 7.3-1, 64-bit) https://www.enthought.com. Версия Python – 2.7.3. MKL версия – 10.3-1. Результаты получены на 4 Intel Core i7 2.6 GHz процессоре с 8GB RAM. Однако были использованы только 2 нити (по умолчанию в MKL). Отметим, что реализация написана на Python, и скорость программы может быть значительно увеличена с помощью использования C или Fortran.

Для дискретизации (4.14) для простоты была использована схема Нистрема (4.7) на $n \times n \times n$ сетках.

Сравнение с QTT алгоритмом. Для начала сравним cross-conv алгоритм с алгоритмом [63], который основан на матрично-векторном произведении в QTT формате (далее QTT алгоритм). Мы использовали MATLAB реализацию, которая доступна как часть TT-toolbox [129], а также заменили DMRG алгоритм матрично-векторного произведения [103], использованный в исходной работе, на более эффективный алгоритм AMEn [30, 31]. QTT алгоритм имеет логарифмическую сложность по размеру мод. Однако QTT ранги могут оказаться значительно больше таккеровских. Поэтому существует интервал мод, где cross-conv алгоритм является более быстрым, несмотря на тот факт, что асимптотически он проигрывает. Для иллюстрации этого факта мы рассмотрим вычис-

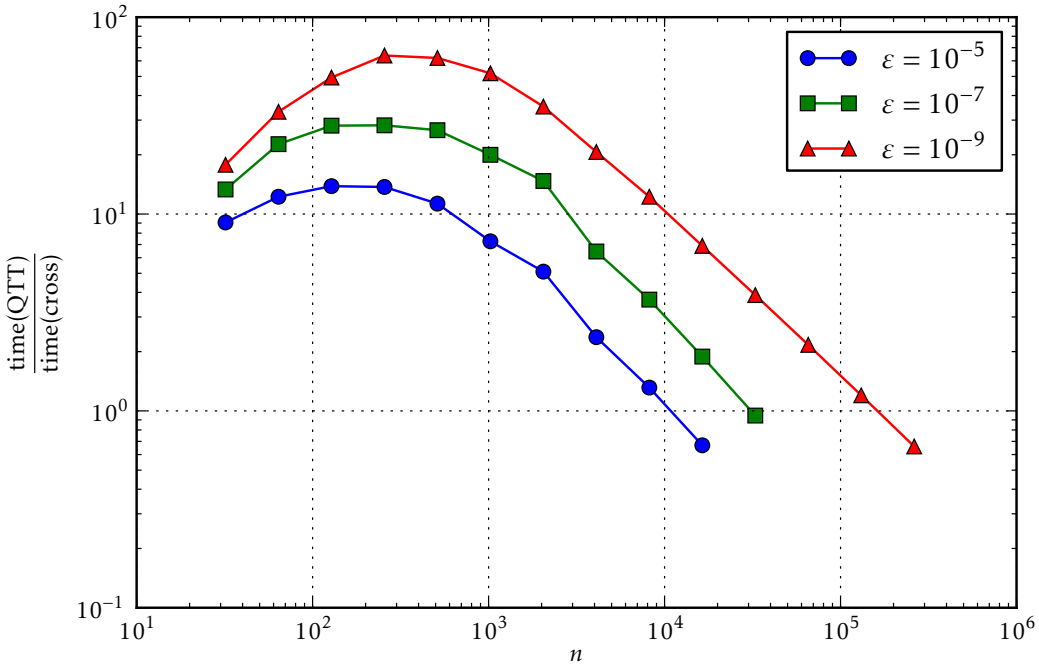


Рис. 4.1: Соотношение времени QTT и cross-conv алгоритмов как функция от n

ление потенциала Ньютона от функции Слейтера $f(y) = e^{-\zeta|y|}$ с $\zeta = 1$. Рисунок 4.1 демонстрирует соотношение времен вычислений в зависимости от размера мод. Абсолютные значения времен приведены в Таблице 4.2. Ясно, что чем более точны вычисления или больше ранги, тем быстрее cross-conv алгоритм по сравнению с QTT алгоритмом. Более того, cross-conv алгоритм быстрее для практически интересных значений мод до $n \sim 10^4 - 10^5$.

Потенциал Ньютона различных молекул. Приведем результаты расчетов сверток с различными молекулярными плотностями ρ. Для этой цели мы использовали заранее вычисленные значения ρ в формате Таккера. Размер сетки по каждой моде равняется $n = 5121$. Времена сверток с различными молекулами представлены в Таблице 4.3.

Алгоритм локальной фильтрации, предложенный в [123], имеет сложность $\mathcal{O}(nr^2 + r^5)$ в случае свертки функции в каноническом формате с функцией в формате Таккера. В случае, когда оба тензора заданы в формате Таккера, сложность этого алгоритма составляет $\mathcal{O}(nr^2 + r^6)$ в сравнении со сложностью $\mathcal{O}(nr^2 + r^4)$ cross-conv алгоритма. Мы реализовали Таккер-Таккер случай

