Федеральное государственное бюджетное учреждение науки
Институт вычислительной математики Российской академии наук
(ИВМ РАН)

На правах рукописи

Долгов Сергей Владимирович

Алгоритмы и применения тензорных разложений для численного решения многомерных нестационарных задач

01.01.07 — Вычислительная математика

Диссертация
на соискание ученой степени
кандидата физико-математических наук

Научный руководитель
чл.-корр. РАН, проф. Тыртышников Е. Е.

Москва 2014

Оглавление

Введение

Объект исследования и актуальность работы

Эта диссертация посвящена численному решению многомерных задач методами тензорных разложений. Что мы подразумеваем под задачами высокой размерности, и как они возникают на практике? В линейной алгебре рассматриваются векторы и матрицы, и под "размерностью" обычно понимается размер, т.е. количество элементов в векторе. Оно может быть классифицировано как "высокое", например по сравнению с имеющейся компьютерной памятью. Однако, под термином "многомерный" мы понимаем нечто иное. В качестве основных приложений мы выделяем квантовые и вероятностные физические модели, такие как уравнения Фоккера-Планка, управляющее уравнение, или уравнение Шредингера. Чтобы понять, в каком смысле они являются многомерными, начнем с иллюстрации на следующем примере.

Предположим, что задан не конкретный вектор, а *класс*, или семейство векторов, так что элементы подчиняются определенным независимым вычислительным правилам. Пусть правило для каждого элемента может давать конечное число различных значений. Все *экземпляры* такого класса могут быть также собраны в вектор: мы просто перечисляем все возможные комбинации реализаций. Если каждый исходный элемент может принимать n значений, число комбинации двух элементов составляет уже $n \cdot n = n^2$, и число реализаций класса из d элементов принимает значение $n \cdot n \cdots n = n^d$. Это огромное количество: всего лишь 80 элементов (требующие 640 байт для хранения их с двойной точностью) при 10 возможных значениях каждого из них дают 10^{80} комбинаций – качественная оценка числа всех атомов во Вселенной. Этот простейший пример иллюстрирует тем не менее два ключевые момента в данной работе: способ, каким мы получили огромное количество значений из относительно небольшого числа исходных элементов, будет возникать в наших основных приложениях, а концепция и понимание того, что мы можем хранить не все 10^{80} экземпляров, а только 80 векторов по 10 значений в каждом, определяющих исходные правила, будут лежать в основе вычислительных методов.

Идея пространства экземпляров, или если говорить более строго, *пространства состояний*, является краеугольным камнем в квантовых и стохастических моделях. Система многих тел может быть описана обыкновенными дифференциальными уравнениями, которые определяют эволюцию d координат позиций частиц, или других степеней свободы. Основные проблемы численного решения проистекают из нелинейной формы физических законов, но само хранение реше-

ния в компьютере не является проблемой, так как в настоящее время даже рабочая станция может с легкостью оперировать миллиардом неизвестных.

Однако это становится не так, как только появляется случайность. Невозможно определенно предсказать положение случайно блуждающей частицы. Тем не менее, можно измерить *вероятность* того, что частица в данный момент находится в определенной области пространства.

Разумная идея может состоять в том, чтобы сравнить такие вероятности для различных областей. Мы должны разделить все пространство на пронумерованные клетки (возможно, бесконечно малые), и ввести функцию, которая возвращает количественное значение вероятности для данного номера клетки. Для хранения вероятностного описания в компьютере, мы ограничимся конечным количеством областей. Предположим, что частица живет в обычном трехмерном изотропном пространстве. Поскольку предпочтительное направление не определено, можно выбрать некоторое количество разбиений n по каждой из трех осей. В итоге, мы получаем n^3 значений вероятности для трех независимых координат.

Если мы хотим одновременно описать несколько взаимодействующих под действием случайных сил тел, совместная функция плотности вероятности задается в d-мерном пространстве, и, как правило, требуют n^d значений после дискретизации, где d составляет количество всех координат всех частиц. Таким образом, под "размерностью" мы будем иметь в виду количество *конфигурационных координат* d в пространстве состояний системы, в то время как n обозначает количество возможных точек по каждой координате. В принципе, даже случаи $d = 3$ или $d = 2$ можно рассматривать как "многомерные" , если n очень велико. Ситуация становится еще более драматичной, если физическая или математическая модель предусматривает работу с размерностями порядка десятков, сотен и более. Если исключить тривиальный случай, когда $n = 1$ для большинства координат (в этом случае нет смысла рассматривать соответствующие переменные вообще), экспоненциальный рост вычислительной сложности с размерностью делает невозможными непосредственные расчеты на любой суперкомпьютере. Например, в квантовом мире, частицы со спином $1/2$ (в определенных условиях, например в магнитно-резонансных экспериментах, электроны и ядра могут быть рассмотрены только с точки зрения спиновых эффектов) обладают только $n = 2$ состояниями для каждой частицы, "спин вверх " и "спин вниз". Однако, простая линейная цепочка из $d = 100$ взаимодействующих спинов (что рассматривается как модельная задача в квантовой физике) описывается уже волновой функцией с $2^{100} \sim 10^{30}$ неизвестными.

Это явление экспоненциального роста сложности в зависимости от количества конфигурационных координат называется *проклятием размерности* с работы [23]. Таким образом, единственным способом решения таких задач является работа только с небольшой *эффективной* частью дискретной информации, что требует гораздо меньше памяти и вычислительных операций, чем начальный массив n^d чисел. Вместе с этим предложением, естественно ожидать, что нам на самом деле и не нужны все n^d элементов. Квантовое, а также стохастическое моделирование, как правило, проводится для выявления высокоточной *статистики*, или *наблюдаемых* величин, таких как среднее, дисперсия, энергия, и др., которые уже являются низкоразмерными, и требуют многомерные данные только на промежу-

точном этапе их вычисления.

Среди всех подходов для сжатия информации, так называемых методов с *разреженными данными*, мы можем выделить проблемно-ориентированные и общие классы. Первый класс предполагает и существенно использует специфические свойства задачи, такие как гладкость соответствующих функций, определенные правила вычисления статистических величин, и так далее. Среди наиболее известных и широко применяемых методов, например, следующие: методы Монте Карло [177, 74, 60, 19, 63] (вместе с большим количеством улучшений, таких как квази Монте Карло [186, 226, 169], Монте Карло на Марковской цепи [111], и др.), разреженные сетки Смоляка [9, 36, 98, 75], радиальные базисные функции [33, 34], вейвлеты и другие наилучшие N-term аппроксимации [39, 180], а также специальные методы редукции и базисы, разработанные с использованием физической интуиции для конкретных задач. В качестве одного из наиболее успешных подходов последнего типа, мы можем упомянуть Гауссовы орбитали [64, 179] с расширениями до более общих базисов с использованием сеточных квадратур [147, 236, 148, 133, 134, 136], Coupled Cluster подход [20, 237] для коррекции решения уравнения Хартри-Фока, или State Space Restriction [165, 228] в спиновой динамике.

Общие методы не используют в явном виде физический смысл задачи или выходных данных, полагаясь вместо этого на чисто математические инструменты для представления всего многомерного объекта с помощью правильно выбранного отображения небольшого количества данных. Ради справедливости стоит отметить, что доказательство *применимости* и полезности таких методов может часто потребовать подробного вникания в детали задачи. Кроме того, специализированные методы имеют больше шансов оказаться вычислительно эффективнее общих методов. Тем не менее, потенциальная возможность получения *любой* части информации (возможно, приближенной) о многомерном объекта, а также общность интерфейса для входных данных оставляет для таких методов важную роль. Например, можно проверить такой подход на любой новой задаче без существенного изменения алгоритмов, и увидеть, работает ли он в принципе, или использовать его для верификации какого-либо другого метода (который, возможно, в итоге и окажется быстрее).

Примечательно эффективным представителем класса общих методов сжатия данных является концепция разделения переменных в *тензорных произведениях*, испытавшая быстрое развитие в последнее десятилетие, разрабатываемая и в настоящее время, в том числе в текущей работе. Общая идея состоит в том, чтобы представить (или приблизить) большой многомерный массив с помощью комбинации произведений и сложений небольших массивов с меньшим количеством степеней свободы. Важно отметить, что существуют определенные разложения и методы, которые требуют только исходных данных и используют только алгебраические инструменты (например, сингулярное разложение) для выделения редуцированного набора параметров. Очевидно, что они не требуют физического понимания смысла конкретных данных, хотя реальная эффективность сжатия конечно зависит от функциональных свойств, таких например как гладкость. Интересно, что, сужая допустимые условия на входные данные, мы можем придти к

тому, что различные методы ведут себя сходно и в теории, и на практике. В качестве примера можно привести сравнение методов разреженных сеток и разделения переменных в работе [99].

Ключевым моментом в разделении переменных является представление многомерной функции (или ее дискретного аналога, *тензора*) в виде произведения одномерных объектов, т.е.

$$x(i_1, \ldots, x_d) = x^{(1)}(i_1)x^{(2)}(i_2)\cdots x^{(d)}(i_d).$$

Если это разложение в *прямое произведение* не выполняется точно, можно рассматривать его как *словарь*, и приблизить более общий объект посредством комбинации нескольких прямых произведений. Широко используемым вычислительным подходом являются "жадные" (greedy) методы. Основная идея описана, например, в книге [235]. Значительный вклад в концепцию "жадных" алгоритмов с *тензорными произведениями*, которые вычисляют линейную комбинацию прямых произведений, был осуществлен в работах [184, 166, 188, 73, 42, 37, 86]. Этот подход уменьшает ошибку решения (например $\|x - \tilde{x}\|^2$, или другую функцию, такую как невязка или отношение Рэлея) путем последовательного извлечения наилучших (или почти наилучших) приближений в форме прямого произведения. Можно провести анализ сходимости для "жадных" методов (см. ссылки, приведенные выше), при условии, что каждая оптимизация компонентов прямого произведения производится с гарантированной точностью. Однако именно это требование трудно удовлетворить на практике. Во-первых, невязка становится все более и более осциллирующей на последних итерациях, и ее приближение прямым произведением (даже оптимальное) дает все более низкую точность. Во-вторых, в реальном численном методе трудно достичь оптимальности приближения. Как правило, это и является причиной того, почему "жадные" тензорные методы стагнируют на каком-то уровне ошибки, который часто оказывается неудовлетворительно большим.

Более надежный способ построения сумм прямых произведений, а также и важная часть теоретического обоснования тензорных разложения проистекает из *аналитического* разделения переменных, которое, как правило, пишется в виде сходящихся рядов. Одним из самых замечательных примеров операторов, записывающихся непосредственно в виде суммы прямых произведений, является обратный оператор Лапласа [78, 79, 104, 105] и связанные с ним функции Грина [138, 142].

Сумма R прямых произведений называется *каноническим* разложением ранга R:

$$x(i_1, i_2, \ldots, i_d) = \sum_{\alpha=1}^{R} x_\alpha^{(1)}(i_1)x_\alpha^{(2)}(i_2)\cdots x_\alpha^{(d)}(i_d).$$

Кроме "жадных" методов, можно применять методы минимизации общего вида для непосредственного вычисления элементов канонических *факторов* $x^{(1)}, \ldots, x^{(d)}$, такие как метод Ньютона [154, 68, 3] или наименьших квадратов с переменными направлениями [110, 38, 32, 40, 29, 30]. Однако, в случае $R > 1$ и $d > 2$, задача оптимизации ошибки может оказаться некорректно поставленной [46]: можно построить такой тензор, и такую последовательность канонических факторов, что

ошибка аппроксимации будет стремится к нулю, тогда как сами элементы разложения будут расходится к неопределенности $\infty - \infty$.

Малоранговое *разложение матрицы* ($d = 2$) имеет существенное отличие: задача приближения матрицей малого ранга корректна, и может быть эффективно решена с использованием сингулярного разложения (Singular Value Decomposition, SVD) [88]. Сингулярное разложение обеспечивает минимизацию ошибки в евклидовой норме на множестве матриц ранга R. Более того, для него существуют очень надежные численные алгоритмы [87], оптимизированные в течение десятилетий развития библиотеки LAPACK.

В более высоких размерностях существует несколько обобщений сингулярного разложения. Одна идея заключается в вычислении сингулярных разложений независимо по каждой координате. Это дает представление *Таккера* [240]:

$$x(i_1, \ldots, i_d) = \sum_{\gamma_1, \ldots, \gamma_d = 1}^{r_1, \ldots, r_d} x^{(c)}(\gamma_1, \ldots, \gamma_d) x^{(1)}_{\gamma_1}(i_1) \cdots x^{(d)}_{\gamma_d}(i_d).$$

Заметим, что каждый *фактор Таккера* $x^{(k)}$ обладает своим ранговым индексом γ_k, в отличие от общего индекса α в каноническом представлении. Эта независимость позволяет решать задачу аппроксимации с помощью так называемого *Higher Order SVD* (HOSVD) алгоритма [43, 44, 45], который берет в качестве Таккеровских факторов наборы старших сингулярных векторов определенных матриц, связанных с исходным массивом x. Это дает вычислительную надежность и квазиоптимальное соотношение точность/ранг.

Метод наименьших квадратов с переменными направлениями также может быть использован для элементов разложения Таккера. Этот метод был разработан в основном как инструмент для вычисления регрессионных моделей в виде малорангового представления, и был предложен в [163], а затем расширен в [44, 47, 147]. Можно сказать, что HOSVD было создано с целью решения задачи *анализа главных компонент*, но применение его как общего метода *сжатия данных* долгое время рассматривалось как второстепенная цель.

Идея сжатия данных начала быстро развиваться, когда представления тензорными произведениями начали использоваться для структурированного решения многомерных уравнений в частных производных [29]. В связи с этой областью применения, был открыт ключевой эффект: для гладких функций имеет место $r \ll n$, что было проверено как численно, так и аналитически, с помощью теории полиномиальной интерполяции, см. [79, 138, 104]. Этот эффект также существует и при наличии небольшого количества разделенных особенностей [146]. Кроме того, функциональное понимание стимулировало разработку вычислительных алгоритмов, в том числе комбинированных. Например, многосеточные схемы могут давать значительное ускорение сходимости методов переменных направлений [147], для уменьшения затрат памяти в разложении Таккера было предложено смешанное каноническое-Таккеровское представление [138, 146, 147], и др. Важные результаты родились из приложений, связанных с интегральными уравнениями [8, 7, 200, 146, 142] и расчетами электронных структур [148, 147, 246, 132, 139, 236, 202, 133, 136, 135, 134, 215, 201, 89].

Ссылки, приведенные выше, рассматривают в основном трехмерные задачи. В более высоких размерностях, Таккеровское *ядро* $x^{(c)}$ все еще содержит недопустимо много $\mathcal{O}(r^d)$ элементов. В качестве альтернативы были предложены *рекуррентные* двумерные разложения. Идея заключается в следующем: мы вводим редуцированный базис по одной переменной, затем объединяем его с другим индексом, определяем редуцированный базис уже в двух переменных, и так далее. Эта процедура может быть проведена в соответствии со сбалансированным бинарным деревом, что дает так называемое *Иерархическое разложение Таккера* (Hierarchical Tucker, HT) [108, 94, 164], или вдоль линейного дерева, что дает представление в виде Tensor Train (TT) [203, 194, 197], или даже с использованием более общих тензорных сетей (Tensor Tree Networks, TTN) [71], где каждая размерность может соединяться несколькими ветвями с другими. Интересно, что TT разложение было разработано задолго до появления названия "Tensor Train" и использовалось уже для многих задач, что отражается существованием нескольких независимых названий для этой конструкции: *valence bond states* [211], *matrix product states* (MPS) [70, 152, 175] и *density matrix renormalization group* (DMRG) [249] в квантовой физике конденсированных состояний, и наконец, термин "tensor train" появился в 2009 году в численной линейной алгебре [203, 197].

В принципе, разложение Таккера может также рассматриваться как TTN с d ветвями. Мы видим, что основная идея, представление многомерной функции полилинейной комбинацией одномерных функций, является общей для всех тензорных разложений, однако конкретные правила того, как именно вычисляется исходный тензор, так называемые *тензорные форматы*, могут существенно отличаться как в формулировке, так и в численных свойствах.

Естественно ожидать, что определенный вид дерева будет наиболее эффективным для определенной задачи. Тем не менее, более сложные тензорные сети требуют и более сложных и длинных описаний. Однако, основные концепции не зависят явно от структуры дерева, и поэтому мы будем придерживаться TT представления, чтобы сделать презентацию более простой и элегантной.

TT формат может рассматриваться как промежуточный между каноническим и Таккеровским разложениями:

$$x(i_1, \ldots, i_d) = \sum_{\alpha_1, \ldots, \alpha_{d-1}=1}^{r_1, \ldots, r_{d-1}} x^{(1)}_{\alpha_1}(i_1) x^{(2)}_{\alpha_1,\alpha_2}(i_2) \cdots x^{(d-1)}_{\alpha_{d-2},\alpha_{d-1}}(i_{d-1}) x^{(d)}_{\alpha_{d-1}}(i_d).$$

Каждый *блок* $x^{(k)}$ в правой части представляет собой трехмерный массив размером $\mathcal{O}(nr^2)$, поэтому общий объем данных $\mathcal{O}(dnr^2)$ дает возможность избежать проклятие размерности, при условии, что ранг r не очень большой.

Дискретизация многомерных уравнений в частных производных может потребовать мелких сеток, что приводит к большим *модовым размерам* n. Интересно, что дальнейшее сжатие может быть достигнуто в той же самой TT концепции. Линейный вклад размерности в общую сложность TT формата наталкивает на мысль об *увеличении* числа переменных. Идея *квантизации* предлагает разбить каждый из индексов $i_1, \ldots, i_d$ на подиндексы в соответствии с простыми множителями n. Применяя TT приближение к полученному тензору с размерностью $\mathcal{O}(d \log n)$, но

небольшими модовыми размерами ($\sim$ 2–3), получаем так называемый *квантизованный TT* (Quantized TT, QTT) формат [140, 144]. Важно заметить, что оценки рангов для многих одномерных функций после дискретизации и квантизации могут быть получены аналитически [144, 198], что гарантирует логарифмический уровень сжатия данных по с равнению с первоначальным объемом. Такой же подход применим и для сжатия матриц [195].

В противоположность форматам Таккера и одномерному формату QTT, общие предположения о гладкости для многомерных функций в TT формате приводят к пессимистическим оценкам на зависимость рангов от точности, см. например, [220], где требование к объему памяти содержит член вида $\mathcal{O}(|\log\varepsilon|^d)$. Однако, в реальности ситуация обычно существенно лучше. Дополнительное сжатие и ускорение может быть получено с *комбинированными* форматами. Например, *расширенный TT* [204] представляет собой формат Таккера с TT представлением для ядра, а *QTT-Tucker* разложение [50] использует вдобавок и QTT приближение для Таккеровских факторов.

Любая тензорная структура требует численных методов для аппроксимации данных и других операций. Возможно, именно существенная направленность на практические приложения в физическом сообществе и дала нам много разносторонних вычислительных алгоритмов, особенно в TT/MPS формате. Во-первых, как и в методе HOSVD для разложения Таккера, TT представление можно вычислить с помощью сингулярных разложений с гарантированной точностью. Во-вторых, концепция переменных направлений, разработанная в квантовой физике, является существенно более мощной, чем метод наименьших квадратов с переменными направлениями для, например, формата Таккера. Алгоритмы Numerical Renormalization Group (NRG) and *Density Matrix Renormalization Group* (DMRG) широко использовались для моделирования волновых функций спиновых систем в форматах тензорных произведений с 1970-х годов [253], и с тех пор были разработаны многие впечатляющие модификации и улучшения, которые впоследствии были приняты и в обществе численного анализа [116]. Соответствующий (далеко не полный) список литературы включает в себя, например, [249, 250, 206, 251, 126, 247, 252, 245, 207, 221, 222]. Эффективность методов переменных направлений проистекает из *линейности* тензорных форматов по отношению к фиксированному блоку элементов. Таким образом, задачи аппроксимации, решения линейной системы или задачи на собственные значения, сформулированные в терминах исходных тензоров, редуцируются на элементы тензорного формата в той же формулировке, и следовательно могут быть решены с использованием стандартных методов. Это резко отличается от одновременной оптимизации всех элементов формата сразу, которая может быть существенно нелинейной и невыпуклой.

Тем не менее, последнее явление снижает надежность также и методов переменных направлений. Даже самые простые функции ошибки вида $\|x - x_\star\|^2$, по отношению к элементам тензорного формата для x, могут иметь многочисленные локальные минимумы. Оптимизация по переменным направлениям является быстрой, но это также оказывается и недостатком. Поскольку на каждом шаге рассматривается лишь часть формата, простые линейные схемы переменных направлений и алгоритмы DMRG с высокой вероятностью возвращают локальный

но не глобальный минимизатор ошибки. Во многих случаях это нежелательный результат, так как мы хотели бы решить первоначальную физическую задачу, поставленную в многомерном пространстве, с достаточной степенью точности приближения глобального решения. Это побуждает нас принять во внимание еще один подход.

Когда формат имеет надежную процедуру аппроксимации, которая позволяет сжимать любые данные с квази-оптимальным объемом памяти для заданного порога точности, можно думать о реализации классических итерационных алгоритмов с приближенной тензорной *арифметикой*, например, методов решения линейных систем [8, 107, 201, 160, 150, 137, 141, 161, 13, 52, 16] или частичных задач на собственные значения [168, 162, 118, 243, 173]. Мы можем думать о тензорных форматах таким же образом, как о числах в компьютере: сложения и умножения начальных данных переписываются для блоков тензорного формата, эти операции обычно увеличивают количество элементов, а затем аппроксимация, как и численное *округление*, обеспечивает ограниченность объема памяти.

Однако, вопрос заключается в том, какой именно объем памяти необходим. Для чисел, фиксированная длина мантиссы всегда обеспечивает гарантированную точность. Это не так в случае с методами тензорных произведений: объем памяти зависит как от точности, так и от слабо формализуемой "структуры" данных. К сожалению, классические (например, Крыловские) методы существенно основаны на достаточно точном представлении невязки, и страдают от того же эффекта, что и "жадные" методы: чем ближе решение к точному, тем сложнее структура ошибки и других вспомогательных векторов, которые требуют либо грубого приближения, либо большого объема памяти для хранения их тензорных форматов. Стратегии релаксации, разработанные в теории неточных Крыловских методов [224], улучшают ситуацию до некоторой степени за счет возможности огрубления точности на последних итерациях, но все равно такие методы далеки от надежных для достаточно широкого класса задач.

Цели диссертационной работы

1. Разработка нового вычислительного метода для решения больших систем линейных уравнений с представлением данных в формате тензорных произведений. Теоретический анализ его свойств сходимости.

2. Применение и проверка нового метода на задачах расчета стохастической химической кинетики (основное кинетическое уравнение) и задаче моделирования Фарлей-Бунемановской неустойчивости в плазме ионосферы Земли (уравнение Власова).

3. Анализ результатов, сравнение нового алгоритма с ранее существующими техниками, сравнение модельных численных данных с имеющимися в литературе экспериментами.

Основные положения диссертации

Главным результатом данной диссертации является вычислительный метод, который сочетает в себе сильные стороны как оптимизационных тензорных алгоритмов переменных направлений, так и классических приближенных итерационных схем [56, 57]. В процессе DMRG итерации, мы *расширяем* тензорный формат решения с помощью тензорного формата приближенной невязки. Это обеспечивает замечательную взаимную поддержку классических методов и DMRG методов переменных направлений. Так как решение адаптируется в процессе DMRG оптимизации по элементам формата, даже очень грубые приближения невязки (причем без последующих Крыловских векторов) дают высокую точность решения. С другой стороны, подключение информации о глобальной невязке в локальных шагах процесса переменных направлений обеспечивает последний правильными градиентными направлениями, помогая ему избегать локальных минимумов. Как уже отмечалось, стагнации в локальных минимумах являются широко известной проблемой вариационных методов на тензорных многообразиях. Напротив, новый AMEn метод (Alternating Minimal Energy) обладает доказанной геометрической сходимостью с точки зрения глобальных элементов тензора, аналогично методу градиентного спуска. Название AMEn было принято в результате следующей цепочки соображений:

$$\text{Minimal Residual} \;\rightarrow\; \begin{array}{c}\textbf{Minimal Energy}\\ \text{(steepest descent)}\end{array} \;\rightarrow\; \begin{array}{c}\text{Alternating}\\ \text{Minimal Energy}\end{array}$$
$$\begin{array}{c}\textbf{Alternating}\\ \text{Least Squares}\end{array} \;\nearrow$$

Примечательно, что практическая скорость сходимости оказывается намного быстрее, чем теоретические оценки на основе градиентного спуска, что делает этот метод надежным даже для несимметричных линейных систем, наподобие метода полной ортогонализации, см. [213].

Приложения, рассматриваемые в этой диссертации, отвечают нашей дискуссии о случайно блуждающей частице и соответствующем распределении вероятностей. Мы применяем методы тензорных произведений для дифференциальных и разностных уравнений, определяющих функции вероятности.

Первым приложением является *уравнение Власова*, возникающее в гибридной модели *Фарлей-Бунемановской неустойчивости* в плазме ионосферы земли (см. раздел 1.1). Плазма состоит из электронов и положительно заряженных ионов. Условия в E-слое (90–130 км) ионосферы (величины фонового электрического поля и магнитного поля земли) позволяют использовать так называемое жидкостное уравнение на плотность концентрации электронов, представляющее собой обычное двух- или трехмерное уравнение диффузии-адвекции. Мы рассматриваем конкретные параметры, используемые в предыдущей работе [5], которые позволяют ограничиться двумя пространственными координатами (в плоскости перпендикулярной магнитному полю). Однако, поведение ионов требует кинетического описания, которое дается четырехмерным уравнением Власова (две пространственные, две скоростные координаты). Решение этого уравнения является функцией распределения плотности ионов по скоростям в каждой пространственной точке, и

требует очень больших объемов памяти при непосредственном хранении. Тем не менее, динамика распределения не сильно далеко уходит от максвелловского. Это дает возможность разделения переменных с помощью TT формата с разумными TT рангами, и сокращения затрат памяти в десятки раз.

Другое приложение имеет решающую роль в высокоточном стохастическом моделирования клеточных явлений, жизненных циклов вирусов и других микробиологических процессов, где случайные флуктуации вносят существенный вклад. Это возникает в случае, когда количество реагирующих молекул очень мало по сравнению с объемом всей системы, и столкновения между молекулами происходят в случайные моменты времени. В результате кинетика подчиняется дискретному марковскому процессу, который опять же может быть описан многомерной функцией вероятности. В отличие от уравнения Власова, конфигурационные состояния здесь с самого начала дискретные, и функция вероятности подчиняется разностному *основному кинетическому уравнению* [244], см. раздел 1.2.

Обе упомянутые модели описывают эволюцию функции вероятности во времени. Таким образом, в качестве третьего приложения, мы покажем, как *схемы интегрирования по времени*, например, Эйлера или Кранка-Николсон, могут быть ускорены методами тензорных произведений. Мы рассматриваем переменную времени в качестве дополнительного измерения. Затем используем QTT формат, что дает логарифмическую вычислительную сложность в зависимости от количества временных шагов, см. раздел 1.3.

Представленные задачи в конечном итоге сводятся к большим линейным системам. Мы выведем несколько аналитических тензорных представлений, показывая, что матрицы и правые части в этих системах могут быть предоставлены или эффективно приближены в тензорных форматах. Остается вопрос, как теперь вычислить их решения. В численных экспериментах мы демонстрируем, что новый итерационный алгоритм AMEn является более быстрым и точным, чем ранее известные методы, и, наконец, может претендовать на роль многоцелевого инструмента для различных задач, которые в принципе подходят для концепции разделения переменных.

Достоверность научных положений и методология

Поскольку результатом диссертации является математический инструмент (алгоритм), основными способами оценки достоверности работы и результатов являются математические методы. Так, для подтверждения сходимости алгоритма использовалось измерение текущей невязки, что известным образом дает информацию о корректности решения, а также сравнение самих решений, полученных в результате запусков алгоритмов с разными допусками точности. В тех примерах прикладных задач, для которых в литературе были доступны результаты экспериментов или моделирования другими методами, проводилось сравнение их с расчетами, проведенными автором. Проведенные исследования подтверждают корректность и применимость новых предложенных методов, при их меньших вычислительных затратах.

Научная новизна

Предложенный метод AMEn является первым алгоритмом для решения системы уравнений в представлении тензорными произведениями, который обладает теоретически доказанной оценкой глобальной сходимости, и при этом является эффективным на практике. В основе AMEn алгоритма лежит сочетание метода градиентного спуска и метода переменных направлений (DMRG). Однако, в отдельности метод градиентного спуска сходится весьма медленно и требует достаточно точной аппроксимации невязки (градиента), что приводит к большой вычислительной сложности. Отдельные шаги методов DMRG весьма эффективны, но итоговый результат может давать очень плохое приближение для точного решения.

С помощью нового метода было впервые получено полное решение основного кинетического уравнения для λ-фага с высокой точностью. Классический способ аппроксимации решения основного кинетического уравнения состоит в усреднении большого набора случайных реализаций [81]. Он хорошо применим в случаях, когда требуется невысокая точность. С ростом числа реализаций M метод сходится с достаточно медленной скоростью $\mathcal{O}(M^{-1/2})$, что отрицательно сказывается на вычислительной сложности в высокоточных расчетах. В качестве альтернативы, были предложены и методы тензорных представлений. Однако ранее существовавшие алгоритмы (например [123]) позволяли моделировать только небольшие промежутки времени, что непригодно для реальных задач в системной биологии.

Новый метод также принес существенных вклад в ускорение расчета Фарлей-Бунемановской неустойчивости, позволив провести моделирование четырехмерного уравнения Власова на персональном компьютере, тогда как затраты памяти на хранение решения в виде обычного четырехмерного массива неизбежно требовали применения распределенных высокопроизводительных систем.

Научная и практическая значимость полученных результатов

Предложенный подход может применяться в различных задачах моделирования стохастических и квантовых систем как метод достаточной степени общности. Так, для того, чтобы проверить некоторый новый случай системы уравнений, нужно уметь представлять матрицу и правую часть в тензорных произведениях, но сам алгоритм ее решения не требует изменений. Поэтому, даже при наличии специализированных эффективных методов для конкретных задач, методы тензорных представлений могут использоваться для проверочных целей.

Другой особенностью малоранговых аппроксимаций являются весьма умеренные требования к памяти, хотя число операций, требуемых для решения задачи, может быть большим. Примером является моделирование Фарлей-Бунемановской неустойчивости в данной диссертации, где вычислительные времена с использованием стандартной схемы расщепления и разделения переменных являются качественно сравнимыми, но затраты памяти отличаются на полтора порядка. Важное применение предложенных методов ожидается в задачах магнитного резонанса и квантовом управлении. Предварительные результаты [69] моделирования ядерного магнитного резонанса белков (задача сводится к расчету квантовых конфигу-

раций спиновых систем) показывают высокую эффективность представлений тензорными произведениями и AMEn алгоритма для решения уравнений с большим количеством переменных.

Апробация результатов диссертации и публикации

Основные результаты работы апробированы в следующих докладах на конференциях и семинарах.

[1] Solution of the chemical master equation by the separation of variables and alternating optimization methods. *European Conference on Mathematical and Theoretical Biology*, Gothenburg, June 16, 2014.

[2] A new family of the alternating linear optimization schemes in tensor product representations. *Seminar of the group Computational Methods in Systems and Control Theory*, Max Planck Institute Magdeburg, March 04, 2014.

[3] Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems. *ENUMATH Conference*, EPFL Lausanne, August 26-30, 2013.

[4] Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems. *MAFELAP*, Brunel University, London, June 11, 2013.

[5] Fast adaptive alternating linear schemes in higher dimensions. Part 2: eigenproblems. *NASCA Conference*, University of Calais, June 24, 2013.

[6] Fast adaptive tensor product approach to eigenvalue problems in higher dimensions. *Seminar of the Department of Chemistry*, University of Southampton, June 27, 2013.

[7] Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems. *Swiss numerics colloquium*, EPFL, Lausanne, April 05, 2013.

[8] Fast adaptive alternating linear solvers. Implementation hints and application to Fokker-Planck and master equations. *Workshop on algorithms for high-dimensional problems in quantum chemistry*, University Southampton, February 26, 2013.

[9] Alternating minimal residual methods for linear systems in higher dimensions. Part II: heuristics and experiments. *29 GAMM Seminar on Uncertainty Quantification*, MPI MIS, Leipzig, January 22, 2013.

[10] Advanced tensor representation and solution techniques with application to Fokker-Planck and master equations. *CMAM-5*, Humbolt University, Berlin, August 16–18, 2012.

[11] Advantages and difficulties of use of tensor methods in solution to the Fokker-Planck equation. *28 GAMM Seminar on Analysis and Numerical Methods in Higher Dimensions*, MPI MIS, Leipzig, January 16–18, 2012.

[12] A gray-box DMRG algorithm for tensor structured solution to linear systems. *17th Conference of the International Linear Algebra Society*, University Braunschweig, August 26, 2011.

[13] On a solution to a parabolic equation in the QTT format using the DMRG approach. *4th Workshop on High-Dimensional Approximation*, University Bonn, June 27, 2011.

[14] Use of the DMRG scheme for structured linear system solution. *3rd International Conference on Matrix Methods in Mathematics and Applications*, ИВМ РАН, Москва, 24 июня, 2011.

[15] Linear solvers in TT formats. *27th GAMM-Seminar on Approximation of Multiparametric functions*, MPI MIS, Leipzig, January 26–28, 2011.

[16] TT-GMRES: о решении систем линейных уравнений в тензорных форматах. *53 научная конференция МФТИ*, Москва, Ноябрь 26–28, 2010.

По результатам работы опубликовано 7 статей в международных рецензируемых журналах: [41, 59, 51, 52, 50, 53, 55], и 1 в материалах конференций [58].

Содержание диссертации

Диссертация состоит из введения, пяти глав, заключения, списка обозначений и списка литературы. Текст изложен на 161 странице, содержит 25 рисунков. Список литературы включает 253 наименования.

В главе 1 мы описываем три основных приложения: уравнение Власова для Фарлей-Бунемановской неустойчивости, основное кинетическое уравнение для стохастической химической кинетики, и схема одновременной дискретизации динамических уравнений в пространстве и времени.

Глава 2 посвящена описанию различных методов разделения переменных. Дается обзор существующих тензорных разложений: канонического (CP), Таккеровского формата, представлений иерархический Таккер и MPS/TT, а также их свойств и связанных с ними алгоритмов. В конце главы предлагается новое комбинированное тензорное представление (QTT-Tucker), объединяющее аналитические и практические преимущества Таккеровского, TT и QTT форматов.

В главе 3 мы объединяем прикладные задачи, изложенные в первой главе, и методы тензорных представлений из второй главы. Мы выведем несколько явных разложений низкого TT/QTT ранга для матриц и векторов (тензоров), имеющих

отношение к дифференциальным и разностным операторам и некоторым типичным функциям в рассматриваемых приложениях. Среди них, например, матрицы конечных разностей (дискретные градиенты), операторы попарных взаимодействий, и матрицы перехода в явных схемах расщепления по времени.

В Главе 4 изложены методы решения линейных уравнений в тензорных форматах: классические итерационные методы с тензорными округлениями, итерационные методы оптимизации переменных направлений, разработанные в квантовой физике (DMRG) и вычислительной математике (метод наименьших квадратов в переменных направлениях), в также новый улучшенный алгоритм, которая дополняет схему переменных направлений классическим градиентным шагом с использованием глобальной невязки системы. Мы анализируем его сходимость, доказываем геометрическую скорость сходимости, и обсуждаем некоторые практические аспекты.

В Главе 5 мы исследуем все вышеизложенное на практике. Рассмотрены несколько важных биологических систем, описываемых с помощью основного кинетического уравнения, а также гибридная многомерная модель плазменной неустойчивости, исследованы их особенности, связанные с концепцией тензорных произведений, проведено сравнение вычислительных алгоритмов. Данными численными экспериментами подтверждено, что тензорные форматы и методы применимы в качестве эффективных и высокоточных инструментов для обсуждаемых приложений.

В заключении, мы обобщаем основные моменты, наблюдения, и возможные пути дальнейшего развития области структурированных тензорных вычислений.

Благодарности

Эта диссертация содержит основные результаты моей научной работы с 2011 по 2014 год. Прежде всего, я хотел бы выразить благодарность моему научному руководителю, профессору Евгению Евгеньевичу Тыртышникову за внимательное руководство и неизменную поддержку в институте вычислительной математики.

Я глубоко признателен моим коллегам Ивану Валерьевичу Оселедцу и Дмитрию Валерьевичу Савостьянову за очень плодотворное сотрудничество и обмен идеями. Их высокий профессионализм и поддержка были неоценимы в процессе исследования многомерных областей математики.

Я особо благодарен гостеприимству математического института им. Макса Планка в Лейпциге, и лично профессору Борису Николаевичу Хоромскому за вдохновляющую атмосферу и уникальные условия, великолепно способствующие научной деятельности.

Я высоко ценю замечательное и дружелюбное сотрудничество, установившееся с нашими коллегами в различных группах, занимающихся как прикладными задачами, так и разработками вычислительных методов, в особенности профессору Илье Купрову, University of Southampton, профессору Александру Павловичу Смирнову, факультет вычислительной математики и кибернетики МГУ, профессору Геннадию Алексеевичу Бочарову, институт вычислительной математики РАН, профессорам Томасу Шульте-Хербрюггену и Томасу Хукле, Technical University

Munich, профессорам Удо Райхлю и Питеру Беннеру, институт сложных динамических систем им. Макса Планка, Магдебург. Объективная оценка текущей работы и дальнейших направлений исследований были бы невозможны без мотивирующих дискуссий о последних математических инструментах и интригующих практических приложениях.

Глава 1

Многомерные вероятностные уравнения

1.1 Модель Фарлей-Бунемановской неустойчивости в ионосфере Земли

В данной главе мы описываем модель Фарлей-Бунемановской неустойчивости, предложенную в [5], и использованную также в работе автора [59] по применению тензорных представлений к данной модели.

Фарлей-Бунемановская (ФБ) неустойчивость возникает в слабо ионизированной плазме E-области ионосферы Земли. Эта неустойчивость порождается в плазме с замагниченными электронами и незамагниченными ионами в электрическом поле, направленном перпендикулярно геомагнитному полю [48]. В E-области электроны подвержены заметному влиянию геомагнитного поля, в отличие от ионов, которые не замагничиваются из-за частых столкновений с нейтральными частицами газа. В результате, посредством скорости дрейфа, обусловленной электрическим полем, распределение электронов по скоростям сдвигается относительно ионного распределения. Соответствующие условия для возникновения неустойчивости возникают в экваториальных и полярных зонах E-области ионосферы Земли на высотах порядка 90–130 км, где нестабильность проявляется как низкочастотные колебания плазмы с длинами волн порядка метра.

Первые работы, которые исследовали Фарлей-Бунемановскую неустойчивость, были независимо опубликованы Фарлеем [72] и Бунеманом [35]. Они использовали линейную теорию. Используя кинетические уравнения, Фарлей показал, что сильное внешнее электрическое поле приводит к нестабильности и появлению волн в плазме. Бунеман с помощью жидкостной теории получил дисперсионное соотношение, которое показывает, что рост нестабильности возможен только тогда, когда скорость дрейфа электронов превышает некоторый порог, т.е. внешнее электрическое поле должно быть достаточно большим.

Линейная теория позволяет вывести пороговые оценки, давая необходимые условия для развития неустойчивости, но она неприменима для описания процесса насыщения. Последний может быть проанализирован только на базе нелинейной теории, которая разрабатывалась в течение длительного времени [225, 233, 109],

но все еще имеет ограниченное применение.

Нелинейные модели ФБ неустойчивости основаны на нелинейных двух- и трехмерных уравнениях в частных производных. Количественное решение этих уравнений Фарлей-Бунемановской неустойчивости требует проведения компьютерного моделирования, по результатам которого можно оценивать значимость и применимость тех или иных теорий.

Первые компьютерные моделирования ФБ неустойчивости были основаны на жидкостной теории [185]. С развитием компьютеров были предложены более продвинутые модели и методы. Так, в [174, 218, 125, 189, 192, 193] был использован метод частиц, а комбинированный метод на основе метода частиц и жидкостных уравнений был применен в [190, 191, 170]. Использование жидкостной модели и для электронов, и для ионов приводит к нефизичному результату: темп роста неустойчивости бесконечно увеличивается с ростом волнового числа.