Таблица 4.2: Ньютоновский потенциал функции Слейтера

n / ϵ	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}	2^{18}
Cross-conv времена (сек)												
10^{-5}	0.03	0.04	0.063	0.12	0.2	0.5	1.0	2.2				
10^{-7}	0.061	0.091	0.13	0.24	0.41	1.1	2.3	5.2	11.5			
10^{-9}	0.14	0.19	0.3	0.5	0.96	2.0	4.1	8.6	17.5	35.3	70.9	142.3
QTT времена (сек)												
10^{-5}	0.42	0.56	0.71	0.87	1.0	1.2	1.3	1.5				
10^{-7}	1.7	2.5	3.6	4.8	6.0	7.2	8.6	9.8	10.9			
10^{-9}	7.1	12.1	18.4	25.9	34.0	41.9	50.3	59.0	67.7	76.4	85.1	93.8
3D БПФ времена (сек)												
	1.3	12.6	118.7	1120	3 hours							

Таблица 4.3: Времена вычисления ньютоновского потенциала различных молекул при $n^3 = 5121^3$

Молекула	Точность	$\mathcal{F}_q$ ранги	$\widetilde{\mathcal{W}}$ ранги	Времена (сек)
CH_4	10^{-5}	$26 \times 26 \times 26$	$22 \times 22 \times 22$	1.3
	10^{-7}	$39 \times 39 \times 39$	$39 \times 39 \times 39$	4.1
	10^{-9}	$52 \times 52 \times 52$	$58 \times 58 \times 58$	6.4
C_2H_6	10^{-5}	$19 \times 30 \times 27$	$15 \times 23 \times 20$	1.2
	10^{-7}	$28 \times 49 \times 40$	$24 \times 42 \times 39$	3.9
	10^{-9}	$42 \times 66 \times 57$	$39 \times 66 \times 60$	6.2
C_2H_5OH	10^{-5}	$43 \times 42 \times 43$	$28 \times 28 \times 29$	2.3
	10^{-7}	$66 \times 67 \times 69$	$50 \times 50 \times 51$	7.5
	10^{-9}	$91 \times 90 \times 94$	$78 \times 79 \times 81$	19.8
$C_2H_5NO_2$	10^{-5}	$24 \times 60 \times 60$	$15 \times 33 \times 33$	2.6
	10^{-7}	$35 \times 93 \times 96$	$26 \times 61 \times 62$	9.4
	10^{-9}	$45 \times 126 \times 133$	$42 \times 97 \times 100$	18.4

из [123, 151] и обнаружили, что таккеровские ранги после локальной фультрации остались достаточно большими. Например, для случая ньютоновского потенциала молекулы C_2H_6, ранги оказались равными $361 \times 589 \times 532$ до фильтрации и $82 \times 144 \times 140$ после, в то время, как истинное значение рангов есть $19 \times 31 \times 28$. Из-за сильной зависимости от ранга это приводит к значительно большему времени вычислений. Отсюда следует, что cross-conv алгоритм является более устойчивым, чем алгоритм, базирующийся на локальной фильтрации факторов.

4.6 Выводы по главе

В настоящей главе был представлен эффективный cross-conv алгоритм для приближенного вычисления многомерной свертки в малоранговых тензорных форматах. Метод опирается на новый алгоритм метода крестовой аппроксимации, имеющий меньшую вычислительную сложность, чем аналоги. Результаты численных расчетов показывают, что cross-conv алгоритм является более эффективным, чем недавно предложенный QTT алгоритм вычисления свертки для интересных на практике размеров мод. Предложенный алгоритм используется в Главе 3 для вычисления оператора Фока для уравнения Хартри-Фока и теории функционала плотности.

Заключение

В настоящей работе разработаны новые тензорные методы решения многомерных частичных задач на собственные значения.

- Для поиска целевого собственного значения предложено обобщение метода Якоби-Дэвидсона при ограничении на тензорный ранг решения. Изучены свойства возникающих в этом методе систем линейных уравнений. Показана сходимость регуляризованного метода. Из полученных уравнений для метода Якоби-Дэвидсона получено обобщение метода обратной итерации. Показано преимущества метода в случае, когда возникающие линейные системы решаются неточно.

- Предложен ALS II метод, базирующийся на ALS подходе и методе обратной итерации. Получен результат о локальной сходимости метода через сходимость ALS итерации для минимизации отношения Релея. Для ALS оптимизации получена новая теория локальной сходимости, явно показывающая связь с мультипликативным методом Шварца.

- Предложена концепция предобуславливания на многообразии для поиска нескольких собственных значений, примененная к LOBPCG методу. С помощью предложенных итераций рассчитан спектр молекулы ацетонитрила с высокой точностью. Показано, что предлагаемый метод превосходит по точности и памяти существующие аналоги для расчета колебательного спектра.

- Рассмотрена задача на собственные значения с нелинейным оператором, возникающим в уравнениях Хартри-Фока и Кона-Шэма. Предложен полностью сеточный метод решения этих уравнений в формате Таккера. Сложность метода линейно зависит от размера сетки по каждому координатному направлению. Проведен точный расчет ряда атомов, молекул

и кластеров. Получены результаты, превышающие по точности подход, использующий глобальные базисные функции. Для кластеров с регулярным расположением атомов/молекул предложенный метод является более быстрым, чем базисный подход. Ключевыми особенностями метода являются пересчет матрицы Фока без производных, а также быстрое вычисление многомерной свертки.

- Для быстрого вычисления многомерной свертки предлагается cross-conv алгоритм, базирующийся на методе крестовой аппроксимации. Показано, что для размеров сетки, интересных на практике, cross-conv алгоритм превосходит по скорости существующие аналоги.

Литература

1. Absil P-A, Mahony Robert, Sepulchre Rodolphe. Optimization algorithms on matrix manifolds. — Princeton University Press, 2009.

2. Absil P-A, Mahony Robert, Trumpf Jochen. An extrinsic look at the Riemannian Hessian // Geometric science of information. — Springer, 2013. — P. 361–368.

3. Absil P. A., Oseledets I. V. Low-rank retractions: a survey and new results // Comput. Optim. Appl. — 2014. — URL: http://sites.uclouvain.be/absil/2013.04.

4. Arbenz Peter, Kressner Daniel, Zürich DME. Lecture notes on solving large scale eigenvalue problems // D-MATH, EHT Zurich. — 2012. — Vol. 2.

5. Avila Gustavo, Carrington Jr Tucker. Using a pruned basis, a non-product quadrature grid, and the exact Watson normal-coordinate kinetic energy operator to solve the vibrational Schrödinger equation for C2H4 // J. Chem. Phys. — 2011. — Vol. 135, no. 6. — P. 064101. — URL: http://dx.doi.org/10.1063/1.3617249.

6. Avila Gustavo, Carrington Jr Tucker. Using nonproduct quadrature grids to solve the vibrational Schrödinger equation in 12D // J. Chem. Phys. — 2011. — Vol. 134, no. 5. — P. 054126. — URL: http://dx.doi.org/10.1063/1.3549817.

7. Bacic Z, Light John C. Theoretical methods for rovibrational states of floppy molecules // Annu. Rev. Phys. Chem. — 1989. — Vol. 40, no. 1. — P. 469–498.

8. Ballani Jonas, Grasedyck Lars, Kluge Melanie. Black box approximation of tensors in hierarchical Tucker format // Linear Algebra Appl. — 2013. — Vol. 428. — P. 639–657.

9. Baye D., Heenen P.-H. Generalised meshes for quantum mechanical problems // J. Phys. A: Math. Gen. — 1986. — Vol. 19, no. 11. — P. 2041–2059.

10. Bebendorf M. Approximation of boundary element matrices // Numer. Math. — 2000. — Vol. 86, no. 4. — P. 565–589.

11. Bebendorf M. Adaptive cross approximation of multivariate functions // Const. Approx. — 2011. — Vol. 34, no. 2. — P. 149–179.

12. Benoit David M. Fast vibrational self-consistent field calculations through a reduced mode–mode coupling scheme // J. Chem. Phys. — 2004. — Vol. 120, no. 2. — P. 562–573. — URL: http://dx.doi.org/10.1063/1.1631817.

13. Berns-Müller Jörg, Graham Ivan G, Spence Alastair. Inexact inverse iteration for symmetric matrices // Linear Algebra Appl. — 2006. — Vol. 416, no. 2. — P. 389–413.

14. Beylkin Gregory, Garcke Jochen, Mohlenkamp Martin. Multivariate regression and machine learning with sums of separable functions // SIAM J. Sci. Comput. — 2009. — Vol. 31, no. 3. — P. 1840–1857.

15. Beylkin G., Mohlenkamp M. J. Numerical operator calculus in higher dimensions // Proc. Nat. Acad. Sci. USA. — 2002. — Vol. 99, no. 16. — P. 10246–10251.

16. Beylkin G., Mohlenkamp M. J. Algorithms for numerical analysis in high dimensions // SIAM J. Sci. Comput. — 2005. — Vol. 26, no. 6. — P. 2133–2159.

17. Billaud-Friess Marie, Nouy Anthony, Zahm Olivier. A tensor approximation method based on ideal minimal residual formulations for the solution of high-dimensional problems // ESAIM-Math. Model. Num. — 2014. — Vol. 48, no. 6. — P. 1777–1806.

18. Bischoff Florian A, Valeev Edward F. Low-order tensor approximations for electronic wave functions: Hartree–Fock method with guaranteed precision // J. Chem. Phys. — 2011. — Vol. 134, no. 10. — P. 104104.

19. Bowman Joel M, Gazdy Bela. A truncation/recoupling method for basis set calculations of eigenvalues and eigenvectors // J. Chem. Phys. — 1991. — Vol. 94, no. 1. — P. 454–460. — URL: http://dx.doi.org/10.1063/1.460361.

20. Chaudhury Anwesha, Oseledets Ivan, Ramachandran Rohit. A computationally efficient technique for the solution of multi-dimensional PBMs of granulation // Comput. Chem. Eng. — 2014. — Vol. 61, no. 11. — P. 234–244.

21. Chelikowsky James R, Troullier N, Saad Y. Finite-difference-pseudopotential method: electronic structure calculations without a basis // Phys. Rev. Lett. — 1994. — Vol. 72, no. 8. — P. 1240.

22. Chiu Jiawei, Demanet Laurent. Sublinear randomized algorithms for skeleton decompositions // SIAM J. Matrix Anal. Appl. — 2013. — Vol. 34, no. 3. — P. 1361–1383.

23. Chu Eleanor, George Alan. Inside the FFT black box: serial and parallel fast Fourier transform algorithms. — CRC Press, 1999.

24. Computation of extreme eigenvalues in higher dimensions using block tensor train format / S. V. Dolgov, B. N. Khoromskij, I. V. Oseledets, D. V. Savostyanov // Computer Phys. Comm. — 2014. — Vol. 185, no. 4. — P. 1207–1216.