В противоположность этому, кинетическое затухание Ландау стабилизирует коротковолновые компоненты. И электроны, и ионы подвержены затуханию Ландау, что приводит к подавлению неустойчивости, но электронное затухание действует только в коротковолновом, высокочастотном диапазоне, и одного ионного затухания Ландау достаточно, чтобы эффективно сдерживать рост волн. Это позволяет использовать для электронов упрощенную жидкостную модель.

В этой диссертации мы используем гибридную модель для ФБ неустойчивости, предложенную в [158, 156, 157, 155]. Модель основана на следующих уравнениях: двумерное жидкостное уравнение для электронной плотности, четырехмерное кинетическое уравнения для ионов, и двумерное уравнение Пуассона для потенциала электрического поля. Рост и насыщение Фарлей-Бунемановского процесса в гибридной модели с использованием дискретизации на многомерных сетках было численно исследовано в [158]. Большую часть компьютерного времени занимало численное решение ионного кинетического уравнения на четырехмерной сетке в фазовом пространстве (2D-пространство, 2D-скорость). Размер четырехмерного массива, содержащего решение кинетического уравнения, достигает 10^9–10^{12} байт. Такие высокие требования к оперативной памяти, а также соответствующее количество компьютерных операций на каждом шаге по времени, обязывали использовать высокопроизводительные параллельные системы. Моделирование ФБ неустойчивости, описанное в [158], было реализовано в таком параллельном программном коде для суперкомпьютера Blue-Jean P.

В качестве альтернативы, в данной работе мы используем аппроксимации тензорными произведениями для вычисления и хранения четырехмерного кинетического уравнения, а именно, разделение пространственных и скоростных переменных. В итоге, объем требуемой памяти может быть сокращен в 20 и более раз, и вычисления становятся возможными на персональном компьютере с помощью последовательного MATLAB кода.

1.1.1 Постановка задачи

В принципе, математическое описание плазмы в весьма общем случае может быть дано уравнением Власова-Фоккера-Планка [11]. Однако, условия E-области в ионо-

сфере позволяют ввести некоторые упрощения, такие как жидкостную модель для электронов и оператор столкновений BGK для ионов.

Координатные оси вводятся параллельно электрическому и магнитному полю, а именно:

$$\begin{aligned}\mathbf{B}_0 &= \begin{bmatrix}0 & 0 & B_0\end{bmatrix}\\ \mathbf{E}_0 &= \begin{bmatrix}0 & E_0 & 0\end{bmatrix}\\ \mathbf{V}_0 &= \begin{bmatrix}V_0 & 0 & 0\end{bmatrix} = \mathbf{E}_0 \times \frac{\mathbf{B}_0}{B_0^2},\end{aligned}$$

где геомагнитное поле обозначено как $\mathbf{B}_0$, внешнее электрическое поле как $\mathbf{E}_0$, и $\mathbf{V}_0$ это скорость дрифта.

Рассматриваемый нами диапазон параметров (в частности, движущее поле E_0) позволяет нам использовать двумерную модель только в плоскости x, y, и в предположении постоянной температуры для электронов. Во-первых, предположение об изотермических электронах является допустимым для слабых (по отношению к порогу Фарлей-Бунемана) электрических полей, когда нагрев электронов выражен несущественно. Во-вторых, в этом случае волновые векторы Фарлей-Бунемановской неустойчивости в значительной степени лежат перпендикулярно магнитному полю, так что вклад z-координаты в дополнительное электрическое поле примерно на два порядка меньше, чем x, y компоненты (см. [5]).

Жидкостная модель для электронов

Мы предполагаем, что поведение электронов регулируется стандартными уравнениями непрерывности:

$$\begin{aligned}\frac{\partial n_e}{\partial t} &= -\nabla(n_e \mathbf{V}_e),\\ m_e \frac{d\mathbf{V}_e}{dt} &= -e(\mathbf{E}_0 - \nabla\Phi + \mathbf{V}_e \times \mathbf{B}_0) - \frac{\nabla(n_e T_e)}{n_e} - m_e \mathbf{V}_e \nu_{en},\end{aligned} \tag{1.1}$$

где $n_e = n_e(x, y)$ обозначает концентрацию электронов, $\mathbf{V}_e$ является полной скоростью электрона, Φ это электрический потенциал, T_e – температура электронов (о влиянии эффекта нагрева электронов см. [156]), ν_{en} – средняя частота столкновений электронов с нейтральными частицами, и m_e и e обозначают массу и заряд электрона, соответственно.

Мы будем обезразмеривать все величины на характерные масштабы модели, как показано в Таблице 1.1. Помимо этого, введем следующие обобщающие безразмерные величины:

$$\gamma = \frac{\nu_{en}\nu_{in}}{\Omega_e \Omega_i}, \qquad \theta = \left(\frac{m_e \nu_{en}}{m_i \nu_{in}}\right)^{1/2}, \tag{1.2}$$

где ν_{in} это средняя частота столкновений ионов с нейтральными частицами, за $\Omega_{e,i}$ обозначены циклотронные частоты электронов и ионов, соответственно, и m_i обозначает массу иона. Так как в E-области выполняется $\omega \ll \nu_{en}$, где ω это плазменная частота, мы можем пренебречь инерцией электронов в (1.1). Принимая во внимание все соображения, представленные выше, уравнение для плотности

Таблица 1.1: Характерные масштабы и правила обезразмеривания

Исходная величина		Масштаб		Безразмерная величина
Время	t	$1/\nu_{in}$	[с]	$t := t\nu_{in}$
Пространство(x)	x	$l = v_{T_i}/\nu_{in}$	[м]	$x := x/l$
Пространство(y)	y	$l = v_{T_i}/\nu_{in}$	[м]	$y := y/l$
Скорость(x)	v	$v_{T_i} = \sqrt{T_i/m_i}$	[м/с]	$v := v/v_{T_i}$
Скорость(y)	w	$v_{T_i} = \sqrt{T_i/m_i}$	[м/с]	$w := w/v_{T_i}$
Температура(эл.)	T_e	T_i	[Дж]	$T_e := T_e/T_i$
Эл. потенциал	Φ	T_i/e	[В]	$\phi = e\Phi/T_i$

электронов можно записать следующим образом:

$$\begin{aligned} \frac{1}{\gamma\sqrt{T_e}}\frac{\partial n_e}{\partial t} &= \Delta(T_e n_e) + \frac{\partial}{\partial x}\left(\frac{V_0}{v_{T_i}\gamma\sqrt{T_e}} - \frac{1}{\theta\sqrt{T_e\gamma}}\frac{\partial\phi}{\partial y} - \frac{\partial\phi}{\partial x}\right) n_e \\ &+ \frac{\partial}{\partial y}\left(\frac{eE_0 v_{T_i}}{T_i\nu_{in}} + \frac{1}{\theta\sqrt{T_e\gamma}}\frac{\partial\phi}{\partial x} - \frac{\partial\phi}{\partial y}\right) n_e, \end{aligned} \tag{1.3}$$

где ϕ это безразмерный электрический потенциал, возникающий из-за неравномерного распределения плотностей электронов и ионов (см. Таблицу 1.1), и T_i это начальная температура ионов. Обратите внимание, что температура измеряется в Джоулях, т.е. $T_* = k\hat{T}_*$ если $\hat{T}_*$ задана в Кельвинах.

Поскольку на больших масштабах мы предполагаем пространственную однородность плазмы, уравнение (1.3) решается в квадратной области $[0, L]^2$ с периодическими граничными условиями.

Уравнение Пуассона для электростатического потенциала

Принимая во внимание изотермичность электронов, мы переходим сразу к формулировке модели на потенциал электрического поля.

Пусть задано распределение заряда ρ, тогда электрический потенциал удовлетворяет уравнению Пуассона,

$$\Delta\Phi = \frac{1}{\varepsilon_0}\rho.$$

В нашем случае, ρ появляется из-за разделения зарядов между электронами и ионами, $\rho = e(n_e - n_i)$. Таким образом, после обезразмеривания, получим окончательное уравнение

$$\Delta\phi = \frac{e^2}{\varepsilon_0 m_i \nu_{in}^2}(n_e - n_i). \tag{1.4}$$

Как и в предыдущем случае, мы ставим периодические граничные условия на $[0, L]^2$.

Кинетическое описание ионов

В отличие от электронов, для ионов требуется модель, учитывающая распределение по скоростям, поскольку мы не можем пренебречь ионным затуханием Лан-

дау. Полный Больцмановский интеграл столкновений, описывающий столкновения ионов с нейтральными частицами, очень сложен и требует упрощения. В соответствии с [158, 156, 157, 155], в ионном кинетического уравнения мы будем использовать упрощенный Bhatnagar-Gross-Krook (BGK) член для моделирования столкновений. Таким образом, уравнение Власова-BGK пишется следующим образом:

$$\frac{\partial\psi(x,y,v,w,t)}{\partial t} + \mathbf{v}\cdot\nabla_{x,y}\psi + \frac{e(\mathbf{E}_0 - \nabla\Phi)}{m_i}\cdot\nabla_{v,w}\psi = -\nu_{in}(\psi - \psi_0),$$

где ψ_0 это распределение нейтральных частиц, которое предполагается максвелловским по отношению к скоростям, и $\mathbf{v} = \begin{bmatrix} v & w \end{bmatrix}$ обозначает вектор скорости, соответствующий пространственному вектору $\begin{bmatrix} x & y \end{bmatrix}$. В безразмерных величинах получаем

$$\frac{\partial\psi}{\partial t} + v\frac{\partial\psi}{\partial x} + w\frac{\partial\psi}{\partial y} - \frac{\partial\phi}{\partial x}\frac{\partial\psi}{\partial v} + \left(\frac{eE_0}{m_i v_{T_i}\nu_{in}} - \frac{\partial\phi}{\partial y}\right)\frac{\partial\psi}{\partial w} = \psi_0 - \psi, \tag{1.5}$$

где

$$\psi_0 = \frac{n_i}{2\pi}\exp\left(-\frac{v^2+w^2}{2}\right), \qquad n_i = \int\limits_{\mathbb{R}^2}\psi(x,y,v,w)dvdw \tag{1.6}$$

определяет правило вычисления нейтрального распределения: оно соответствует распределению плотности ионов в пространстве, но предполагается максвелловским по скоростям.

Обработка результатов модели

Есть несколько наблюдаемых величин, которые могут быть предсказаны нашей моделью и проверены экспериментально. Во-первых, полное электрическое поле вычисляется как

$$\mathbf{E}_{tot} = \mathbf{E}_0 + \nabla\Phi,$$

и *добавочное* поле $\mathbf{E}_{add} = \nabla\Phi$, возникающее в результате неравномерно дрейфующих зарядов можно рассматривать как возмущение к исходному *сигналу* $\mathbf{E}_0$ (например, окружающему электрическому полю). Поэтому в качестве выходных параметров модели интересно проследить среднюю величину дополнительного поля,

$$E_{add} = \sqrt{(\partial\Phi/\partial x)^2 + (\partial\Phi/\partial y)^2} = \frac{T_i}{le}\sqrt{(\partial\phi/\partial x)^2 + (\partial\phi/\partial y)^2}$$

(в последнем выражении x и y безразмерные).

Поскольку дополнительное поле изменяется в пространстве и времени, еще одной характеристикой являются главные длины волн. Как обсуждалось в [158, 5], несмотря на то, что начальное поле $\mathbf{E}_0$ направлено вдоль оси y, вектор дрейфа поворачивается с развитием процесса Фарлей-Бунемана. Пространственные спектральные интенсивности вычисляются как квадрат модуля двумерного преобразования Фурье добавочного электрического поля $\nabla\Phi$,

$$\hat{E}^2(k_x,k_y) = |\hat{E}_x|^2 + |\hat{E}_y|^2, \quad \hat{E}_i(k_x,k_y) = \int E_i(x,y)\mathrm{e}^{-\mathrm{i}k_x x}\mathrm{e}^{-\mathrm{i}k_y y}dxdy.$$

1.1.2 Дискретизация по пространству и скоростям

Для пространственных переменных x, y, а также для скоростей v, w в уравнении (1.5) мы используем разностные схемы, принимая во внимание периодические граничные условия. Хотя областью определения для скоростей является вся плоскость, на практике функция распределения ψ в (1.5), будучи возмущенным распределением Максвелла, быстро убывает с ростом v и w, и мы можем ограничить область до квадрата $(v, w) \in [-v_{max}, v_{max}]^2$, накладывая периодические граничные условия.

Для краткости, мы будем записывать разностные схемы с помощью матричных произведений. Нам понадобятся следующие $n \times n$ матрицы:

$$G_n = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & -1 & 1 \\ 1 & 0 & \cdots & 0 & -1 \end{bmatrix}, \quad L_n = G_n^\top G_N = \begin{bmatrix} 2 & -1 & 0 & \cdots & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & 0 & \cdots & -1 & 2 \end{bmatrix}. \quad (1.7)$$

Кроме того, под I_n мы будем понимать единичную матрицу размером n, а символ $\otimes$ будет обозначать Кронекеровское произведение, $A \otimes B = [AB_{i,j}]_{i,j=1}^n$.

Диффузия электронной плотности

Для члена $T_e \Delta n_e$ в (1.3) мы используем дискретизацию оператора Лапласа на 5-точечном равномерном шаблоне методом конечных разностей с числом узлов по каждой переменной n_x. С учетом периодических граничных условий, получаем симметричную неположительно определенную матрицу

$$\Delta_h = -\frac{\gamma (T_e)^{3/2}}{h_x^2} \left(L_{n_x} \otimes I_{n_x} + I_{n_x} \otimes L_{n_x} \right), \quad h_x = \frac{L}{n_x},$$

которая, к тому же, является двухуровневым циркулянтом. Поэтому она легко диагонализуется двумерным дискретным преобразованием Фурье. Таким образом, не составляет труда вычислить любую матричную функцию от Δ_h и применить ее к вектору. Например, нам потребуется матричная экспонента:

$$\Delta_h = (F \otimes F)\mathrm{diag}(\lambda)(F^* \otimes F^*), \quad \exp(\delta t \Delta_h) n_e = (F \otimes F)\mathrm{diag}(\exp(\delta t \lambda))(F^* \otimes F^*) n_e.$$

Эта операция требует четыре преобразования Фурье сложности $\mathcal{O}(n_x^2 \log n_x)$ и одно умножение на диагональную матрицу со сложностью $\mathcal{O}(n_x^2)$. Неположительность Δ_h и использование матричной экспоненты гарантирует абсолютную устойчивость шага диффузии.

Конвекция электронной плотности

Для стадии конвекции электронов мы используем схему дискретизации Мак-Кормака второго порядка точности в пространстве и времени. С использованием покоординатного расщепления, первый член конвекции в (1.3) может быть представлен

в виде

$$\frac{\partial n_e}{\partial t} + \frac{\partial U(\phi) n_e}{\partial x} = 0, \quad U(\phi) = -\gamma\sqrt{T_e}\left(\frac{V_0}{v_{T_i}\gamma\sqrt{T_e}} - \frac{1}{\theta\sqrt{T_e\gamma}}\frac{\partial\phi}{\partial y} - \frac{\partial\phi}{\partial x}\right), \tag{1.8}$$

и аналогично для y. Введем такую же равномерную пространственную сетку, как и для дискретизации диффузионной части: $x_i = ih_x$, $y_j = jh_x$, $h_x = L/n_x$, и полагаем $(n_e)_{i,j} = n_e(x_i, y_j)$. Теперь, пусть задан временной шаг δt и приближение концентрации электронов на предыдущем слое по времени $n_e^0 = n_e(t)$, один шаг Мак-Кормака пишется следующим образом:

$$\begin{aligned}
(\bar{n}_e)_{i,j} &= (n_e^0)_{i,j} - \frac{\delta t}{h_x}\left(U_{i+1,j}(\phi)(n_e^0)_{i+1,j} - U_{i,j}(\phi)(n_e^0)_{i,j}\right), \\
(n_e^1)_{i,j} &= \frac{1}{2}\left((n_e^0)_{i,j} + (\bar{n}_e)_{i,j} - \frac{\delta t}{h_x}\left(U_{i,j}(\phi)(\bar{n}_e)_{i,j} - U_{i-1,j}(\phi)(\bar{n}_e)_{i-1,j}\right)\right),
\end{aligned}$$

после чего мы берем n_e^1 за приближение концентрации электронов на следующем слое по времени, $n_e(t+\delta t) \approx n_e^1$.

В матричных произведениях, схему можно записать как $n_e^1 = M_x^{ec}(\phi, \delta t) n_e^0$, где *матрица перехода* выражается формулой

$$\begin{aligned}
M_x^{ec}(\phi, \delta t) = I_{n_x^2} &+ \frac{\delta t}{2h_x}\left((G_{n_x}^\top - G_{n_x}) \otimes I_{n_x}\right)\operatorname{diag}(U(\phi)) \\
&- \frac{\delta t^2}{2h_x^2}(G_{n_x}^\top \otimes I_{n_x})\operatorname{diag}(U(\phi))(G_{n_x} \otimes I_{n_x})\operatorname{diag}(U(\phi))
\end{aligned}$$

с использованием (1.7), и $\operatorname{diag}(U)$ обозначает $n_x^2 \times n_x^2$ диагональную матрицу со значениями $U(x_i, y_j)$ на диагонали. Аналогично, для конвекции по y можно обозначить

$$V(\phi) = -\gamma\sqrt{T_e}\left(\frac{eE_0 v_{T_i}}{T_i \nu_{in}} + \frac{1}{\theta\sqrt{T_e\gamma}}\frac{\partial\phi}{\partial x} - \frac{\partial\phi}{\partial y}\right), \tag{1.9}$$

и построить матрицу перехода

$$\begin{aligned}
M_y^{ec}(\phi, \delta t) = I_{n_x^2} &+ \frac{\delta t}{2h_x}\left(I_{n_x} \otimes (G_{n_x}^\top - G_{n_x})\right)\operatorname{diag}(V(\phi)) \\
&- \frac{\delta t^2}{2h_x^2}(I_{n_x} \otimes G_{n_x}^\top)\operatorname{diag}(V(\phi))(I_{n_x} \otimes G_{n_x})\operatorname{diag}(V(\phi)).
\end{aligned}$$

После этого, полный шаг конвекции можно записать как $n_e^1 = M_x^{ec} M_y^{ec} n_e^0$, или $n_e^1 = M_y^{ec} M_x^{ec} n_e^0$. Мы будем использовать оба варианта в двуцикличной схеме расщепления. Для устойчивости схемы требуется выполнение условий Куранта, $\delta t|U|/h_x \leqslant 1$, $\delta t|V|/h_x \leqslant 1$.

Сама по себе схема Мак-Кормака может приводить к ложным осцилляциям на негладких решениях, однако в нашем случае мы получаем некоторую стабилизацию от диффузионного члена, так как число Пекле сравнительно невелико для используемых сеток.

Конвекция распределения ионов

Конвекционная часть в ионном уравнении (1.5) обладает одним важным свойством. Во всех четырех членах, скорость конвекции не зависит от той переменной, по которой применяется градиент. Другими словами, каждый одномерный дрейф можно рассматривать как набор переносов с постоянными скоростями. Такая задача может быть решена аналитически с помощью метода характеристик,

$$\psi(x, y, v, w, t + \delta t) = \psi(x - \delta t v, y, v, w, t),$$

и аналогично по y, v, w. Для краткости мы приводим процедуру только для переменной x. Для других координат мы повторяем ее аналогично, подразумевая дополнительное расщепление ионной конвекции по направлениям.

Итак, пусть заданы сетки

$$x_i = ih_x, \quad y_j = jh_x, \quad v_k = -v_{max} + kh_v, \quad w_m = -v_{max} + mh_v,$$

и рассматриваются значения распределения в узлах, $\psi_{i,j,k,m} = \psi(x_i, y_j, v_k, w_m)$, в частности, $\psi^0 = \psi(t)$. Для дискретизации, сдвиг по характеристике заменяется 5-точечной интерполяцией, так называемой схемой "крест" [158, 5]:

$$\psi^1_{i,j,k,m} = \alpha_{-2}\psi^0_{i-2,j,k,m} + \alpha_{-1}\psi^0_{i-1,j,k,m} + \alpha_0\psi^0_{i,j,k,m} + \alpha_1\psi^0_{i+1,j,k,m} + \alpha_2\psi^0_{i+2,j,k,m},$$

где

$$\begin{aligned} \alpha_{-2} &= \frac{c(c^2-1)(c+2)}{24}, & \alpha_2 &= \frac{c(c^2-1)(c-2)}{24}, \\ \alpha_{-1} &= \frac{-c(c+1)(c^2-4)}{6}, & \alpha_1 &= \frac{-c(c-1)(c^2-4)}{6}, \\ \alpha_0 &= \frac{(c^2-1)(c^2-4)}{4}, & & \text{и} \\ h_x &= x_i - x_{i-1} = \frac{L}{n_x}, & c &= \frac{v\delta t}{h_x} \end{aligned} \tag{1.10}$$

обозначают шаг пространственной сетки и число Куранта, соответственно. Если x_i является граничной точкой, соседние значения $x_{\pm1}, x_{\pm2}$ берутся с другого края области в соответствии с периодичностью. Таким образом, один шаг конвекции для распределения ионов можно записать как $\psi^1 = M_x(\delta t)\psi^0$, где M_x является 5-диагональной циркулянтной матрицей по отношению к x, но зависящей от v как от параметра (см. также секцию 3.2). Как обычно, для перехода на следующий шаг по времени полагаем $\psi(t + \delta t) \approx \psi^1$. Эта схема также получается второго порядка точности и по пространству, и по времени (при условии выполнения условия Куранта $c < 1$). Кроме того, такой метод оказывается более устойчив против возникновения паразитных осцилляций, чем схема Мак-Кормака. Это важно для уравнения Власова, так как оно не содержит диффузии или иной стабилизации по переменным x, y.

Для полного шага конвекции получаем $\psi^1 = M_x(\delta t)M_y(\delta t)M_v(\phi, \delta t)M_w(\phi, \delta t)\psi^0$, где выражения для матриц перехода M_x, M_y, M_v, M_w приведены в секции 3.2. Схема является устойчивой при условии $c \leqslant 1$.

Реакционный член в ионном уравнении

Наконец, часть задачи, соответствующую реакции BGK, $\frac{\partial \psi}{\partial t} = \psi_0 - \psi$, можно рассматривать как линейную автономной систему дифференциальных уравнений с матрицей, приближенно являющейся ортогональным проектором. Это позволяет интегрировать по времени быстро и точно. Обратим внимание, что скоростная часть ψ является возмущением к функции Гаусса, которая имеет ограниченный (точнее: быстро убывающий) Фурье спектр. Это означает, что простая квадратурная формула прямоугольников для (1.6) становится точна (в пренебрежении усеченной частью функции распределения за пределами выбранной области) начиная с некоторой сетки, достаточно разрешающей все гармоники в спектре. Поэтому, дискретное ψ_0 считается следующим образом:

$$(\psi_0)_{i,j,k,m} = \left(\sum_{\tilde{k},\tilde{m}=1}^{n_v} \psi_{i,j,\tilde{k},\tilde{m}} h_v^2 \right) \cdot \frac{1}{2\pi} \exp\left(-\frac{v_k^2 + w_m^2}{2} \right),$$

где

$$h_v = v_i - v_{i-1} = \frac{2v_{max}}{n_v}$$

обозначает шаг сетки по скоростным переменным. Предыдущее выражение может быть записано как матричное произведение:

$$\psi_0 = E\psi, \quad E = I_{n_x^2} \otimes \left(\frac{h_v}{\sqrt{2\pi}} \mathbf{e}\mathbf{1}^\top \right) \otimes \left(\frac{h_v}{\sqrt{2\pi}} \mathbf{e}\mathbf{1}^\top \right),$$

где $\mathbf{e} = \left[\exp(-\frac{v_i^2}{2}) \right]_{i=1}^{n_v}$, и $\mathbf{1}$ это вектор, состоящий из единиц. Поскольку

$$\frac{h_v}{\sqrt{2\pi}} \mathbf{1}^\top \mathbf{e} \approx \frac{1}{\sqrt{2\pi}} \int \exp\left(-\frac{v^2}{2} \right) dv = 1,$$

то следовательно

$$E^2 = E + \mathcal{O}\left(\int_{|\mathbf{v}|>v_{max}} \psi(x,y,v,w) d\mathbf{v} \right), \quad \psi(x,y,v_{max},v_{max}) \sim \mathrm{e}^{-v_{max}^2} \ll 1.$$

Т.е., E приближенно равен ортопроектору, также как и его дополнение $E_\perp = I - E$ в реакционном уравнении

$$\frac{\partial \psi}{\partial t} = (E - I)\psi = -E_\perp \psi, \qquad -E_\perp \leqslant 0.$$

Точное решение последней задачи выписывается как $\psi(t+\delta t) = \exp(-\delta t E_\perp)\psi(t)$, и поскольку $E_\perp^k = E_\perp$ для любой степени $k \geqslant 1$, экспоненциальный ряд упрощается: $\psi^1 = M_r \psi^0$, где

$$M_r(\delta t) = \exp(-\delta t E_\perp) = I + \sum_{k\geqslant 1} \frac{(-\delta t)^k E_\perp^k}{k!} = I + E_\perp \sum_{k\geqslant 1} \frac{(-\delta t)^k}{k!} = I + E_\perp(\exp(-\delta t) - 1).$$

Поскольку $E_\perp \geqslant 0$, этот шаг схемы является абсолютно устойчивым.

1.1.3 Расщепление по времени

Для решения нелинейных уравнений модели, мы используем расщепления по физическим процессам и координатам:

1. Уравнение (1.3) для электронов, оператор диффузии.
2. Уравнение (1.3) для электронов, оператор конвекции.
3. Уравнение Пуассона (1.4).
4. Уравнение для ионов (1.5), конвективные члены.
5. Уравнение для ионов (1.5), член реакции (BGK релаксации).

Такая же схема расщепления была использована в [158, 5]. Для получения второго порядка аппроксимации по времени, мы используем двуцикличную схему Марчука-Стрэнга для *квазилинейных* задач [232, 6]: по предыдущему временному слою $\psi^0 = \psi(t)$, $n_e^0 = n_e(t)$ вычисляем коэффициенты,

$$\begin{aligned} (n_i^0)_{i,j} &= \sum_{k,m} \psi^0_{i,j,k,m} h_v^2, \\ \phi &= \frac{1}{\gamma (T_e)^{3/2}} \Delta_h^{-1} \frac{e^2}{\varepsilon_0 m_i \nu_{in}^2} (n_e^0 - n_i^0), \end{aligned} \tag{1.11}$$

затем находим линеаризацию с первым порядком точности,

$$\begin{aligned} \tilde{n}_e &= \exp(\delta t \Delta_h) M_x^{ec}(\phi, \delta t) M_y^{ec}(\phi, \delta t) n_e^0, \\ \tilde{\psi} &= M_r(\delta t) M_x(\delta t) M_y(\delta t) M_v(\phi, \delta t) M_w(\phi, \delta t) \psi^0, \\ (\tilde{n}_i)_{i,j} &= \sum_{k,m} \tilde{\psi}_{i,j,k,m} h_v^2, \\ \tilde{\phi} &= \frac{1}{\gamma (T_e)^{3/2}} \Delta_h^{-1} \frac{e^2}{\varepsilon_0 m_i \nu_{in}^2} (\tilde{n}_e - \tilde{n}_i), \end{aligned} \tag{1.12}$$

и наконец выполняем двуцикличную линеаризованную схему,

$$\begin{aligned} \psi^{1/4} &= M_x(\delta t) M_y(\delta t) M_v(\tilde{\phi}, \delta t) M_w(\tilde{\phi}, \delta t) \psi^0, \\ \psi^{2/4} &= M_r(\delta t) \psi^{1/4}, \\ n_e^{1/4} &= \exp(\delta t \Delta_h) n_e^0, \\ n_e^{2/4} &= M_x^{ec}(\tilde{\phi}, \delta t) M_y^{ec}(\tilde{\phi}, \delta t) n_e^{1/4}, \\ n_e^{3/4} &= M_y^{ec}(\tilde{\phi}, \delta t) M_x^{ec}(\tilde{\phi}, \delta t) n_e^{2/4}, \\ n_e^{1} &= \exp(\delta t \Delta_h) n_e^{3/4}, \\ \psi^{3/4} &= M_r(\delta t) \psi^{2/4}, \\ \psi^{1} &= M_w(\tilde{\phi}, \delta t) M_v(\tilde{\phi}, \delta t) M_y(\delta t) M_x(\delta t) \psi^{3/4}, \end{aligned} \tag{1.13}$$

полагая новое приближение $\psi(t + 2\delta t) \approx \psi^1$, $n_e(t + 2\delta t) \approx n_e^1$.

Оценки аппроксимации и устойчивости отдельных шагов расщепления, а также теорема о сходимости двуцикличной схемы [6] позволяют сформулировать следующее утверждение.

Утверждение 1.1.1. Пусть выполняются условия Куранта:

$$\max_{i,j} \frac{|U_{i,j}|\delta t}{h_x} \leqslant 1, \quad \max_{i,j} \frac{|V_{i,j}|\delta t}{h_x} \leqslant 1,$$

где U и V определены в (1.8) и (1.9), соответственно, и

$$\frac{v_{max}\delta t}{h_x} \leqslant 1, \quad \max_{i,j} \left| \left(\frac{\partial \phi}{\partial x}\right)_{i,j} \right| \frac{\delta t}{h_v} \leqslant 1, \quad \max_{i,j} \left| \frac{eE_0}{m_i v_{T_i} \nu_{in}} - \left(\frac{\partial \phi}{\partial y}\right)_{i,j} \right| \frac{\delta t}{h_v} \leqslant 1.$$

Тогда схема (1.11)–(1.13) обладает вторым порядком точности аппроксимации и является устойчивой.

Адаптация численной схемы под различные временные масштабы

Характерные временные масштабы в уравнениях (1.3) и (1.5) обусловлены массами электронов и ионом, и поэтому заметно различаются В работе [158, 5] было предложено разделить базовый временной шаг (используется для ионного уравнения (1.5)) на 20–40 меньших подинтервалов при решении уравнения для электронной плотности. Основной причиной для этого является более сильное условие Куранта для конвекционных шагов в (1.3), по сравнению с ионным уравнением (1.5). В данной работе мы используем такой же подход. Обратите внимание, что в схеме расщепления (1.13) шаги, соответствующие электронному уравнению, не зависят от ионной части (при условии зафиксированного предиктора $\tilde{\phi}$). Таким образом, мы можем дополнительно разделить их на $N_{ext} \sim 40$ интервалов, не затрагивая ионные шаги. То же самое выполняется для $\tilde{n}_e$, $\tilde{\psi}$: мы отдельно решаем ионное уравнение с шагом по времени δt, а затем выполняем N_{ext} шагов длиной $\delta t / N_{ext}$ каждый для электронной плотности. Порядок этих шагов не имеет значения, так как для $\tilde{\psi}$ и $\tilde{n}_e$ достаточна точность $\mathcal{O}(\delta t)$.

1.1.4 Начальные состояния плотности электронов и распределения ионов

До сих пор в обсуждении Фарлей-Бунемановской неустойчивости мы не рассматривали начальное состояние системы, т.е. $n_e(0)$ и $\psi(0)$. По скоростям, мы предполагаем Максвелловское распределение,

$$\psi(x, y, v, w, 0) = n_i(x, y, 0) \cdot \frac{1}{2\pi} \exp(-\frac{v^2}{2}) \exp(-\frac{w^2}{2}), \tag{1.14}$$

и распределение концентрации в пространстве берется случайным образом в каждой точке сетки по следующему правилу:

$$n_e(x_i, y_j, 0) = n_0 + M \cdot \text{rand}, \quad n_i(x_i, y_j, 0) = n_0 + M \cdot \text{rand}, \quad \text{rand} \in \mathcal{N}(0, 1),$$

где n_0 это средняя концентрация, $\mathcal{N}(0, 1)$ обозначает стандартное нормальное распределение, и M это масштабирующая константа, выбираемая таким образом, чтобы добавочное поле удовлетворяло $E_{add} = \frac{1}{10} E_0$. Этот выбор основан на двух

соображений. Во-первых, мы хотели бы иметь все возможные гармоники в системе, так что белый шум является хорошим кандидатом для начального состояния. Во-вторых, мы воссоздаем ситуацию имеющую место и в природе, когда Фарлей-Бунемановский процесс развивается от случайных возмущений плазмы. Обратите внимание, что M зависит от размера сетки и конкретных реализаций генератора rand, однако поведение системы в целом остается тем же, при условии что отношение $\frac{E_{add}}{E_0}$ фиксировано.

1.2 Основное кинетическое уравнение для стохастической химической кинетики

Эта секция содержит введение в основное кинетическое уравнение. Соответствующие ссылки см. по тексту.

Изучение биологических систем и молекулярной биологии переживает в последние годы бурное развитие и демонстрирует впечатляющие достижения в понимании генома, поведения клеток, дизайне вакцины и других важных задачах. Правильное описание процессов в живых организмах является требует больших усилий как с экспериментальной точки зрения (точные измерения количеств различных веществ и их разделение), так и в связи с необходимостью полного описания сложных систем. Последнее означает, что раздельное рассмотрение компонентов системы (генов или белков) недостаточно, чтобы понять комплексные явления, имеющие зачастую нетривиальный и контр-интуитивный характер (например, в клеточной дифференцировке). Компоненты системы находятся в сложной взаимосвязи, что выражается нелинейной динамической эволюцией, которая должна рассматриваться на уровне всей системы.

Системная биология, область исследований, посвященная одновременно молекулярной биологии и теории систем, учитывает специфические свойства компонентов и их взаимодействий в системе, с целью выявления и понимания биологических законов, и, наконец, разработки новых биологических систем, эффективного производства вакцин и лекарств или лечения заболеваний.

В этой области, математическое моделирование имеет решающее значение. Каждый эксперимент *in vivo* может быть очень сложным и дорогим. Кроме того, математическая модель позволяет изолировать некоторые явления, чтобы понять их вклад в общую картину, или поставить систему в условия, которые были бы невозможны в реальной жизни. Конечно, расхождения между предсказаниями модели и фактическими измерениями могут свидетельствовать как о неподходящей модели так и о неточных экспериментах. Принципиальную важность имеет совместное проведение экспериментальных исследований реальных биологических систем и моделирования *in silico*, при котором и возможен обмен идеями и поправками из одного подхода в другой.

Различные масштабы дают разные уровни и методы моделирования. Примерно, они могут быть разделены на четыре категории.

- Макроскопический (детерминистский) масштаб,

- мезоскопический (например, клеточный) масштаб,
- классический микроскопический (молекулярный) масштаб, и
- квантовые описания и соответствующие масштабы.

Первый подход подходит если число молекул достаточно велико, порядка числа Авогадро. В этом случае, квантовые и стохастические колебания незначительны, и химическая кинетика или биологическая динамика может быть описана в терминах макроскопических концентраций с помощью обыкновенных дифференциальных уравнений (ОДУ) с удовлетворительной точностью. Для построения полного портрета системы можно использовать стандартные аналитические или численные инструменты.

Существенно микромасштабные процессы на молекулярном и атомном уровнях приходится рассматривать с помощью молекулярной динамики, принимая во внимание все координаты, скорости и физические силы (например, с помощью уравнения Фоккера-Планка). Этот способ достаточно точен, однако может быть слишком вычислительно затратен для больших молекул, типичных для биологических систем, например ДНК, РНК или белков.

Еще ближе к реальности квантовое моделирование. Так, было обнаружено, что уравнение Шредингера может быть в чрезвычайно точном соответствии с экспериментальными результатами для небольших систем: например, уровни энергии для гелия были предсказаны с достоверностью восемь и более десятичных цифр [62]. Тем не менее, из-за вычислительной сложности пока еще невозможно использовать квантовые модели для расчета белковых реакций.

На *мезоскопическом* масштабе, описание системы по-прежнему более или менее феноменологическое (скорости реакций, как правило, оценивается по экспериментальным данным и интуиции, а не *ab initio* квантовым моделям), но стохастический шум вносит уже значительный вклад, который не может быть должным образом учтен в детерминистских ОДУ. Во внутриклеточных системах, количество молекул химических веществ часто находится в пределах сотен. При таких малых концентрациях, стохастические колебания числа молекул могут достигать уровня $\mathcal{O}(10^{-1})$ [230, 14], так как столкновения между частицами и соответствующие реакции возникают принципиально случайным образом. Более того, сам по себе такой биологический шум играет важную роль в межклеточных и внутриклеточных функциях. Такие системы могут демонстрировать неожиданные поведения, например, наличие нескольких метастабильных состояний [76, 183].

Поэтому, для таких систем больше подходит стохастическая кинетика, так как она принимает во внимание стохастические шумы. *Состояние системы* определяется как вектор количеств копий различных веществ, взаимодействующих через несколько биохимических *реакционных каналов*. Состояния являются принципиально дискретными: количества копий веществ всегда целые, и любая реакция может изменить состояние также только на целое число молекул. Скорость реакции определяет теперь “склонность”, т.е. вероятность того, что данная реакция произойдет в следующий бесконечно малый промежуток времени.

Таким образом, динамическую эволюцию стохастической системы реакций можно рассматривать как дискретный марковский процесс. Достаточно точным стоха-

стическим описанием является так называемое *основное кинетическое уравнение* (ОКУ, the Chemical Master Equation, CME), которое моделирует распределение вероятностей на всех возможных состояниях системы [244, 81, 84].

В истории моделирования химической кинетики в биологии, были разработаны различные подходы. Методы Монте-Карло основаны на статистически большом ансамбле реализаций случайного процесса, связанного с ОКУ. Наиболее известным является алгоритм стохастического моделирования (Stochastic Simulation Algorithm, SSA) [81]. На каждом шаге, на основе скоростей реакций и случайного числа, SSA выбирает, какая реакция произойдет в следующий шаг, и выполняет соответствующий переход в новое состояние. Тем не менее, SSA часто очень вычислительно затратен по нескольким причинам. Во-первых, в связи со случайностью одной реализации, требуется моделировать много траекторий (10^6–10^8 и более), чтобы получить правильные статистические результаты. Это особенно сложно, если требуется оценить вероятности редких событий, так как только небольшая часть траекторий может выйти на соответствующие режимы системы.

Во-вторых, даже феноменологическая биологическая модель может содержать значительно различающиеся по масштабам времени и концентраций реакции и вещества. Например, быстрые реакции быстро стабилизируют соответствующую часть системы, в то время как медленные реакции могут потребовать гораздо большие времена моделирования для удовлетворительного описания данных процессов. Часто интересует именно медленная часть динамики. Однако, в большинстве реализаций SSA будет выбирать быстрые реакции, что приводит к очень большому числу шагов по времени.

Некоторые улучшения включают в себя продвинутые методы отбора реализаций [114], так называемый метод τ-прыжков [82], или гибридные методы разбиения системы [92, 113, 124]. Кроме того, для систем с высокими концентрациями можно использовать кинетическое уравнение Фоккера-Планка [83]. Уравнение Фоккера-Планка может быть дискретизировано на сравнительно грубой сетке, содержащей меньшее число неизвестных, чем исходное ОКУ.

Основной альтернативой методам типа Монте-Карло является решение основного кинетического уравнения непосредственно в виде линейного ОДУ. Для многих систем, распределение вероятностей быстро убывает вне ограниченной области. Таким образом, можно редуцировать пространство состояний до конечной области без существенных погрешностей в решении [181].

Тем не менее, такая постановка страдает от “проклятия размерности”: объем даже редуцированного пространства состояний, как правило, очень большой и растет экспоненциально с числом различных веществ. Таким образом, необходимы аппроксимации с помощью методов малопараметрического хранения данных. Одним из первых в этом направлении был метод разреженных сеток (sparse grids) [227]. За ним последовали представления в тензорных произведениях, которые сразу продемонстрировали свой потенциал даже с применением “жадных” алгоритмов в каноническом тензорном формате [12, 73, 166, 37, 42, 112]. Другим типом тензорных подходов был так называемый динамический метод *Дирака-Френкеля* на Таккеровском многообразии (см. [153, 65, 199] и [123], где этот метод был применен непосредственно для решения ОКУ). Мы используем более продвинутые

тензорные форматы и методы, которые позволяют моделировать системы высокой сложности.