25. Davidson Ernest R. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices // J. Comput. Phys. — 1975. — Vol. 17, no. 1. — P. 87–94.

26. Dawes Richard, Carrington Jr Tucker. How to choose one-dimensional basis functions so that a very efficient multidimensional basis may be extracted from a direct product of the one-dimensional functions: Energy levels of coupled systems with as many as 16 coordinates // J. Chem. Phys. — 2005. — Vol. 122, no. 13. — P. 134101. — URL: http://dx.doi.org/10.1063/1.1863935.

27. Direct minimization for calculating invariant subspaces in density functional computations of the electronic structure / Reinhold Schneider, Thorsten Rohwedder, Alexej Neelov, Johannes Blauert // J. Comp. Math. — 2009.

28. Dolgov S. V. TT-GMRES: solution to a linear system in the structured tensor format // Russ. J. Numer. Anal. Math. Model. — 2013. — Vol. 28, no. 2. — P. 149–172.

29. Dolgov S. V., Khoromskij B. N., Savostyanov D. V. Superfast Fourier transform using QTT approximation // J. Fourier Anal. Appl. — 2012. — Vol. 18, no. 5. — P. 915–953.

30. Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems : arXiv preprint : 1301.6068 ; Executor: S. V. Dolgov, D. V. Savostyanov : 2013. — URL: http://arxiv.org/abs/1301.6068.

31. Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems : arXiv preprint : 1304.1222 ; Executor: S. V. Dolgov, D. V. Savostyanov : 2013. — URL: http://arxiv.org/abs/1304.1222.

32. Dolgov S. V., Savostyanov D. V. Alternating minimal energy methods for linear systems in higher dimensions // SIAM J. Sci. Comput. — 2014. — Vol. 36, no. 5. — P. A2248–A2271.

33. Dolgov S. V., Savostyanov D. V. Corrected one-site density matrix renormalization group and alternating minimal energy algorithm // Numerical Mathematics and Advanced Applications — ENUMATH 2013. — Vol. 103. — 2015. — P. 335–343.

34. Eckart Carl, Young Gale. The approximation of one matrix by another of lower rank // Psychometrika. — 1936. — Vol. 1, no. 3. — P. 211–218.

35. Frommer A., Nabben R., Szyld D. B. Convergence of stationary iterative methods for Hermitian semidefinite linear systems and applications to Schwarz methods // SIAM J. Matrix Anal. Appl. — 2008. — Vol. 30, no. 2. — P. 925–938.

36. Fully adaptive algorithms for multivariate integral equations using the non-standard form and multiwavelets with applications to the Poisson and bound-state Helmholtz kernels in three dimensions / Luca Frediani, Eirik Fossgaard,

Tor Flå, Kenneth Ruud // Molecular Physics. — 2013. — Vol. 111, no. 9-11. — P. 1143–1160.

37. General atomic and molecular electronic structure system / Michael W Schmidt, Kim K Baldridge, Jerry A Boatz et al. // J. Comput. Chem. — 1993. — Vol. 14, no. 11. — P. 1347–1363.

38. Golub Gene H, Ye Qiang. Inexact inverse iteration for generalized eigenvalue problems // BIT Numer. Math. — 2000. — Vol. 40, no. 4. — P. 671–684.

39. Goreinov S. A. On cross approximation of multi-index array // Doklady Math. — 2008. — Vol. 420, no. 4. — P. 404–406.

40. Goreinov S. A., Oseledets I. V., Savostyanov D. V. Wedderburn rank reduction and Krylov subspace method for tensor approximation. Part 1: Tucker case // SIAM J. Sci. Comput. — 2012. — Vol. 34, no. 1. — P. A1–A27.

41. Goreinov S. A., Tyrtyshnikov E. E. The maximal-volume concept in approximation by low-rank matrices // Contemporary Mathematics. — 2001. — Vol. 280. — P. 47–51.

42. Goreinov S. A., Tyrtyshnikov E. E., Zamarashkin N. L. A theory of pseudo–skeleton approximations // Linear Algebra Appl. — 1997. — Vol. 261. — P. 1–21.

43. Grasedyck L. Hierarchical singular value decomposition of tensors // SIAM J. Matrix Anal. Appl. — 2010. — Vol. 31, no. 4. — P. 2029–2054.

44. Grasedyck L., Kressner D., Tobler C. A literature survey of low-rank tensor approximation techniques // GAMM-Mitt. — 2013. — Vol. 36, no. 1. — P. 53–78.

45. Hackbusch W. Fast and exact projected convolution for non-equidistant grids // Computing. — 2007. — Vol. 80, no. 2. — P. 137–168.

46. Hackbusch W. Efficient convolution with the Newton potential in d dimensions // Numerische Mathematik. — 2008. — Vol. 110, no. 4. — P. 449–489.

47. Hackbusch W., Khoromskij B. N. Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. I. Separable approximation of multivariate functions // Computing. — 2006. — Vol. 76, no. 3-4. — P. 177–202.

48. Hackbusch W., Khoromskij B. N. Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. II. HKT representation of certain operators // Computing. — 2006. — Vol. 76, no. 3-4. — P. 203–225.

49. Hackbusch W., Kühn S. A new scheme for the tensor representation // J. Fourier Anal. Appl. — 2009. — Vol. 15, no. 5. — P. 706–722.

50. Hartree Douglas Rayner. The calculation of atomic structures. — J. Wiley, 1957.

51. Higher-order adaptive finite-element methods for Kohn–Sham density functional theory / Phani Motamarri, Michael R Nowak, Kenneth Leiter et al. // J. Comput. Phys. — 2013. — Vol. 253. — P. 308–343.

52. Hitchcock F. L. Multiple invariants and generalized rank of a p-way matrix or tensor // J. Math. Phys. — 1927. — Vol. 7, no. 1. — P. 39–79.

53. Holtz S., Rohwedder T., Schneider R. On manifolds of tensors of fixed TT–rank // Numer. Math. — 2012. — Vol. 120, no. 4. — P. 701–731.

54. Holtz S., Rohwedder T., Schneider R. The alternating linear scheme for tensor optimization in the tensor train format // SIAM J. Sci. Comput. — 2012. — Vol. 34, no. 2. — P. A683–A713.

55. 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.

56. Improved Roothaan–Hartree–Fock wave functions for atoms and ions with $N \leq 54$ / Toshikatsu Koga, Shinya Watanabe, Katsutoshi Kanayama et al. // J. Chem. Phys. — 1995. — Vol. 103, no. 8. — P. 3000–3005.

57. Ipsen Ilse CF. Computing an eigenvector with inverse iteration // SIAM rev. — 1997. — Vol. 39, no. 2. — P. 254–291.

58. Ishida Toshimasa, Ohno Koichi. On the asymptotic behavior of Hartree-Fock orbitals // Theoretica Chimica Acta. — 1992. — Vol. 81, no. 6. — P. 355–364.

59. Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients / Mike C Payne, Michael P Teter, Douglas C Allan et al. // Rev. Mod. Phys. — 1992. — Vol. 64, no. 4. — P. 1045.

60. Jensen Frank. Introduction to computational chemistry. — John Wiley & Sons, 2013.

61. Joyce D. C. Survey of extrapolation processes in numerical analysis // SIAM Rev. — 1971. — Vol. 13, no. 4. — P. 435–490.

62. Kalos MH. Monte Carlo calculations of the ground state of three-and four-body nuclei // Physical Review. — 1962. — Vol. 128, no. 4. — P. 1791.

63. Kazeev V., Khoromskij B., Tyrtyshnikov E. Multilevel Toeplitz Matrices Generated by Tensor-Structured Vectors and Convolution with Logarithmic Complexity // SIAM J. Sci. Comput. — 2013. — Vol. 35, no. 3. — P. A1511–A1536.

64. Khoromskaia V. Computation of the Hartree-Fock exchange by tensor-structured methods // Comput. Methods Appl. Math. — 2008. — Vol. 10, no. 2.

65. Khoromskaia V. Numerical solution of the Hartree-Fock equation by multilevel tensor-structured methods : Ph. D. thesis / V. Khoromskaia ; TU Berlin. — 2010. — URL: http://opus.kobv.de/tuberlin/volltexte/2011/2948/.

66. Khoromskaia V. Black-Box Hartree–Fock Solver by Tensor Numerical Methods // Computational Methods in Applied Mathematics. — 2014. — Vol. 14, no. 1. — P. 89–111.

67. Khoromskaia Venera, Khoromskij Boris N. Tensor numerical methods in quantum chemistry: from Hartree–Fock to excitation energies // Physical Chemistry Chemical Physics. — 2015.

68. Khoromskaia V., Khoromskij B. N., Schneider R. QTT representation of the Hartree and exchange operators in electronic structure calculations // Comput. Methods Appl. Math. — 2011. — Vol. 11, no. 3. — P. 327–341.

69. Khoromskij B. N. Structured rank-$(r_1,\dots,r_d)$ decomposition of function-related operators in $\mathbb{R}^d$ // Comput. Methods Appl. Math. — 2006. — Vol. 6, no. 2. — P. 194–220.

70. Khoromskij B. N. Tensor-structured preconditioners and approximate inverse of elliptic operators in $\mathbb{R}^d$ // Constr. Approx. — 2009. — Vol. 30. — P. 599–620.

71. Khoromskij B. N. Fast and accurate tensor approximation of multivariate convolution with linear scaling in dimension // J. Comp. Appl. Math. — 2010. — Vol. 234, no. 11. — P. 3122–3139.

72. Khoromskij B. N. $\mathcal{O}(d\log N)$–Quantics approximation of N–d tensors in high-dimensional numerical modeling // Constr. Approx. — 2011. — Vol. 34, no. 2. — P. 257–280.

73. Khoromskij B. N., Khoromskaia V. Low rank Tucker-type tensor approximation to classical potentials // Central European journal of mathematics. — 2007. — Vol. 5, no. 3. — P. 523–550.

74. Khoromskij B. N., Khoromskaia V. Multigrid accelerated tensor approximation of function related multidimensional arrays // SIAM J. Sci. Comput. — 2009. — Vol. 31, no. 4. — P. 3002–3026.

75. Khoromskij B. N., Khoromskaia V., Flad. H.-J. Numerical solution of the Hartree–Fock equation in multilevel tensor-structured format // SIAM J. Sci. Comput. — 2011. — Vol. 33, no. 1. — P. 45–65.