Теперь перейдем к формулировке и базовым свойствам основного кинетического уравнения. Предположим, что d различных активных химических соединений $S_1, \ldots, S_d$ в хорошо перемешиваемой среде могут реагировать посредством M реакционных каналов. Каждый канал задается *стехиометрическим вектором* $\boldsymbol{z}^m \in \mathbb{Z}^d$, и функцией *скорости* $w^m(\boldsymbol{i}) : \mathbb{R}_+^d \to \mathbb{R}_+$, $m = 1, \ldots, M$, $\mathbb{R}_+ = \{x \in \mathbb{R} : x \geqslant 0\}$, так что m-я реакция может быть записана в классическом виде

$$a_1^m S_1 + \cdots + a_d^m S_d \xrightarrow{w^m} (a_1^m + z_1^m)S_1 + \cdots + (a_d^m + z_d^m)S_d,$$

где $a_1^m, \ldots, a_d^m \in \mathbb{Z}$ обозначают числа молекул, необходимых для начала реакции.

Чтобы ввести стохастическое описание, обозначим состояния мультииндексом $\boldsymbol{i} = (i_1, \ldots, i_d)$, и всегда будем иметь в виду числа молекул, так что i_k является неотрицательным целым числом, $i_k \in (\{0\} \cup \mathbb{N})$. Вероятностная роль функции скорости следующая: для бесконечно малого интервала времени dt,

$$W^m(\boldsymbol{i}, t, dt) = w^m(\boldsymbol{i})dt$$

является вероятностью того, что при числе молекул $\boldsymbol{i}$ в момент времени t, в последующем интервале $[t, t + dt)$ в системе произойдет одна реакция по каналу m.

Количественной характеристикой состояния $\boldsymbol{i}$ является вероятность того, что количества молекул веществ $S_1, \ldots, S_d$ в момент времени t принимают значения $i_1, \ldots, i_d$,

$$\Psi(\boldsymbol{i}, t) : (\{0\} \cup \mathbb{N})^d \cup [0, T] \to \mathbb{R}_+.$$

Эта вероятность на самом деле условная и зависит также от начального состояния, $\boldsymbol{i}|_{t=0}$, но для краткости мы его не будем указывать явно.

Теперь, принимая dt достаточно малым, так что вероятность того, что больше, чем одна реакция будет происходить в интервале $[t, t + dt)$ пренебрежимо мала, можно написать распределение в конце интервала, $\Psi(\boldsymbol{i}, t + dt)$, используя законы сложения и умножения вероятностей для независимых и взаимоисключающих событий:

$$\Psi(\boldsymbol{i}, t + dt) = \underbrace{\Psi(\boldsymbol{i}, t)\left(1 - \sum_{m=1}^{M} w^m(\boldsymbol{i})dt\right)}_{\substack{\text{было состояние } \boldsymbol{i}, \text{ и} \\ \text{не произошло реакций} \\ \equiv \texttt{not}(\text{реакция произошла})}} + \sum_{m=1}^{M} \underbrace{\Psi(\boldsymbol{i} - \boldsymbol{z}^m, t) \cdot w^m(\boldsymbol{i} - \boldsymbol{z}^m)dt}_{\substack{\text{реакция } m \text{ произошла} \\ \text{с переходом в } \boldsymbol{i}, \text{ следовательно,} \\ \text{из начального состояния } \boldsymbol{i} - \boldsymbol{z}^m}}.$$

Комбинируя члены $\Psi(\boldsymbol{i}, t+dt) - \Psi(\boldsymbol{i}, t)$ в левой части, и переходя к пределу $dt \to 0$, получаем [84, 85] основное кинетическое уравнение:

$$\frac{d\Psi(\boldsymbol{i}, t)}{dt} = \sum_{m=1}^{M} w^m(\boldsymbol{i} - \boldsymbol{z}^m)\Psi(\boldsymbol{i} - \boldsymbol{z}^m, t) - w^m(\boldsymbol{i})\Psi(\boldsymbol{i}, t). \tag{1.15}$$

Потенциально возможны любые числа молекул, т.е. уравнение (1.15) является бесконечномерным ОДУ. Разумеется, для проведения численного моделирования

нам нужно ограничить его до конечной задачи. Так называемый алгоритм *проекции в конечное пространство* (Finite State Projection, FSP) [181] использует тот факт, что очень большие количества молекул имеют очень малую вероятность возникнуть за конечное время:

$$\Psi(\boldsymbol{i}, t) \to 0, \ |\boldsymbol{i}| \to \infty.$$

Таким образом, мы полагаем что каждый i_k лежит в конечном диапазоне, $i_k = 0, \ldots, n_k - 1$, и выбираем n_k достаточно большими, так что, например, при $i_k > n_k$ вероятность $\Psi(\boldsymbol{i}, t)$ меньше машинной точности, и можно пренебречь ошибкой, внесенной при ограничении пространства состояний.

Даже если каждый $n_k = \mathcal{O}(n)$ не превышает десятков, общее число степеней свободы растет как n^d, и сжатое хранение и обработка для Ψ принципиально необходимы. Мы отложим это до следующих глав, когда мы введем тензорные методы. Сейчас сосредоточимся на алгебраических и спектральных свойствах оператора ОКУ.

Для начала, введем более удобную запись (1.15) с помощью матриц сдвига. Обозначим

$$J^z = \begin{bmatrix} 0 & & & & \\ \vdots & \ddots & & & \\ 1 & & \ddots & & \\ & \ddots & & \ddots & \\ & & 1 & \cdots & 0 \end{bmatrix} \leftarrow \quad \text{строка} \quad z+1, \quad \text{если } z \geqslant 0, \tag{1.16}$$

и для $z < 0$ определим $J^z = (J^{-z})^\top$. Теперь запишем проекцию в конечное пространство (FSP) уравнения (1.15) как линейное ОДУ:

$$\frac{d\psi(t)}{dt} = A\psi(t), \quad A = \sum_{m=1}^{M} (\mathbf{J}^{\boldsymbol{z}^m} - \mathbf{J}^0) \operatorname{diag}(w^m), \quad \psi(t) \in \mathbb{R}_+^{\prod_{k=1}^d n_k}, \tag{1.17}$$

где многомерный оператор сдвига определен как прямое произведение одномерных,

$$\mathbf{J}^{\boldsymbol{z}} = J^{z_1} \otimes \cdots \otimes J^{z_d},$$

$w^m = \{w^m(\boldsymbol{i})\}$ и $\psi(t) = \{\psi(\boldsymbol{i}, t)\}$, $\boldsymbol{i} \in \bigotimes_{k=1}^{d} \{0, \ldots, n_k - 1\}$, суть векторы, содержащие значения w^m и ψ, $\operatorname{diag}(w^m)$ конструирует диагональную матрицу из элементов w^m, вытягивая их вдоль диагонали, и $\otimes$ обозначает Кронекеровское произведение (см. выражение (1.24) ниже). Заметим, что $\mathbf{J}^0$ является обычной единичной матрицей подходящих размеров. Как правило, конечное решение $\psi(\boldsymbol{i}, t)$ не совпадает с точным распределением $\Psi(\boldsymbol{i}, t)$ (в общих точках), даже если начальное состояние проецировалось точно, $\psi(\boldsymbol{i}, 0) = \Psi(\boldsymbol{i}, 0)$. Тем не менее, погрешность может быть количественно оценена, см. теорему 1.2.1.

Граничные значения функций скоростей w^m требует внимательного рассмотрения. Предположим, что реакции, уменьшающей количество молекул определенного типа, $z_k^m < 0$, разрешается протекать и в случае, когда количество соответствующего компонента недостаточно: $w^m(\boldsymbol{i}) > 0$ при $i_k < |z_k^m|$. Это приведет к

нефизичному явлению: с ненулевой вероятностью возникнут отрицательные i_k. Чтобы избежать такой ситуации, мы всегда будем задавать *граничные условия*:

$$w^m(\boldsymbol{i}) = 0 \quad \text{если любой элемент } \boldsymbol{i} + \boldsymbol{z}^m < 0. \tag{1.18}$$

Эти граничные условия *непротекания* достаточны, чтобы бесконечномерное уравнение (1.15) было корректно в физическом и вероятностном смысле, то есть отрицательные количества веществ никогда не возникают, вероятность Ψ неотрицательна, а нормировка $\sum_{\boldsymbol{i}} \Psi(\boldsymbol{i}, t)$ сохраняется при эволюции системы во времени (при условии, что $\Psi(\boldsymbol{i}, 0)$ подчиняется этим свойствам). Тем не менее, это может быть не так, если FSP применяется непосредственно без изменения w^m.

Основные свойства приближения FSP были установлены в [181]. Во-первых, если $\psi(\boldsymbol{i}, 0) \geqslant 0$ и $w^m(\boldsymbol{i}) \geqslant 0$, то сохраняется условие $\psi(\boldsymbol{i}, t) \geqslant 0$. Во-вторых, ошибка в решении контролируемо связана с потерей нормировки вероятности.

Теорема 1.2.1 ([181], Теорема 2.2)**.** Пусть $\psi(\boldsymbol{i}, 0) = \Psi(\boldsymbol{i}, 0) \geqslant 0$, $w^m(\boldsymbol{i}) \geqslant 0$. Если для некоторого $\varepsilon > 0$ и $t \geqslant 0$ выполняется

$$\sum_{\boldsymbol{i}} \psi(\boldsymbol{i}, t) \geqslant 1 - \varepsilon,$$

тогда

$$\psi(\boldsymbol{i}, t) \leqslant \Psi(\boldsymbol{i}, t) \leqslant \psi(\boldsymbol{i}, t) + \varepsilon, \quad \boldsymbol{i} \in \bigotimes_{k=1}^{d} \{0, \ldots, n_k - 1\}.$$

Другой вариант анализа, связанный с регулярностью функции вероятности, был приведен в [77].

Как было показано в [121], все собственные значения оператора ОКУ в (1.17) имеют неположительные вещественные части. В самом деле, каждая строчная сумма $\mathbf{J}^{\boldsymbol{z}^m} - \mathbf{J}^0$ равна либо -1, либо 0, и $\operatorname{diag}(w^m)$ неотрицательна, так как диагональные элементы и строчные суммы матрицы A неположительны. По теореме Гершгорина, все собственные числа лежат в левой части комплексной плоскости. Это обеспечивает стабильность динамики ОКУ. Однако, если и w^m, и ψ отличны от нуля в таких точках i_k, что $i_k + z_k^m \geqslant n_k$, все собственные значения имеют строго отрицательные действительные части, и норма решения $\psi(t)$ уменьшается со временем. Это приводит к накоплению ошибки как показано в теореме 1.2.1, и свидетельствует о необходимости подбора достаточно большого n_k, чтобы отброшенная часть ψ была ничтожно мала.

Несложно восстановить сохранение нормировки и для конечного уравнения [122]. Все, что для этого требуется, это изменить функцию скорости следующим образом:

$$w^m(\boldsymbol{i}) = 0 \quad \text{если любой } i_k + z_k^m \geqslant n_k, \quad k = 1, \ldots, d, \tag{1.19}$$

т.е. накладывая дополнительные граничные условия, помимо естественных (1.18). Теперь, следуя [122], можно заметить, что

$$\sum_{\boldsymbol{i}} w^m(\boldsymbol{i} - \boldsymbol{z^m})\psi(\boldsymbol{i} - \boldsymbol{z^m}) = \sum_{\boldsymbol{i} + \boldsymbol{z^m}} w^m(\boldsymbol{i})\psi(\boldsymbol{i}) = \sum_{\boldsymbol{i}} w^m(\boldsymbol{i})\psi(\boldsymbol{i}),$$

и следовательно $\boldsymbol{e}^\top(\mathbf{J}^{\boldsymbol{z}^m} - \mathbf{J}^0)\operatorname{diag}(w^m)\psi = 0$, где $\boldsymbol{e}$ суть вектор из всех единиц, соответствующий суммированию по всем допустимым $\boldsymbol{i}$.

Лемма 1.2.2. Пусть для конечного ОКУ заданы и левые (1.18), и правые (1.19) граничные условия. Тогда минимальное сингулярное число A равно нулю, вектор $\boldsymbol{e}$ является соответствующим левым сингулярным вектором, и правый сингулярный вектор $\psi_\star$ является стационарным решением, $\frac{d\psi_\star}{dt} = 0$.

Второе утверждение следует сразу из того, что $A\psi = 0$ обеспечивает $\frac{d\psi}{dt} = 0$.

Замечание 1.2.3. *Кратность* ядра A требует дополнительного рассмотрения. Как было показано в [101], определенная неизбыточность набора веществ и реакций гарантирует единственность стационарного решения. Тем не менее, часто размерность ядра может достигать n и более (например, в случае обратимой реакции преобразования $S_1 + S_2 \leftrightarrows S_3$, протекающей с постоянной скоростью в обе стороны). В этом случае важное значение имеет начальное состояние, определяющее конкретный вектор из ядра, к которому сходится процесс.

Теперь оценим максимальное сингулярное значение оператора ОКУ.

Лемма 1.2.4.

$$\|A\|_2 \leqslant 2\sum_{m=1}^{M} \max(w^m).$$

Эту оценку можно доказать с помощью простого неравенства матричных норм, примененного к (1.17): $\|\mathbf{J}^{\boldsymbol{z}^m} - \mathbf{J}^0\|_2 \leqslant 2$ (более детальную аргументацию см. в следующей секции), и того факта, что нормой диагональной матрицы является ее максимальный элемент.

Обратим внимание, что матрица может быть довольно плохо обусловленной: если w^m является многочленом степени p, его максимальный элемент может достигать значений порядка n^p. Поэтому может потребоваться много мелких временных шагов, чтобы приблизить динамику системы достаточно точно. В следующем разделе представлен эффективный подход как решению этого вопроса.

Наконец, мы можем отметить, что неположительность оператора ОКУ помогает подавлять высокочастотный шум, возникающий из-за тензорных аппроксимаций на каждом шаге. Можно было бы ожидать роста ошибок тензорных приближений на каждом шаге по времени, но, как мы заметили на практике, даже при моделировании на больших временных масштабах, уровень ошибок поддерживается на постоянном уровне.

1.3 Схемы интегрирования эволюционных уравнений

В данной секции излагаются классические схемы интегрирования ОДУ в новой интерпретации, позволяющей сокращать вычислительную сложность при использовании тензорных аппроксимаций. Эта интерпретация является авторской и приводится в соответствии с работами [53, 51].

1.3.1 Одновременная дискретизация в пространстве и времени

Предположим, что в пространстве или в переменных состояний уже введена некоторая дискретизация, тогда уравнение в частных производных сводится к ОДУ

$$\begin{aligned} &\frac{dx(t)}{dt} + Ax(t) = f(t) \in \mathbb{C}^N \\ &x(0) = v, \quad t \in [0, T], \end{aligned} \tag{1.20}$$

возможно, очень большого размера $N = n^d$.

Для простоты будем считать, что матрица A не зависит от t и x, хотя обобщение рассматривается аналогично. Для определенности, рассмотрим известную схему Кранка-Николсон (для линейного ОДУ она всегда применима при достаточно малых шагах по времени). Зададим во времени равномерную сетку,

$$t \in \{t_p\}_{p=0}^{N_t}, \quad t_p = p\delta t, \quad p = 0, \ldots, N_t, \quad f_p = f(t_p), \quad T = N_t \delta t,$$

тогда приближенное решение (1.20) можно вычислить, решая следующую линейную систему на каждом шаге по времени:

$$\left(I + \frac{\delta t}{2} A\right) x_{p+1} = \left(I - \frac{\delta t}{2} A\right) x_p + \frac{\delta t}{2}(f_p + f_{p+1}), \quad p = 0, \ldots, N_t - 1, \tag{1.21}$$

при условии $x_0 = v$. Если матрица A неотрицательно определена, спектральная норма матрицы *перехода* ограничена единицей,

$$\left\| \left(I + \frac{\delta t}{2} A\right)^{-1} \left(I - \frac{\delta t}{2} A\right) \right\|_2 \leqslant 1,$$

поэтому метод является абсолютно устойчивым [131].

Схема обладает вторым порядком аппроксимации, $\|x(t_p) - x_p\|_2 = \mathcal{O}(\delta t^2)$. Тем не менее, если матрица A жесткая, количество временных шагов для обеспечения разумной точность в длительной динамике может быть довольно большим. С другой стороны, чем меньше шаг по времени, тем меньшая ошибка допустима при решении линейной системы (1.21) – но именно этого мы хотели бы избежать, и сохранить этот порог “не слишком маленьким”, для достижения хорошей сжимаемости решения в тензорном формате.

В качестве нестандартной альтернативы, мы можем посмотреть на время просто как на еще одну координату, и собрать все временные шаги в одну глобальную линейную систему:

$$\begin{bmatrix} I + \frac{\delta t}{2} A & & & \\ -I + \frac{\delta t}{2} A & I + \frac{\delta t}{2} A & & \\ & \ddots & \ddots & \\ & & -I + \frac{\delta t}{2} A & I + \frac{\delta t}{2} A \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_t} \end{bmatrix} = \begin{bmatrix} v - \frac{\delta t}{2} A v \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \frac{\delta t}{2} g, \tag{1.22}$$

где $g = \begin{bmatrix} f_0 + f_1 & f_1 + f_2 & \cdots & f_{N_t-1} + f_{N_t} \end{bmatrix}^\top$. Конечно, эта система не несет смысла, если мы будем держать ее в стандартном виде – пошаговое решение (1.21) является оптимальным методом для двухдиагональной матрицы. Тем не менее, предположим что можно вычислить ее решение более эффективно в *структурированном* представлении. Тогда становится заманчиво получить сразу всю историю динамики во времени, и использовать очень мелкие шаги по времени, чтобы гарантировать достаточную точность для всех спектральных компонент матрицы A.

Естественно ожидать, что структурированное представление является эффективным при определенных предположениях о решении. Например, подобная постановка рассматривалась в подходе *разреженных сеток* [100, 248], где количества пространственных и временных степеней свободы вносят аддитивный вклад в вычислительную сложность. Эти свойства доказаны в предположениях об определенной гладкости решения.

В этой работе мы используем методы приближения тензорными произведениями. Как правило, соотнести их эффективность с гладкостью решения непросто. Существующие оценки сравнимы с результатами в методах разреженных сеток (см. [220, 99]), но в большинстве случаев реальная сложность оказывается значительно ниже теоретически предсказываемой. Более того, тензорные алгоритмы могут вполне успешно применяться и для негладких функций. Малоранговое разделение пространственных и временных переменных также было использовано в [231]. С многомерными тензорными методами, мы можем ввести дополнительные виртуальные переменные, и в конечном итоге аппроксимировать и пространственную, и временную части системы (1.22) с логарифмической редукцией вычислительной сложности, $\mathcal{O}(\log(N)\log(N_t))$, которая действительно подтверждается в численных экспериментах.

Схема решения будет описано более подробно ниже, после того как будет введен формализм тензорных произведений. Сейчас сосредоточимся на спектральных свойствах системы (1.22). Прежде всего, запишем более удобный аналог с помощью Кронекеровских произведений:

$$\mathcal{A}x = \mathcal{F}, \quad \mathcal{A} = I \otimes G_t + A \otimes \frac{\delta t}{2} M_t, \quad \mathcal{F} = \left(v - \frac{\delta t}{2} A v \right) \otimes e_1 + \frac{\delta t}{2} g, \tag{1.23}$$

где Кронекеровское произведение $\otimes$ определяется как следующая блочная матрица:

$$A \otimes B = \begin{bmatrix} AB_{1,1} & \cdots & AB_{1,n} \\ \vdots & & \vdots \\ AB_{m,1} & \cdots & AB_{m,n} \end{bmatrix}, \tag{1.24}$$

e_1 суть первый единичный вектор, и G_t, M_t обозначают матрицы жесткости и массы для дискретизации по времени,

$$G_t = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}, \quad M_t = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix}.$$

Напоминаем, что $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_{N_t} \end{bmatrix}^\top$ определяет вектор глобального решения, содержащий все временные слои исходной схемы Кранка-Николсон.

Поскольку G_t и M_t треугольные, можно сразу сказать, что все их собственные значения равны 1. Для обеспечения устойчивости системы, предполагается, что спектр A лежат в правой комплексной полуплоскости. При этом условии мы можем доказать корректность постановки (1.22).

Теорема 1.3.1. Пусть $\mathrm{Re}\lambda(A) \geqslant 0$, тогда экстремальные сингулярные значения матрицы $\mathcal{A}$ в (1.23) оцениваются следующим образом:

$$\sigma_{\max}(\mathcal{A}) \leqslant 2 + \delta t \|A\|_2, \quad \sigma_{\min}(\mathcal{A}) \geqslant \frac{1}{N_t}, \tag{1.25}$$

так что $\mathrm{cond}(\mathcal{A}) \leqslant 2N_t + T\|A\|_2$.

Доказательство. Поскольку и $\mathrm{Re}\lambda(A) \geqslant 0$, и $\lambda(M_t) > 0$, можно утверждать, что $\|\mathcal{A}z\| \geqslant \|(I \otimes G_t)z\|$ для любого вектора z. В частности, используя также свойства Кронекеровского произведения, можно оценить $\sigma_{\min}(\mathcal{A}) \geqslant \sigma_{\min}(G_t)$. Последняя величина вычисляется с помощью спектральной нормы G_t^{-1}, которая может быть ограничена с помощью следующего неравенства:

$$\|M\|_2 \leqslant \sqrt{\|M\|_1 \|M\|_\infty}, \quad \|M\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{m} |M_{i,j}|, \quad \|M\|_\infty = \max_{i=1,\ldots,m} \sum_{j=1}^{n} |M_{i,j}|$$

для любой матрицы M размеров $m \times n$. Можно непосредственно проверить, что

$$G_t^{-1} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ \vdots & \ddots & \ddots & \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

где и строчная, и столбцовая суммы достигают N_t. Следовательно,

$$\sigma_{\min}(\mathcal{A}) \geqslant \sigma_{\min}(G_t) = \frac{1}{\|G_t^{-1}\|_2} \geqslant \frac{1}{\sqrt{\|G_t^{-1}\|_1 \|G_t^{-1}\|_\infty}} = \frac{1}{N_t}.$$

Второе утверждение доказано.

Первая оценка выводится аналогично: неравенство треугольника дает

$$\sigma_{\max}(\mathcal{A}) = \|\mathcal{A}\|_2 \leqslant \|G_t\|_2 + \frac{\delta t}{2} \|M_t\|_2 \|A\|_2,$$

тогда как

$$\|G_t\|_1 = \|G_t\|_\infty = \|M_t\|_1 = \|M_t\|_\infty = 2,$$

так что спектральные нормы этих матриц не выше 2. Наконец, вспоминая, что $\delta t = T/N_t$, мы получаем и последнее утверждение. □

Из-за линейности задачи было естественно ожидать, что число обусловленности линейно зависит от всех основных свойств системы: количество временных шагов, длина временного интервала, и норма пространственной матрицы. Конечно, эта теорема носит весьма общий характер, и, накладывая дополнительные требования на A, можно доказать более аккуратные результаты. Есть ли другие способы, чтобы уменьшить число обусловленности? Одним из основных используемых подходов является предобуславливание. Ослабить влияние дискретизации по времени достаточно просто. А именно, мы умножаем (1.23) на матрицу $I \otimes G_t^{-1}$, и приходим к системе

$$\begin{aligned} \tilde{\mathcal{A}} x = \tilde{\mathcal{F}}, \quad \tilde{\mathcal{A}} = I \otimes I + A \otimes \frac{\delta t}{2}(G_t^{-1} M_t), \\ \tilde{\mathcal{F}} = \left(v - \frac{\delta t}{2} A v \right) \otimes e + \frac{\delta t}{2}(I \otimes G_t^{-1}) g, \end{aligned} \tag{1.26}$$

где $e = G_t^{-1} e_1$ является вектором из одних единиц.

Теорема 1.3.2. Пусть $\mathrm{Re}\lambda(A) \geqslant 0$, тогда экстремальные сингулярные значения матрицы $\tilde{\mathcal{A}}$ в (1.26) оцениваются следующим образом:

$$\sigma_{\max}(\tilde{\mathcal{A}}) \leqslant 1 + T\|A\|_2, \quad \sigma_{\min}(\tilde{\mathcal{A}}) \geqslant 1, \tag{1.27}$$

так что $\mathrm{cond}(\tilde{\mathcal{A}}) \leqslant 1 + T\|A\|_2$.

Доказательство. Используя те же аргументы, что и раньше, получаем $\sigma_{\min}(\tilde{\mathcal{A}}) \geqslant 1$. Поскольку и G_t, и M_t треугольные, матрица $G_t^{-1} M_t$ тоже треугольная, и содержит на диагонали собственные значения, равные 1, что обеспечивает выполнение свойства $\mathrm{Re}\lambda(A \otimes (G_t^{-1} M_t)) \geqslant 0$. Следовательно, верхняя оценка пишется так:

$$\sigma_{\max}(\tilde{\mathcal{A}}) \leqslant 1 + \frac{\delta t}{2}\|G_t^{-1}\|_2 \|M_t\|_2 \|A\|_2 \leqslant 1 + \delta t N_t \|A\|_2 = 1 + T\|A\|_2.$$

□

В такой схеме, выбирая интервал времени, мы можем сделать пространственно-временную матрицу сколь угодно хорошо обусловленной.

Замечание 1.3.3. Обратим внимание, что интервал времени $[0, T]$ может не соответствовать всему диапазону времен, необходимому в данном приложении. Мы не обязаны отказываться от идеи временных шагов полностью – глобальные схемы (1.22) – (1.26) можно рассматривать как методы для выполнения "больших" шагов по времени, размера T каждый. Разделим желаемый интервал $[0, \hat{T}]$ на подинтервалы $[0, T], [T, 2T], \ldots, [\hat{T} - T, \hat{T}]$. Затем, используем (1.22) – (1.26) с *рестартами*. Когда решение системы на данном интервале $[(q-1)T, qT]$ найдено $(q = 1, \ldots, \hat{T}/T)$, мы извлекаем последний слой x_{N_t}, и для системы на следующем интервале используем его в качестве начального состояния. Мы можем адаптировать T, добиваясь наиболее быстрого режима вычислений.

Замечание 1.3.4. Если матрицы A несимметрична, действительность спектра матрицы $G_t^{-1}M_t$ является важным компонентом эффективности такого типа предобуславливания. Предположим, что и $\lambda(A)$, и $\lambda(G_t^{-1}M_t)$ комплексные. Тогда могло бы оказаться, что $\mathrm{Re}\left(\lambda(A)\lambda(G_t^{-1}M_t)\right) < 0$ для некоторых собственных чисел. Это могло бы дать еще худшую обусловленность, чем в системе (1.23), если величина $1 + \mathrm{Re}\lambda(A \otimes (G_t^{-1}M_t))$ слишком близка к нулю.

Комплексные собственные значения временных матриц могут появиться при использовании дискретизаций методом Галеркина, или спектральным [223, 128, 239]. Преимущества спектральных методов вытекают из их быстрой сходимости с увеличением числа степеней свободы – когда в большинстве случаев достаточно $\mathcal{O}(40)$ базисных функций, не проблема использовать систему (1.23) без предобуславливания.

1.3.2 Нахождение стационарного решения неявным методом Эйлера

В некоторых задачах, записанных изначально в виде динамических уравнений, требуется только стационарное решение. Предположим для простоты, что система имеет точечный аттрактор (например, в случае линейного ОДУ $dx/dt + Ax = 0$), который принадлежит ядру оператора, $Ax_\star = 0$. Однако, вычисление $x_\star$ в виде решения задачи на собственные или сингулярные значения имеет несколько недостатков. Во-первых, для несимметричной матрицы задача на собственные значения сложна, особенно в тензорных форматах, поскольку для нее не существует вариационной формулировки. Алгоритм для вычисления частичного сингулярного разложения в тензорных произведениях мог бы улучшить ситуацию, но пока неясно какими именно свойствами он будет обладать.

Более важна вторая проблема: как выделить несколько векторов ядра, и отфильтровать их от более высоких собственных/сингулярных векторов в случае очень малых спектральных промежутков. Как мы уже отмечали в Замечании 1.2.3, ядро с размерностью больше единицы – это естественная ситуация в основном кинетическом уравнении, и важно выбрать правильную проекцию исходного состояния на нуль-пространство.

Именно последнее соображение дает нам разумную идею, как решить эту проблему. Все что нам нужно, это провести динамическую эволюцию, начиная с желаемого начального приближения, пока невязка (обычно $|x(t+\delta t)-x(t)|$ или $|Ax(t)|$) не упадет ниже заданного порога. В этом случае, дискретизация на мелкой сетке в блочной схеме (1.22) избыточна. Вместо этого мы используем неявную схему Эйлера, также известную как обратный степенной метод,

$$(I + \delta t A)\, x_p = x_{p-1}, \quad p = 1, \ldots, N_t. \tag{1.28}$$

Заметим, что здесь шаг δt может быть существенно больше, чем в (1.22). Промежуточные решения плохо приближают переходные процессы, но как только метод сходится, он восстанавливает правильную компоненту собственного подпространства A, соответствующего собственному числу с минимальной действительной ча-

стью. Как было отмечено, количество шагов N_t выбирается так, чтобы выполнялось

$$\eta = \frac{\|x_{N_t+1} - x_{N_t}\|}{\delta t \|x_{N_t}\|} \leqslant \epsilon, \quad \text{или} \quad \eta = \frac{\|Ax_{N_t}\|}{\|x_{N_t}\|} \leqslant \epsilon. \tag{1.29}$$

При условии, что система ОДУ устойчива, то есть $A \geqslant 0$, можно сделать вывод, что число обусловленности матрицы в (1.28) ограничено $1 + \delta t \|A\|_2$. В степенном методе скорость сходимости $(1 + \delta t |\mathrm{Re}\lambda_2(A)|)^{-1}$ определяется спектральным интервалом, и большие δt приводят к более быстрой сходимости, но надо принимать во внимание и сложность обращения сдвинутой матрицы.

Так как последняя операция обычно делается итерационным методом до некоторой точности (и это единственный подход в рамках тензорных аппроксимаций), мы предлагаем следующий способ ускорения вычислений. Мы вычисляем невязку, пользуясь одним из определений (1.29). Если η велико, нам не нужно решать (1.28) очень точно. Когда η уменьшается, точность может быть улучшена. На практике, мы используем правило вида $\varepsilon = c\eta$ (например, $c = 10^{-1}$), где ε задает порог тензорных аппроксимаций и критерий останова для решения системы (1.28). Такой подход существенно уменьшает сложность промежуточных итераций.

В дальнейшем мы будем ссылаться на этот метод как на метод (неявных) итераций Эйлера. Мы избегаем двусмысленности, используя схему Кранка-Николсон в глобальных пространственно-временных постановках (1.22)–(1.26).

Глава 2

Представления и аппроксимации тензорными произведениями

Многомерные массивы возникают во многих приложениях, но непосредственное их хранение и обработка невозможны при существенно высоких размерностях, поскольку обычно мы неизбежно сталкиваемся с экспоненциальной зависимостью числа неизвестных, необходимых для поддержания заданной точности, от размерности.

Однако, в реальных приложениях массивы (или тензоры) возникают из некоторой физической задачи и обладают какой-то скрытой структурой. Для эффективной работы с такими объектами важно выявить эту структуру и использовать ее для малопараметрического представления данных. Иногда такая структура очевидно следует из модели. Широко известными примерами являются разреженное хранение (например, матриц), а также случаи тензоров, инвариантных к сдвигам, таких как Теплицевы или Ганкелевы [15]. Тем не менее, эти частные классы недостаточно широки для наших целей, и мы будем рассматривать другие виды представлений.

Первая часть данной главы посвящена введению в представления тензорными произведениями и обзору существующих методов. Основным источником для этого являются обзоры [221, 222, 154, 102, 145]. Автором предложено новое представление QTT-Tucker и операции с ним в секциях 2.2.2–2.2.5.

2.1 Разделение переменных в двух и многих размерностях

В данной работе используется метод разделения переменных. Пусть дан тензор $x = [x(i_1, \ldots, i_d)]$, где $i_k = 1, \ldots, n_k \leqslant n$, $k = 1, \ldots, d$, так что количество элементов в x оценивается числом n^d. Однако, предположим, что x может быть записан в виде прямого произведения одномерных массивов (векторов), т.е.

$$x(i_1, \ldots, i_d) = x^{(1)}(i_1) x^{(2)}(i_2) \cdots x^{(d)}(i_d). \tag{2.1}$$

Обратите внимание, что каждый член $x^{(k)}$ требует хранения только n элементов, но они определяют любой элемент из x. Таким образом, эффективные затраты памяти сокращаются до значения $nd \ll n^d$. Разумеется, идеальный вариант разделения переменных (2.1) слишком узок на практике. Чтобы построить применимое обобщение (2.1), нам потребуется два компонента: во-первых, представление должно допускать суммирование нескольких тензоров вида прямого произведения, и во-вторых, оно должно давать возможность эффективных вычислений с аппроксимациями.

2.1.1 Малоранговое разложение матрицы

Для последовательного понимания основных идей, рассмотрим в начале случай двух переменных. Пусть задана матрица $X = \big[x(i_1, i_2)\big]$ (массив двух переменных), тогда естественный способ разделения переменных следующий:

$$x(i_1, i_2) = \sum_{\alpha=1}^{n} x_\alpha^{(1)}(i_1) x_\alpha^{(2)}(i_2) \quad \Leftrightarrow \quad X = X^{(1)}(X^{(2)})^\top.$$

Это равенство называется скелетным разложением, и (приближенное) сжатие данных получается путем ограничения диапазона *рангового индекса* α с n до некоторого значения $r < n$, так что

$$\big\|X - X^{(1)}(X^{(2)})^\top\big\| \leqslant \varepsilon, \quad X^{(k)} = \Big[x_\alpha^{(k)}(i_k)\Big]_{\alpha, i_k=1}^{r, n}.$$

Теперь, $x^{(1)}$ и $x^{(2)}$ содержат $2nr$[1] значений, и мы можем предлагать выбор этих *факторов* в целях оптимизации зависимости ошибки ε от *ранга* r. Замечательное свойство матричного случая заключается в том, что *оптимальное* разложение дается очень конструктивным способом, и может быть надежно посчитано численно с использованием широко известной и эффективной библиотеки LAPACK.

Теорема 2.1.1. Любая матрица X имеет *сингулярное разложение* (SVD),

$$x(i_1, i_2) = \sum_{\alpha=1}^{n} U_\alpha(i_1) \sigma_\alpha V_\alpha(i_2), \tag{2.2}$$

где $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$ называются сингулярными значениями, $\sigma_i^2 \in \lambda(X^* X)$, а U и V являются ортогональными матрицами сингулярных векторов. Кроме того, *неполное* сингулярное разложение ранга r дает оптимальную ранг-r аппроксимацию, т.е.

$$\left\| X - \sum_{\alpha=1}^{r} U_\alpha \sigma_\alpha V_\alpha^\top \right\|_2 = \min_{\operatorname{rank}(Y)=r} \|X - Y\|_2 = \varepsilon.$$

[1] В дальнейшем, для краткой записи асимптотических оценок, мы будем подразумевать $n_k \leqslant n$, и т.д.

Очевидно, для скелетного разложение можно положить $X^{(1)} = U, X^{(2)}_\alpha = \sigma_\alpha V_\alpha$ или $X^{(1)}_\alpha = U_\alpha \sigma_\alpha, X^{(2)} = V$, получая соответственно *лево-* или *право-ортогональные* разложение. Понятие ортогональности будет принципиально для вычислительных алгоритмов в дальнейшем.

Хотя в общем случае зависимость $r(\varepsilon)$ неизвестна, если x задается путем дискретизации гладкой функции, некоторые разумные оценки существуют, обычно в следующем логарифмическом виде [10, 242, 106]:

$$r = \mathcal{O}(\log^\beta(1/\varepsilon)\log^\gamma(n)), \quad \beta, \gamma > 0.$$

Один недостаток сингулярного разложения, это, пожалуй, его сложность: оно требует рассмотрения всех элементов матрицы, и $\mathcal{O}(mn\min(m,n))$ вычислительных операций, что может представлять проблему при n порядка десятков тысяч и выше. В определенных случаях, может быть достаточно квази-оптимального, но значительно более быстрого метода: исключения Гаусса, также известного как метод крестовой интерполяции [2, 91, 1, 90, 241]. Он основывается на выборе "важных" строк и столбцов матрицы, т.е.

$$X(i_1, i_2) \approx \tilde{X}(i_1, i_2) = \sum_{\alpha,\alpha'=1}^{r} X(i_1, \mathcal{J}^2_\alpha) M_{\alpha,\alpha'} X(\mathcal{J}^1_{\alpha'}, i_2), \tag{2.3}$$

где $M = (X(\mathcal{J}^1, \mathcal{J}^2))^{-1}$, и $\mathcal{J}^1 \subset \{1, \ldots, n_1\}$, $\mathcal{J}^2 \subset \{1, \ldots, n_2\}$ обозначают выбранные множества индексов размером r. Для того, чтобы разложение (2.3) имело хорошие аппроксимирующие свойства, важно выбирать эти "крестовые" индексы правильно. Так, в работе [90] был доказан следующий *принцип наибольшего объема.*

Лемма 2.1.2. Если $\mathcal{J}^1$ и $\mathcal{J}^2$ взяты так, что $\det X(\mathcal{J}^1, \mathcal{J}^2)$ максимален среди всех $r \times r$ подматриц X, тогда

$$\|X - \tilde{X}\|_C \leqslant (r+1) \min_{\mathrm{rank}(Y)=r} \|X - Y\|_2 = (r+1)\varepsilon.$$

Хотя точная максимизация объема (детерминанта) является NP-сложной задачей, существует эвристический квази-оптимальный метод, предложенный в [117] (так называемый алгоритм *maxvol*), дающий удовлетворительные результаты в большинстве случаев. Существуют также отдельные оценки для качества крестовой аппроксимации в применении непосредственно к гладким функциям, см. например [219].

2.1.2 Канонический формат и формат Таккера

Обобщение предыдущих соображений на многомерный случай не столь очевидно. Простейшая идея это взять сумму ранг-1 компонентов (2.1), точно так же как мы действуем в размерности два.

Определение 2.1.3. Говорят, что тензор x представлен (или аппроксимирован) в *каноническом* формате, если выполняется

$$x(i_1, i_2, \ldots, i_d) \approx \sum_{\alpha=1}^{R} x^{(1)}_\alpha(i_1) x^{(2)}_\alpha(i_2) \cdots x^{(d)}_\alpha(i_d). \tag{2.4}$$

Это представление, или *формат*, известен с 1920 годов [115]. Диапазон суммирования R называется *каноническим рангом* тензора, и $x^{(k)} \in \mathbb{C}^{n_k \times R}$ образуют канонические *факторы*. В дальнейшем этот подход использовался многократно для структурирования данных, и стал известен и под другими названиями, такими как CANDECOMP, Canonical Polyadic (CP) формат или PARAFAC, см. обзор [154] и ссылки в нем, например [110, 38, 32, 40, 29, 30].

Для некоторых классов тензоров (обычно возникающих при дискретизации интегральных операторов) можно доказать существование малоранговых канонических приближений аналитически [10, 106, 238, 29, 30, 79, 104, 105, 93, 103, 141, 171]. Такие доказательства часто конструктивны и позволяют строить аппроксимации с небольшим R (в пределах десятков–сотен), так что канонический формат обеспечивает сжатие данных до разумных объемов $\mathcal{O}(dnR)$.

Тем не менее, аналитические соображения имеют ограниченную применимость, и обычно дают квази-оптимальные оценки. Более того, не существует надежного метода вычисления CP формата для произвольного заданного тензора (процедуры *сжатия данных*), так как это представление может быть неустойчивым [46]. Для сравнительно узкого класса данных, итерационные методы минимизации ошибки все же применимы (см. [68, 3] и обзоры [30, 154]), но они могут сходиться медленно, и в добавок требовать задания ранга аппроксиманта априори.

Таким образом, мы подходим к главной задаче тензорных аппроксимаций: по заданным элементам тензора (или процедуре, вычисляющей элемент по заданным индексам), построить малопараметрическую аппроксимацию. Поскольку надежный подход известен только в двумерном случае (скелетное, сингулярное разложение), были разработаны различные пути обобщения его на случай многих переменных.