76. Knyazev Andrew V. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method // SIAM J. Sci. Comput. — 2001. — Vol. 23. — P. 517–541.

77. Knyazev Andrew V, Neymeyr Klaus. A geometric theory for preconditioned inverse iteration III: A short and sharp convergence estimate for generalized eigenvalue problems // Linear Algebra Appl. — 2003. — Vol. 358, no. 1-3. — P. 95–114.

78. Kohn Walter, Sham Lu Jeu. Self-consistent equations including exchange and correlation effects // Phys. Rev. — 1965. — Vol. 140, no. 4A. — P. A1133.

79. Kolda T. G., Bader B. W. Tensor decompositions and applications // SIAM Rev. — 2009. — Vol. 51, no. 3. — P. 455–500.

80. Kressner Daniel, Steinlechner Michael, Vandereycken Bart. Preconditioned low-rank Riemannian optimization for linear systems with tensor product structure // SIAM J. Sci. Comput. — 2016. — Vol. 38, no. 4. — P. A2018–A2044.

81. Kressner D., Tobler C. Krylov Subspace Methods for Linear Systems with Tensor Product Structure // SIAM J. Matrix Anal. Appl. — 2010. — Vol. 31. — P. 1688–1714.

82. Kressner D., Tobler C. Preconditioned low-rank methods for high-dimensional elliptic PDE eigenvalue problems // Computational Methods in Applied Mathematics. — 2011. — Vol. 11, no. 3. — P. 363–381.

83. Kwok Yue Kuen, Leung Kwai Sun, Wong Hoi Ying. Efficient options pricing using the fast Fourier transform // Handbook of computational finance. — Springer, 2012. — P. 579–604.

84. Lebedeva O. S. Block tensor conjugate gradient-type method for Rayleigh quotient minimization in two-dimensional case // Comput. Math. Math. Phys. — 2010. — Vol. 50, no. 5. — P. 749–765.

85. Lebedeva O. S. Tensor conjugate-gradient-type method for Rayleigh quotient minimization in block QTT-format // Russ. J. Numer. Anal. Math. Modelling. — 2011. — Vol. 26, no. 5. — P. 465–489.

86. Leclerc Arnaud, Carrington Tucker. Calculating vibrational spectra with sum of product basis functions without storing full-dimensional vectors or matrices // J. Chem. Phys. — 2014. — Vol. 140, no. 17. — P. 174111. — URL: http://dx.doi.org/10.1063/1.4871981.

87. Lee John M. Introduction to smooth manifolds. — 2001.

88. Lipnikov K, Svyatskiy D, Vassilevski Y. Anderson acceleration for nonlinear finite volume scheme for advection-diffusion problems // SIAM J. Sci. Comput. — 2013. — Vol. 35, no. 2. — P. A1120–A1136.

89. Lubich Christian, Oseledets Ivan, Vandereycken Bart. Time integration of tensor trains // SIAM J. Numer. Anal. — 2015. — Vol. 53, no. 2. — P. 917–941. — URL: http://arxiv.org/abs/1407.2042.

90. Mach T. Computing Inner Eigenvalues of Matrices in Tensor Train Matrix Format // Numerical Mathematics and Advanced Applications 2011. — Springer Berlin Heidelberg, 2013. — P. 781–788.

91. Mortensen Jens Jørgen, Hansen Lars Bruno, Jacobsen Karsten Wedel. Real-space grid implementation of the projector augmented wave method // Phys. Rev. B. — 2005. — Vol. 71, no. 3. — P. 035109.

92. Multidimensional Quantum Dynamics: MCTDH Theory and Applications / Ed. by H.-D. Meyer, F. Gatti, G. A. Worth. — Weinheim : Wiley-VCH, 2009.

93. Multiresolution quantum chemistry: Basic theory and initial applications / Robert J Harrison, George I Fann, Takeshi Yanai et al. // J. Chem. Phys. — 2004. — Vol. 121, no. 23. — P. 11587–11598.

94. Multiresolution quantum chemistry: Basic theory and initial applications / Robert J Harrison, George I Fann, Takeshi Yanai et al. // The Journal of chemical physics. — 2004. — Vol. 121, no. 23. — P. 11587–11598.

95. Multiresolution quantum chemistry in multiwavelet bases: Analytic derivatives for Hartree-Fock and density functional theory / Takeshi Yanai, George I Fann, Zhengting Gan et al. // The Journal of chemical physics. — 2004. — Vol. 121, no. 7. — P. 2866–2876.

96. Neymeyr Klaus. A geometric theory for preconditioned inverse iteration I: Extrema of the Rayleigh quotient // Linear Algebra Appl. — 2001. — Vol. 322, no. 1-3. — P. 61–85.

97. Notay Yvan. Combination of Jacobi–Davidson and conjugate gradients for the partial symmetric eigenproblem // Numer. Linear Algebra Appl. — 2002. — Vol. 9, no. 1. — P. 21–44.

98. Notay Yvan. Inner iterations in eigenvalue solvers // Report GANMN 05. — 2005. — Vol. 1.

99. Ortega J. M., Rheinboldt W. C. Iterative Solution of Nonlinear Equations in Several Variables. — New York : Academic Press, 1970. — P. xx+572.

100. Oseledets Ivan, Muravleva Ekaterina. Fast orthogonalization to the kernel of the discrete gradient operator with application to Stokes problem // Linear Algebra and its Applications. — 2010. — Vol. 432, no. 6. — P. 1492–1500.

101. Oseledets I., Rakhuba M., Uschmajew A. Alternating least squares as moving subspace correction // INS Preprint. — 2017.

102. Oseledets I. V. Approximation of $2^d \times 2^d$ matrices using tensor decomposition // SIAM J. Matrix Anal. Appl. — 2010. — Vol. 31, no. 4. — P. 2130–2145.

103. Oseledets I. V. DMRG approach to fast linear algebra in the TT–format // Comput. Meth. Appl. Math. — 2011. — Vol. 11, no. 3. — P. 382–393.

104. Oseledets I. V. Tensor-train decomposition // SIAM J. Sci. Comput. — 2011. — Vol. 33, no. 5. — P. 2295–2317.

105. Oseledets I. V., Dolgov S. V. Solution of linear systems and matrix inversion in the TT-format // SIAM J. Sci. Comput. — 2012. — Vol. 34, no. 5. — P. A2718–A2739.

106. Oseledets I. V., Savostianov D. V., Tyrtyshnikov E. E. Tucker dimensionality reduction of three-dimensional arrays in linear time // SIAM J. Matrix Anal. Appl. — 2008. — Vol. 30, no. 3. — P. 939–956.

107. Oseledets I. V., Savostyanov D. V., Tyrtyshnikov E. E. Linear algebra for tensor problems // Computing. — 2009. — Vol. 85, no. 3. — P. 169–188.

108. Oseledets I. V., Tyrtyshnikov E. E. Breaking the curse of dimensionality, or how to use SVD in many dimensions // SIAM J. Sci. Comput. — 2009. — Vol. 31, no. 5. — P. 3744–3759.

109. Oseledets I. V., Tyrtyshnikov E. E. TT-cross approximation for multidimensional arrays // Linear Algebra Appl. — 2010. — Vol. 432, no. 1. — P. 70–88.

110. Östlund S., Rommer S. Thermodynamic limit of Density Matrix Renormalization // Phys. Rev. Lett. — 1995. — Vol. 75, no. 19. — P. 3537–3540.

111. Perdew John P, Zunger Alex. Self-interaction correction to density-functional approximations for many-electron systems // Phys. Rev. B. — 1981. — Vol. 23, no. 10. — P. 5048.

112. Pižorn Iztok, Verstraete Frank. Variational Numerical Renormalization Group: Bridging the Gap between NRG and Density Matrix Renormalization Group // Phys. Rev. Lett. — 2012. — Vol. 108, no. 067202.

113. Rakhuba Maxim, Oseledets Ivan. Calculating vibrational spectra of molecules using tensor train decomposition // J. Chem. Phys. — 2016. — Vol. 145. — P. 124101.

114. Rakhuba M. V., Oseledets I. V. Fast multidimensional convolution in low-rank tensor formats via cross approximation // SIAM J. Sci. Comput. — 2015. — Vol. 37, no. 2. — P. A565–A582.

115. Rakhuba M. V., Oseledets I. V. Grid-based electronic structure calculations: the tensor decomposition approach // J. Comp. Phys. — 2016. — URL: http://arxiv.org/abs/1508.07632.

116. Robbin JW, Salamon DA. Introduction to Differential Geometry // Lecture Notes, ETH. — 2011.

117. Rohwedder T., Uschmajew A. On Local Convergence of Alternating Schemes for Optimization of Convex Problems in the Tensor Train Format // SIAM J. Num. Anal. — 2013. — Vol. 51, no. 2. — P. 1134–1162.

118. Russell D. Johnson III. NIST Computational Chemistry Comparison and Benchmark Database Number 101 Release 16a. — August 2013. — URL: http://webbook.nist.gov.

119. Sameh Ahmed H, Wisniewski John A. A trace minimization algorithm for the generalized eigenvalue problem // SIAM J. Numer. Anal. — 1982. — Vol. 19, no. 6. — P. 1243–1259.

120. Quasioptimality of maximum–volume cross interpolation of tensors : arXiv preprint : 1305.1818 ; Executor: D. V. Savostyanov : 2013. — URL: http://arxiv.org/abs/1305.1818.

121. Savostyanov D. V. Quasioptimality of maximum–volume cross interpolation of tensors // Linear Algebra Appl. — 2014. — Vol. 458. — P. 217–244.

122. Savostyanov D. V., Oseledets I. V. Fast adaptive interpolation of multi-dimensional arrays in tensor train format // Proceedings of 7th International Workshop on Multidimensional Systems (nDS). — IEEE, 2011.

123. Savostyanov D. V., Tyrtyshnikov E. E. Approximate multiplication of tensor matrices based on the individual filtering of factors // J. Comp. Math. Math. Phys. — 2009. — Vol. 49, no. 10. — P. 1662–1677.

124. Savostyanov D. V., Tyrtyshnikov E. E., Zamarashkin N. L. Fast truncation of mode ranks for bilinear tensor operations // Numer. Linear Algebra Appl. — 2012. — Vol. 19, no. 1. — P. 103–111.

125. Schollwöck U. The density-matrix renormalization group in the age of matrix product states // Annals of Physics. — 2011. — Vol. 326, no. 1. — P. 96–192.