Первым представлением такого типа был формат Таккера [240].

Определение 2.1.4. Говорят, что тензор x представлен (или аппроксимирован) в формате *Таккера*, если выполняется

$$x(i_1, \ldots, i_d) \approx \sum_{\gamma_1, \ldots, \gamma_d} x^{(c)}(\gamma_1, \ldots, \gamma_d) x^{(1)}_{\gamma_1}(i_1) \cdots x^{(d)}_{\gamma_d}(i_d). \tag{2.5}$$

Ранговые индексы γ_k меняются в диапазонах $\gamma_k = 1, \ldots, r_k$, где $r_k = r_k(x)$ называются *рангами Таккера*, или *мультилинейными* рангами, тензор $x^{(c)} \in \mathbb{C}^{r_1 \times \cdots \times r_d}$ называется *ядром Таккера*, и $x^{(k)} \in \mathbb{C}^{n_k \times r_k}$ это *Таккеровские факторы*.

На первый взгляд, этот формат по-прежнему подвержен проклятию размерности: для хранения ядра требуется $\mathcal{O}(r^d)$ элементов. Поэтому, довольно долгое время, представление Таккера использовалось только в маломерных (трех-, четырехмерных) задачах. Однако, формат Таккера обладает несколькими полезными свойствами. Проблема хранения ядра будет решена отдельно, мы рассмотрим ее в Секциях 2.1.3 и 2.2.2. Сейчас обратим внимание на то, почему Таккер формат принципиально ближе к двумерному разложению, чем каноническое разложение.

Первое важное свойство состоит в том, что формат Таккера обладает универсальной процедурой аппроксимации, основанной на сингулярном разложении, см. [43, 44, 45]. Она следует из того, что каждый Таккеровский ранг является на самом

деле рангом определенной матрицы, построенной из элементов исходного тензора. Чтобы показать это, мы сначала введем несколько определений.

Определение 2.1.5. Под *мультииндексом* $\boldsymbol{i} = \boldsymbol{i}_{1,\ldots,k} = \overline{i_1, \ldots, i_k}$ будем понимать индекс, который принимает все возможные сочетания допустимых значений $i_1, \ldots, i_k$, т.е. если $i_m = 1, \ldots, n_m$, $m = 1, \ldots, k$, то[2]

$$\overline{i_1, \ldots, i_k} = i_1 + (i_2 - 1)n_1 + \cdots + (i_k - 1)n_1 \cdots n_{k-1}.$$

Концепция группировки индексов имеет решающее значение в описании многомерных тензоров. Напомним соображения из первой главы: дискретное решение уравнения в частных производных можно рассматривать как тензор $x(i_1, \ldots, i_d)$, но итерационные методы для решения линейной системы на те же данные пишутся с использованием вектора $x(\boldsymbol{i}) = x(\overline{i_1, \ldots, i_d})$.

Семантика $\boldsymbol{i}_{1,\ldots,k}$ позволяет писать образующие наборы индексов более кратко, например,

$$\boldsymbol{i}_{\neq k} = \overline{i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_d}, \qquad \boldsymbol{i}_{>k} = \overline{i_{k+1}, \ldots, i_d}, \tag{2.6}$$

и так далее. Теперь, мы можем определить *матрицы развертки.*

Определение 2.1.6. Пусть дан тензор $x(i_1, \ldots, i_d)$. k-тая Таккеровская матрица развертки задается следующим образом:

$$\begin{aligned} x^{[k]} &= \left[x^{[k]}_{i_k, \boldsymbol{i}_{\neq k}}\right] \in \mathbb{C}^{n_k \times n_1 \cdots n_{k-1} n_{k+1} \cdots n_d}, \quad \text{где} \\ x^{[k]}_{i_k, \boldsymbol{i}_{\neq k}} &= x^{[k]}(i_k, \overline{i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_d}) = x(i_1, \ldots, i_d). \end{aligned}$$

Предположим, что разложение Таккера (2.5) выполняется точно. Введем вспомогательную матрицу

$$V^{[k]}(\gamma_k, \boldsymbol{i}_{\neq k}) = \sum_{\boldsymbol{\gamma}_{\neq k}} x^{(c)}(\boldsymbol{\gamma}) x^{(1)}_{\gamma_1}(i_1) \cdots x^{(k-1)}_{\gamma_{k-1}}(i_{k-1}) x^{(k+1)}_{\gamma_{k+1}}(i_{k+1}) \cdots x^{(d)}_{\gamma_d}(i_d),$$

тогда можно видеть, что разложение Таккера является скелетным разложением для каждой матрицы развертки,

$$x^{[k]} = x^{(k)} V^{[k]}, \qquad x^{(k)} \in \mathbb{C}^{n_k \times r_k}, \quad V^{[k]} \in \mathbb{C}^{r_k \times n_1 \cdots n_{k-1} n_{k+1} \cdots n_d}. \tag{2.7}$$

Это сразу дает первое следствие: если разложение Таккера точно, ранги матриц развертки ограничены Таккеровскими рангами, $\operatorname{rank}(x^{[k]}) \leqslant r_k$, $k = 1, \ldots, d$. На самом деле, мы можем выбрать минимальное представление и превратить последнее выражение в равенство. Второе следствие еще более важно, так как оно решает сформулированную выше задачу аппроксимации.

[2] В данной работе мы используем *little-endian* конвенцию, как, например, в арабских цифрах, Fortran и MATLAB индексации. В большинстве случаев конкретный порядок индексов не важен, достаточно того, что Кронекеровское произведение, введенное в (1.24), согласуется с little-endian порядком группировки индексов.

Теорема 2.1.7. Пусть дан тензор $y(i_1, \ldots, i_d)$, задача минимизации ошибки в формате Таккера с рангами $\boldsymbol{r} = (r_1, \ldots, r_d)$ имеет решение, т.е.[3]

$$\exists x_{min} : \ \|x_{min} - y\| = \inf_x \|x - y\| = \min_x \|x - y\| = \varepsilon,$$

где x (и x_{min}) представляются в виде (2.5) с рангами $r_1, \ldots, r_d$.

Квази-оптимальное приближение x_*, дающее точность

$$\|x_* - y\| \leqslant \sqrt{d}\varepsilon,$$

вычисляется с помощью d сингулярных разложений матриц развертки.

Доказательство. Данная теорема была доказана в [43, 44]. Тем не менее, мы приведем доказательство здесь, т.к. в дальнейшем нам потребуется его обобщение на другие тензорные форматы. Для доказательства первой части, вспомним, что "inf" означает, что существует последовательность x_s такая что $\lim\limits_{s\to\infty} \|y - x_s\| = \varepsilon$. Поскольку все элементы x_s ограничены, существует подпоследовательность x_{s_t}, сходящаяся поточечно к x_{min}. Из представления Таккера (2.5) следует то же самое для матриц развертки, $x^{[k]}_{s_t} \to x^{[k]}_{min}$. Поскольку последовательность матриц с равномерно ограниченным рангом не может сходиться к матрице большего ранга, выполняется $\mathrm{rank}(x^{[k]}_{min}) \leqslant r_k$, и в то же время $\|y^{[k]} - x^{[k]}_{min}\| = \varepsilon$. Это означает, что минимизатор существует в пространстве тензоров с Таккеровскими рангами ограниченными r_k, т.е. доказано первое утверждение теоремы.

Практический вычислительный алгоритм, так называемый *HOSVD* [43, 44], воспроизводит (2.7) в "обратном" направлении. Для каждой матрицы развертки, мы вычисляем неполное сингулярное разложение,

$$y^{[k]}_{i_k, \boldsymbol{i}_{\neq k}} \approx x^{[k]}_{i_k, \boldsymbol{i}_{\neq k}} = \sum_{\gamma_k=1}^{r_k} x^{(k)}_{\gamma_k}(i_k)\sigma_{\gamma_k} V^{[k]}(\gamma_k, \boldsymbol{i}_{\neq k}),$$

с требованием $\|y^{[k]} - x^{[k]}\| \leqslant \varepsilon_k \leqslant \varepsilon$. Ортогональная матрица левых сингулярных векторов принимается за k-й Таккеровский фактор, и на последнем шаге, ядро Таккера восстанавливается как проекция,

$$x^{(c)}(\gamma_1, \ldots, \gamma_d) = \sum_{\boldsymbol{i}} \left(x^{(1)}_{\gamma_1}(i_1)\right)^* \cdots \left(x^{(d)}_{\gamma_d}(i_d)\right)^* y(i_1, \ldots, i_d).$$

Введем следующие матрицы ортогонального проектирования,

$$P_k = \underbrace{I \otimes \cdots \otimes I}_{k-1} \otimes \left(x^{(k)} \left(x^{(k)}\right)^*\right) \otimes \underbrace{I \otimes \cdots \otimes I}_{d-k}.$$

[3]Если не указано иное, мы используем Фробениусову (евклидову) норму для тензоров и векторов, $\|x\| = \|x\|_2 = \|x\|_F = \sqrt{\sum\limits_{\boldsymbol{i}} |x(\boldsymbol{i})|^2}$.

Теперь результат HOSVD алгоритма $x_*(\boldsymbol{i})$ может быть записан как $x_* = P_1 \cdots P_d y$[4]. Добавляя и вычитая некоторые члены, с использованием ортогональности, получаем

$$\begin{aligned} \|y - x_*\|^2 &= \|(y - P_1 y) + (P_1 y - P_1 P_2 y) + (P_1 P_2 y - P_1 P_2 P_3 y) + \cdots \\ &+ (P_1 \cdots P_{d-1} y - P_1 \cdots P_d y)\|^2 \\ &\leqslant \|y - P_1 y\|^2 + \|y - P_2 y\|^2 + \cdots + \|y - P_d y\|^2 \\ &\leqslant \varepsilon_1^2 + \cdots + \varepsilon_d^2, \end{aligned}$$

что немедленно дает второе утверждение теоремы. □

Как и в двухмерном случае, фактическая зависимость рангов Таккера от точности зависит от конкретных данных. Формат Таккера имеет долгую историю применения в обработке данных и изображений (основополагающая статья [240] была посвящена хемометрике; много ссылок можно найти в обзоре [154]).

Важным вкладом в развитие тензорных численных методов для уравнений в частных производных явилось теоретическое обоснование экспоненциальной скорости сходимости приближения в Таккер-формате для тензоров, возникающих при дискретизации аналитических функций [79, 138, 104], в том числе и с конечным количеством точечных особенностей [146]. Так, в последней статье такая экспоненциальная сходимость демонстрируется численно, в частности, в задачах расчета электронной структуры. Сложностью приближений в каноническом и Таккеровском формате при численном решении уравнений в частных производных являются большие модовые размеры n, порядка 10^4–10^5, возникающие при дискретизации функций и операторов на мелких сетках. В этом случае применение стандартных алгоритмов, описанных в [154], затруднительно. В качестве альтернативы были разработаны, например, многосеточные алгоритмы для Таккеровской и канонической аппроксимаций [147].

Для оценки рангов Таккера с использованием свойств гладкости, рассмотрим $f(q_1, \ldots, q_d)$ как одномерную функцию, зависящую от других переменных как параметров, $f^{[k]}_{\boldsymbol{q}\neq k}(q_k) = f(q_1, \ldots, q_d)$. Тогда мы можем применить полиномиальное приближение по q_k, которое для аналитических функций имеет экспоненциально быструю сходимость с увеличением степени многочлена [27, 234]. С другой стороны, $f^{[k]}$ порождает соответствующую матрицу развертки, с ε-рангом ограниченным степенью полинома. Эти соображения позволили доказать следующее общее утверждение для многомерного случая [138, 79].

Теорема 2.1.8. Пусть дана аналитическая функция $f(q_1, \ldots, q_d)$, $\boldsymbol{q} \in \Omega = [-1, 1]^d$, и тензор $x(i_1, \ldots, i_d) = f(q_1(i_1), \ldots, q_d(i_d))$ получен как ее дискретизация на тензорном произведении одномерных сеток. Пусть f по каждой переменой q_k допускает продолжение в эллипс Бернштейна с радиусом ρ_k,

$$\mathcal{E}_{\rho_k} = \left\{ z \in \mathbb{C} : \ |1 + z| + |1 - z| \leqslant \rho_k + \frac{1}{\rho_k} \right\}.$$

Тогда существует ε-аппроксимация к x в формате Таккера, с оценками на ранги вида $r_k \leqslant C|\log(\varepsilon)|/\log(\rho_k)$.

[4]Обратите внимание, что здесь мы использовали те же обозначения для данных, представленных в виде вектора $y = [y(\boldsymbol{i})] \in \mathbb{C}^{n_1 \cdots n_d}$, чтобы сделать матричное произведение корректным.

Мы приведем доказательство и усиленный результат для нового формата на основе Таккера в секции 2.2.2. Формат Таккера успешно применялся для дискретизации и решения интегральных уравнений [8, 7, 146, 142] и задач квантовой химии [148, 147, 246, 132, 139, 236, 202, 133, 136, 135, 134]. Также было разработано много улучшенных методов для Таккер-аппроксимации общего вида, имеющих меньшую вычислительную сложность по сравнению с алгоритмом HOSVD:

- сжатие матриц методом наименьших квадратов [120],
- метод Ньютона для минимизации ошибки [238],
- сжатие из канонического формата в представление Таккера путем комбинации метода наименьших квадратов и многосеточных техник [147], разложения Холецкого [215], а также
- "жадные" и минимизационные алгоритмы, адаптированные для арифметики в формате Таккера (см. секцию 2.1.5), т.е. процедуры редукции ранга для случая, когда тензор является произведением или суммой Таккеровских форматов [200, 214, 201, 89].

Интересно, что точно такое же многомерное представление долгое время использовалось независимо в химическом сообществе под названием Multi Configuration Time Dependent Hartree (MCTDH) метод [178].

Однако, чтобы действительно избежать проклятие размерности, нужен несколько другой подход, основанный на *рекуррентных* разложениях. Исчерпывающая информация изложена в последних книгах и обзорах, см. [102, 97, 145, 143].

2.1.3 Рекуррентные тензорные представления

Формат Таккера отделяет каждую переменную от всех остальных, стартуя всякий раз с исходного тензора, что приводит к d-мерному ядру. Чтобы избавиться от проклятия размерности, нам нужен способ сжатия нескольких переменных в структурированный вид. На первом шаге мы используем *группировку индексов*, чтобы уменьшить число переменных формально, а затем находим для них более узкие подпространства, что и дает фактическое сокращение числа неизвестных. Мы покажем два основных подхода, разработанных в численной линейной алгебре, и основанных на последовательных *вложенных* Таккеровских и скелетных разложениях.

В первом случае, мы можем собрать все индексы попарно, $\boldsymbol{i}_{k,k+1} = \overline{i_k, i_{k+1}}$, получая $d/2$-мерный тензор, и применить разложение Таккера, с целью получения ядра меньшего размера, если итоговые ранги Таккера невелики. Если объем данных $r^{d/2}$ является удовлетворительным, дальнейшие действия не требуются, в противном случае мы повторяем процедуру для *ядра*. Мы группируем в нем индексы попарно и снова вычисляем разложение Таккера, что дает новое ядро, уже размерности $d/4$, и так далее, пока не достигнем тензора небольших размеров, например, двумерного. Из-за такого вложенного применения формата Таккера, полученное представление было названо *Иерархическим Таккером* [108, 94, 164],

или многоуровневым MCTDH (ML-MCTDH) [178]. Поскольку индексы исходного тензора могут иметь особый смысл (например, дискретизованные координаты), первый шаг разложения Таккера выполняется обычным способом, без бинарного группирования.

Определение 2.1.9. Говорят, что тензор x представлен (или аппроксимирован) в (бинарном) иерархическом формате Таккера (HT), если существует следующая цепочка тензоров, каждый из которых является ядром Таккера для последующего:

$$x^{(L+1)}(\gamma_1^{L+1}, \ldots, \gamma_{d_{L+1}}^{L+1}) \approx \sum_{\gamma_1^1, \ldots, \gamma_{d_L}^L} \prod_{l=1}^{L} \prod_{k_l=1}^{d_l} x_{\gamma_{k_l}^l}^{(l,k_l)}(\gamma_{2k_l-1}^{l+1}, \gamma_{2k_l}^{l+1}), \tag{2.8}$$

где *уровни* меняются в диапазоне $L = 1, \ldots, \lceil \log_2 d \rceil - 1$, причем последний уровень отвечает исходному ядру Таккера в разложении (2.5), $x^{(\lceil \log_2 d \rceil)} = x^{(c)}$; размерности редуцируются по рекурсии $d_{\lceil \log_2 d \rceil} = d$, $d_L = \lceil d_{L+1}/2 \rceil$, и определяющие тензоры $x^{(l,k_l)} \in \mathbb{C}^{r_{k_l}^l \times r_{2k_l-1}^{l+1} \times r_{2k_l}^{l+1}}$ называются *иерархическими (HT) факторами*, с *HT рангами* $r_{k_l}^l$. Если $2k_l < d_{l+1}$, считается, что последний индекс принимает единственное значение $\{1\}$.

Факторы являются главным элементом такого разложения, поскольку они хранятся в качестве всего лишь трехмерных массивов. Они определяют отображение пространства размером $r_{2k_l-1}^{l+1} r_{2k_l}^{l+1}$, заданного прямым произведением двух переменных, в пространство размера $r_{k_l}^l$ на предыдущем уровне. Каждый ранг $r_{k_l}^l$ является таким образом рангом скелетного разложения между некоторой группой исходных переменных (отвечающих данной ветви дерева) и всеми остальными индексами. Если ранги могут быть ограничены разумной константой, иерархический формат требует $\mathcal{O}(dnr + dr^3)$ неизвестных (первый член отвечает исходным Таккеровским факторам, второй иерархическим), так что затраты памяти, по сравнению с n^d, существенно сокращаются.

Разумеется, ранги зависят от порядка включения размерностей в дерево (т.е. конкретной группировки индексов). Легко построить примеры, когда один порядок приводит к значительно меньшим рангам, чем другой. Тем не менее, в любом дереве есть ребра, соответствующие разделению переменных примерно напополам, которые, с большой вероятностью, будут отвечать наибольшим значениям ранга. Практический опыт также позволяет полагать, что принципиально важным является *порядок* индексов (а он может быть определен посредством физических соображений, или с использованием адаптивных алгоритмов, см. например [17]), тогда как группировку размерностей в большинстве случаев достаточно вести в соответствии с деревом *линейного* вида.

Определение 2.1.10. Говорят, что тензор x представлен (или аппроксимирован) в формате *Matrix Product States* (MPS), или *Tensor Train* (TT), если выполняется

$$x(i_1, \ldots, i_d) \approx \sum_{\alpha_1, \ldots, \alpha_{d-1}} x_{\alpha_1}^{(1)}(i_1) x_{\alpha_1, \alpha_2}^{(2)}(i_2) \cdots x_{\alpha_{d-2}, \alpha_{d-1}}^{(d-1)}(i_{d-1}) x_{\alpha_{d-1}}^{(d)}(i_d), \tag{2.9}$$

где $x^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}$ называются ТТ *ядрами* (*блоками*, или *факторами*), диапазоны $r_k = r_k(x) \leqslant r(x)$ *ранговых индексов* $\alpha_k = 1, \ldots, r_k$ называются *ТТ рангами*. Для однообразия записи можно полагать $r_0 = r_d = 1$.

Любопытно, что это представление было независимо открыто несколько раз: как MPS в квантовой физике с 1980'х [211, 70, 152, 249, 175], и как ТТ в 2009 в вычислительной линейной алгебре [203, 194, 197]. Термин Matrix Product States проистекает из того, что если мы зафиксируем индексы $i_1, \ldots, i_d$, ТТ блоки превращаются в $r_{k-1} \times r_k$ матрицы, так что элемент x записывается в виде матричного произведения,

$$x(i_1, \ldots, i_d) = x^{(1)}(i_1) \cdots x^{(d)}(i_d).$$

Обратите внимание, что прямое произведение (тензор ранга 1) является тензором ранга 1 во всех форматах: если $r_{k-1} = r_k = 1$, предыдущее представление является произведением d чисел, и совпадает с каноническим форматом (2.4) с рангом $R = 1$, и с форматом Таккера (2.5) с рангами 1 и ядром $x^{(c)} = 1$. С другой стороны, фиксируя ранговые индексы, можем записать

$$x = \sum_{\alpha_1, \ldots, \alpha_{d-1}} x^{(1)}_{\alpha_1} \otimes x^{(2)}_{\alpha_1, \alpha_2} \otimes \cdots \otimes x^{(d-1)}_{\alpha_{d-2}, \alpha_{d-1}} \otimes x^{(d)}_{\alpha_{d-1}}, \tag{2.10}$$

где $\otimes$ обозначает Кронекеровское произведение (см. 1.24).

Кроме линейной структуры, еще одно интересное отличие от (иерархического) формата Таккера заключается в том, что ни один из ТТ блоков не может быть однозначно классифицирован как "фактор" или "ядро" (таким образом, термины "блок", "фактор" и "ядро" здесь взаимозаменяемы), так как все они несут и исходные, и ранговые индексы. Это упрощает как аналитические построения, так и реализации алгоритмов. Асимптотические затраты памяти для ТТ формата составляют $\mathcal{O}(dnr^2)$. Для небольших диапазонов исходных индексов i_k (т.н. *модовых размеров* n) и более высоких рангов, ТТ формат содержит меньше неизвестных, чем НТ представление, что делает его особенно эффективным для описания, например, спиновых систем. Если модовые размеры велики, можно применить ТТ формат только для Таккеровского ядра, так называемое *расширенное ТТ* разложение [204], которое является частным случаем НТ формата с линейным деревом.

В дальнейшем, мы будем рассматривать существующие и предлагать новые вычислительные методы и построения в тензорных форматах. За исключением разделов, посвященных новому комбинированному формату (секции 2.2.2–2.2.5), мы будем описывать анализ и алгоритмы на базе ТТ формата. Его простота обеспечивает определенную элегантность обозначений, что особенно важно в многомерных задачах из-за обилия индексов.

2.1.4 Обозначения для работы с тензорными форматами

Для начала, рассмотрим задачу умножения матрицы на вектор, который может быть ассоциирован с многомерным тензором:

$$y = Ax, \quad x, y \in \mathbb{C}^{n^d}, \text{ т.е. } \quad y(\boldsymbol{i}) = \sum_{\boldsymbol{j}} A(\boldsymbol{i}, \boldsymbol{j}) x(\boldsymbol{j}),$$

где $\boldsymbol{i} = \overline{i_1, \dots, i_d}$, и $\boldsymbol{j} = \overline{j_1, \dots, j_d}$. Пусть x представлен в ТТ формате. Мы хотели бы получить результат y также в ТТ формате с разумной сложностью. В каком представлении следует записать A? Простейшая идея могла бы состоять в использовании $2d$-мерного ТТ формата, $A^{(1)}(i_1) \cdots A^{(2d)}(j_d)$. Однако, при этом во всех нетривиальных случаях (даже для единичной матрицы $A = I$) средний ТТ ранг будет равен матричному рангу, т.е. $r_d = \operatorname{rank}(A) = n^d$, что делает вычисления невозможными.

Таким образом, ТТ формат для матриц, также известный как *Matrix Product Operator* [222], вводится с использованием перестановки индексов,

$$A(i_1, \dots, i_d, j_1, \dots, j_d) = \sum_{\gamma_1, \dots, \gamma_{d-1}} A^{(1)}_{\gamma_1}(i_1, j_1) A^{(2)}_{\gamma_1, \gamma_2}(i_2, j_2) \cdots A^{(d)}_{\gamma_{d-1}}(i_d, j_d). \tag{2.11}$$

Тогда произведение матрицы на вектор пишется независимо для каждого ТТ блока: результат получается в аналогичном ТТ формате $y(\boldsymbol{i}) = y^{(1)}(i_1) \cdots y^{(d)}(i_d)$, где

$$y^{(k)}_{\beta_{k-1}, \beta_k} = A^{(k)}_{\gamma_{k-1}, \gamma_k} x^{(k)}_{\alpha_{k-1}, \alpha_k}, \quad \text{т.е.} \quad y^{(k)}_{\beta_{k-1}, \beta_k}(i_k) = \sum_{j_k} A^{(k)}_{\gamma_{k-1}, \gamma_k}(i_k, j_k) x^{(k)}_{\alpha_{k-1}, \alpha_k}(j_k), \tag{2.12}$$

причем $\beta_k = \overline{\alpha_k, \gamma_k} = 1, \dots, r_k(A) r_k(x)$, $k = 0, \dots, d$. Более того, эта концепция естественным образом сводится к стандартному Кронекеровскому произведению матриц, если все ТТ ранги равны 1. В противном случае, ТТ ранги матрицы и вектора перемножаются, то есть результат записывается с рангами $r_k(y) = r_k(A) r_k(x)$. Если ранги сомножителей $r(A)$ и $r(x)$ небольшие, ТТ формат для y также требует разумного объема памяти.

В дальнейшем, для единого описания форматов (2.9) и (2.11), мы будем использовать следующее определение, являющееся несколько расширенной версией обозначений из работы [212].

Определение 2.1.11. Пусть даны ТТ форматы $\{x^{(k)}(i_k)\}$ или $\{A^{(k)}(i_k, j_k)\}$. Векторное *ТТ отображение* раскрывает ТТ представление в полный тензор следующим образом:

$$\tau(x^{(p)}, \dots, x^{(q)}) : \{x^{(k)}\}_{k=p}^{q} \to x^{(p, \dots, q)} \in \mathbb{C}^{r_{p-1} \times n_p \cdots n_q \times r_q}, \quad \text{где}$$

$$x^{(p, \dots, q)}_{\alpha_{p-1}, \alpha_q}(\overline{i_p, \dots, i_q}) = \sum_{\alpha_p, \dots, \alpha_{q-1}} x^{(p)}_{\alpha_{p-1}, \alpha_p}(i_p) \cdots x^{(q)}_{\alpha_{q-1}, \alpha_q}(i_q),$$

при $1 \leqslant p \leqslant q \leqslant d$. Для граничных случаев мы можем дополнительно обозначать

$$x^{(1, \dots, q)} = x^{(\leqslant q)} = x^{(<q+1)} \in \mathbb{C}^{n_1 \cdots n_q \times r_q}, \quad x^{(p, \dots, d)} = x^{(\geqslant p)} = x^{(>p-1)} \in \mathbb{C}^{r_{p-1} \times n_p \cdots n_d},$$

и наконец $x^{(1, \dots, d)} = x \in \mathbb{C}^{n_1 \cdots n_d}$.

Матричное ТТ отображение пишется как

$$\tau(A^{(p)}, \dots, A^{(q)}) : \{A^{(k)}\}_{k=p}^{q} \to A^{(p, \dots, q)} \in \mathbb{C}^{r_{p-1} \times n_p \cdots n_q \times m_p \cdots m_q \times r_q}, \quad \text{где}$$

$$A^{(p, \dots, q)}_{\gamma_{p-1}, \gamma_q}(\overline{i_p, \dots, i_q}, \overline{j_p, \dots, j_q}) = \sum_{\gamma_p, \dots, \gamma_{q-1}} A^{(p)}_{\alpha_{p-1}, \alpha_p}(i_p, j_p) \cdots A^{(q)}_{\gamma_{q-1}, \gamma_q}(i_q, j_q),$$

с аналогичными граничными соглашениями.

Отображение τ можно использовать для извлечения TT *куска*. Эта операция будет широко применяться в оптимизационных методах переменных направлений. Кроме того, обратите внимание, что в образе τ мы всегда объединяем начальные индексы i_k в мультииндекс, что позволяет писать корректные матричные произведения (в частности, граничные куски являются просто матрицами).

Еще один полезный тип группировки индексов может быть связан с концепцией подпространств в HT формате (2.8): вспомним, что фактор $x^{(l,k_l)}$ задает отображение $\mathbb{C}^{r^{l+1}_{2k_l-1} r^{l+1}_{2k_l}} \to \mathbb{C}^{r^l_{k_l}}$, которое в свою очередь может быть описано матрицей, а не 3-мерным массивом. То же соображение может быть применено к TT факторам. Матрицы соответствующих отображений будем обозначать следующим образом.

Определение 2.1.12. Для данного TT блока $x^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}$, определим следующие группировки элементов:

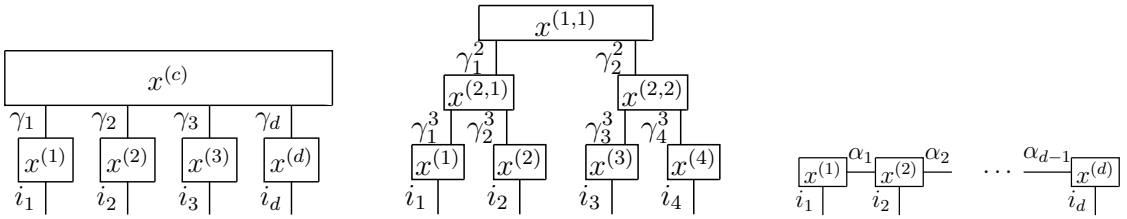
- *Левая развертка блока*: $x^{|k\rangle}(\overline{\alpha_{k-1}, i_k}, \alpha_k) = x^{(k)}_{\alpha_{k-1},\alpha_k}(i_k), \quad x^{|k\rangle} \in \mathbb{C}^{r_{k-1} n_k \times r_k}$.
- *Правая развертка блока*: $x^{\langle k|}(\alpha_{k-1}, \overline{i_k, \alpha_k}) = x^{(k)}_{\alpha_{k-1},\alpha_k}(i_k), \quad x^{\langle k|} \in \mathbb{C}^{r_{k-1} \times n_k r_k}$.
- *Центральная развертка*: $x^{|k|}(i_k, \overline{\alpha_{k-1}, \alpha_k}) = x^{(k)}_{\alpha_{k-1},\alpha_k}(i_k), \quad x^{|k|} \in \mathbb{C}^{n_k \times r_{k-1} r_k}$.

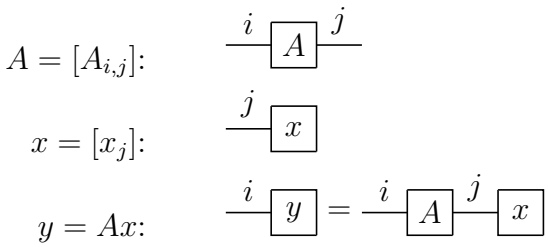
Здесь важно понимать, что все развертки указывают на одни и те же данные, хранящиеся в TT блоке – сравните с известной концепцией *указателя* в программировании. Это позволит нам, например, писать равенства вида $x^{|k\rangle} = Q$, опуская избыточные переопределения $x^{(k)}_{\alpha_{k-1},\alpha_k}(i_k) = x^{|k\rangle}(\overline{\alpha_{k-1}, i_k}, \alpha_k)$, поскольку любая из разверток задает все остальные автоматически.

В качестве четвертой, "скрытой" группировки данных, мы будем использовать одно и то же обозначение $x^{(k)}$ для элементов TT блока как в виде 3-мерного массива, так и вытянутых в вектор $\left[x^{(k)}(\overline{\alpha_{k-1}, i_k, \alpha_k})\right]$, точно так же, как мы не различаем $x(\boldsymbol{i})$ и $x(i_1, \ldots, i_d)$. Конкретное значение $x^{(k)}$ однозначно определяется из контекста. Так, $x^{(k)}(i_k)$ в TT представлении (2.9) по-прежнему является двумерной срезкой трехмерного тензора, тогда как матричное произведение вида $A_k x^{(k)}$ подразумевает, что элементы $x^{(k)}$ вытянуты в столбец.

В физическом сообществе популярен другой типа обозначений, т.н. *диаграммы тензорных сетей* [250, 175, 119]. Они особенно удобны для работы со сложными представлениями. В формализме диаграмм, любой массив обозначается в виде блока (например, прямоугольника), любой индекс соответствует линии, и если линия соединяет два блока, это означает, что мы умножаем элементы соответствующих массивов и суммируем по данному общему индексу. Если индекс присоединен только к одному блоку, он является "свободным", и если мы пишем уравнение, предполагается что оно выполняется для всех возможных значений свободных индексов. Например, матрица, вектор, и их произведение могут быть записаны следующим образом:

Рис. 2.1: Форматы Таккера (слева), HT (посередине), и TT/MPS (справа) в диаграммах тензорных сетей





Теперь мы можем перейти к более сложным тензорным форматам. Так, разложения Таккера, HT и TT можно описать как показано на рис. 2.1.

2.1.5 Основные операции в TT формате

Данная секция является обзорной. Построения и алгоритмы приведены в соответствии с [221, 222, 197].

В предыдущей секции мы видели, как пишется матричное произведение для TT представлений (2.12). Другие полилинейные операции могут быть также перенесены по аналогии с канонического формата. Например, чтобы умножить тензор на скаляр, достаточно умножить любой из его TT блоков.

Сумма двух тензоров объединяет диапазоны ранговых индексов, так что $z = x + y$ представляется в TT формате со следующими блоками:

$$z^{(1)}(i_1) = \begin{bmatrix} x^{(1)}(i_1) & y^{(1)}(i_1) \end{bmatrix}, \quad z^{(d)}(i_d) = \begin{bmatrix} x^{(d)}(i_d) \\ y^{(d)}(i_d) \end{bmatrix}, \quad z^{(k)}(i_k) = \begin{bmatrix} x^{(k)}(i_k) & \\ & y^{(k)}(i_k) \end{bmatrix}$$

для $k = 2, \ldots, d-1$. Очевидно, суммирование складывает TT ранги, $r_k(z) = r_k(x) + r_k(y)$, $k = 1, \ldots, d-1$.

Замечание 2.1.13. Эта процедура может быть использована для преобразования канонического формата (2.4) в TT. Действительно, каждый ранг-1 член канонического формата можно рассматривать как TT тензор ранга 1, а суммирование записать как показано выше. Отсюда следует, что если x задан в канонической форме, для него существует и TT формат с ограниченными рангами, $r_k(x) \leqslant R$.

Диагональная TT матрица строится из вектора в TT формате без изменения TT рангов, путем вытягивания вектора в диагональ по каждому индексу i_k,

$$A^{(k)}(i_k, j_k) = x^{(k)}(i_k)\delta_{i_k,j_k}, \quad A = \tau(A^{(1)}, \ldots, A^{(d)}) = \operatorname{diag}(x) = \operatorname{diag}\left(\tau\left(x^{(1)}, \ldots, x^{(d)}\right)\right).$$

Алгоритм 1 Скалярное произведение в TT формате [197]

Ввод: Тензоры (векторы) x, y в TT формате.
Вывод: Произведение $s = (x, y)$.

1: Положить $s_0 = 1$.
2: **for** $k = 1, \ldots, d$ **do**
3: $\quad z^{\langle k|} = s_{k-1} y^{\langle k|} \in \mathbb{C}^{r_{k-1}(x) \times n_k r_k(y)}$.
4: $\quad s_k = \left(x^{|k\rangle}\right)^* z^{|k\rangle} \in \mathbb{C}^{r_k(x) \times r_k(y)}$.
5: **end for**
6: **return** $s = s_d \in \mathbb{C}^{r_d(x) \times r_d(y)} = \mathbb{C}$.

Выполняя эту операцию наоборот, мы также можем извлечь диагональ из TT-матрицы в формате TT-вектора.

Поточечное (Адамарово) произведение векторов $z = x \odot y$ может быть записано как произведение диагональной матрицы на вектор, $x \odot y = \operatorname{diag}(x) y$, т.е., применяя (2.12), получаем:

$$z = \tau\left(z^{(1)}, \ldots, z^{(d)}\right), \quad z^{(k)}_{\beta_{k-1}, \beta_k} = x^{(k)}_{\gamma_{k-1}, \gamma_k} \odot y^{(k)}_{\alpha_{k-1}, \alpha_k},$$

что, как и в случае матричного произведения, приводит к перемножению TT рангов, $\beta_k = \overline{\alpha_k, \gamma_k} = 1, \ldots, r_k(z) = r_k(x) r_k(y)$.

Скалярное (внутреннее) произведение двух векторов, $s = (x, y)$, равно произведению всех TT элементов обоих векторов с последующим суммированием по всем ранговым и исходным индексам. Однако, произвольный порядок вычислений может потребовать сложности $\mathcal{O}(dnr^2(x)r^2(y)) = \mathcal{O}(dnr^4)$. Правильно используя развертки блоков (Определение 2.1.12) и вспомогательные тензоры, скалярное произведение можно вычислить за $\mathcal{O}(dn(r^2(x)r(y) + r(x)r^2(y)) = \mathcal{O}(dnr^3)$ операций, как показано в Алгоритме 1.

Теперь давайте сосредоточимся на главном преимуществе тензорных деревьев по сравнению с каноническим форматом – процедуре **рекомпрессии** или **округления**. Мы видели, что алгебраические операции могут увеличивать TT ранги, при этом они могут получаться завышенными для данных тензоров и точностей. TT округление является аналогом алгоритма HOSVD для понижения рангов до квазиоптимальных значений при заданной точности. Пусть дан тензор x, его приближение (обозначим здесь как тензор y) будем писать так:

$$y = \mathcal{T}_\varepsilon(x), \quad y = \mathcal{T}_r(x), \quad y = \mathcal{T}_{\varepsilon, r}(x),$$

где индекс $\mathcal{T}$ обозначает стратегию сжатия:

- точность ε: требуем $\|y - x\| \leqslant \varepsilon \|x\|$, ранги наименьшие возможные, но в принципе не ограничены.
- ранг r: ограничиваем $r(y) \leqslant r$, и ищем приближение квазиоптимальной точности.
- точность/ранг ε, r: где возможно, удерживаем точность, но ограничиваем $r_k(y) = r$ если порог ε требует большего значения.

Итак, опишем процедуру TT округления в соответствии с [205].

Мы начнем с TT версии определения матриц развертки.

Определение 2.1.14. Пусть дан тензор $x(i_1, \dots, i_d)$. k-той матрицей развертки для TT формата называем матрицу

$$\begin{aligned} x^{\{k\}} &= \left[x^{\{k\}}_{\boldsymbol{i}_{\leqslant k}, \boldsymbol{i}_{>k}}\right] \in \mathbb{C}^{n_1 \cdots n_k \times n_{k+1} \cdots n_d}, \quad \text{где} \\ x^{\{k\}}_{\boldsymbol{i}_{\leqslant k}, \boldsymbol{i}_{>k}} &= x^{\{k\}}(\overline{i_1, \dots, i_k}, \overline{i_{k+1}, \dots, i_d}) = x(i_1, \dots, i_d). \end{aligned}$$

Если TT представление (2.9) точно, полилинейность сразу дает равенство $r_k = \operatorname{rank}(x^{\{k\}})$. В приближенных вычислениях, пусть дан $x(i_1, \dots, i_d) = x^{(1)}(i_1) \cdots x^{(d)}(i_d)$ с TT рангами $\hat{r}_k$ (возможно, завышенными, или вообще полными), стоит задача определения (квази) оптимальных рангов r_k, дающих ε-аппроксимацию для k-той матрицы развертки. Начиная с $k = d$, представляем

$$x^{\{d\}}_{\boldsymbol{i}_{<d}, i_d} = \sum_{\alpha_{d-1}=1}^{\hat{r}_{d-1}} x^{(<d)}_{\alpha_{d-1}}(\boldsymbol{i}_{<d}) x^{(d)}_{\alpha_{d-1}}(i_d), \quad x^{(<d)} = \tau(x^{(1)}, \dots, x^{(d-1)}). \tag{2.13}$$

Если бы мы имели QR разложение

$$x^{(<d)} = QR, \quad Q^*Q = I,$$

достаточно было бы вычислить сингулярное разложение от матрицы меньших размеров $(\hat{r}_{d-1} \times n_d)$,

$$\left(Rx^{\langle d|}\right) \approx U\Sigma V.$$

Оставляя только r_{d-1} старших сингулярных значений, получаем немедленно $y^{\langle d|} = V$. Однако, $x^{(<d)}$ является матрицей очень большого размера, и ее ортогонализацию следует вычислять в структурированном виде.

Определение 2.1.15. Говорят, что TT блок $x^{(k)}$ *лево-* или *право-ортогонален*, если верно

$$\left(x^{|k\rangle}\right)^* x^{|k\rangle} = I, \quad \text{или} \quad x^{\langle k|} \left(x^{\langle k|}\right)^* = I,$$

соответственно.

Обратите внимание, что для первого и последнего блоков, левая и правая ортогональности означает обыкновенные ортогональности по тензорным индексам i_1 и i_d, так же, как и в скелетном разложении матрицы. Поскольку TT представление не единственно,

$$x(\boldsymbol{i}) = \left(x^{(1)}(i_1)R_1\right)\left(R_1^{-1}x^{(2)}(i_2)R_2\right)R_2^{-1} \cdots R_{d-1}\left(R_{d-1}^{-1}x^{(d)}(i_d)\right)$$

для любых невырожденных матриц R_k подходящих размеров, требования ортогональности можно удовлетворить без изменения исходного тензора. В самом деле, соседние TT блоки $x^{|k\rangle}x^{\langle k+1|}$ составляют скелетное разложение, которое может быть ортогонализовано или как

$$q^{|k\rangle}\left(Rx^{\langle k+1|}\right), \quad \text{, или как} \quad \left(x^{|k\rangle}L\right)q^{\langle k+1|},$$

Алгоритм 2 Левая TT ортогонализация [197]

Ввод: Тензор x в TT формате.

Вывод: Тензор x со всеми лево-ортогональными блоками кроме $x^{(d)}$.

1: **for** $k = 1, \ldots, d-1$ **do**
2: QR разложение $x^{|k\rangle} = q^{|k\rangle} R$, $\left(q^{|k\rangle}\right)^* q^{|k\rangle} = I$.
3: Заменить $x^{\langle k+1|} = R x^{\langle k+1|}$, $x^{|k\rangle} = q^{|k\rangle}$.
4: **end for**

где $q^{|k\rangle} R = x^{|k\rangle}$, и $Lq^{\langle k+1|} = x^{\langle k+1|}$, соответственно. Повторяя эту процедуру для всех блоков (см. Алгоритмы 2 и 3), получаем TT представление с соответствующей ортогональностью. Эта процедура требует всего $\mathcal{O}(dnr^3)$ операций, но гарантирует ортогональность TT кусков любой длины.

Лемма 2.1.16. Если в TT тензоре блоки $1, \ldots, d-1$ лево-ортогональны, выполняется

$$\left(x^{(<p)}\right)^* x^{(<p)} = I, \quad p = 2, \ldots, d.$$

Доказательство. Произведение

$$\left(\left(x^{(<p)}\right)^* x^{(<p)}\right)_{\alpha_{p-1}, \beta_{p-1}} = \sum_{i_1, \ldots, i_{p-1}} \bar{x}^{(<p)}_{\alpha_{p-1}}(\overline{i_1, \ldots, i_{p-1}}) \cdot x^{(<p)}_{\beta_{p-1}}(\overline{i_1, \ldots, i_{p-1}})$$

может быть вычислено как незаконченное скалярное произведение (Алг. 1) с $y = x$, при остановке процесса на s_{p-1}. Если $x^{(1)}$ лево-ортогональный, из первого шага следует $s_1 = I$. Пусть $s_{k-1} = I$, тогда $z^{\langle k|} = x^{\langle k|}$, где $x^{|k\rangle}$ также лево-ортогонален для $k < p$. Следовательно, s_k является единичной матрицей, и утверждение теоремы следует по индукции. □

Теперь ясно, как построить процедуру округления в TT формате: на первом шаге, мы запускаем Алг. 2, накладывая условия левой ортогональности на весь TT формат, в частности, $x^{(<d)}$ в (2.13). После этого, вычисляем SVD небольшого размера, $\left(Rx^{\langle d|}\right) \approx U\Sigma V$ и перебрасываем $U\Sigma$ на блок $x^{(d-1)}$. Инициализация $y^{\langle d|} = V$ обеспечивает его право-ортогональность, и также по-прежнему присутствует левая ортогональность $x^{(<d-1)}$. Так что мы можем использовать сингулярное разложение на $x^{\langle d-1|}$, и так далее. Полная процедура приведена в Алгоритме 4.

В диаграммах, левая, и соответственно правая ортогональность в TT формате может быть показана как на рис. 2.2: при суммировании блока, умноженного самого на себя, по индексам, выходящих из темной части прямоугольника, получаем единичную матрицу по индексам, выходящим из белой части блока. На рис. 2.2 (справа), уменьшенные размеры блоков обозначают редуцированные TT ранги после процедуры аппроксимации.

Также как и QR разложение, каждое SVD от $\hat{r} \times n\hat{r}$ матрицы требует $\mathcal{O}(n\hat{r} \cdot \hat{r}^2)$ операций, что приводит к общей сложности $\mathcal{O}(dn\hat{r}^3)$. Квазиоптимальность погрешности $\|x - y\|^2 \leqslant \|x\|^2 \sum \varepsilon_k^2$ может быть доказано с помощью тех же самых проекционных аргументов, которые были использованы в HOSVD теореме 2.1.7.

Алгоритм 3 Правая TT ортогонализация [197]

Ввод: Тензор x в TT формате.

Вывод: Тензор x со всеми право-ортогональными блоками кроме $x^{(1)}$.

1: **for** $k = d, d-1, \ldots, 2$ **do**
2: LQ разложение $x^{\langle k|} = Lq^{\langle k|}$, $q^{\langle k|}\left(q^{\langle k|}\right)^* = I$.
3: Заменить $x^{|k-1\rangle} = x^{|k-1\rangle}L, \quad x^{\langle k|} = q^{\langle k|}$.
4: **end for**

Алгоритм 4 TT округление (справа налево) [197]

Ввод: Тензор x в TT формате, точности ε_k, $k = 1, \ldots, d-1$.

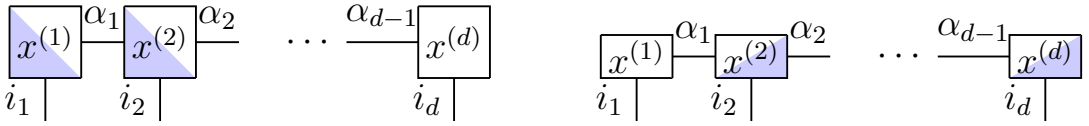
Вывод: Тензор $y : \|x - y\|^2 \leqslant \|x\|^2 \sum \varepsilon_k^2$ с оптимальными рангами.

1: Выполнить ортогонализацию x слева, используя Алг. 2.
2: **for** $k = d, d-1, \ldots, 2$ **do**
3: Вычислить SVD $x^{\langle k|} = U \operatorname{diag}(\sigma) V$.
4: Подобрать ранг r_{k-1} : $\sum_{\beta > r_{k-1}} \sigma_\beta^2 \leqslant \varepsilon_{k-1}^2 \|\sigma\|^2$.
5: Выделить старшие сингулярные вектора и числа: $\tilde{U}_\beta = U_\beta$, $\tilde{V}_\beta = V_\beta$, $\tilde{\sigma}_\beta = \sigma_\beta$, $\beta = 1, \ldots, r_{k-1}$.
6: Заменить $x^{|k-1\rangle} = x^{|k-1\rangle} \cdot \tilde{U} \operatorname{diag}(\tilde{\sigma}), \quad y^{\langle k|} = \tilde{V}$.
7: **end for**
8: $y^{|1\rangle} = x^{|1\rangle}$.

Если все элементы тензора даны, и мы хотели бы сжать его в TT формате, можно использовать адаптированную версию Алгоритма 4, опуская шаг левой ортогонализации (он обеспечится автоматически в процессе сингулярного разложения) и заменяя блок $x^{\langle k|}$ на большой тензор $X^{\langle k|} = \left[X^{\langle k|}(\overline{i_1, \ldots, i_{k-1}}, \overline{i_k, \alpha_k})\right]$ для $k = 1, \ldots, d-1$, и $X^{\langle d|} = x^{\{d\}}$.

Однако, алгоритм на основе SVD может быть слишком медленным в задаче сжатия не только полного тензора, но и уже TT-структурированного, при условии очень высоких рангов. Такой случай может возникать, например, при умножении матрицы на вектор (2.12) с $r(A) \sim r(x) \lesssim r$, так что $r(y) \sim r^2$, и процедура округления требует асимптотически $\mathcal{O}(dnr^6)$ операций. Для этой ситуации существует более быстрые, но эвристические методы. Например, можно использовать различные многомерные обобщением крестовой интерполяции (2.3) для восстановления тензора по небольшому количеству элементов [205, 217, 67, 22, 18, 216]. Если доступна быстрая процедура для вычисления *неполного* скалярного произведения исходного тензора и аппроксиманта (аналогичная использованной в Лемме 2.1.16), например, при вычислении приближенного произведения матрицы на вектор [196], может быть предложен более надежный подход наименьших квадратов в переменных направлениях (Alternating Least Squares, ALS). Эта концепция оказалось очень плодотворной, но для создания по-настоящему надежного и адаптивного алгоритма требуются дополнительные действия. Мы рассмотрим их подробно в главе 4.2.

Рис. 2.2: Диаграммы состояний TT формата после Алг. 2 (слева) и Алг. 4 (справа).



2.2 Квантизованные тензорные аппроксимации

2.2.1 Формат QTT: Quantized Tensor Train

Данная секция представляет собой введение в подход искусственной тензоризации, предложенный Оселедцем [195] и Хоромским [144].

Хотя тензорные форматы изначально были разработаны как инструмент для существенно многомерных массивов, линейная сложность TT формата по отношению к размерности побуждает применить его и для данных "низкой" размерности, например, возникающих при дискретизации одномерных или двумерных уравнений в частных производных. Мы уже видели, как тензоры, порождаемые функциями многих переменных, могут рассматриваться как векторы или матрицы (см. Определение 2.1.12). Это стандартная операция развертки, т.е. группировки индексов. Однако, та же процедура может быть проведена и в обратном направлении: можно смотреть на вектор как на многомерный тензор и приблизить его в малоранговом формате. Какая степень сжатия может быть при этом достигнута?

Рассмотрим для краткости одномерный вектор, $x = [x(i)]$. Пусть число допустимых значений для i составляет степень 2, т.е. $n = 2^L$. Рассмотрим позиционную запись i в двоичной системе исчисления, т.е.

$$i = \sum_{l=1}^{L} i_l \cdot 2^{l-1}, \quad i_l \in \{0, 1\}. \tag{2.14}$$

Это соответствует перегруппировке вектора в тензор с L виртуальными (*квантизованными*) размерностями. Теперь можно применить TT разложение:

$$x(i) = x(i_1, \ldots, i_L) \approx x^{(1)}(i_1) \cdots x^{(L)}(i_L).$$

Если TT ранги этого тензора малы, требуемый объем памяти составляет примерно *логарифм* от исходного, $\mathcal{O}(L \cdot 2 \cdot r^2) = \mathcal{O}(\log n)$. Этот метод был предложен в [195] для $2^L \times 2^L$ матриц, и его эффективность подтверждена численными экспериментами. Для векторов и тензоров более высоких размерностей, подход был расширен и теоретически обоснован в [140, 144], под названием *Quantized Tensor Train* (QTT) формат. Название Quantized связано с термином "квант", минимально возможный долей информации, получаемой при задании каждой цифры i_l.

При работе с малопараметрическими представлениями данных, неизбежно задается вопрос о конкретных значения TT (или QTT) рангов, r. Для многих одномерных векторов, порождаемых функциями, были обнаружены элегантные аналитические QTT представления. Мы приведем здесь наиболее простые и важные примеры, обеспечивающие теоретическую основу QTT приближений. Более

сложные аналитические TT структуры, полученные автором в ходе собственных исследований, будут показаны в следующей главе.

1. Экспоненциальный вектор: все QTT ранги 1 [144]:

$$\exp(\kappa i) = \mathrm{e}^{(1)}(i_1)\cdots\mathrm{e}^{(L)}(i_L), \quad \mathrm{e}^{(l)}(i_l) = \exp(\kappa i_l \cdot 2^{l-1}),$$

индексы $i_l \in \{0,1\}$ соответствуют (2.14). Т.о., требуется всего $2L = 2\log_2 n$ чисел для точного задания всех 2^L элементов вектора.

2. Единичный вектор (QTT ранги 1) [144]:

$$e_j(i) = \delta_{i,j} = \delta^{(1)}(i_1)\cdots\delta^{(L)}(i_L), \quad \delta^{(l)}(i_l) = \delta_{i_l,j_l}, \quad j = \sum_{l=1}^{L} j_l \cdot 2^{l-1}, \quad \delta_{i,j} = \begin{cases} 1, & i=j, \\ 0, & i\neq j. \end{cases}$$

3. Синус $\sin(\kappa i + \phi) = s^{(1)}(i_1)\ldots s^{(L)}(i_l)$ имеет QTT ранги 2 [144]:

$$s^{(1)}(i_1) = \begin{bmatrix}\sin(\kappa i_1 + \phi) & \cos(\kappa i_1 + \phi)\end{bmatrix}, \quad s^{(L)}(i_L) = \begin{bmatrix}\cos(\kappa i_L \cdot 2^{L-1}) \\ \sin(\kappa i_L \cdot 2^{L-1})\end{bmatrix},$$
$$s^{(l)}(i_l) = \begin{bmatrix}\cos(\kappa i_l \cdot 2^{l-1}) & -\sin(\kappa i_l \cdot 2^{l-1}) \\ \sin(\kappa i_l \cdot 2^{l-1}) & \cos(\kappa i_l \cdot 2^{l-1})\end{bmatrix}, \quad \text{for} \quad l = 2,\ldots,L-1.$$

4. Полином $\sum\limits_{m=0}^{p} a_m i^m = P^{(1)}(i_1)\cdots P^{(L)}(i_L)$ (QTT ранги $p+1$) [144], [198], [95]:

$$P^{(1)}(i_1) = \begin{bmatrix}\sum\limits_{m=0}^{p} a_m C_m^0 i_1^m & \sum\limits_{m=1}^{p} a_m C_m^1 i_1^{m-1} & \cdots & \sum\limits_{m=p-1}^{p} a_m C_m^{p-1} i_1^{m-p+1} & a_p\end{bmatrix},$$
$$P^{(L)}(i_L) = \begin{bmatrix}1 \\ i_L \\ i_L^2 \\ \vdots \\ i_L^p\end{bmatrix}, \quad P^{(l)}(i_l) = \begin{bmatrix}C_0^0 & & & & \\ C_1^1 i_l^1 & C_1^0 & & & \\ C_2^2 i_l^2 & C_2^1 i_l^1 & C_2^0 & & \\ \vdots & & & \ddots & \\ C_p^p i_l^p & & \cdots & C_p^1 i_l^1 & C_p^0\end{bmatrix},$$

для $l = 2,\ldots,L-1$, где $C_m^k = m!/(k!(m-k)!)$.

Последний пример расширяет применимость QTT для почти любой гладкой функции, которая может быть аппроксимирована с помощью полиномиальной интерполяции (аналогично теореме 2.1.8 для оценки рангов Таккера). Например, вектор с элементами $x(i) = 1/(1+i)$ может быть численно приближен в QTT формате с рангами не более 8 до точности 10^{-10}, независимо от длины 2^L.

Кроме того, QTT формат в применении к матрицам (напомним понятие матричного TT (2.11)) позволяет строить простые представления основных операторов (дискретизации Лапласа, градиента на равномерной сетке, или сдвига), см. [129] и секцию 3.1, а также дальнейшее развитие концепции аналитических построений в [50] и секции 3.3.1.

Логарифмическая сложность по сравнению с количеством элементов в исходном тензоры делает QTT формат очень перспективным инструментом для больших задач. Были разработаны алгоритмы, принципиально основанные на двоичной структуре QTT, например, супер быстрый метод преобразования Фурье [54], свертки [127] и вейвлет-преобразований [149, 130] с логарифмической сложностью. Используя QTT формат по временной переменной в схеме одновременной дискретизации в пространстве-времени, представленной в главе 1.3, мы получаем очень эффективный метод интеграции эволюционных уравнений, см. также раздел 3.1 и статьи [53, 80].

2.2.2 QTT-Tucker: двухуровневое разделение исходных и виртуальных переменных

Формат QTT-Tucker предложен автором. Изложение ведется в соответствии с [50].

Конечно, QTT разложение не ограничивается одномерным случаем. Оно может быть использовано в дополнение к "традиционным" форматам, когда разделены только исходные индексы $i_1, \ldots, i_d$ с (возможно) большими модовыми размерами $n_1, \ldots, n_d$. В самом деле, каждый индекс i_k может быть дополнительно закодирован в двоичной системе (2.14),

$$i_k = \sum_{l=1}^{L_k} i_{k,l} \cdot 2^{l-1}, \quad i_{k,l} \in \{0, 1\},$$

так что глобальный индекс принимает вид

$$\boldsymbol{i} = \overline{i_{1,1}, \ldots, i_{1,L_1}, i_{2,1}, \ldots, i_{d,L_d}}.$$

Нумеруя элементы тензора с помощью индексов $\{i_{k,l}\}$, получаем тензор размерности $\sum L_k \leqslant dL$, который затем сжимается в малоранговый формат.

Какой формат подходит для этой цели? Так как все *новые* модовые размеры невелики, $\#\{i_{k,l}\} = 2$, TT (или QTT, если мы подчеркиваем использование квантизации; термин QTT будем также использовать и для dL-мерных тензоров) представление, дающее наименьшую асимптотическую сложность $\mathcal{O}(dLr^2)$ кажется на первый взгляд идеальным. Действительно, такое представление отлично работает во многих случаях, особенно если TT ранги небольшие. Однако, ранги, разделяющие виртуальные размерности (особенно в середине TT цепи), могут меняться довольно сильно в зависимости от точности. Для некоторых классов тензоров, например, решений уравнения Фоккера-Планка, оказывается более эффективным пожертвовать линейной структурой формата, и переключиться на потенциально более высокую асимптотическую сложность $\mathcal{O}(r^3)$, которая все же предпочтительнее, если новые ранги r будут меньше.

Итак, в основе нового формата будет лежать разложение Таккера:

$$x(i_1, \ldots, i_d) = \sum_{\gamma_1, \ldots, \gamma_d} x^{(c)}(\gamma_1, \ldots, \gamma_d) x^{(1)}_{\gamma_1}(i_1) \cdots x^{(d)}_{\gamma_d}(i_d). \tag{2.15}$$

Проклятие размерности снимается за счет хранения Таккеровского ядра в TT формате:

$$x^{(c)}(\gamma_1, \dots, \gamma_d) = \sum_{\alpha_1, \dots, \alpha_{d-1}} x^{c(1)}_{\alpha_1}(\gamma_1) x^{c(2)}_{\alpha_1, \alpha_2}(\gamma_2) \cdots x^{c(d)}_{\alpha_{d-1}}(\gamma_d). \tag{2.16}$$

Наконец, если Таккеровские факторы $x^{(k)}_{\gamma_k}(i_k)$ из-за больших размеров n_k занимают слишком много памяти, для индекса i_k можно ввести квантизацию:

$$x^{(k)}_{\gamma_k}(i_k) = \sum_{\gamma_{k,L}, \dots, \gamma_{k,1}} x^{f(k,L)}_{\gamma_{k,L-1}}(i_{k,L}) x^{f(k,L-1)}_{\gamma_{k,L-1}, \gamma_{k,L-2}}(i_{k,L-1}) \cdots x^{f(k,1)}_{\gamma_{k,1}, \gamma_k}(i_{k,1}). \tag{2.17}$$

Чтобы уменьшить и без того большое количество индексов, в дальнейшем в этой главе будем считать, что все $L_k = L$ одинаковы. Обратите внимание, что QTT блок $x^{f(k,1)}$ в правой части содержит *два типа* ранговых индексов: если в стандартном TT формате (2.9), γ_k соответствовал бы $\alpha_d \in \{1\}$, здесь он перечисляет векторы в Таккеровских факторах, которые хранятся *одновременно* в одном и том же QTT формате. Мы могли бы также разместить индекс Таккеровского ранга в последнем QTT блоке $x^{f(k,L)}$, но нам будет удобнее держать его в первом блоке, т.к. это дает больше однообразности в обозначениях, например, посредством эквивалентности $\gamma_{k,0} = \gamma_k$.

Собирая вместе все введенные представления, т.е. подставляя *внутренние* уровни структуры (2.16), (2.17) во *внешнее* разложение Таккера (2.15), мы окончательно получаем тензорную сеть, названную *QTT-Tucker* [50].

Определение 2.2.1. Говорят, что тензор x представлен (или аппроксимирован) в формате QTT-Tucker, если выполняется

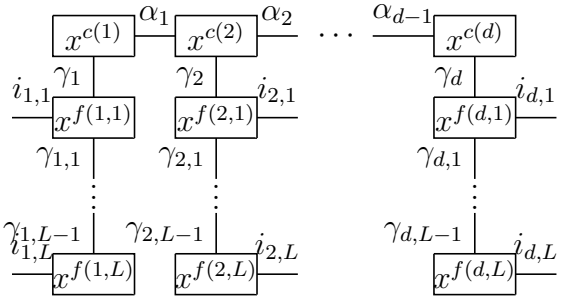
$$x(i_1, \dots, i_d) \approx \sum_{\gamma_k, \gamma_{k,l}, \alpha_k} \prod_{k=1}^{d} x^{c(k)}_{\alpha_{k-1}, \alpha_k}(\gamma_k) \prod_{l_k=1}^{L_k} x^{f(k,l_k)}_{\gamma_{k,l_k}, \gamma_{k,l_k-1}}(i_{k,l_k}), \tag{2.18}$$

где мы полагаем, что $\gamma_k = \gamma_{k,0}$, $n_k = 2^{L_k}$, и $i_k = \sum_{l_k=1}^{L_k} i_{l_k} \cdot 2^{l_k - 1}$. Параметр d называется *физической* размерностью, или *размерностью ядра*, L_k являются *квантизованными*, или *факторными* размерностями, $x^{c(k)} \in \mathbb{C}^{r_{k-1} \times R_k \times r_k}$ задает *k-ый блок ядра*, $x^{f(k,l)} \in \mathbb{C}^{R_{k,l} \times n_{k,l} \times R_{k,l-1}}$ это *k,l-ый блок фактора*, и индексы меняются в следующих диапазонах:

- $i_{k,l} = 0, \dots, n_{k,l} - 1, \quad n_{k,l} \leqslant n_Q$ (*квантизованный модовый* размер, например, $n_Q = 2$),
- $\alpha_k = 1, \dots, r_k, \quad r_k \leqslant r_C$ (ранг *ядра*),
- $\gamma_k = \gamma_{k,0} = 1, \dots, R_k, \quad R_k = R_{k,0} \leqslant r_T$ (ранг *Таккера*), и
- $\gamma_{k,l} = 1, \dots, R_{k,l}, \quad R_{k,l} \leqslant r_F$ (ранг *фактора*).

Для пограничных рангов мы полагаем, что $r_0 = r_d = R_{k,L_k} = 1$.

В диаграммах тензорных сетей, формат QTT-Tucker выглядит как показано в уравнении (2.19).



(2.19)

С использованием некоторого переопределения обозначений, будем обозначать исходные Таккеровские блоки (в разложении (2.5)) как $x^{f(k)} = \left[x_{\gamma_k}^{f(k)}(i_k)\right]$. Это делается с двоякой целью: во-первых, мы получаем более согласованную запись с квантизованными факторами $x^{f(k,l)}$, во-вторых, мы отличаем Таккеровские факторы от TT факторов, для которых используется вид $x^{(k)}$.

Суммируя размеры блоков в QTT-Tucker, получаем требования к памяти для этого формата.

Лемма 2.2.2. Сложность хранения формата QTT-Tucker оценивается как

$$dLn_Q r_F^2 + dr_T r_C^2, \quad \text{или} \quad \mathcal{O}(\log(n) dr^2 + dr^3), \tag{2.20}$$

в предположении, что все ранги ограничены константой r, и $n = n_Q^L$.

В случае высоких рангов, вклад Таккеровских ядер, $\mathcal{O}(r^3)$, может преобладать в оценке (2.20). Если для заданной точности тензор требует одинаковых рангов в форматах QTT и QTT-Tucker, мы можем потерять выигрыш в производительности. Например, если мы применяем иерархический формат (HT) непосредственно к $2 \times \cdots \times 2$-тензору, он будет состоять из $\mathcal{O}(d \log(n))$ блоков размеров $r \times r \times r$ вместо $r \times 2 \times r$ в простом QTT (называемом также *линейным QTT* форматом). Это приведет к большему объему памяти $\mathcal{O}(d \log(n) r + d \log(n) r^3)$. Подобные накладные расходы были обнаружены на примере спиновой системы в [162]. Чтобы сделать HT разложение эффективным, приходилось искусственно собирать несколько спиновых переменных в одну.

Можно привести следующие эвристические аргументы в пользу формата QTT-Tucker. Во-первых, только d блоков содержат три ранговых индекса, в то время как $d \log(n)$ блоков обладают квадратичной зависимостью числа элементов от рангов, как в линейном QTT формате. Кроме того, размеры этих d блоков регулируются TT и Таккеровскими рангами, которые, как правило, меньше, чем ранги между виртуальными размерностями в QTT разложении. Во-вторых, для поддержания той же точности, QTT ранги в Таккеровских факторах требуют меньших значений, чем соответствующие QTT ранги в глобальном линейном формате.

Последнее соображение можно аргументировать с помощью предположения, что тензор x возникает как дискретизация гладкой функции. Можно рассматривать исходный TT блок $x_{\alpha_{k-1},\alpha_k}^{(k)}(i_k)$ как набор r_C^2 векторов размера n. Эти векторы

получаются из дискретизации одномерных гладких функций, и мы можем рассчитывать на хорошую сжимаемость каждого из них в QTT формате,

$$x^{(k)}_{\alpha_{k-1},\alpha_k}(i_k) = \sum_{\alpha_{k,1}\dots,\alpha_{k,L-1}} x^{(k,1)}_{\alpha_{k-1},\alpha_{k,1}}(i_{k,1}) \cdots x^{(k,L)}_{\alpha_{k,L-1},\alpha_k}(i_{k,L})$$

где $\alpha_{k,l} = 1, \dots, \tilde{r}$ при сравнительно небольшом $\tilde{r}$. Если все векторы "независимы" друг от друга, итоговый QTT ранг для x оценивается как $r_C^2\tilde{r} = \mathcal{O}(r^2\tilde{r})$. Следовательно, QTT формат требует $\mathcal{O}(d\log(n)r^4\tilde{r}^2)$ памяти.

Применяя аналогичные рассуждения к факторам Таккера, мы можем ожидать более низкой оценки на факторные ранги, $r_F = \mathcal{O}(r_T\tilde{r})$. Если ранги Таккера такого же порядка, как и TT ранги, $r_F \sim r_C \sim r$ (что, как правило, имеет место на практике), общая сложность формата QTT-Tucker может быть оценена как $\mathcal{O}(d\log(n)r^2\tilde{r}^2 + dr^3)$, что уже меньше, чем в линейном QTT.

Еще одной интересной особенностью нового формата является то, что он может наследовать оценки рангов из формата Таккера, который могут быть гораздо легче доказуемы, чем оценки в TT формате, за счет использования полиномиальной интерполяции. Например, мы можем обобщить теорему 2.1.8 для QTT-Tucker формата.

Теорема 2.2.3. Пусть дана аналитическая функция $f(q_1, \dots, q_d)$ в области $\Omega = [-1, 1]^d$, и тензор $x(i_1, \dots, i_d) = f(q_1(i_1), \dots, q_d(i_d))$ построен как ее дискретизация на тензорном произведении равномерных одномерных сеток. Пусть f допускает продолжение в эллипс Бернштейна с радиусом ρ_k,

$$\mathcal{E}_{\rho_k} = \left\{ z \in \mathbb{C} : \ |1+z| + |1-z| \leqslant \rho_k + \frac{1}{\rho_k} \right\},$$

по каждой переменной q_k, так что $\mathcal{M} = \max\limits_{\boldsymbol{z} \in \mathcal{E}_{\rho_1} \otimes \cdots \otimes \mathcal{E}_{\rho_d}} |f(\boldsymbol{z})| < \infty$. Тогда Таккеровские ранги ε-аппроксимации ограничиваются $R_k \leqslant C|\log(\varepsilon)|/\log(\rho_k)$, а QTT ранги Таккеровских факторов равны $R_{k,l} = R_k + 1$, $l = 1, \dots, L-1$.

Доказательство. Функцию f можно рассматривать как одномерную функцию $f^{[k]}(q_k)$, зависящую от всех остальных переменных как параметров. Таким образом, можно применить следующий результат из теории аппроксимаций: если $f^{[k]}$ допускает аналитическое продолжение в эллипс Бернштейна, существует интерполяция P_p полиномом степени не выше p с точностью

$$\|f^{[k]}(z) - P_p(z)\|_\infty \leqslant C\log(n)\frac{M_k}{1-\rho_k}\rho_k^{-p}, \quad z \in \mathcal{E}_{\rho_k}, \ \rho_k > 1, \quad M_k = \max_{z \in \mathcal{E}_{\rho_k}} |f^{[k]}(z)|,$$

где n это число точек сетки, и константа C не зависит от p, n, M_k, ρ_k [27, 234]. Однако, константа M_k зависит от остальных переменных $q_1, \dots, q_d$, за исключением q_k. Чтобы избавиться от них, предположим равномерную ограниченность, $\mathcal{M} = \max_k M_k$. Теперь можно сказать, что для каждой $f^{[k]}$ существует полином $P^{[k]}_{p_k}$ такой, что

$$\varepsilon = \|f^{[k]}(q_k) - P^{[k]}_{p_k}(q_k)\|_\infty \leqslant C\rho_k^{-p_k}, \quad \text{или} \quad p_k = C|\log(\varepsilon)|/\log(\rho_k).$$

Таким образом, можно взять тензорное произведение полиномов степеней p_k как новый базис. Очевидно, коэффициенты в этом базисе составляют $p_1 \times \cdots \times p_d$-тензор, который может рассматриваться в качестве ядра Таккер, а набор p_k многочленов на сетке является k-ым Таккеровским фактором. Оценка для R_k доказана.

Вспоминая, что все многочлены степени p_k на равномерной сетке одновременно представимы в QTT формате с рангами p_k+1 (см. предыдущую секцию), получаем второе утверждение теоремы. □

2.2.3 Преобразования из TT в расширенный TT и QTT-Tucker форматы

Имея нескольких тензорных представлений, разумно разработать процедуру для переноса данные из одного формата в другой. Например, полный тензор разворачивается из TT формата как сумма Кронекеровских произведений (2.10). Если тензор дан в формате QTT-Tucker (или расширенный TT), блоки стандартного TT представления вычисляются с помощью суммирования по индексам Таккеровских рангов,

$$x^{(k)}_{\alpha_{k-1},\alpha_k}(i_k) = \sum_{\gamma_{k,L-1},\ldots,\gamma_{k,1},\gamma_k} x^{f(k,L)}_{\gamma_{k,L-1}}(i_{k,L}) \cdots x^{f(k,1)}_{\gamma_{k,1},\gamma_k}(i_{k,1}) x^{c(k)}_{\alpha_{k-1},\alpha_k}(\gamma_k). \tag{2.21}$$

Очевидно, что в правой части предыдущего уравнения стоит также TT формат. Поэтому, в работе [50] это выражение было названо *расширенным фактором*. В то же время, этот массив в свою очередь является блоком другого TT формата. Таким образом, уравнение (2.21) показывает двухуровневую QTT-Tucker структуру с другой точки зрения.

Обратная операция может потребовать приближенных вычислений. Рассмотрим *центральную развертку* (см. определение 2.1.12) k-го TT блока $x^{|k|}$. Предположим, что TT блоки $1, \ldots, k-1$ лево-ортогональны, и блоки $k+1, \ldots, d$ право-ортогональны, так что возмущение, вносимое в $x^{|k|}$, распространяется без изменения нормы на весь x. Теперь, мы можем выполнить (неполное) SVD разложение

$$x^{|k|} \approx U \Sigma V,$$

и присвоить либо $x^{f(k)} = U$, $x^{c|k|} = \Sigma V$, либо $x^{f(k)} = U\Sigma$, $x^{c|k|} = V$, в зависимости от того, какой тип ортогональности мы хотели бы обеспечить. Ранг этого разложения дает k-й Таккеровский ранг. Непосредственно из размеров блоков можно написать оценку сверху на новый ранг, $R_k \leqslant \min(n_k, r_{k-1}r_k) = \mathcal{O}(\min(n, r^2))$ (см. аналогичное сравнение TT и HT форматов в [96]), а также сложность $\mathcal{O}(nr^2 \min(n, r^2))$.

В точном случае, такое разложение можно провести аналитически. Давайте покажем это на сумме одномерных компонент (например, оператор Лапласа, сумма сеточных координат $q_1 + \cdots + q_d$, и т.п.),

$$x(i_1, \ldots, i_d) = a(i_1) \cdot b(i_2) \cdots b(i_d) + \cdots + b(i_1) \cdots b(i_{d-1}) \cdot a(i_d).$$

Точное TT представление ранга 2 [129] пишется так:

$$x(\boldsymbol{i}) = \begin{bmatrix} a(i_1) & b(i_1) \end{bmatrix} \begin{bmatrix} b(i_2) & 0 \\ a(i_2) & b(i_2) \end{bmatrix} \cdots \begin{bmatrix} b(i_{d-1}) & 0 \\ a(i_{d-1}) & b(i_{d-1}) \end{bmatrix} \begin{bmatrix} b(i_d) \\ a(i_d) \end{bmatrix}.$$

Поскольку каждый блок содержит только 2 линейно независимых элемента, все факторы Таккера имеют один вид,

$$x^{f(k)}(i_k) = \begin{bmatrix} a(i_k) & b(i_k) \end{bmatrix},$$

и ядро (тензор $2 \times \cdots \times 2$) состоит только из нулей и единиц:

$$x^{(c)}(\boldsymbol{\gamma}) = \begin{bmatrix} e_0(\gamma_1) & e_1(\gamma_1) \end{bmatrix} \begin{bmatrix} e_1(\gamma_2) & 0 \\ e_0(\gamma_2) & e_1(\gamma_2) \end{bmatrix} \cdots \begin{bmatrix} e_1(\gamma_{d-1}) & 0 \\ e_0(\gamma_{d-1}) & e_1(\gamma_{d-1}) \end{bmatrix} \begin{bmatrix} e_1(\gamma_d) \\ e_0(\gamma_d) \end{bmatrix},$$

где $e_0 = \begin{bmatrix} 1 & 0 \end{bmatrix}$, и $e_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$.

Замечание 2.2.4. Другие тензорные сети могут быть преобразованы сначала в TT формат (или сумму TT тензоров, например, тензорную цепь), а затем в QTT-Tucker. Это может быть полезным в задачах квантовой физики, если модель предоставляет исходные данные в виде сложной тензорной сети.

2.2.4 Операции в формате QTT-Tucker

Аналоги алгебраических операций в тензорном формате, описанные в разделе 2.1.5, могут быть распространены и на QTT-Tucker.

Так, **сумма** QTT-Tucker тензоров $z = x + y$ выполняется как объединение факторов,

$$z^{f(k,L)} = \begin{bmatrix} x^{f(k,L)}(i_{k,L}) & y^{(k,L)}(i_{k,L}) \end{bmatrix}, \quad z^{f(k,l)} = \begin{bmatrix} x^{f(k,l)}(i_{k,l}) & \\ & y^{(k,l)}(i_{k,l}) \end{bmatrix}$$

для $l = L-1, \ldots, 1$, и блоков ядра,

$$z^{c(k)}(\gamma_k) = \begin{cases} \begin{bmatrix} x^{c(k)}(\gamma_k) & 0 \\ 0 & 0 \end{bmatrix}, & \text{если} \quad \gamma_k = 1, \ldots, R_k(x), \\ \begin{bmatrix} 0 & 0 \\ 0 & y^{c(k)}(\gamma_k - R_k(x)) \end{bmatrix}, & \text{если} \quad \gamma_k = R_k(x)+1, \ldots, R_k(x)+R_k(y). \end{cases}$$

Матричное произведение $y = Ax$ переносится как соответствующее произведение на факторы, но с объединением индексов Таккеровских рангов $\gamma_k(A)$, $\gamma_k(x)$,

$$y^{f(k,l)}(i_{k,l}) = \sum_{j_{k,l}} A^{f(k,p)}(i_{k,l}, j_{k,l}) \otimes x^{f(k,l)}(j_{k,l}),$$

где стандартные Кронекеровские произведения $\otimes$ выполняются для ранговых индексов (напомним, что при фиксированных $i_{k,l}, j_{k,l}$, блоки становятся обычными матрицами), и 3-мерное Кронекеровское произведение применяется для блоков ядра,

$$y^{c(k)}(\overline{\gamma_k(A), \gamma_k(x)}) = A^{c(k)}(\gamma_k(A)) \otimes x^{c(k)}(\gamma_k(x)).$$

Все ранги перемножаются: $r(y) = r(A)r(x)$. Заметим, что матричный QTT-Tucker формат содержит два исходных индекса только в блоках факторов, но блоки ядра остаются 3-мерными, как и в представлении для векторов.

Скалярное произведение начинается с TT алгоритма 1, примененного к факторам. Разница в том, что мы начинаем не с первого, а с L-го блока фактора, и получаем на выходе не число, а матрицу $s_k \in \mathbb{C}^{R_k(x) \times R_k(y)}$, $s_k(\gamma_k(x), \gamma_k(y)) = \left(x^{f(k)}_{\gamma_k(x)}, y^{f(k)}_{\gamma_k(y)}\right)$. Когда все факторы соответственно перемножены таким образом, матрицы s_k, $k = 1, \ldots, d$, перемножаются с одним из Таккеровских ядер, что дает новый промежуточный тензор z таких же размеров, как ядро другого тензора. Без нарушения общности, пусть это будет $y^{(c)}$, тогда

$$z = \tau(z^{(1)}, \ldots, z^{(d)}), \qquad z^{|k|} = s_k y^{c|k|}, \quad k = 1, \ldots, d.$$

После этого, результат вычисляется как еще одно скалярное произведение в TT формате, $(x, y) = \left(x^{(c)}, z\right)$. Можно проверить непосредственными вычислениями, что этот результат является произведением элементов всех ядер, просуммированным по всем индексам. Сложность такого алгоритма составляет $\mathcal{O}(d \log(n) r_F^3) + \mathcal{O}(d r_F^2 r_C^2) + \mathcal{O}(d r_F r_C^3)$, где первый член отвечает произведению факторов, второй соответствует построению z, и последний возникает из-за произведения ядер.

2.2.5 Округление в формате QTT-Tucker

Хотя вывод QTT-Tucker формата был основан на свойствах разложения Таккера, далее будет удобнее описывать алгоритмы переменных направлений (процедура уменьшения ранга также принадлежит к этому семейству) с точки зрения двухуровневого TT представления: исходный TT формат (2.9), каждый блок которого в свою очередь записан в виде расширенного фактора (2.21). Это позволяет ссылаться на TT версии многих алгоритмов.

Как только все TT блоки, кроме k-го должным образом ортогональны, мы можем приблизить $x^{(k)}$, с помощью процедуры TT аппроксимации, примененной к расширенному фактору

$$\sum_{\gamma_{k,L-1}, \ldots, \gamma_{k,1}, \gamma_k} x^{f(k,L)}_{\gamma_{k,L-1}}(i_{k,L}) \cdots x^{f(k,1)}_{\gamma_{k,1}, \gamma_k}(i_{k,1}) x^{c|k|}_{\gamma_k}(\overline{\alpha_{k-1}, \alpha_k}).$$

Обратите внимание, что последний блок здесь является центральной разверткой блока ядра $x^{c(k)}$.

В свою очередь, QR-разложение, скажем, блока $x^{|k\rangle}$, может быть эффективно вычислено с помощью алгоритма левой ортогонализации 2, примененного для расширенного фактора, с последующим QR-разложением одного блока ядра

$$q^{c|k\rangle} R = x^{c|k\rangle} \in \mathbb{C}^{r_{k-1} R_k \times r_k}.$$

Полная процедура описана в алгоритме 5.

Сложность алгоритма 5 следует непосредственно из оценок для TT алгоритмов. А именно, ортогонализация и сжатие (SVD) требует $\mathcal{O}(d L r_F^3)$ операций для факторов, и $\mathcal{O}(d r_C^4)$ операция для ядра, давая в итоге оценку $\mathcal{O}(d L r_F^3 + d r_C^4)$.