126. Sleijpen Gerard L. G., Van der Vorst Henk A. A Jacobi–Davidson iteration method for linear eigenvalue problems // SIAM Rev. — 2000. — Vol. 42, no. 2. — P. 267–293.

127. Steinlechner Michael. Riemannian Optimization for Solving High-Dimensional Problems with Low-Rank Tensor Structure. — 2016.

128. Stewart G. W. On the perturbation of pseudo-inverses, projections and linear least squares problems // SIAM review. — 1977. — Vol. 19, no. 4. — P. 634–662.

129. Oseledets I. V., Dolgov S., Kazeev V. et al. TT-Toolbox. — https://github.com/oseledets/TT-Toolbox. URL: https://github.com/oseledets/TT-Toolbox.

130. Tensor decomposition in electronic structure calculations on 3D Cartesian grids / B. N. Khoromskij, V. Khoromskaia, S. R. Chinnamsetty, H.-J. Flad // J. Comput. Phys. — 2009. — Vol. 228, no. 16. — P. 5749–5762.

131. Tensor product approximation (DMRG) and coupled cluster method in quantum chemistry / Örs Legeza, Thorsten Rohwedder, Reinhold Schneider, Szilárd Szalay // Many-Electron Approaches in Physics, Chemistry and Mathematics. — Springer, 2014. — P. 53–76.

132. Thomas Phillip S, Carrington Jr Tucker. Using Nested Contractions and a Hierarchical Tensor Format to Compute Vibrational Spectra of Molecules with Seven Atoms // J. Phys. Chem. A. — 2015. — P. 13074–13091.

133. Tucker L. R. Some mathematical notes on three-mode factor analysis // Psychometrika. — 1966. — Vol. 31. — P. 279–311.

134. Tyrtyshnikov E. E. Optimal and superoptimal circulant preconditioners // SIAM J. Matrix Anal. Appl. — 1992. — Vol. 13, no. 2. — P. 459–473.

135. Tyrtyshnikov E. E. Incomplete cross approximation in the mosaic–skeleton method // Computing. — 2000. — Vol. 64, no. 4. — P. 367–380.

136. Uschmajew André, Vandereycken Bart. Greedy rank updates combined with Riemannian descent methods for low-rank optimization // International Conference on Sampling Theory and Applications (SampTA). — 2015. — P. 420–424.

137. Use of tensor formats in elliptic eigenvalue problems / W. Hackbusch, B. N. Khoromskij, S. A. Sauter, E. E. Tyrtyshnikov // Numer. Linear Algebra Appl. — 2012. — Vol. 19, no. 1. — P. 133–151.

138. Vahtras O, Almlöf J, Feyereisen MW. Integral approximations for LCAO-SCF calculations // Chem. Phys. Lett. — 1993. — Vol. 213, no. 5. — P. 514–518.

139. Vandereycken Bart. Low-rank matrix completion by Riemannian optimization // SIAM J. Optim. — 2013. — Vol. 23, no. 2. — P. 1214–1236.

140. Wang Haobin, Thoss Michael. Multilayer formulation of the multiconfiguration time-dependent Hartree theory // J. Chem. Phys. — 2003. — Vol. 119, no. 3. — P. 1289–1299. — URL: http://dx.doi.org/10.1063/1.1580111.

141. White S. R. Density matrix formulation for quantum renormalization groups // Phys. Rev. Lett. — 1992. — Vol. 69, no. 19. — P. 2863–2866.

142. Woods John W. Multidimensional signal, image, and video processing and coding. — Academic Press, 2006.

143. de Lathauwer L., de Moor B., Vandewalle J. On best rank-1 and rank-$(R_1, R_2, ..., R_N)$ approximation of high-order tensors // SIAM J. Matrix Anal. Appl. — 2000. — Vol. 21. — P. 1324–1342.

144. de Lathauwer L., de Moor B., Vandewalle J. A multilinear singular value decomposition // SIAM J. Matrix Anal. Appl. — 2000. — Vol. 21. — P. 1253–1278.

145. de Silva V., Lim L.-H. Tensor rank and the ill-posedness of the best low-rank approximation problem // SIAM J. Matrix Anal. Appl. — 2008. — Vol. 30, no. 3. — P. 1084–1127.

146. Агошков В. И. Методы оптимального управления и сопряженных уравнений в задачах математической физики // М.: ИВМ РАН. — 2003.

147. Горейнов С. А. О крестовой аппроксимации многоиндексного массива // Докл. РАН. — 2008. — Т. 420, № 4. — С. 439–441.

148. Горейнов С. А., Тыртышников Е. Е., Замарашкин Н. Л. Псевдоскелетная аппроксимация матриц // Докл. РАН. — 1995. — Т. 342, № 2. — С. 151–152.

149. Оселедец И. В. О новом тензорном разложении // ДАН. — 2009. — Т. 427, № 2. — С. 168–169.

150. Оселедец И. В. О приближении матриц логарифмическим числом параметров // ДАН. — 2009. — Т. 428, № 1. — С. 23–24.

151. Савостьянов Д. В., Тыртышников Е. Е. Приближенное умножение тензорных матриц на основе индивидуальной фильтрации факторов // Ж. вычисл. матем. и матем. физ. — 2009. — Т. 49, № 10. — С. 1741–1756.

152. Тыртышников Е. Е. Методы численного анализа. — М.: “Академия”, 2007.