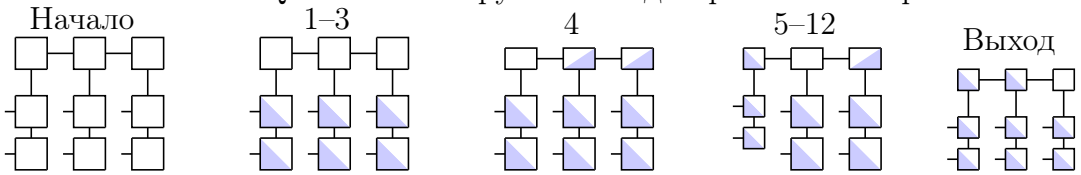
Алгоритм 5 Округление в QTT-Tucker формате

Ввод: Тензор x в формате QTT-Tucker, пороги точности ε_k, η.

Вывод: Тензор y в формате QTT-Tucker с условием $\|y - x\|^2 \leqslant \sum \varepsilon_k^2 + \eta^2$ и возможно меньшими рангами.

1: **for** $k = 1, \ldots, d$ **do**
2: Алг. 2 для левой ортогонализация k-го расширенного фактора, так что $\left(x^{f|k\rangle}\right)^* x^{f|k\rangle} = I$.
3: **end for**
4: Найти сжатое ядро $y^{(c)}$ с помощью Алг. 4: $\|y^{(c)} - x^{(c)}\| \leqslant \eta \|x^{(c)}\|$
5: **for** $k = 1, \ldots, d$ **do**
6: Найти сжатый расширенный фактор $y^{(k)}$ с помощью Алг. 4: $\|y^{(k)} - x^{(k)}\| \leqslant \varepsilon_k \|x^{(k)}\|$
7: **if** $k < d$ **then**
8: Алг. 2 для левой ортогонализации $y^{f|k\rangle}$.
9: Найти QR разложение $y^{c|k\rangle} = q^{c|k\rangle} R$.
10: Записать $y^{c|k\rangle} = q^{c|k\rangle}, \quad y^{c\langle k+1|} = R y^{c\langle k+1|}$.
11: **end if**
12: **end for**

Рис. 2.3: Шаги QTT-Tucker округления в диаграммах тензорных сетей



Интересно, что ортогонализации/округление расширенных факторов (строки 2 и 6 алгоритма) обеспечивает выполнение шагов Таккер-HOSVD алгоритма: разница заключается в использовании только одного *блока* ядра для каждого QR/SVD разложения, вместо всего d-мерного ядра. Диаграммы состояний формата QTT-Tucker после определенных шагов алгоритма 5 показаны на рис. 2.3: заполненные части блоков обозначают ортогональность и размеры блоков пропорциональны их рангам.

Перед тем как закончить с процедурой округления, приведем теорему о квазиоптимальности алгоритма 5.

Теорема 2.2.5. Пусть каждое приближение в QTT-Tucker факторах выполняется сингулярным разложением с относительными порогами во Фробениусовой норме $\varepsilon_{k,l}$, Таккеровские ранги определяются с порогами ε_k, а каждый блок ядра приближается с точностью η_k. Тогда относительная Фробениусова норма ошибки во всем тензоре оценивается следующим образом:

$$\frac{\|y - x\|^2}{\|x\|^2} \leqslant \sum_{k,l=1}^{d,L-1} \varepsilon_{k,l}^2 + \sum_{k=1}^{d} \varepsilon_k^2 + \sum_{k=1}^{d-1} \eta_k^2. \tag{2.22}$$

Мы опускаем доказательство, так как оно в значительной степени повторяет

доказательство HOSVD теоремы 2.1.7: мы связываем каждое сингулярное разложение с соответствующим ортогональным проектором в терминах исходных векторов, а затем с помощью сложения и вычитания одинаковых членов и попарной ортогональности векторов вида $x = Pz$ и $y = z - Pz$, получаем результат.

Глава 3

Представления основных функций, векторов и матриц в тензорных произведениях

В процессе эксплуатации различных тензорных форматов, полезно уметь строить простые и широко используемые тензоры явно, задавая непосредственно элементы формата. Для TT и QTT форматов эта работа была начата в [144, 198, 129, 50], а для канонического формата в [79, 104, 105]. Особенно удобно, если тензор имеет точное разложение малого ранга – его сжатие из полного, или даже канонического формата может быть довольно затруднительно вычислительно, и приводить к потере точности. Основным инструментом для получения точных тензорных структур является индуктивный аналитический аналог TT-SVD алгоритма 4. На каждом шаге вместо вычисления сингулярного разложения, мы извлекаем интуитивно простые линейно независимые элементы, чтобы сформировать текущий TT блок. Если остающиеся тензоры (вида $X^{\langle k|}(\overline{i_1, \ldots, i_{k-1}}, \overline{i_k, \alpha_k})$, см. конец раздела 2.1.5) имеет один и тот же аналитический вид для любого k, это означает, что индукция может быть продолжена далее, и все внутренние TT блоки строятся по одному и тому же правилу.

Другой способ, который во многих случаях является единственным методом для вычисления *приближенных* тензорных структур, состоит в сборке сложного тензора из более простых с помощью тензорной алгебры. В основном используется канонический формат, так как обычно теория приближений дает результаты в виде сходящихся рядов. Если каждый член ряда имеет простое (например, ранга 1) тензорное представление, мы можем взять конечное число слагаемых, и получить канонический или TT формат (см. замечание 2.1.13) с контролируемыми оценками на соотношение ранг–точность.

Все представления в этой главе, за исключением обозначенных отдельно (например, обращение дискретного оператора Лапласа в секции 3.4), являются авторскими.

3.1 Тензорные представления в блочной временной схеме

Одним из важных вкладов этой работы является одновременная пространственно-временная схема Кранка-Николсон (1.23), или (1.26). Для ее эффективного решения в тензорных форматах нам нужно три компонента: представимость входных данных в тензорных форматах (т.е. матрицы, правой части и начального состояния), представимость решения в тензорных форматах, и алгоритм для собственно вычисления решения, при условии, что все данные хорошо структурированы в принципе.

Вторая часть довольно сложна для теоретического анализа, и часто приходится исследовать свойства решения численно. Надежное решение третьего вопроса будет представлено в следующей главе. Теперь мы сосредоточимся на первой части: исследуем отдельно вклад переменной времени, при условии, что другие входные данные уже заданы в тензорном формате, а затем покажем несколько примеров, когда это предположение действительно имеет место.

3.1.1 Тензорная структура блочной пространственно-временной матрицы

Рассматривая сначала непредобусловленный вариант (1.23), предположим, что правая часть g равен нулю, матрица A записана в TT формате (2.11), и начальное состояние v также дается в TT формате (2.9). Тогда, глобальная матрица легко строится как $d+1$-мерный TT формат:

$$\mathcal{A} = \sum_{\boldsymbol{\alpha}} \mathcal{A}^{(1)}_{\alpha_1} \otimes \mathcal{A}^{(2)}_{\alpha_1,\alpha_2} \otimes \cdots \otimes \mathcal{A}^{(d)}_{\alpha_{d-1},\alpha_d} \otimes \mathcal{A}^{(d+1)}_{\alpha_d},$$

$$\mathcal{A}^{(k)}_{\alpha_{k-1},\alpha_k} = \begin{cases} A^{(k)}_{\alpha_{k-1},\alpha_k}, & \alpha_{k-1} = 1,\ldots,r_{k-1}(A), \\ & \alpha_k = 1,\ldots,r_k(A), \\ I, & \alpha_{k-1} = r_{k-1}(A)+1, \text{ и} \\ & \alpha_k = r_k(A)+1, \\ 0, & \alpha_{k-1} = r_{k-1}(A)+1,\ \alpha_k \leqslant r_k(A), \text{ или} \\ & \alpha_k = r_k(A)+1,\ \alpha_{k-1} \leqslant r_{k-1}(A), \end{cases}$$

для $k = 1,\ldots,d$, и

$$\mathcal{A}^{(d+1)}_1 = \frac{\delta t}{2} M_t, \quad \mathcal{A}^{(d+1)}_2 = G_t.$$

Можно сразу заметить, что TT ранги $r_1,\ldots,r_{d-1}$ больше на 1 по сравнению с рангами матрицы $\boldsymbol{r}(A)$, и последний ранг $r_d = 2$. Поэтому, в отличие от полного формата (1.22), хранение этой системы в тензорных произведениях ненамного сложнее, чем каждого шага Кранка-Николсон.

Начальное состояние собирается аналогично: пусть дано d-мерное TT представление для $v - \frac{\delta t}{2} Av$, оно объединяется с $d+1$-ым TT блоком, являющимся первым единичным вектором по времени e_1, без изменения TT рангов.

TT формат для предобусловленной схемы (1.26) пишется аналогичным образом: $\tilde{\mathcal{A}}^{(k)} = \mathcal{A}^{(k)}$ для $k = 1, \ldots, d$, и только последний блок меняется следующим образом,

$$\tilde{\mathcal{A}}_1^{(d+1)} = \frac{\delta t}{2} G_t^{-1} M_t, \quad \tilde{\mathcal{A}}_2^{(d+1)} = I.$$

Также, e_1 в правой части заменяется вектором из всех единиц.

3.1.2 Матрицы сдвига и конечных разностей в QTT формате

Количество временных шагов, необходимых для разрешения многомасштабных процессов может быть весьма большим, и использование квантизации по временной переменной очень разумно, см. [53, 51].

Единичная матрица I, вектор из одних единиц e, и единичный вектор e_1 (см. раздел 2.2.1) имеют ранг-1 QTT представления для любого размера. Более интересной является структура матриц жесткости G_t и массы M_t. Как было показано в [127, 129], матрицы сдвига любого порядка и размера обладает QTT представлением ранга 2. В частности, Жорданов блок $\boldsymbol{J} \in \mathbb{R}^{2^L \times 2^L}$ имеет следующий вид [127][1]:

$$\boldsymbol{J}(\boldsymbol{i}, \boldsymbol{j}) = \begin{bmatrix} J_{i_1,j_1} & J_{i_1,j_1}^\top \end{bmatrix} \cdots \begin{bmatrix} I_{i_k,j_k} & \\ J_{i_k,j_k} & J_{i_k,j_k}^\top \end{bmatrix} \cdots \begin{bmatrix} I_{i_L,j_L} \\ J_{i_L,j_L} \end{bmatrix},$$

где $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ и $J = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ являются элементарными единичными и Жордановыми матрицами, соответственно. Заметим, что внутренние блоки строятся одинаковым образом для всех $k = 2, \ldots, L-1$.

В качестве простой иллюстрации, давайте посмотрим, как можно алгоритм TT-округления провести аналитически. Матрицы, возникающие в схема Кранка-Николсон, равны $G_t = I - \boldsymbol{J}^\top$ и $M_t = I + \boldsymbol{J}^\top$, то есть на первый взгляд имеют структуру ранга 3. Однако, устраним линейную зависимость в последних блоках:

$$\begin{bmatrix} I_{i_L,j_L} \\ J_{i_L,j_L}^\top \\ I_{i_L,j_L} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} I_{i_L,j_L} \\ J_{i_L,j_L}^\top \end{bmatrix},$$

и для $k = L-1$ получаем

$$\begin{bmatrix} I_{i_k,j_k} & & \\ J_{i_k,j_k}^\top & J_{i_k,j_k} & \\ & & I_{i_k,j_k} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} I_{i_k,j_k} & \\ J_{i_k,j_k}^\top & J_{i_k,j_k} \\ I_{i_k,j_k} & \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} I_{i_k,j_k} & \\ J_{i_k,j_k}^\top & J_{i_k,j_k} \end{bmatrix}.$$

[1] В данной работе мы используем little-endian конвенцию, т.е. индексы i_1, j_1 меняются быстрее всего, и i_L, j_L являются самыми медленными, см. (2.14). Это отличается от работ [129, 127], где использовалась конвенция big-endian. Поэтому здесь представления QTT пишутся в "обратном" порядке по сравнению с [127].

Поскольку скалярный множитель оказался переброшен на левую сторону, индукция может быть продолжена для всех k, и закончена на первом блоке:

$$\begin{bmatrix} J^\top_{i_1,j_1} & J_{i_1,j_1} & I_{i_1,k_1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} M_{i_1,j_1} & J_{i_1,j_1} \end{bmatrix},$$

где $M = I + J^\top$ суть элементарная матрица масс. Таким образом, и G_t, и M_t представимы *одновременно* в QTT формате, т.е. с одними и теми же блоками $k = 2, \ldots, L$, принадлежащими и QTT представлению транспонированного Жорданова блока, и только в первом QTT блоке происходит разделение исходных матриц. Мы можем записать их в форме, используемой в QTT-Tucker разделе 2.2.2: нулевой TT ранг выступает в качестве индекса, нумерующего объекты, т.е.

$$\begin{bmatrix} G_t(\boldsymbol{i}, \boldsymbol{j}) \\ M_t(\boldsymbol{i}, \boldsymbol{j}) \end{bmatrix} = \begin{bmatrix} G_{i_1,j_1} & J_{i_1,j_1} \\ M_{i_1,j_1} & J_{i_1,j_1} \end{bmatrix} \ldots \begin{bmatrix} I_{i_k,j_k} & \\ J^\top_{i_k,j_k} & J_{i_k,j_k} \end{bmatrix} \ldots \begin{bmatrix} I_{i_L,j_L} \\ J^\top_{i_L,j_L} \end{bmatrix},$$

где $G = I - J^\top$ обозначает элементарную матрицу конечных разностей.

Предобусловленная схема (1.26) может быть проанализирована аналогично. Матрица G_t^{-1} является Теплицевой, образованной от вектора $\begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 1 \end{bmatrix}$ с QTT рангом 1, и следовательно имеет QTT структуру ранга 2 [127], так что матрица $G_t^{-1} M_t$ представима в QTT формате с рангом 4.

3.2 Матрицы перехода для ионного уравнения модели Фарлей-Бунемановской неустойчивости

В данной секции мы рассмотрим как представить решение и матрицы, используемые во временных схемах ("крест" и BGK), в уравнении для ионного распределения (1.5). Поскольку электронное уравнение (1.3) и уравнение Пуассона (1.4) являются двумерными, они не требуют дальнейшей дискуссии. Однако применение тензорных аппроксимаций для ионного уравнения нужно пояснить более подробно.

Поскольку концентрации частиц в зависимости от x, y координаты выглядят псевдослучайным образом, эти переменные не могут быть хорошо разделены в TT формате. Однако, распределение скоростей остается небольшим возмущением к Максвелловскому в течение всего процесса. Таким образом, мы можем ожидать хороший отделимости x, y (как целого) от v, w переменных, и ввести следующую TT структуру:

$$\psi_{i,j,k,m} = \psi^{(1)}(i,j)\psi^{(3)}(k)\psi^{(4)}(m) = \sum_{\alpha_1,\alpha_3} \psi^{(1)}_{\alpha_1}(i,j)\psi^{(3)}_{\alpha_1,\alpha_3}(k)\psi^{(4)}_{\alpha_3}(m). \tag{3.1}$$

Начальное состояние (1.14) обладает при этом TT представлением ранга 1.

Рассмотрим стадию конвекции в ионном уравнении. Как и в секции 1.1.2, мы начинаем с шага $\frac{\partial \psi}{\partial t} + v\frac{\partial \psi}{\partial x} = 0$. Каждый коэффициент в формуле интерполяции

типа "крест" (1.10) действует в соответствии с определенной матрицей периодического сдвига. Обозначим

$$S_1 = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ 1 & 0 & \cdots & & 0 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 0 & 0 & 1 & & & \\ & 0 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 0 & 0 & 1 \\ 1 & 0 & & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & & 0 \end{bmatrix}, \tag{3.2}$$

а также $S_{-2} = S_2^\top$, $S_{-1} = S_1^\top$, и $S_0 = I$. Тогда одномерная матрица интегрирования на один шаг по времени собирается как

$$M_c = \alpha_{-2} S_{-2} + \alpha_{-1} S_{-1} + \alpha_0 S_0 + \alpha_1 S_1 + \alpha_2 S_2.$$

Однако, в нашем случае каждый α_i параметризован значением скорости. То есть, действие α_i задается диагональной матрицей

$$\Lambda_i = [I \otimes I] \otimes \mathrm{diag}(\alpha_i(v)) \otimes I,$$

и в итоге для матрицы перехода в исходном ионном уравнение получаем TT представление ранга 5:

$$\begin{aligned} M_x(\delta t) &= [S_{-2} \otimes I] \otimes \mathrm{diag}(\alpha_{-2}(v)) \otimes I + [S_{-1} \otimes I] \otimes \mathrm{diag}(\alpha_{-1}(v)) \otimes I \\ &+ [S_0 \otimes I] \otimes \mathrm{diag}(\alpha_0(v)) \otimes I \\ &+ [S_1 \otimes I] \otimes \mathrm{diag}(\alpha_1(v)) \otimes I + [S_2 \otimes I] \otimes \mathrm{diag}(\alpha_2(v)) \otimes I, \end{aligned}$$

где Кронекеровские произведения по x и y (в квадратных скобках) вычисляются явно (мы не разделяем пространственные переменные), тогда как остальные подразумеваются неявно, и сомножители хранятся отдельно, как TT блоки. Таким же образом конструируем другие матрицы, соответствующие конвекционным шагам:

$$\begin{aligned} M_y(\delta t) &= [I \otimes S_{-2}] \otimes I \otimes \mathrm{diag}(\alpha_{-2}(w)) + [I \otimes S_{-1}] \otimes I \otimes \mathrm{diag}(\alpha_{-1}(w)) \\ &+ [I \otimes S_0] \otimes I \otimes \mathrm{diag}(\alpha_0(w)) \\ &+ [I \otimes S_1] \otimes I \otimes \mathrm{diag}(\alpha_1(w)) + [I \otimes S_2] \otimes I \otimes \mathrm{diag}(\alpha_2(w)), \end{aligned}$$

$$\begin{aligned} M_v(\phi, \delta t) &= \mathrm{diag}\left(\alpha_{-2}\left(V_v\right)\right) \otimes S_{-2} \otimes I + \mathrm{diag}\left(\alpha_{-1}\left(V_v\right)\right) \otimes S_{-1} \otimes I \\ &+ \mathrm{diag}\left(\alpha_0\left(V_v\right)\right) \otimes S_0 \otimes I \\ &+ \mathrm{diag}\left(\alpha_1\left(V_v\right)\right) \otimes S_1 \otimes I + \mathrm{diag}\left(\alpha_2\left(V_v\right)\right) \otimes S_2 \otimes I, \end{aligned}$$

где $V_v = -\frac{\partial \phi}{\partial x}$, и

$$\begin{aligned} M_w(\phi, \delta t) &= \mathrm{diag}\left(\alpha_{-2}\left(V_w\right)\right) \otimes I \otimes S_{-2} + \mathrm{diag}\left(\alpha_{-1}\left(V_w\right)\right) \otimes I \otimes S_{-1} \\ &+ \mathrm{diag}\left(\alpha_0\left(V_w\right)\right) \otimes I \otimes S_0 \\ &+ \mathrm{diag}\left(\alpha_1\left(V_w\right)\right) \otimes I \otimes S_1 + \mathrm{diag}\left(\alpha_2\left(V_w\right)\right) \otimes I \otimes S_2, \end{aligned}$$

где $V_w = \frac{eE_0}{m_i \nu_{T_i} \nu_{in}} - \frac{\partial \phi}{\partial y}$. Теперь, стадия расщепления "конвекция ионов" (например, $\psi^{1/4}$) в (1.13) выполняется следующим образом:

$$\psi^{1/4} = \mathfrak{T} M_x \mathfrak{T} M_y \mathfrak{T} M_v \mathfrak{T} M_w \psi^0,$$

где $\mathcal{T}$ обозначает операцию округления (в т.ч. с помощью AMEn алгоритма для аппроксимации матрично-векторного произведения, см. раздел 4.4.5) в TT формате.

Реакционная часть проще, так как она включает в себя только одну матрицу с TT структурой ранга 2. Действительно, как было показано в разделе (1.1.2), BGK шаги могут быть вычислены через произведение матрицы на вектор:

$$\psi^{2/4} = \mathcal{T} M_r \psi^{1/4}, \quad \psi^{3/4} = \mathcal{T} M_r \psi^{2/4},$$

где

$$M_r(\delta t) = \mathrm{e}^{-\delta t/2} I \otimes I \otimes I \otimes I + (1 - \mathrm{e}^{-\delta t/2}) I \otimes I \otimes \left(\frac{h_v}{\sqrt{2\pi}} \mathbf{e}\mathbf{1}^\top \right) \otimes \left(\frac{h_v}{\sqrt{2\pi}} \mathbf{e}\mathbf{1}^\top \right).$$

3.3 Тензорные свойства основного кинетического уравнения

В этом разделе мы представляем общий и более тонкий анализ тензорных структур для матрицы в основном кинетическом уравнении. Подчеркнем еще раз, что сами аналитические решения, а также их тензорные свойства редко когда можно вывести строго, большинство таких результатов требует довольно узкого класса входных данных. Однако численные эксперименты говорят в пользу разделения переменных и при гораздо более широких условиях, где теоретических оценок (по крайней мере, пока) не существует. Обычно намного проще исследовать матрицу или начальное состояние, которые могут обладать достаточно простыми представлениями и для не самых тривиальных случаев. Это и будет наша задача в этой секции.

В качестве первого, базового шага, предположим, что *коэффициенты* в операторе ОКУ допускают тензорные представления. Что можно сказать о самом операторе?

Для основного кинетического уравнения (1.17),

$$\frac{d\psi}{dt} = \sum_{m=1}^{M} \left(\boldsymbol{J}^{\boldsymbol{z}^m} - I \right) \mathrm{diag}(w_m) \psi,$$

матрица $\boldsymbol{J}^{\boldsymbol{z}^m} = J^{z_1^m} \otimes J^{z_2^m} \otimes \cdots \otimes J^{z_d^m}$ является многоуровневой матрицей сдвига с порядками $(z_1^m, \ldots, z_d^m)$, т.е. матричным TT форматом с TT рангами 1. Применяя правила тензорных алгебраических операций (секция 2.1.5), можем заключить, что TT ранги всего оператора ОКУ ограничены следующим образом:

$$r \leqslant \sum_{m=1}^{M} 2 r(w_m) \leqslant 2 M r_w, \quad \text{если } r(w_m) \leqslant r_w.$$

Фактор 2 возникает из сложения матриц $\boldsymbol{J}^{\boldsymbol{z}^m}$ и $-I$, которые имеют TT ранги 1 каждая.

3.3.1 Матрица ОКУ для случая цепи каскадных реакций

Оценка выше дает прямую связь тензорной структуры оператора ОКУ с тензорной структурой входных данных (функций скорости реакций), которая в общем случае выражается суммой по всем реакциям. Однако пару самых важных случаев *системных взаимодействий* стоит проанализированы более подробно.

Простейший пример это сумма независимых действий, или операторов, описанная и в секции 2.2.3:

$$\begin{aligned} x(\boldsymbol{i}) &= a(i_1) \cdot b(i_2) \cdots b(i_d) + \cdots + b(i_1) \cdots b(i_{d-1}) \cdot a(i_d) \\ &= \begin{bmatrix} a(i_1) & b(i_1) \end{bmatrix} \begin{bmatrix} b(i_2) & 0 \\ a(i_2) & b(i_2) \end{bmatrix} \cdots \begin{bmatrix} b(i_{d-1}) & 0 \\ a(i_{d-1}) & b(i_{d-1}) \end{bmatrix} \begin{bmatrix} b(i_d) \\ a(i_d) \end{bmatrix}. \end{aligned}$$

Следующим по сложности является *попарное*, или *каскадное* взаимодействие, которое также приводит к хорошо представимой матрице ОКУ в ТТ формате.

Лемма 3.3.1 ([51])**.** Пусть даны элементарные векторы или матрицы $E_k(i_k)$, $F_k^k(i_k)$, $F_k^{k+1}(i_k)$ для $k = 1, \ldots, d$. Сумма попарных произведений

$$H = F_1^1 \otimes \left(\bigotimes_{j=2}^{d} E_j \right) + \sum_{k=2}^{d} \left(\bigotimes_{j=1}^{k-2} E_j \right) \otimes F_{k-1}^k \otimes F_k^k \otimes \left(\bigotimes_{j=k+1}^{d} E_j \right) \tag{3.3}$$

обладает точным ТТ разложением $H(\boldsymbol{i}) = H^{(1)}(i_1) \cdots H^{(d)}(i_d)$ ранга 3, вида

$$\begin{aligned} H^{(1)}(i_1) &= \begin{bmatrix} E_1(i_1) & F_1^2(i_1) & F_1^1(i_1) \end{bmatrix}, \quad H^{(d)}(i_d) = \begin{bmatrix} 0 \\ F_d^d(i_d) \\ E_d(i_d) \end{bmatrix}, \\ H^{(k)}(i_k) &= \begin{bmatrix} E_k(i_k) & F_k^{k+1}(i_k) & 0 \\ 0 & 0 & F_k^k(i_k) \\ 0 & 0 & E_k(i_k) \end{bmatrix}, \quad \text{если } k = 2, \ldots, d-1. \end{aligned} \tag{3.4}$$

Разложение Таккера также выполняется с рангами не более 3.

Доказательство. Первый шаг индукции в разделении переменных пишется так:

$$H(\boldsymbol{i}) = \begin{bmatrix} E_1(i_1) & F_1^2(i_1) & F_1^1(i_1) \end{bmatrix} \begin{bmatrix} \tilde{H}_2(i_2, \ldots, i_d) \\ F_2^2(i_2) E_3(i_3) \cdots E_d(i_d) \\ E_2(i_2) \cdots E_d(i_d) \end{bmatrix},$$

где первый сомножитель дает в точности первый ТТ блок. Остальную часть обозначим

$$\tilde{H}_k = F_k^{k+1} \otimes F_{k+1}^{k+1} \otimes E_{k+1} \otimes \cdots \otimes E_d + \cdots + E_k \otimes \cdots \otimes E_{d-2} \otimes F_{d-1}^d \otimes F_d^d.$$

В такой общей записи второго члена, мы аналогичным образом отделяем k-ю размерность в каждой строке:

$$\begin{bmatrix} \tilde{H}_k(i_k, \ldots, i_d) \\ F_k^k(i_k) E_{k+1}(i_{k+1}) \cdots E_d(i_d) \\ E_k(i_k) \cdots E_d(i_d) \end{bmatrix} = H^{(k)}(i_k) \begin{bmatrix} \tilde{H}_{k+1}(i_{k+1}, \ldots, i_d) \\ F_{k+1}^{k+1}(i_{k+1}) E_{k+2}(i_{k+2}) \cdots E_d(i_d) \\ E_{k+1}(i_{k+1}) \cdots E_d(i_d) \end{bmatrix},$$

и получаем k-й TT блок в качестве первого фактора. Обращая внимание на второй фактор, заключаем, что индуктивное правило теперь установлено, и мы можем закончить процесс на двух последних блоках:

$$\begin{bmatrix} F^d_{d-1}(i_{d-1})F^d_d(i_d) \\ F^{d-1}_{d-1}(i_{d-1})E_d(i_d) \\ E_{d-1}(i_{d-1})E_d(i_d) \end{bmatrix} = \begin{bmatrix} F^d_{d-1}(i_{d-1}) & 0 \\ 0 & F^{d-1}_{d-1}(i_{d-1}) \\ 0 & E_{d-1}(i_{d-1}) \end{bmatrix} \begin{bmatrix} F^d_d(i_d) \\ E_d(i_d) \end{bmatrix}.$$

Наконец, для большей однородности записи, дополним второй член нулем до вида $H^{(d)}$ в (3.4). Это придаст блоку $H^{(d-1)}$ общий вид $H^{(k)}$.

Мы видим, что все TT ранги равны 3, что доказывает первое утверждение леммы. Для получения оценки на ранг Таккера, достаточно отметить, что каждый TT блок содержит только 3 независимых элемента, и применить процедуру преобразования TT в Таккер из секции 2.2.3. □

Замечание 3.3.2. В выражении (3.3), каждое слагаемое является тензором (TT) ранга 1. Тем не менее, достаточно просто обобщить эту структуру на случай, когда соседние члены суммируются из нескольких компонентов:

$$\sum_{\alpha_{k-1}=1}^{r_{k-1}} F^k_{k-1,\alpha_k} \otimes F^k_{k,\alpha_k} \quad \text{вместо} \quad F^k_{k-1} \otimes F^k_k,$$

т.е. TT ранг каждого слагаемого (например, функция скорости в ОКУ) больше 1. В этом случае, мы можем собрать соответственно строчные и столбцовые векторы

$$F^k_{k-1}(i_{k-1}) = \begin{bmatrix} F^k_{k-1,1}(i_{k-1}) & \cdots & F^k_{k-1,r_{k-1}}(i_{k-1}) \end{bmatrix}, \quad F^k_k(i_k) = \begin{bmatrix} F^k_{k,1}(i_k) \\ \vdots \\ F^k_{k,r_{k-1}}(i_k) \end{bmatrix},$$

и выражения (3.4) можно рассматривать как блочные матрицы с размерами (и, соответственно, TT рангами) $(2 + r_{k-1}) \times (2 + r_k)$. Подсчитывая число линейно независимых элементов в каждом TT факторе, мы можем заключить, что k-й Таккеровский ранг ограничен $1 + r_{k-1} + r_k$.

Замечание 3.3.3. Для простоты мы рассмотрели только векторный случай, т.е. когда $H = [H(\boldsymbol{i})]$ зависит от одного индекса $\boldsymbol{i}$. Тем не менее, все те же рассуждения справедливы и для матриц, то есть мы можем свободно заменить $\boldsymbol{i}$ на $\boldsymbol{i}, \boldsymbol{j}$, каждый i_k на i_k, j_k, и так далее. Это обобщает результат до вида, непосредственно пригодного для построения оператора ОКУ для каскада реакций.

3.4 Обращение дискретного оператора Лапласа и преобразование Фурье

В некоторых случаях не только конкретное решение, но и вся обратная матрица системы уравнений может быть построена в структурированной форме. Одним из первых и замечательных примеров был предложен оператор Лапласа [29, 78,

93, 79, 104, 105, 28]. Итак, приведем метод построения обратной матрицы для дискретизации оператора Лапласа в соответствии с [93, 79].

Пусть дана положительно определенная матрица A, ее обращение может быть записано в интегральном виде

$$A^{-1} = \int_0^\infty \exp(-tA)dt,$$

что можно проверить путем умножения правой части на A. Теперь, пусть

$$A = A^{(1)} \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes A^{(d)},$$

тогда экспонента разлагается в тензорное произведение "одномерных" матриц,

$$A^{-1} = \int_0^\infty \exp(-tA^{(1)}) \otimes \exp(-tA^{(2)}) \otimes \cdots \otimes \exp(-tA^{(d)})dt.$$

Конструктивная аппроксимация A^{-1} в каноническом тензорном формате получается путем замены интеграла конечной квадратурой. Например, в правиле Штенгера [93] вычисляется

$$\begin{aligned} A^{-1} &\approx \sum_{m=-M}^{M} c_m \bigotimes_{k=1}^{d} \exp(-t_m A^{(k)}), \\ t_m &= \log\left(\exp(mh) + \sqrt{1+\exp(2mh)}\right), \quad h = \pi^2/\sqrt{M}, \\ c_m &= h/\sqrt{1+\exp(-2mh)}, \end{aligned}$$

гарантируя почти экспоненциальную сходимость

$$\left\| A^{-1} - \sum_{m=-M}^{M} c_m \bigotimes_{k=1}^{d} \exp(-t_m A^{(k)}) \right\|_2 \leqslant C(4+2\|A\|_2) \exp\left(-\pi\sqrt{2M}\right).$$

Очевидно следует оценка $R = (2M+1) = \mathcal{O}(\log^2(1/\varepsilon))$ на канонический ранг. Существуют также sinc-квадратуры со сходимостью $\mathcal{O}(\exp(-cM/\log M))$, см. например обзор [143].

Интересно что качество приближения слабо зависит от малых возмущений в квадратурном правиле. Для вычисления матричных экспонент $B_m^k = \exp(-t_m A^{(k)})$ с помощью быстрых рекуррентных формул, например, $B_{m+1}^k = (B_m^k)^2$, в работе [151] была предложена модифицированная sinc-квадратура. Пусть дана оригинальная sinc-схема

$$h = \pi/\sqrt{M}, \quad t_m = \mathrm{e}^{mh}, \quad c_m = ht_m,$$

мы формально полагаем $h = \log 2$, так что $t_m = 2^m = 2t_{m-1}$. Точное правило потребовало бы нецелого $M = (\pi/h)^2 \approx 20.54$, так что мы выбираем $M = 21$, что тем не менее дает разумное приближение, особенно для задачи предобусловливания. Такая же процедура может быть применена и для более общих матриц в форме

Кронекеровских произведений, например, вытекающих из уравнения Ляпунова [24].

Если требуется не весь обратный оператор, а только решение линейной системы $Ax = y$, может быть более эффективно избегать полных матриц $\exp(-t_m A^{(k)})$, вычисляя напрямую произведения $\exp(-t_m A^{(k)})y^{(k)}$, с использованием быстрого преобразования Фурье. Многомерное дискретное преобразование Фурье пишется по определению

$$\hat{y}(j_1, \dots, j_d) = \sum_{i_1,\dots,i_d} \exp\left(-\frac{2\pi \mathrm{i}}{n_1} i_1 j_1\right) \cdots \exp\left(-\frac{2\pi \mathrm{i}}{n_d} i_d j_d\right) y(i_1, \dots, i_d),$$

и представляет собой произведение многоуровневой матрицы TT ранга 1 на вектор,

$$\hat{y} = \left(F^{(1)} \otimes \cdots \otimes F^{(d)}\right) y, \quad F^{(k)}_{j_k, i_k} = \exp\left(-\frac{2\pi \mathrm{i}}{n_k} i_k j_k\right).$$

Однако обобщение этой формулы на QTT формат невозможно, поскольку одномерная матрица Фурье имеет неуменьшаемый QTT ранг 2^L для любого уровня точности вплоть до $1 - \mathcal{O}(2^{-L})$. Тем не менее, *вычисление* преобразования Фурье может быть выполнено в формате QTT со сложностью $\mathcal{O}(r^3 L^2)$, см. [54]. Однако в процессе такого алгоритма возникают промежуточные векторы, которые могут потенциально требовать большие QTT ранги, даже если и входные, и выходные данные хорошо структурированы.

Глава 4

Итерационные методы в тензорных форматах

До сих пор мы рассматривали только явные операции, которые могут быть выполнены за конечное число шагов с гарантированным результатом. Даже обратная матрица в последней секции была построена аналитически. Однако разнообразие тензорных операций не ограничивается базовой полилинейной алгеброй. Не меньшее значение имеет получение *неявных* решений, что во многих случаях может быть выполнено только приближенно с помощью итерационных методов. Успешное моделирование в высоких размерностях (например, уравнения Власова или основного кинетического) требует решения линейных систем (стационарных задач и неявных схем во времени), задач на собственные значения, более сложных матричных функций, таких как экспонента, и т.д. В этой работе, матричные функции и задачи на собственные значения остаются в стороне, так как для всех приложений, описанных в первой главе, достаточно решения линейной системы, или даже только матричного произведения (см. секцию 3.2). Мы посвящаем эту главу краткому обзору известных методов, и построению семейства новых алгоритмов для решения линейных систем.

Для сокращения формул, в этой главе мы зарезервируем "анонимное" обозначение тензорных рангов r, r_k для тензора решения x, т.е. $r = r(x)$, $r_k = r_k(x)$. Другие величины будем обозначать как и предложено в определении 2.1.10, например, $r_k(A)$, $r_k(b)$.

В данной главе, секции 4.1 и 4.2 посвящены обзору существующих итерационных алгоритмов в тензорных произведениях. Алгоритмы, начиная с секции 4.3.3 и до конца главы предложены автором. Их анализ проведен совместно с Д. Савостьяновым.

4.1 Итерационные методы с приближенными тензорными операциями

Тензорная линейная алгебра вместе с процедурой округления позволяет мыслить в терминах классических алгоритмов, и первые попытки были связаны с использо-

Алгоритм 6 Метод Ричардсона в TT формате [150]

Ввод: Матрица A в TT формате (2.11), начальное приближение x_0 и правая часть b в TT форматах (2.9), размер шага λ, точность аппроксимации ε.

Вывод: Улучшенное решение x_1.

1: **for** до сходимости **do**
2: Вычислить произведение $w = Ax_0$ в TT формате.
3: Аппроксимировать $w = \mathcal{T}_\varepsilon(w)$. {Опционально}
4: Вычислить $x_1 = x_0 + \lambda b - \lambda w$ в TT формате.
5: Аппроксимировать $x_1 = \mathcal{T}_\varepsilon(x_1)$.
6: **end for**

ванием именно традиционных итерационных методов с использованием тензорной арифметики:

- алгоритмов Ричардсона и Ньютона [8, 107, 137, 141, 150, 13, 26],
- сопряженных и би-сопряженных градиентов [167, 168, 160, 161, 13, 26, 31], и также
- различных реализаций GMRES [16, 52].

Самый простой метод итераций Ричардсона с тензорными приближениями показан в Алг. 6.

Предобуславливатель B может быть внесен в схему путем повторения для него шагов 2 и 3, т.е. $w = \mathcal{T}_\varepsilon\left(B\mathcal{T}_\varepsilon(Ax_0)\right)$, и b в строке 4 алгоритма заменяется на Bb.

Для базового анализа ошибок, мы полагаем, что решение возмущается за счет тензорных аппроксимаций (строка 5 в Алг. 6), но невязка пока вычисляется точно, т.е. $w = Ax_0$.

Лемма 4.1.1. Пусть дана линейная система $Ax = b$, точное решение которой обозначим $x_\star$. Пусть начальная ошибка $e_0 = x_\star - x_0$, а ошибка после одного шага метода Ричардсона 6 с возмущением решения с порогом ε равна $e_1 = x_\star - x_1$. Предположим, что для невозмущенного метода имеет место убывание ошибки $\|e_1\|_2/\|e_0\|_2 \leqslant \Omega < 1$. Тогда для Алг. 6 убывание ошибки на одном шаге оценивается как

$$\frac{\|e_1\|_2}{\|e_0\|_2} \leqslant \Omega\left(1 + \varepsilon\frac{\|x_0\|_2}{\|e_0\|_2}\right). \tag{4.1}$$

Таким образом, Алг. 6 сходится до тех пор, пока не достигнута следующая окрестность точного решения:

$$\frac{\|e_0\|_2}{\|x_0\|_2} \leqslant \varepsilon\frac{\Omega}{1-\Omega}.$$

Доказательство. Рассмотрим вариант Алг. 6, когда аппроксимация решения проводится на первом шаге, то есть следующую последовательность операций

1. аппроксимировать начальное приближение, $u = \mathcal{T}_\varepsilon(x_0)$;
2. вычислить поправку к решению, $x_1 = u + \lambda(b - Au)$.

Алгоритм 7 TT-GMRES(m) [52]

Ввод: Правая часть b, начальное приближение x_0 в TT формате, матрица A в виде процедуры умножения на вектор, $y = \mathcal{T}_{\varepsilon,r}(Ax)$, точность ε и/или максимальный TT ранг r.

Вывод: Приближенное решение $x_j : \ \|Ax_j - b\|/\|b\| \leqslant \varepsilon$.

Начало: вычислить $z_0 = \mathcal{T}_{\varepsilon,r}(b - Ax_0)$, $\beta = \|z_0\|$ $v_1 = z_0/\beta$.
Итерации:
for $j = 1, 2, \ldots, m$ **do**
  Посчитать огрубленную точность $\delta = \dfrac{\varepsilon}{\|\tilde{z}_{j-1}\|/\beta}$.
  $w = \mathcal{T}_{\delta,r}(Av_j)$: новый Крыловский вектор.
  for $i = 1, 2, \ldots, j$ **do**
    $h_{i,j} = (w, v_i)$,
    $w = w - h_{i,j}v_i$, {ортогонализация}
  end for
  $w = \mathcal{T}_{\delta,r}(w)$. {TT-сжатие}
  $h_{j+1,j} = \|w\|$, $v_{j+1} = w/h_{j+1,j}$.
  Собрать матрицу $\bar{H}_j = [h_{i,k}]$, $k = 1, \ldots, j$, $i = 1, \ldots, j+1$.
  Решить редуцированную систему: $y_j = \arg\min_y \|\beta e_1 - \bar{H}_j y\|$.
  Вычислить невязку $\|\tilde{z}_j\| = \|\beta e_1 - \bar{H}_j y_j\|$: если $\|\tilde{z}_j\|/\|b\| \leqslant \varepsilon$, то стоп.
end for
Коррекция решения: положить $x_j = x_0$,
for $i = 1, 2, \ldots, j$ **do**
  $x_j = x_j + y_j(i)v_i$ {поправка}
end for
$x_j = \mathcal{T}_{\varepsilon,r}(x)$ {TT-сжатие}
Рестарт: если $\|\tilde{z}_j\|/\|b\| > \varepsilon$, то положить $x_0 = x_j$, и перейти к 1.

Очевидно, такая пересортировка шагов не меняет промежуточных итераций алгоритма. Тогда изменение ошибки на одном шаге алгоритма выражается как

$$\frac{\|x_\star - x_1\|}{\|x_\star - x_0\|} = \frac{\|x_\star - x_1\|}{\|x_\star - u\|} \cdot \frac{\|x_\star - u\|}{\|x_\star - x_0\|} \leqslant \Omega \cdot \frac{\|x_\star - u\|}{\|x_\star - x_0\|}.$$

Подставляем возмущенный вектор $u = x_0 + \eta$, $\|\eta\| \leqslant \varepsilon\|x_0\|$, и получаем

$$\frac{\|x_\star - x_1\|}{\|x_\star - x_0\|} \leqslant \Omega \cdot \frac{\|x_\star - x_0\| + \|\eta\|}{\|x_\star - x_0\|} \leqslant \Omega \left(1 + \varepsilon \left(\frac{\|x_\star - x_0\|}{\|x_0\|}\right)^{-1}\right).$$

Скорость убывания ошибки меньше единицы, если $\|e_0\|/\|x_0\| \geqslant \varepsilon \cdot \Omega/(1 - \Omega)$, что и обеспечивает сходимость метода.

□

Формула (4.1) не только показывает, что сходимость ухудшается, когда процесс приближается к $\mathcal{O}(\varepsilon)$-окрестности точного решения. Она также дает нам представление о понятии *релаксации*: мы не обязаны держать один и тот же ε для всех

итераций, но можно допускать менее точные и более быстрые вычисления на некоторых этапах процесса. Как правило, в первые итерациях ошибка $\|e_0\|$ велика, и мы можем использовать довольно грубый порог ε не теряя сходимости метода.

Замечание 4.1.2. Скорость *геометрической* сходимости Ω, как правило, оценивается с помощью спектра матрицы A, например в виде $\Omega \lesssim 1 - 1/\mathrm{cond}(A)$, где $\mathrm{cond}(A)$ это число обусловленности матрицы. Фактор возрастания ошибки, таким образом, может достигать величины порядка числа обусловленности, $\Omega/(1-\Omega) \sim \mathrm{cond}(A)$. Это может быть несущественно в стандартной машинной арифметике с точностью $\mathcal{O}(10^{-16})$, но расчеты в тензорных форматах вносят гораздо большие погрешности, например 10^{-3}—10^{-8}. Таким образом, очень важно использование спектрально эквивалентного предобуславливателя.

Не любой метод предобуславливания легко адаптируется для тензорных структурированных представлений. Например, неполное LU разложение требует доступа ко всем элементам матрицы, и соответственно полному формату хранения тензора, что по нашим предположениям невозможно. Среди доступных методов можно выделить многосеточные (в основном геометрические [16]; алгебраические версии не должны исследовать всю матрицу целиком. Среди таковых, например, вариант BPX из [13]), и приближенные обратные матрицы, имеющие явную низкоранговую структуру. Замечательным примером последнего является обратным дискретный оператор Лапласа (см. раздел 3.4), нашедший применение в [137, 141, 150, 52].

Отсутствие эффективного предобусловливателя общего вида может быть частично компенсировано за счет использования более продвинутых итерационных методов. Например, метод GMRES с приближенными тензорными вычислениями Крыловских векторов может быть реализован как показано в Алг. 7. Отметим, что точность векторов Крылова огрубляется в соответствии с текущей невязкой (строка 4). Как показано в теории неточных Крыловских методов [224], такая релаксация не нарушает сходимость. Точнее, имеет место следующее утверждение.

Утверждение 4.1.3 (Следствие 4.1 [52] из Теоремы 5.3 [224])**.** Пусть проведено m итераций GMRES. Если для любого $i \leqslant m$ относительная погрешность, внесенная в произведение матрицы на вектор, удовлетворяет

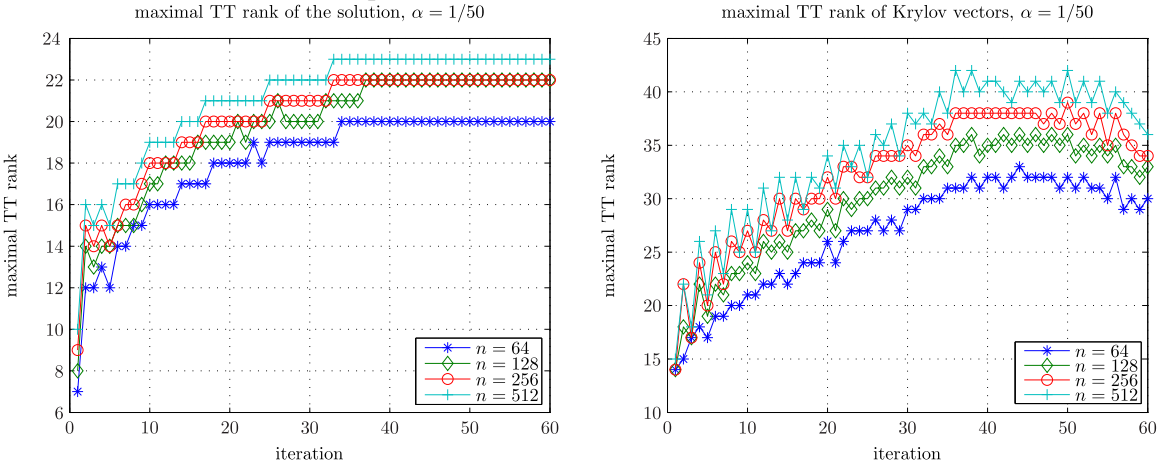
$$\frac{\|w\|}{\|Av_j\|} \leqslant \frac{1}{m \cdot \mathrm{cond}(A)} \frac{1}{\|\tilde{z}_{i-1}\|/\|b\|}\epsilon, \tag{4.2}$$

то реальная относительная невязка z_m связана с невязкой в редуцированной GMRES системе $\tilde{z}_m$ следующим образом:

$$\frac{\|z_m\|}{\|b\|} \leqslant \frac{\|\tilde{z}_m\|}{\|b\|} + \epsilon.$$

Однако, Крыловские векторы требуют обычно больших тензорных рангов, даже несмотря на релаксацию точности, см. рис. 4.1. Это явление можно объяснить тем наблюдением, что GMRES последовательно корректирует спектральные гармоники в решении, и чем более гладко и точно приближенное решение, тем более

Рис. 4.1: Пример из [52]: алгоритм TT-GMRES дает решение с быстро насыщающимися TT рангами (слева), в то время как TT ранги последнего Крыловского вектора (справа) достигают значительно больших значений, и релаксация уменьшает их только в конце процесса.



осциллирующий вид имеют невязка и Крыловские векторы. Все более и более сложная структура старших Крыловских векторов приводит к росту тензорных рангов, необходимых для обеспечения их приближения с достаточной точностью. Это ограничивает применимость Крыловских методов с тензорными форматами.

4.2 Оптимизация на элементах тензорных форматов

4.2.1 Классические итерации и методы переменных направлений

В предыдущем разделе мы видели, что классические итерационные методы с тензорной арифметикой не очень надежны, поскольку различные вспомогательные векторы могут требовать больших тензорных рангов, даже если матрица, правая часть и решение хорошо представимы в формате. Чтобы избавиться от дополнительных векторов, другое семейство методов предлагает решать уравнения напрямую на элементы тензорного формата.

Постановка задачи оптимизации на элементах тензорных форматов начинается с целевой функции. Самый простой выбор это ошибка во Фробениусовой норме в терминах исходных элементов тензоров:

$$\mathtt{E}_{x_\star} = \|x_\star - \tau(x^{(1)}, \ldots, x^{(d)})\|^2 \to \min \tag{4.3}$$

по TT блокам $x^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}$. Обратите внимание, что в связи с полилинейностью TT отображения τ, функция ошибки, будучи квадратичной для элементов тензора, становится существенно нелинейной и невыпуклой относительно TT элементов. Хотя прямая оптимизация и разработана до некоторой степени (см.,

например, методы Ньютона и квазиньютоновские для канонического формата [154, 238, 68, 3], или обобщенное собственное разложение [61]), этот подход обычно применяется к ограниченному классу задач, и не очень надежен в более общем случае.

Чтобы ослабить нелинейность, было предложено семейство методов *переменных направлений*. Общий подход состоит в замене задачи (4.3) последовательностью квадратичных оптимизаций по элементам каждого TT блока. С 1980-х годов, метод *наименьших квадратов* (Alternating Least Squares, ALS) для форматов Таккера [163] и канонического [38] стал обширно применяться для классификации данных моделями низкого ранга в психометрии и другие подобных задачах.

Несмотря на невероятно медленную сходимость во многих случаях, метод ALS для тензорных форматов обладает одним важным преимуществом: поскольку формат линеен по отношению к каждому из своих блоков, целевая функция, ограниченная на элементы блока, сохраняет ту же полиномиальную степень, какая была присуща исходной функции на элементах полного тензора. Это делает редуцированную задачу гораздо проще, чем одновременная оптимизация (4.3). Тому, как восстановить быструю сходимость и избежать стагнации в локальных минимумах, и будет посвящен этот и следующий разделы.

Сначала покажем, что TT формат действительно является линейным относительно выбранного блока. Напомним определение TT отображения 2.1.11 для частичного TT представления:

$$\begin{aligned} x^{(<k)} &= \tau(x^{(1)}, \dots, x^{(k-1)}) \in \mathbb{C}^{n_1 \cdots n_{k-1} \times r_{k-1}}, \\ x^{(>k)} &= \tau(x^{(k+1)}, \dots, x^{(d)}) \in \mathbb{C}^{r_k \times n_{k+1} \cdots n_d}. \end{aligned}$$

Теперь мы можем определить *фрейм*-матрицу.

Определение 4.2.1. Пусть дан TT формат $x^{(1)}, \dots, x^{(d)}$, введем k-ю фрейм-матрицу следующим образом:

$$X_{\neq k} = x^{(<k)} \otimes I_{n_k} \otimes \left(x^{(>k)}\right)^\top \in \mathbb{C}^{n_1 \cdots n_d \times r_{k-1} n_k r_k}. \tag{4.4}$$

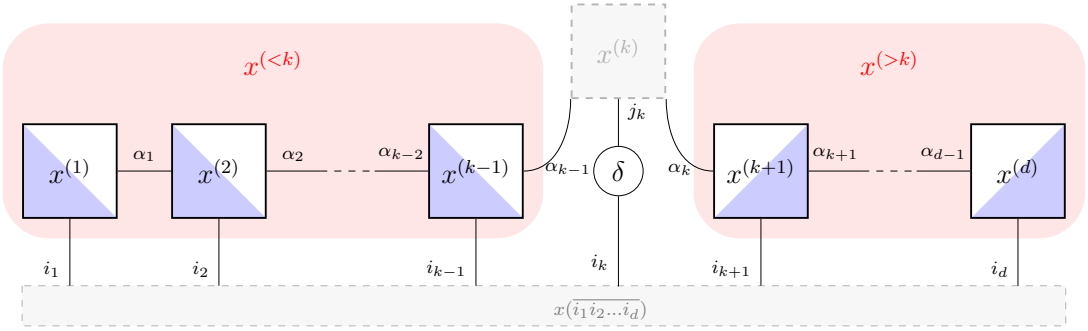
Утверждение о линейности следует прямо из определений TT формата и фрейм-матрицы.

Утверждение 4.2.2. Фрейм-матрица задает линейное отображение из элементов TT блока в элементы начального тензора:

$$x = X_{\neq k} x^{(k)}. \tag{4.5}$$

Кроме того, вводя условия ортогональности на TT блоки, мы можем сделать и всю фрейм-матрицу ортогональной (унитарной). В самом деле, напомним, что по лемме 2.1.16 кусок $x^{(<k)}$ лево-ортогонален, если таковыми являются соответствующие TT блоки $x^{(1)}, \dots, x^{(k-1)}$. Аналогичным образом, правая ортогональность блоков $x^{(k+1)}, \dots, x^{(d)}$ обеспечивает ортогональность правого куска $x^{(>k)}$, и вместе они дают унитарность фрейм-матрицы, как показано на рис. 4.2.

Рис. 4.2: Фрейм-матрица (4.4) отображает TT блок (сверху) в полный вектор (снизу).



4.2.2 Решение задач линейной алгебры с помощью оптимизации

Исходная алгебраическая задача, такая как аппроксимация, решение линейной системы или симметричной задачи на собственные значения, может часто рассматриваться как оптимизация некоторой квадратичной целевой функции. Наиболее важными являются:

1. ошибка $\mathtt{E}_{x_\star}(x) = \|x_\star - x\|^2 = \mathtt{J}_{I,x_\star}$,
2. энергия $\mathtt{J}_{A,b}(x) = \|x_\star - x\|_A^2 = (x, Ax) - 2\mathrm{Re}(x, b) + \mathrm{const}$, где $b = Ax_\star$,
3. невязка $\mathtt{R}_{A,b}(x) = \|Ax - b\|^2 = \mathtt{J}_{A^*A,A^*b}(x) + \mathrm{const}$, и
4. отношение Рэлея $\mathtt{Q}_A(x) = (x, Ax)/(x, x)$.

Минимум отношения Рэлея достигается на решении экстремальной собственной задачи $Ax = \lambda x$ при $\lambda = \min \mathtt{Q}_A(x)$, и в текущей работе не рассматривается, в то время как остальные три функции связаны с решением некоторой (может быть, с единичной матрицей) линейной системы. Функция энергии определяется для симметричной положительно определенной (SPD) матрицы, $A = A^* > 0$, с A-скалярным произведением, и A-нормой, введенными обычным образом: $(x, y)_A = (x, Ay)$, $\|x\|_A = \sqrt{(x, x)_A}$.

Теперь напомним метод наименьших квадратов в переменных направлениях (ALS), также названный *линейная схема* в переменных направлениях в [116], где было показано, что ALS схема совпадает с (одноблочным) *Density Matrix Renormalization Group* (DMRG) подходом из квантовой физики. Оригинальная DMRG схема [249, 250] была предложена для вычисления основного состояния системы путем минимизации отношения Рэлея. Позднее она была применена [126] для решения SPD линейной системы $Ax = b$, где A и b заданы в TT формате, путем минимизации функции энергии. Мы будем работать с последней задачей линейной системы, хотя связь с "оригинальным" DMRG проста, и большинство идей может быть использовано в обеих задачах.

Так, мы ограничиваем оптимизацию $\mathtt{J}_{A,b}(x)$ на векторы $x = \tau(x^{(1)}, \ldots, x^{(d)})$, которые представлены в TT формате с *фиксированными* TT рангами $\boldsymbol{r} = (r_1, \ldots, r_{d-1})$,

и выполняем фактические вычисления посредством последовательности *микрошагов*, т.е. *последовательных* оптимизаций по TT блокам $x^{(k)}$. Каждая такая *локальная задача* ставится следующим образом:

$$u^{(k)} = \arg\min_{x^{(k)}} \mathtt{J}_{A,b}(\tau(x^{(1)}, \dots, x^{(d)})) \quad \text{по} \quad x^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}. \tag{4.6}$$

TT блок $x^{(k)}$ затем заменяется на $u^{(k)}$, и процесс переходит к следующему ядру; обычно блоки пересчитываются в последовательном проходе по размерностям, например $k = 1, \dots, d$ (*прямой полу–проход*), или $k = d, \dots, 1$ (*обратный полу–проход*), и так далее вплоть до сходимости x.

Линейность TT формата (4.5) позволяет переписать (4.6) в виде

$$\begin{aligned} u^{(k)} &= \arg\min_{x^{(k)}} \mathtt{J}_{A_k,b_k}(x^{(k)}) \quad \text{по} \quad x^{(k)} \in \mathbb{C}^{r_{k-1} n_k r_k}, \\ A_k = X_{\neq k}^* A X_{\neq k} \in \mathbb{C}^{(r_{k-1} n_k r_k) \times (r_{k-1} n_k r_k)}, &\qquad b_k = X_{\neq k}^* b \in \mathbb{C}^{r_{k-1} n_k r_k}. \end{aligned} \tag{4.7}$$

Здесь единственный минимум обеспечивается решением *локальной линейной системы* $A_k u^{(k)} = b_k$,[1] которая имеет разумный размер и может быть решена стандартным методом, например, исключением Гаусса или итерационным [213]. Как показано на рис. 4.3, A_k и b_k могут быть собраны из TT блоков $A = \tau(\{A^{(k)}\})$, $x = \tau(\{x^{(k)}\})$, и $b = \tau(\{b^{(k)}\})$, без использования массивов исходного (большого) размера $\mathcal{O}(n^d)$.

Точность полученного решения критически зависит от обусловленности локальной системы. Ее можно контролировать с помощью ограничений ортогональности на TT блоки, и, следовательно, ортогональности фрейм-матрицы. В последовательном проходе по TT блокам, оптимизация по переменным направлениям может быть синхронизирована с шагами ортогонализации Алгоритмов 2 и 3, как показано в Алг. 8. Если $X_{\neq k}$ ортогональна, спектр A_k лежит внутри спектрального диапазона A. Действительно,

$$\begin{aligned} \lambda_{\min}(A_k) = \lambda_{\min}(X_{\neq k}^* A X_{\neq k}) &= \min_{\|v\|=1} (X_{\neq k} v, A X_{\neq k} v) = \min_{\substack{\|u\|=1 \\ u \in \operatorname{span} X_{\neq k}}} (u, Au) \\ &\geqslant \min_{\|u\|=1} (u, Au) = \lambda_{\min}(A), \end{aligned}$$

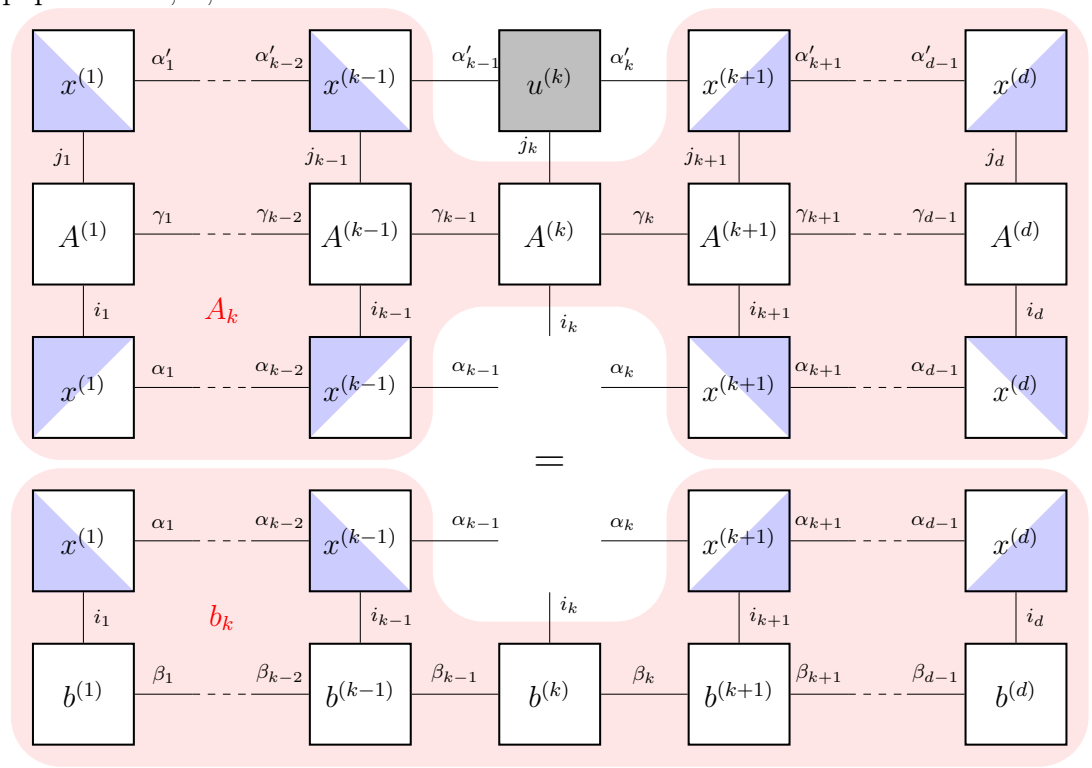
и аналогично $\lambda_{\max}(A_k) \leqslant \lambda_{\max}(A)$. Поэтому, *числа обусловленности* удовлетворяют условию $\operatorname{cond}(A_k) \leqslant \operatorname{cond}(A)$, т.е. локальная система (4.7) обусловлена не хуже, чем $Ax = b$.

4.2.3 Проблема адаптация рангов и двухблочный DMRG

Недостатком одноблочного алгоритма DMRG является то, что TT ранги остаются постоянными в процессе расчета. Таким образом, мы должны угадывать правильные значения рангов решения *априори*, что может быть трудно; если мы

[1] Напомним определение 2.1.12 и замечание после него: произведение $A_k u^{(k)}$ подразумевает "векторный" вид TT блока, $u^{(k)} \in \mathbb{C}^{r_{k-1} n_k r_k}$.

Рис. 4.3: Линейная система $A_k u^{(k)} = b_k$, определенная в (4.7), собирается из блоков TT форматов A, x, и b.



недооцениваем их, приближенное решение будет далеко от точного; если мы переоцениваем их, сложность локальных задач будет сильно завышена. Кроме того, сходимость даже к квази-оптимальному решению при данных TT рангах может быть очень медленной.

Со времен первых статей [249, 250], *двухблочный* DMRG стал классическим методом для адаптивного *изменения* (обычно *увеличения*) TT рангов в процессе расчета. Хороший обзор различных методов DMRG можно найти в работе [221], и ее "втором издании" [222]. Сообществу численной линейной алгебры двухблочный DMRG был показан в [116] под названием *модифицированная линейная схема переменных направлений* (Modified Alternating Linear Scheme, MALS).

Данный метод работает с вектором в следующем виде:

$$x = \tau(x^{(1)}, \ldots, x^{(k-1)}, x^{(k,k+1)}, x^{(k+2)}, \ldots, x^{(d)}), \tag{4.8}$$

где $x^{(k,k+1)} = x^{(k,\ldots,k+1)} = \tau(x^{(k)}, x^{(k+1)})$ пишется в соответствии с определением 2.1.11. Шаг локальной оптимизации проводится аналогично (4.6), но по элементам *суперядра* $x^{(k,k+1)}$ следующим образом:

$$u^{(k,k+1)} = \arg\min_{x^{(k,k+1)}} \mathrm{J}_{A_{k,k+1}, b_{k,k+1}}(x^{(k,k+1)}) \quad \text{по} \quad x^{(k,k+1)} \in \mathbb{C}^{r_{k-1} n_k n_{k+1} r_{k+1}}, \tag{4.9}$$

что эквивалентно решению *двухблочной локальной системы* $A_{k,k+1} u^{(k,k+1)} = b_{k,k+1}$,

Алгоритм 8 Одношаговый DMRG для $Ax = b$ (прямой полу–проход) [116]

Ввод: Начальное приближение $t = \tau(\{t^{(k)}\})$ в TT формате (2.9).
Вывод: Новое приближение $x = \tau(\{x^{(k)}\})$ с условием $\mathtt{J}_{A,b}(x) \leqslant \mathtt{J}_{A,b}(t)$.

```
Скопировать x^(k) = t^(k), k = 1, ..., d.
for k = d, ..., 2 do {Ортогонализация справа налево}
    Найти LQ разложение x^⟨k| = LQ, QQ* = I.
    Заменить x^⟨k| := Q, и x^|k−1⟩ := x^|k−1⟩ L.
end for
for k = 1, ..., d do {Оптимизация TT блоков}
    Собрать A_k и b_k как показано в (4.7).
    Решить A_k u^(k) = b_k. {Если используется итерационный метод, взять x^(k) для
    начального приближения}
    Заменить x^(k) := u^(k).
    if k ≠ d then {Ортогонализация левого интерфейса}
        Найти QR разложение x^|k⟩ = QR, Q*Q = I.
        Заменить x^|k⟩ := Q, и x^⟨k+1| := R x^⟨k+1|.
    end if
end for
return x = τ(x^(1), ..., x^(d)).
```

где

$$\begin{aligned} A_{k,k+1} &= X^*_{\neq k,k+1} A X_{\neq k,k+1} \in \mathbb{C}^{(r_{k-1}n_k n_{k+1} r_{k+1}) \times (r_{k-1}n_k n_{k+1} r_{k+1})}, \\ b_{k,k+1} &= X^*_{\neq k,k+1} b \in \mathbb{C}^{r_{k-1}n_k n_{n+1} r_{k+1}}, \end{aligned} \tag{4.10}$$

а $X_{\neq k,k+1}$ обозначает *двухблочную фрейм-матрицу*

$$X_{\neq k,k+1} = x^{(<k)} \otimes I_{n_k} \otimes I_{n_{k+1}} \otimes \left(x^{(>k+1)}\right)^\top. \tag{4.11}$$

Как и ранее, мы предполагаем ортогональность $X^*_{\neq k,k+1} X_{\neq k,k+1} = I_{r_{k-1}n_k n_{k+1} r_{k+1}}$, так что обусловленность локальной задачи не хуже, что обусловленность исходной системы $Ax = b$.

Решение системы (4.9) $u^{(k,k+1)}$ затем разделяется обратно на $u^{(k)}$ и $u^{(k+1)}$ для восстановления первоначальной TT структуры: мы перенумеровываем элементы в виде матрицы

$$u^{(k,k+1)}_{\alpha_{k-1},\alpha_{k+1}}(\overline{i_k i_{k+1}}) = u^{\langle k,k+1\rangle}(\overline{\alpha_{k-1} i_k}, \overline{i_{k+1}\alpha_{k+1}}),$$

и вычисляем приближенное скелетное разложение с помощью, например, сингулярного разложения (2.2) или крестового метода (2.3):

$$u^{\langle k,k+1\rangle} \approx \tilde{u}^{\langle k,k+1\rangle} = u^{|k\rangle} u^{\langle k+1|}, \quad u^{|k\rangle} \in \mathbb{C}^{r_{k-1}n_k \times r'_k}. \tag{4.12}$$

Возмущение, внесенное в $u^{\langle k,k+1\rangle}$ на этом шаге разложения влияет на весь вектор x, и ортогональность фрейм-матрицы $X_{\neq k,k+1}$ гарантирует одинаковость Фробениусовых норм локального (в $u^{\langle k,k+1\rangle}$) и глобального (в x) возмущений. Кроме того,

Алгоритм 9 Двухблочный DMRG для $Ax = b$ (прямой полу–проход) [116]

Ввод: Начальное приближение $t = \tau(\{t^{(k)}\})$, точность ε или ранг r.
Вывод: Новое приближение $x = \tau(\{x^{(k)}\})$.

```
Скопировать x^(k) = t^(k), k = 1, ..., d.
for k = d, ..., 2 do {Ортогонализация справа налево}
    Найти LQ разложение x^⟨k| = LQ, QQ* = I.
    Заменить x^⟨k| := Q, и x^|k-1⟩ := x^|k-1⟩ L.
end for
for k = 1, ..., d-1 do {Оптимизация TT блоков}
    Построить A_{k,k+1} и b_{k,k+1} как в (4.10).
    Решить A_{k,k+1} u^(k,k+1) = b_{k,k+1}. {Взять x^(k,k+1) как начальное приближение}
    Разделить u^(k,k+1) на u^(k) и u^(k+1) посредством (4.12) так, что (u^|k⟩)* u^|k⟩ = I.
    Выбрать такой r'_k, что удовлетворяется r'_k ⩽ r и/или ||u^⟨k,k+1⟩ - ũ^⟨k,k+1⟩|| ⩽ ε||u^⟨k,k+1⟩||.
    Заменить x^(k) := u^(k) и x^(k+1) := u^(k+1).
end for
return x = τ(x^(1), ..., x^(d)).
```

ортогональность фрейм-матриц можно легко поддерживать в процессе итерации, так как сингулярное разложение возвращает ортогональные сингулярные вектора, которые достаточно взять в качестве $u^{|k\rangle}$. Процедура представлена в Алг. 9.

Как и в процедуре TT округления (раздел 2.1.5), есть три стратегии выбора нового ранга r'_k в разложении (4.12):

- *ограничение ранга* $r'_k \leqslant r$ (равенство достигается с вероятностью 1, если $r \leqslant \min\{r_{k-1}n_k, n_{k+1}r_{k+1}\}$),
- требование *относительной точности* ε, и минимального ранга r'_k, обеспечивающего (4.12) с условием $\|u^{\langle k,k+1\rangle} - \tilde{u}^{\langle k,k+1\rangle}\| \leqslant \varepsilon\|u^{\langle k,k+1\rangle}\|$, или
- использование ε-стратегии где возможно, с ограничением $r'_k = r$ если предыдущее условие предполагает большее значение.

Стратегия точности может быть также классифицирована по отношению к конкретной *норме*, используемой в условии фильтрации. Например, если $\|x-\tilde{x}\| = \|u^{\langle k,k+1\rangle} - \tilde{u}^{\langle k,k+1\rangle}\| \leqslant \varepsilon\|u^{\langle k,k+1\rangle}\|$ удовлетворяется в норме Фробениуса, невязка оценивается следующим образом:

$$\frac{\|A\tilde{x} - b\|}{\|b\|} \leqslant \frac{\|A\|\|\tilde{x} - x_\star\|}{\|Ax_\star\|} \leqslant \mathrm{cond}(A)\frac{\|\tilde{x} - x_\star\|}{\|x_\star\|} \leqslant \mathrm{cond}(A)\varepsilon.$$

Чтобы более точно контролировать невязку на выходе, в работе [55] была предложена следующая схема: мы фильтруем сингулярные вектора в разложении (4.12) не по критерию Фробениуса, а так, чтобы удовлетворить

$$\|A_{k,k+1}\tilde{u}^{(k,k+1)} - b_{k,k+1}\| \leqslant \varepsilon\|b_{k,k+1}\|.$$

Конечно, эта эвристика не гарантирует, что глобальная невязка обязательно меньше ε, но, как правило, она близка к затребованному уровню, так как матрица $A_{k,k+1}$ обычно хорошо оценивает спектральный интервал A. Аналогичным образом, можно использовать критерии вида $\|u^{(k,k+1)} - \tilde{u}^{(k,k+1)}\|_{A_{k,k+1}} \leqslant \varepsilon \|u^{(k,k+1)}\|_{A_{k,k+1}}$.

Когда разложение вычислено, TT ядра $x^{(k)}$ и $x^{(k+1)}$ заменяются на $u^{(k)}$ и $u^{(k+1)}$, и TT ранг r_k заменяется на r'_k. Тензорная структура меняется на каждом шаге, и последующая оптимизация проводится по новому *тензорному многообразию*. Как правило, это ускоряет сходимость, но делает процесс более трудным для анализа. Кроме того, даже такой двухблочный DMRG может сходиться не к точному решению $x_\star$, а к локальному минимуму $\mathtt{J}_{A,b}(\tau(\{x^{(k)}\}))$, особенно если метод стартует с начального приближения низкого ранга.

Замечание 4.2.3. И Алгоритм 8 и Алг. 9 могут быть настроены для решения задачи аппроксимации

$$\min \mathtt{E}_{x_\star}(x) = \|x - x_\star\|^2 = \mathtt{J}_{I,x_\star}(x).$$

Если матрицы интерфейсов ортогональны, единичная глобальная матрица порождает единичные локальные матрицы в (4.7) и (4.10): например,

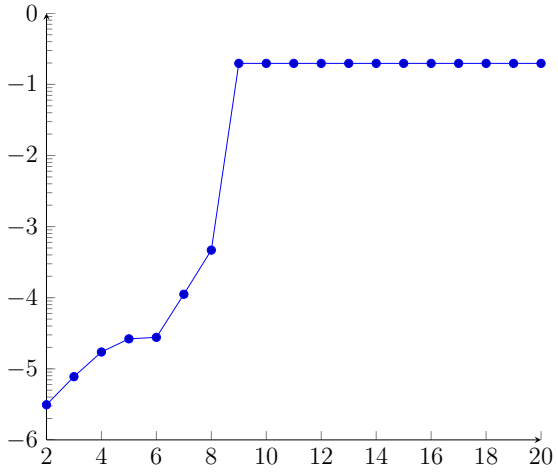
$$A_k = X^*_{\neq k} A X_{\neq k} = X^*_{\neq k} I X_{\neq k} = I.$$

Таким образом, в локальных задачах мы просто присваиваем $u^{(k)} = b_k$ или $u^{(k,k+1)} = b_{k,k+1}$, без изменения остальных шагов алгоритмов. Этот подход особенно полезен для быстрого приближения матричных и адамаровых произведений в TT формате: искомый вектор может иметь вид $x_\star = Ay$ с TT рангами $r(x_\star) = r(A)r(y)$. Его аппроксимация непосредственно SVD алгоритмом вычислительно затратна, тогда как проекция $X^*_{\neq k} A y$ имеет гораздо меньшую сложность, если приближение x действительно обладает небольшими рангами. Именно такая процедура используется для приближенных произведений матрицы на вектор в явной схеме интегрирования уравнения Власова, см. секцию 3.2. Мы вернемся к ней подробнее, когда введем и проанализируем более эффективный метод переменных направлений в следующем разделе.

Замечание 4.2.4. И Алгоритм 8 и Алг. 9 могут применяться для решения несимметричных систем. В этом случае мы рассматриваем несимметричные локальные задачи (4.7) и (4.10) как Галеркинские проекции общего вида. Хотя такой метод не является вариационным (т.е. задача не связана ни с какой монотонной оптимизацией), он работает довольно хорошо в некоторых не очень сложных случаях, см., например, [55, 53, 128, 49].

Интересно, что Галеркинские базисы переменных направлений, построенные по текущему аппроксиманту, могут использоваться не только непосредственно для решения линейной системы, но и для расчета оптимизированных сдвиговых параметров в традиционных методах Alternating Direction Iteration, ADI [25]. Это может помочь взглянуть на схемы переменных направлений с другой точки зрения (например, теорий Чебышевских или Крыловских методов), но в настоящее время неясно, возможен ли строгий анализ подобного рода.

Рис. 4.4: Невязка в методе DMRG в зависимости от размерности



4.3 Адаптивные методы переменных направлений для решения линейных систем высоких размерностей

4.3.1 Понятие расширения формата

Из-за *локального* характера оптимизации, методы DMRG (даже двухблочные) могут терять важную часть информации о направлении к точному решению, и стагнировать в каком-то локальном минимуме, далеко от желаемого порога. Обычно это явление становится более заметным с увеличением размерности. Например, основное кинетическое уравнение для каскада реакций (см. разделы 3.3.1, и 5.1 с более подробным описанием и экспериментами) естественным образом обобщается на произвольную размерность, поэтому рассмотрим поведение двухблочного DMRG с увеличением размерности, см. рис. 4.4. Мы видим, что этот алгоритм хорошо работает до размерности 6, но дальше качество решения резко ухудшается, вплоть до совершенно неверного на размерностях больше 10. Можно ли решить эту проблему и разработать надежные и эффективные схемы для линейных систем и аппроксимации в высоких размерностях? В этом разделе мы дадим положительный ответ.

Еще в работе [55] было отмечено, что на удивление полезным приемом является добавление время от времени в TT формат решения тензора малых рангов со случайно заполненными блоками. Предыдущий уровень точности восстанавливается в следующем же микрошаге (4.10), но помимо этого, в последующих итерациях сходимость становится заметно быстрее. В работе [196], посвященной быстрому методу приближении матрично-векторного произведения на основе DMRG (см. замечание 4.2.3), концепция *случайного расширения* получила дальнейшее развитие: после того, как вычислены новые TT блоки $u^{(k)}, u^{(k+1)}$, строка 11 в алгоритме

9 изменяется следующим образом:

$$x^{|k\rangle} = \begin{bmatrix} u^{|k\rangle} & s^{|k\rangle} \end{bmatrix}, \quad x^{\langle k+1|} = \begin{bmatrix} u^{\langle k+1|} \\ 0 \end{bmatrix}, \tag{4.13}$$

где $s^{|k\rangle} \in \mathbb{C}^{r_{k-1}n_k \times \rho_k}$ это набор векторов со случайными элементами, ортогонализованный к $u^{|k\rangle}$, а нулевой блок в $x^{\langle k+1|}$ имеет размеры $\rho_k \times n_{k+1}r_{k+1}$. *Расширение* (4.13) можно рассматривать как добавление нуля в TT формате: $x^{(k,k+1)} = u^{(k,k+1)} + s^{(k,k+1)}$ меняет TT представление, но на самом деле коррекция всего тензора отсутствует, $s^{(k,k+1)} = \tau(s^{|k\rangle}, 0) = 0$. Поэтому в такой схеме сохраняются $x = \tau(x^{(1)}, \ldots, x^{(d)})$ и $\mathtt{J}_{A,b}(x)$, но не матрица интерфейса $x^{(<k+1)}$. Легко заметить, что

$$x^{(<k+1)} = \begin{bmatrix} u^{(<k+1)} & \tau(u^{(<k)}, s^{(k)}) \end{bmatrix},$$

и следовательно фрейм-матрица расширяется таким образом:

$$X_{\neq k+1,k+2} := \begin{bmatrix} X_{\neq k+1,k+2} & \tau(u^{(<k)}, s^{(k)}) \otimes I \otimes I \otimes (t^{(>k+2)})^\top \end{bmatrix}.$$

Второй член привносит новые базисные компоненты для оптимизации (по крайней мере внутри k-го блока), и может ускорить сходимость.

Однако для сложных задач даже этот прием не предотвращает стагнаций в локальных минимумах. Иногда дополнительные эвристики (например, искусственное уменьшение порога точности [52, секция 5]) улучшают ситуацию в некоторой степени, но все еще не достаточно надежны.

Другая трудность двухблочного DMRG состоит в по меньшей мере кубической сложности относительно модовых размеров n, вызванной шагом сингулярного разложения (4.12), в то время как одноблочный DMRG обладает асимптотической сложностью $\mathcal{O}(n^2)$, или даже $\mathcal{O}(n)$, если используется разреженность TT блоков матрицы $A^{(k)}$. Таким образом, если модовые размеры велики (не в любой задаче эффективно введение QTT формата), одноблочная итерация может быть значительно быстрее, чем двухблочная.

Заметим, что расширение (4.13) дает ранговую адаптивность даже в одноблочной схеме. Это решает первую проблему одноблочного подхода. Вторая проблема (стагнация в локальных минимумах) может быть теперь переформулирована следующим образом: как выбрать блок расширения $s^{(k)}$ так, чтобы и вычислительная эффективность, и *глобальная* сходимость в терминах элементов полных тензоров являлись удовлетворительными.

4.3.2 Метод неточного градиентного спуска и его анализ

Мы видели, что схемы переменных направлений связаны с Галеркинскими проекциями (4.7), (4.10) исходной системы на элементы каждого TT блока. В этом смысле этот подход аналогичен классическим проекционным методам, например, алгоритму GMRES 7 в TT формате; разница в том, что фрейм-матрицы текущего решения не приближают Крыловские базисы, и, следовательно, не гарантируют сходимость метода. Наша главная идея состоит в сочетании классических схем и переменных направлений, путем обогащения фрейм-матриц информацией Крыловского типа.

Так как мы хотели бы избежать использования старших Крыловских векторов с большими TT рангами, рассмотрим простейший метод, который обеспечивает все же геометрическую сходимость – алгоритм *наискорейшего спуска* (градиентного спуска, steepest descent, SD). Традиционный алгоритм градиентного спуска использует только первый Крыловский вектор, т.е. невязку, для коррекции аппроксиманта в сторону точного решения. Так как все методы в оставшейся части этой главы будут *одношаговыми*, удобно несколько изменить классические обозначения, используемые для итерационных алгоритмов.

Итерационный метод начинается с *начального приближения*, обозначаемого обычно x_0, и генерирует последовательность приближений $x_1, x_2, \ldots$, таких, что $x_i \to x_\star = A^{-1}b$. Метод называется *геометрически* сходящимся, если существует равномерная оценка $\Omega < 1$, называемая *скоростью сходимости*, т.е. $\|x_\star - x_i\| \leqslant \|x_\star - x_0\|\Omega^i$, где Ω не зависит от x_0.

Одношаговые методы (ALS и DMRG с точки зрения полных циклов по всем TT ядрам также принадлежат к этому семейству) используют одни и те же формулы на всех итерациях, не зависящие от i или x_i. Таким образом, мы будем оценивать Ω по одной итерации, т.е. $\|x_\star - x_1\|/\|x_\star - x_0\| \leqslant \Omega$ для любого x_0. Нижний индекс номера итерации i здесь становится ненужным, и мы принимаем упрощенные обозначения, характерные при анализе одношаговых методов (см., например, [187]): мы подразумеваем, что

- $t \equiv x_0$ это начальное приближение,
- $x \equiv x_1$ это решение на новом шаге,
- $z = b - At$ невязка,
- $c = x_\star - t$ начальная ошибка, и
- $f = x_\star - x$ ошибка на новом шаге.

Нижний индекс мы зарезервируем для микрошагов $k = 1, \ldots, d$ оптимизаций по TT блокам в схеме переменных направлений.

Итак, начнем с краткого описания классического шага наискорейшего спуска. Он минимизирует функцию энергии в направлении градиента, т.е.

$$\begin{aligned} z &= -\nabla \mathsf{J}_{A,b}(t) = b - At, \\ h &= \arg\min_{h'} \mathsf{J}_{A,b}(t + zh') = \frac{(z,z)}{(z,Az)}. \end{aligned} \tag{4.14}$$

Другими словами, новое решение $x = t + zh$ удовлетворяет *условию Галеркина* на векторе невязки, $(z, b - Ax) = 0$. Новая ошибка выражается так:

$$f = c - zh = c - \frac{z(z,z)}{(z,Az)} = (I - \mathcal{P}_{A,z})c, \qquad \text{где} \quad \mathcal{P}_{A,z} = \frac{zz^*A}{z^*Az}$$

обозначает A–ортогональный проектор на $\mathrm{span}(z)$, и может быть оценена следующим образом:

$$\frac{\mathsf{J}_{A,b}(x)}{\mathsf{J}_{A,b}(t)} = \frac{\|f\|_A^2}{\|c\|_A^2} = \frac{(c, (I - \mathcal{P}_{A,z})^* A (I - \mathcal{P}_{A,z})c)}{(c, Ac)} = 1 - \frac{(c, \mathcal{P}_{A,z}c)_A}{(c,c)_A} = \omega_{z,z}^2. \tag{4.15}$$

Ортогональность проекции гарантирует монотонность,

$$\mathtt{J}_{A,b}(x) = \|f\|_A^2 \leqslant \|c\|_A^2 = \mathtt{J}_{A,b}(t).$$

Для будущего удобства мы вводим понятие *точной скорости сходимости* $\omega_{z,z}$, равной отношению ошибок на предыдущей и следующей итерациях. Конечно, она зависит *апостериори* от текущего начального приближения t. Тем не менее, она равномерно ограничена сверху, в соответствии с *неравенством Канторовича* [4]:

$$\omega_{z,z} = \sqrt{1 - \frac{(z,z)}{(z,Az)}\frac{(z,z)}{(z,A^{-1}z)}} \leqslant \frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)} = \frac{\operatorname{cond}(A) - 1}{\operatorname{cond}(A) + 1} = \Omega < 1, \quad (4.16)$$

где $\lambda_{\min}$ и $\lambda_{\max}$ являются соответственно минимальным и максимальным собственными числами A.

Замечание 4.3.1. Для $z = u_{\min}(A) + u_{\max}(A)$, где $u_{\min}(A)$ и $u_{\max}(A)$ являются нормированными собственными векторами A, отвечающими $\lambda_{\min}$ и $\lambda_{\max}$, неравенство (4.25) превращается в равенство, т.е. оценка $\omega_{z,z} \leqslant \Omega$ строгая.

При обсуждении локальной задачи (4.7) было сказано, что оценка на обусловленность важна для получения решения с достаточной точностью, $\operatorname{cond}(A) \leqslant C < \infty$. Поэтому мы будем всегда предполагать существование *априорной* оценки $\omega_{z,z} \leqslant \Omega < 1$.

Теперь рассмотрим *неточный* шаг градиентного спуска:

$$x = t + \tilde{z}h, \qquad h = \arg\min_{h'} \mathtt{J}_{A,b}(t + \tilde{z}h') = \frac{(\tilde{z}, z)}{(\tilde{z}, A\tilde{z})}, \quad (4.17)$$

который использует приближенную невязку $\tilde{z} \approx z$, и дает возмущенную новую ошибку $\tilde{f} = x_\star - x = (I - \mathcal{P}_{A,\tilde{z}})c$. Принимать во внимание неточность вычисления невязки важно, так как в численно эффективных тензорных методах она не являются пренебрежимо малой.

В отличие от утверждения о сходимости неточного GMRES 4.1.3, где требовались оценки на ошибку $\|z - \tilde{z}\|$, схема наискорейшего спуска позволяет накладывать очень слабые ограничения, основанные на *угле* $\angle(z;\tilde{z})$ между z и $\tilde{z}$.

Утверждение 4.3.2 (Сходимость неточного SD [182])**.** Пусть дана SPD линейная система $Ax = b$ и начальное приближение t, рассмотрим невязку $z = b - At$ и вектор $\tilde{z}$, такой, что $\angle(z;\tilde{z}) \leqslant \theta < \pi/2$. Тогда неточный SD метод (4.17) сходится, и скорость сходимости оценивается следующим образом:

$$\omega_{z,\tilde{z}} = \frac{\|\tilde{f}\|_A}{\|c\|_A} \leqslant \frac{\tilde{\kappa} - 1}{\tilde{\kappa} + 1} = \tilde{\Omega} < 1, \qquad \tilde{\kappa} = \operatorname{cond}(A)\frac{1 + \sin\theta}{1 - \sin\theta}. \quad (4.18)$$

Доказательство. Непосредственно из $\tilde{f} = (I - \mathcal{P}_{A,\tilde{z}})c$ получаем

$$\omega_{z,\tilde{z}}^2 = \frac{\mathtt{J}_{A,b}(x)}{\mathtt{J}_{A,b}(t)} = \frac{\|\tilde{f}\|_A^2}{\|c\|_A^2} = 1 - \frac{|(\tilde{z}, z)|^2}{(\tilde{z}, A\tilde{z})(z, A^{-1}z)}. \quad (4.19)$$

Чтобы ограничить $\omega_{z,\tilde{z}}$ сверху подобно (4.16), мы используем обобщение неравенства Канторовича из [21, Следствие IV]:

$$\frac{(\tilde{z}, A\tilde{z})(z, A^{-1}z)}{\|\tilde{z}\|^2\|z\|^2} \leqslant \frac{((\kappa+1)+(\kappa-1)\sin\theta)^2}{4\kappa}, \quad \kappa = \mathrm{cond}(A). \tag{4.20}$$

Вместе с $\cos\angle(\tilde{z}; z) \geqslant \cos\theta$ и тождеством $\frac{1+\sin\theta}{\cos\theta} = \frac{\cos\theta}{1-\sin\theta} = \sqrt{\frac{1+\sin\theta}{1-\sin\theta}}$ это дает

$$\frac{|(\tilde{z}, z)|^2}{(\tilde{z}, A\tilde{z})(z, A^{-1}z)} \geqslant \frac{4\kappa\cos^2\theta}{(\kappa(1+\sin\theta)+(1-\sin\theta))^2} = \left(\frac{2}{\tilde{\kappa}^{1/2}+\tilde{\kappa}^{-1/2}}\right)^2.$$

Доказательство завершается путем подстановки этого неравенства в (4.19). □

В случае точного вычисления $\tilde{z} = z$, неравенство (4.20) превращается в неравенство Канторовича (4.16). Другой сильной особенностью неточного градиентного спуска является то, что он является сходящимся методом при использовании *любого* шага, не являющегося строго перпендикулярным невязке. В тензорных алгоритмах, $\tilde{z}$ строится обычно ортогональной проекцией z на малоранговый формат (например в TT округлении на базе сингулярного разложения, или ALS, настроенного для задачи аппроксимации, как в замечании 4.2.3):

$$z = \tilde{z} + \delta z, \qquad (\tilde{z}, \delta z) = 0, \qquad \|\delta z\| \leqslant \varepsilon\|z\|. \tag{4.21}$$

Это дает $\sin\angle(\tilde{z}; z) = \|\delta z\|/\|z\| \leqslant \varepsilon$, и следовательно $\angle(\tilde{z}; z) < \pi/2$ при $\varepsilon < 1$. Интересно, что даже очень грубые пороги точности могут обеспечивать разумную оценку для $\tilde{\Omega}$. Действительно, дополнения Ω и $\tilde{\Omega}$ до единицы можно сравнить так:

$$1 \leqslant \frac{1-\Omega}{1-\tilde{\Omega}} \leqslant \frac{\frac{1+\varepsilon}{1-\varepsilon}\kappa+1}{\kappa+1} = \frac{1+\varepsilon\frac{\kappa-1}{\kappa+1}}{1-\varepsilon} \leqslant 3, \quad \text{для} \quad \varepsilon \leqslant \frac{1}{2}.$$

Таким образом, в дополнение к известному Ω, можно считать, что ε выбирается *априори* или контролируются в процессе расчетов таким образом, что $\tilde{\Omega}$ в (4.18) является приемлемой *априорной* оценкой для скорости сходимости неточного алгоритма SD.

Нам также понадобится обобщение *расширенного* наискорейшего спуска, так как интерфейс- и фрейм-матрицы содержат наборы векторов, а не только одну невязку. Пусть дана полноранговая матрица Z подходящего размера, тогда решение на следующем шаге вычисляется как $x = t + Zv$, где

$$v = \arg\min_{v'} \mathrm{J}_{A,b}(t + Zv') = (Z^*AZ)^{-1}Z^*z. \tag{4.22}$$

В сравнении с обычным (4.14), или неточным (4.17) шагами градиентного спуска, оптимизация (4.22) выполняется по векторам в *более широком* многообразии $t + \mathrm{span}(Z)$. Новая ошибка также может быть записана в виде ортогональной проекции: $f = (I - \mathcal{P}_{A,Z})c$, где A–ортогональный проектор $\mathcal{P}_{A,Z}$ определяется для полноранговой матрицы Z таким образом:

$$\mathcal{P}_{A,Z} = Z(Z^*AZ)^{-1}Z^*A. \tag{4.23}$$

Аналогично пишется скорость сходимости расширенного SD метода:

$$\omega^2_{z,Z} = \frac{\mathtt{J}_{A,b}(x)}{\mathtt{J}_{A,b}(t)} = \frac{\|f\|^2_A}{\|c\|^2_A} = 1 - \frac{(c, \mathcal{P}_{A,Z}c)_A}{(c,c)_A}. \tag{4.24}$$

В алгоритмах переменных направлений, Z будет играть роль матрицы интерфейса или фрейм-матрицы. Из линейности ТТ формата (4.5) следует $\tilde{z} = Z_{\neq k} z^{(k)}$, то есть $\tilde{z} \in \mathrm{span}(Z)$, если $Z = Z_{\neq k}$. Таким образом, полезно связать скорости сходимости расширенного и одномерного методов градиентного спуска.

Лемма 4.3.3. Если $\tilde{z} \in \mathrm{span}(Z)$, тогда $\omega_{z,Z} \leqslant \omega_{z,\tilde{z}}$, т.е. расширенный SD метод (4.22) сходится не хуже неточного SD метода (4.17).

Доказательство. Рассматривая дополнения $\omega^2_{z,\tilde{z}}$ и $\omega^2_{z,Z}$ до единицы, имеем

$$\|\mathcal{P}_{A,\tilde{z}}c\|^2_A = \|\mathcal{P}_{A,\tilde{z}}\mathcal{P}_{A,Z}c\|^2_A \leqslant \|\mathcal{P}_{A,Z}c\|^2_A.$$

□

Следствие 4.3.4. Если $\angle(Z;z) = \min_{\tilde{z}\in\mathrm{span}(Z)} \angle(\tilde{z};z) \leqslant \theta < \pi/2$, тогда

$$\omega_{z,Z} \leqslant \frac{\tilde{\kappa}-1}{\tilde{\kappa}+1} = \tilde{\Omega} < 1, \qquad \tilde{\kappa} = \mathrm{cond}(A)\frac{1+\sin\theta}{1-\sin\theta}. \tag{4.25}$$

Доказательство. Применим лемму 4.3.3 для $\tilde{z} \in \mathrm{span}(Z)$, так что $\angle(\tilde{z};z) = \angle(Z;z) \leqslant \theta$, и оценим $\omega_{z,\tilde{z}}$ посредством (4.18). □

Замечание 4.3.5. В практических расчетах можно ожидать, что расширенный SD метод сходится заметно быстрее, чем одномерный неточный SD метод, т.е. $\omega_{z,Z} \ll \omega_{z,\tilde{z}}$ если $\dim(Z) \gg 1$. В общем случае, однако, неравенство в лемме 4.3.3 строгое. Например, $\omega_{z,Z} = \omega_{z,\tilde{z}}$ в случае $Z = \begin{bmatrix}\tilde{z} & s\end{bmatrix}$ с таким s, что $(\tilde{z}, s)_A = 0$ и $(c, s)_A = 0$.

Доказательство. Для $(\tilde{z}, s)_A = 0$ имеем $\mathcal{P}_{A,Z} = \mathcal{P}_{A,\tilde{z}} + \mathcal{P}_{A,s}$, и первое условие $(\tilde{z}, s)_A = 0$ дает $\|\mathcal{P}_{A,Z}c\|^2_A = \|\mathcal{P}_{A,\tilde{z}}c\|^2_A + \|\mathcal{P}_{A,s}c\|^2_A$. Условие $(c, s)_A = 0$ обнуляет второй член и доказывает равенство проекций. □

4.3.3 AMEn: комбинация градиентного спуска и переменных направлений

Все версии метода наискорейшего спуска, представленные выше, являются *вариационными* для функции энергии: на каждом шаги, они ищут минимизатор энергии с ограничениями, см. (4.22). По аналогии с алгоритмом *минимальных невязок* (Minimal Residual, MR) (см., например, [213]), мы можем назвать метод наискорейшего спуска также методом *минимальных энергий* (Minimal Energy, MEn).

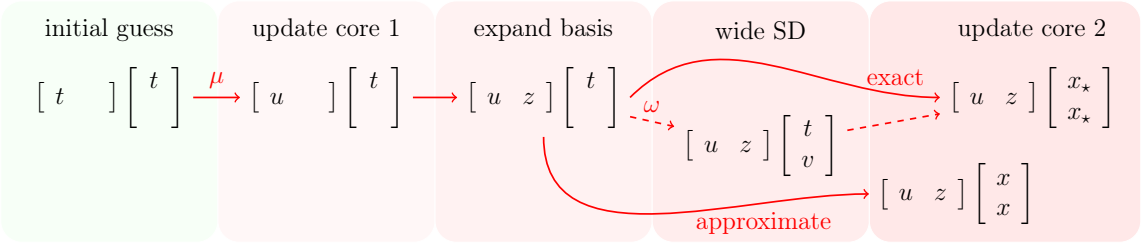
В этом разделе мы предложим алгоритм для решения линейной системы в ТТ формате в стиле DMRG, дополненный шагом расширения (4.13) на подпространство вида расширенного наискорейшего спуска. Это и послужило причиной названия AMEn: *Alternating Minimal Energy*.

Алгоритм 10 AMEn для SPD линейной системы $Ax = b$ (рекуррентная версия)

Ввод: Начальное приближение $t = \tau(\{t^{(k)}\})$, точность ε или ранги $\rho_1, \dots, \rho_{d-1}$
Вывод: Новое приближение $x = \tau(\{x^{(k)}\})$ с рангами $r'_k \leqslant r_k + \rho_k$, и $\mathtt{J}_{A,b}(x) < \mathtt{J}_{A,b}(t)$

1: Построить $A_1 = T^*_{\neq 1} A T_{\neq 1}$, $b_1 = T^*_{\neq 1} b$, и решить $A_1 u^{(1)} = b_1$.
2: Пусть $u = \tau(u^{(1)}, t^{(2)}, \dots, t^{(d)})$ и $z = b - Au$.
3: Приблизить $z \approx \tilde{z} = \tau(\{z^{(k)}\})$, так, что $\|\tilde{z} - z\| \leqslant \varepsilon \|z\|$ или $r_k(\tilde{z}) < \rho_k$
4: Объединить ТТ блоки невязки и решения $x^{(1)} := \begin{bmatrix} u^{(1)} & z^{(1)} \end{bmatrix}$, $t^{(2)} := \begin{bmatrix} t^{(2)} \\ 0 \end{bmatrix}$
5: Рассмотрим $(d-1)$–мерную систему $A_{\geqslant 2} x^{(\geqslant 2)} = b_{\geqslant 2}$ в соответствии с (4.26)
6: **if** $d = 2$ **then**
7: Построить и решить прямым методом $A_{\geqslant 2} x^{(\geqslant 2)} = b_{\geqslant 2}$
8: **else**
9: Решить $A_{\geqslant 2} x^{(\geqslant 2)} = b_{\geqslant 2}$ методом AMEn, получить $x_{\geqslant 2} = \tau(x^{(2)}, \dots, x^{(d)})$
10: **end if**
11: **return** $x = \tau(x^{(1)}, x^{(2)}, \dots, x^{(d)})$

Рис. 4.5: Иллюстрация алгоритма AMEn в двух переменных



Описание и анализ в этом разделе следуют [56, 57]. Хотя можно разработать алгоритм с *глобальным* расширением формата, который меняет все ТТ блоки за раз (так называемый ALS$(t+z)$ метод), мы будем рассматривать AMEn алгоритм с *локальным* расширением (4.13), который на практике оказывается быстрее и точнее.

ТТ формат был введен в качестве рекуррентного обобщения скелетного разложения, и таким же образом мы начинаем представление алгоритма AMEn с двумерного случая. Отметим, что в расширении (4.13) можно эквивалентно писать $x^{(1)}$ вместо $x^{|1\rangle}$, так как левый ранговый индекс отсутствует.

Одноблочный DMRG Алг. 8 с расширением (4.13) после строки 9 можно записать следующим образом (мы опускаем шаги ортогонализации, и считаем, что все фрейм-матрицы $X_{\neq p}$ являются ортогональным).

1. Скопировать $x^{(k)} = t^{(k)}$, $k = 1, \dots, d$.
2. Построить $A_1 = X^*_{\neq 1} A X_{\neq 1} = T^*_{\neq 1} A T_{\neq 1}$, $b_1 = X^*_{\neq 1} b = T^*_{\neq 1} b$.
3. Решить $A_1 u^{(1)} = b_1$.

4. Заменить $x^{(1)} := \begin{bmatrix} u^{(1)} & s^{(1)} \end{bmatrix}$, $x^{(2)} := \begin{bmatrix} t^{(2)} \\ 0 \end{bmatrix}$.

5. Решить (приближенно) $A_{\geqslant 2} x^{(\geqslant 2)} = b_{\geqslant 2}$.

В соответствии с анализом в предыдущем разделе, разумно выбрать расширение $s^{(1)}$ связанное с невязкой. Перед применением расширения, решение пишется в виде $u = \tau(u^{(1)}, t^{(\geqslant 2)})$. Анализ градиентного спуска не содержит локальной ALS оптимизации $t^{(1)} \to u^{(1)}$, поэтому нам придется несколько адаптировать обозначения, и ввести величины, начальные для *расширенного шага SD*, но не для всего алгоритма:

- текущее решение $u = \tau(u^{(1)}, t^{(\geqslant 2)})$,
- невязка $z = b - Au$, и
- ошибка $c = x_\star - u$.

Теперь, если мы вычислили приближенную невязку, $\tilde{z} = \tau(z^{(1)}, z^{(\geqslant 2)}) \approx z$, естественно взять ее интерфейс-матрицу как базис для расширенного SD, $Z = Z_1 = z^{|1\rangle} \otimes I_{n_2 \cdots n_d}$. Для этого, мы добавляем в TT формат решения первое TT ядро невязки, $s^{(1)} = z^{(1)}$. В этом случае, $\tilde{z} \in \operatorname{span}(Z)$, и лемма 4.3.3 будет иметь место, если мы покажем, что ошибка действительно проецируется на подпространство, содержащее Z.

Этот анализ может быть проведен в соответствии со схемой на рис. 4.5. *Редуцированная* система $A_{\geqslant 2} x^{(\geqslant 2)} = b_{\geqslant 2}$ собирается так:

$$\begin{aligned} A_{\geqslant 2} = X^*_{\neq\{2,\ldots,d\}} A X_{\neq\{2,\ldots,d\}}, \quad b_{\geqslant 2} = X^*_{\neq\{2,\ldots,d\}} b, \quad \text{где} \\ X_{\neq\{2,\ldots,d\}} \equiv X_1 = x^{|1\rangle} \otimes I_{n_2 \cdots n_d} \in \mathbb{C}^{(n_1 \cdots n_d) \times (r_1 n_2 \cdots n_d)}. \end{aligned} \tag{4.26}$$

По построению $x^{|1\rangle}$, выполняется $Z_1 \in \operatorname{span}(X_1)$. Новое решение пишется (при условии, что редуцированная система решается точно) следующим образом:

$$x = \tau(x^{(1)}, x^{(\geqslant 2)}) = X_1 A^{-1}_{\geqslant 2} b_{\geqslant 2} = \mathcal{P}_{A,X_1} x_\star,$$

и новая ошибка выражается как проекция точного решения: $f = x_\star - x = (I - \mathcal{P}_{A,X_1}) x_\star$. Однако, заметим что $u = U_1 t^{(\geqslant 2)}$, где $U_1 = u^{|1\rangle} \otimes I_{n_2 \cdots n_d} \in \operatorname{span}(X_1)$. В частности, $\mathcal{P}_{A,X_1} u = u$, так что мы можем записать новую ошибку как проекцию предыдущей:

$$f = (I - \mathcal{P}_{A,X_1}) x_\star - (I - \mathcal{P}_{A,X_1}) u = (I - \mathcal{P}_{A,X_1}) c.$$

Поскольку $\tilde{z} \in \operatorname{span}(Z_1) \subset \operatorname{span}(X_1)$, получаем

$$\frac{\mathtt{J}_{A,b}(x)}{\mathtt{J}_{A,b}(u)} = \frac{\|f\|^2_A}{\|c\|^2_A} = 1 - \frac{(c, \mathcal{P}_{A,X_1} c)_A}{(c,c)_A} = \omega^2_{z,X_1}, \tag{4.27}$$

с *априорными* оценками

$$\omega_{z,X_1} \leqslant \omega_{z,Z_1} \leqslant \omega_{z,\tilde{z}} \leqslant \tilde{\Omega} < 1$$

для $\varepsilon < 1$ (сравним с (4.18)).

Для случая многих размерностей, достаточно повторить предыдущие соображения по индукции. В самом деле, во всех шагах кроме решения системы $A_{\geqslant 2} x^{(\geqslant 2)} = b_{\geqslant 2}$ участвуют только структурированные вычисления в TT формате. Редуцированная система (4.26) также может быть собрана с помощью одноблочных операций: единичная матрица $I_{n_2 \cdots n_d}$ оставляет все TT ядра $A^{(2)}, \ldots, A^{(d)}$ и $b^{(2)}, \ldots, b^{(d)}$ без изменений, и только первые блоки $A^{(1)}$, $b^{(1)}$ нужно спроецировать на $x^{|1\rangle}$. Это дает следующее TT представление системы (4.26):

$$\begin{aligned} A_{\geqslant 2} &= \tau\left(\tau(\mathsf{A}^{<2}, A^{(2)}), A^{(3)}, \ldots, A^{(d)}\right), \quad \mathsf{A}^{<2}_{\gamma_1} = \left(x^{|1\rangle}\right)^* A^{(1)}_{\gamma_1} x^{|1\rangle}, \\ b_{\geqslant 2} &= \tau\left(\tau(\mathsf{b}^{<2}, b^{(2)}), b^{(3)}, \ldots, b^{(d)}\right), \qquad \mathsf{b}^{<2}_{\beta_1} = \left(x^{|1\rangle}\right)^* b^{(1)}_{\beta_1}, \end{aligned} \tag{4.28}$$

где $\gamma_1 = 1, \ldots, r_1(A)$, $\beta_1 = 1, \ldots, r_1(b)$. Обратим внимание, что $\mathsf{b}^{<2} \in \mathbb{C}^{r_1(x) \times r_1(b)}$, и следовательно размеры $\tau(\mathsf{b}^{<2}, b^{(2)}) \in \mathbb{C}^{r_1(x) n_2 \times r_2(b)}$ сравнительно невелики, эти матрицы можно непосредственно хранить в памяти. Аналогично, первый TT блок матрицы $A_{\geqslant 2}$ имеет эффективные размеры порядка одного TT ядра, и (4.28) является $(d-1)$-мерной системой с такими же рангами $r_2, \ldots, r_{d-1}$, как были в начальных данных. Кроме того, поскольку используется ортогональность $x^{|1\rangle}$, обусловленность задачи не ухудшается, $\mathrm{cond}(A_{\geqslant 2}) \leqslant \mathrm{cond}(A)$.

Таким образом, для редуцированной системы можно применить тот же AMEn метод, см. алгоритм 10. Это дает индуктивный вид анализа сходимости.

На первом шаге, следует также учитывать тот факт, что $A_{\geqslant 2} x^{(\geqslant 2)} = b_{\geqslant 2}$ решается неточно, поскольку $x^{(\geqslant 2)}$ вычисляется в рекуррентном запуске AMEn и отличается от точного решения $x^{(\geqslant 2)}_\star = A^{-1}_{\geqslant 2} b_{\geqslant 2}$. Мы можем преобразовать

$$x_\star - x = x_\star - X_1 x^{(\geqslant 2)}_\star + X_1 x^{(\geqslant 2)}_\star - X_1 x^{(\geqslant 2)} = (I - \mathcal{P}_{A, X_1}) x_\star + X_1 (x^{(\geqslant 2)}_\star - x^{(\geqslant 2)}),$$

где второй член A–ортогонален первому, который мы уже оценили в (4.27). Это дает

$$\frac{\|x_\star - x\|^2_A}{\|x_\star - u\|^2_A} = \omega^2_{z, X_1} + \frac{\|X_1 (x^{(\geqslant 2)}_\star - x^{(\geqslant 2)})\|^2_A}{\|x_\star - u\|^2_A}.$$

По построению, существует эквивалентность норм, $\|X_1 v\|_A = \|v\|_{A_{\geqslant 2}}$. Поэтому предыдущее уравнение переписывается следующим образом:

$$\frac{\|f\|^2_A}{\|c\|^2_A} = \omega^2_{z, X_1} + \frac{\|x^{(\geqslant 2)}_\star - x^{(\geqslant 2)}\|^2_{A_{\geqslant 2}}}{\|c\|^2_A} \cdot \frac{\|X_1 (x^{(\geqslant 2)}_\star - t^{(\geqslant 2)})\|^2_A}{\|x^{(\geqslant 2)}_\star - t^{(\geqslant 2)}\|^2_{A_{\geqslant 2}}}, \tag{4.29}$$

где $t^{(\geqslant 2)} = t^{(2, \ldots, d)}$ суть начальное приближение в редуцированной задаче. Аналогично для случая $d = 2$ выполняется $X_1 x^{(\geqslant 2)} = \mathcal{P}_{A, X_1} x_\star$. В то же время,

$$X_1 t^{(\geqslant 2)} = \tau(x^{(1)}, t^{(2)}, \ldots, t^{(d)}) = u = \mathcal{P}_{A, X_1} u,$$

поскольку $u \in \mathrm{span}(X_1)$. Подставим $X_1 (x^{(\geqslant 2)}_\star - t^{(\geqslant 2)}) = \mathcal{P}_{A, X_1}(x_\star - u) = \mathcal{P}_{A, X_1} c$ в (4.29) и сравним полученное выражение с (4.27). Это дает следующую оценку:

$$\frac{\mathsf{J}_{A,b}(x)}{\mathsf{J}_{A,b}(u)} = \omega^2_{z, X_1} + \frac{\|\mathcal{P}_{A, X_1} c\|^2_A}{\|c\|^2_A} \frac{\|x^{(\geqslant 2)}_\star - x^{(\geqslant 2)}\|^2_{A_{\geqslant 2}}}{\|x^{(\geqslant 2)}_\star - t^{(\geqslant 2)}\|^2_{A_{\geqslant 2}}} = \omega^2_{z, X_1} + (1 - \omega^2_{z, X_1}) \frac{\mathsf{J}_{A_{\geqslant 2}, b_{\geqslant 2}}(x^{(\geqslant 2)})}{\mathsf{J}_{A_{\geqslant 2}, b_{\geqslant 2}}(t^{(\geqslant 2)})}. \tag{4.30}$$

Последний член в (4.30) оценивается по индукции, так как он определяется тем же AMEn алгоритмом. Один полу-проход AMEn поэтому можно рассматривать как последовательность вложенных редуцированных задач $A_{\geqslant k} x^{(\geqslant k)} = b_{\geqslant k}$, где

$$\begin{aligned} A_{\geqslant k} &= X_{<k}^* A X_{<k} \in \mathbb{C}^{(r_{k-1} n_k \cdots n_d) \times (r_{k-1} n_k \cdots n_d)}, & b_{\geqslant k} &= X_{<k}^* b, \\ X_{<k} &= x^{|<k\rangle} \otimes I_{n_k \cdots n_d} \in \mathbb{C}^{(n_1 \cdots n_d) \times (r_{k-1} n_k \cdots n_d)}, & X_{<1} &= I_{n_1 \cdots n_d}. \end{aligned} \tag{4.31}$$

Для каждой редуцированной задачи, одношаговые величины определяются следующим образом:

- начальное приближение $t^{(\geqslant k)} = \tau(t^{(k)}, \ldots, t^{(d)})$,
- решение после ALS-оптимизации $u_k = \tau(u^{(k)}, t^{(k+1)}, \ldots, t^{(d)})$,
- решение на новом шаге $x^{(\geqslant k)} = \tau(x^{(k)}, \ldots, x^{(d)})$,
- точное решение $x_\star^{(\geqslant k)} = A_{\geqslant k}^{-1} b_{\geqslant k}$,
- начальная ошибка $c_k = x_\star^{(\geqslant k)} - u_k$,
- невязка $z_k = b_{\geqslant k} - A_{\geqslant k} u_k \approx \tilde{z}_k = \tau(z_k^{(k)}, \ldots, z_k^{(d)})$, и
- ошибка на новом шаге $f_k = x_\star^{(\geqslant k)} - x^{(\geqslant k)}$.

Расширение TT формата решения осуществляется с помощью первого TT блока редуцированной невязки,

$$x^{|k\rangle} = \begin{bmatrix} u^{|k\rangle} & z_k^{|k\rangle} \end{bmatrix}, \quad t^{\langle k+1|} = \begin{bmatrix} t^{\langle k+1|} \\ 0 \end{bmatrix}.$$

Введем следующие определения для отношений ошибок после шагов ALS и расширенного градиентного спуска:

$$\mu_k^2 = \frac{\|x_\star^{(\geqslant k)} - u_k\|_{A_{\geqslant k}}^2}{\|x_\star^{(\geqslant k)} - t^{(\geqslant k)}\|_{A_{\geqslant k}}^2} \leqslant 1, \qquad \omega_k^2 = 1 - \frac{(c_k, \mathcal{P}_{A_{\geqslant k}, X_k} c_k)_{A_{\geqslant k}}}{(c_k, c_k)_{A_{\geqslant k}}} < 1, \tag{4.32}$$

где $X_k = x^{|k\rangle} \otimes I_{n_{k+1} \cdots n_d}$, после объединения $x^{|k\rangle} = \begin{bmatrix} u^{|k\rangle} & z_k^{|k\rangle} \end{bmatrix}$ (и ортогонализации). Теперь можно оценить итоговое убывание ошибки в AMEn алгоритме.

Лемма 4.3.6. В определениях, установленных выше, скорость сходимости на одной итерации (полу-проходе) Алг. 10 оценивается следующим образом:

$$\omega_{\text{AMEn}}^2 = \frac{\|x_\star - x\|_A^2}{\|x_\star - t\|_A^2} = \sum_{k=1}^{d-1} \omega_k^2 \prod_{j=1}^{k-1} (1 - \omega_j^2) \prod_{j=1}^{k} \mu_j^2 = \phi_{(1:d)}^2. \tag{4.33}$$

Доказательство. Для $d = 2$ используем (4.27) и первое определение в (4.32), и получаем

$$\frac{\mathtt{J}_{A,b}(x)}{\mathtt{J}_{A,b}(t)} = \frac{\mathtt{J}_{A,b}(u)}{\mathtt{J}_{A,b}(t)} \frac{\mathtt{J}_{A,b}(x)}{\mathtt{J}_{A,b}(u)} = \mu_1^2 \omega_1^2,$$

что полагает начало индукции. Предположим теперь, что (4.33) выполняется для $d-1$ размерностей и докажем это по индукции для d переменных. Из (4.30) можно видеть, что

$$\frac{\mathtt{J}_{A,b}(x)}{\mathtt{J}_{A,b}(t)} = \mu_1^2 \left(\omega_1^2 + (1-\omega_1^2) \frac{\mathtt{J}_{A_{\geqslant 2},b_{\geqslant 2}}(x^{(\geqslant 2)})}{\mathtt{J}_{A_{\geqslant 2},b_{\geqslant 2}}(t^{(\geqslant 2)})} \right) = \mu_1^2 \left(\omega_1^2 + (1-\omega_1^2)\phi_{(2:d)}^2 \right).$$

Подставляя предположение индукции $\phi_{(2:d)}^2 = \sum_{k=2}^{d-1} \omega_k^2 \prod_{j=2}^{k-1}(1-\omega_j^2) \prod_{j=2}^{k} \mu_j^2$ в последнее уравнение, получаем (4.33). □

Таким же образом, как и в методе наискорейшего спуска, мы вынуждены были обозначить точные отношения ошибок за ω_k, и использовать их для установления *апостериорной* скорости сходимости (4.33). Теперь мы должны ограничить все величины равномерной *априорной* оценкой.

По определению (4.24), $\omega_k = \omega_{z_k, X_k}$, и поскольку $\tilde{z}_k \in \mathrm{span}(Z_k) \subset \mathrm{span}(X_k)$, верхняя граница (4.25) пишется так:

$$\omega_k \leqslant \frac{\tilde{\kappa}_k - 1}{\tilde{\kappa}_k + 1} = \tilde{\Omega}_k < 1, \qquad \tilde{\kappa}_k = \mathrm{cond}(A_{\geqslant k}) \frac{1+\sin\theta_k}{1-\sin\theta_k}, \tag{4.34}$$

где $\theta_k = \angle(z_k; X_k) \leqslant \angle(z_k; \tilde{z}_k)$. Если на всех шагах (Строка 3 в Алг. 10) используется критерий точности $\|z_k - \tilde{z}_k\| \leqslant \varepsilon \|z_k\|$, то оценка на $\sin\theta_k \leqslant \varepsilon$ известна *априори*. Если применяется критерий ограничения ранга, θ_k оценивается соответствующей относительной точностью $\angle(z_k; \tilde{z}_k)$. Поскольку

$$A_{\geqslant k+1} = X_{<k+1}^* A X_{<k+1} = (x^{|k\rangle} \otimes I_{n_{k+1}\cdots n_d})^* A_{\geqslant k} (x^{|k\rangle} \otimes I_{n_{k+1}\cdots n_d}), \tag{4.35}$$

то числа обусловленности $\mathrm{cond}(A_{\geqslant k})$ связаны следующим образом:

$$\mathrm{cond}(A) = \mathrm{cond}(A_{\geqslant 1}) \geqslant \ldots \geqslant \mathrm{cond}(A_{\geqslant k-1}) \geqslant \mathrm{cond}(A_{\geqslant k}) \geqslant \ldots \geqslant \mathrm{cond}(A_{\geqslant d}). \tag{4.36}$$

Подставляя $\theta_k \leqslant \theta = \max_k \angle(z_k; \tilde{z}_k)$ и $\mathrm{cond}(A_{\geqslant k}) \leqslant \mathrm{cond}(A)$ в (4.34), получаем

$$\omega_k \leqslant \tilde{\Omega}_k = \frac{\tilde{\kappa}_k - 1}{\tilde{\kappa}_k + 1} \leqslant \frac{\tilde{\kappa} - 1}{\tilde{\kappa} + 1} = \tilde{\Omega} < 1, \qquad \tilde{\kappa} = \mathrm{cond}(A) \frac{1+\sin\theta}{1-\sin\theta}, \tag{4.37}$$

что дает равномерную верхнюю границу для всех ω_k.

Теорема 4.3.7. AMEn алгоритм 10 сходится, если ошибка аппроксимации, вносимая в строке 3, удовлетворяет $\theta = \max_{k=1,\ldots,d-1} \angle(z_k; \tilde{z}_k) < \pi/2$. Скорость сходимости одной итерации (4.33) ограничена сверху так, что имеет место неравенство

$$\phi_{(1:d)}^2 \leqslant 1 - (1-\tilde{\Omega}^2)^{d-1}, \qquad \tilde{\Omega} = \frac{\tilde{\kappa}-1}{\tilde{\kappa}+1}, \qquad \tilde{\kappa} = \mathrm{cond}(A) \frac{1+\sin\theta}{1-\sin\theta}. \tag{4.38}$$

Доказательство. Заметим, что $\phi_{(1:d)}$ в (4.33) монотонна при $0 \leqslant \omega_k \leqslant \tilde{\Omega}$ и $0 \leqslant \mu_k \leqslant 1$, $k = 1, \ldots, d-1$. Подставляя верхние оценки $\mu_k = 1$ и $\omega_k = \tilde{\Omega}$ из (4.37) в (4.33), получаем доказательство (4.38). □

Перед разбором практических аспектов эффективной реализации алгоритма AMEn, сравним его с другим типом адаптивной одноблочной схемы DMRG.

4.3.4 AMEn и одноблочный DMRG с дополнительной переменной

И двублочный DMRG, и AMEn являются ранг-адаптивными алгоритмами. Тем не менее, изменение рангов осуществляется разными способами. Метод DMRG использует двумерное разделение переменных, и возможное значение ранга ограничено сверху полным размером матрицы, например $r'_k \leqslant r_{k-1} n_k$ или $r'_k \leqslant n_{k+1} r_{k+1}$. Таким образом, ранг может быть увеличен в n раз. Новые проекционные подпространства (основанные на фрейм-матрице $X_{\neq k}$) могут заметно отличаться от старых, так как сингулярное разложение распределяет поправку по всем сингулярным векторам.

В алгоритме AMEn, увеличение ранга *аддитивное*, т.е. $r'_k \leqslant r_k + r_k(\tilde{z})$ за счет объединения TT блоков решения и невязки. То же самое можно сказать и про матрицы интерфейсов: после расширения (4.13), k-й TT блок больше не изменяется до следующей итерации, поэтому и $u^{(k)}$, и $z^{(k)}$ лежат в $\mathrm{span}(x^{(\leqslant k)})$ точно.

Это соображение дает понимание различий между алгоритмами AMEn и ранг-адаптивным одноблочным методом, предложенным в сообществе квантовой физики [252], так называемым *корректированным* одноблочным DMRG. Последний алгоритм может быть описан следующим образом: после того, как получен новый текущий TT блок (см. строку 8 в Алг. 8), мы вычисляем матрицу Грама $G^{(k)} = u^{|k\rangle} \left(u^{|k\rangle}\right)^*$, и возмущаем ее некоторой поправкой, $\tilde{G}^{(k)} = G^{(k)} + aH^{(k)}$, где $a > 0$ это некоторый (эвристический) весовой коэффициент, и $H^{(k)} = p^{|k\rangle} \left(p^{|k\rangle}\right)^*$ является матрицей Грама поправки, подробнее обсуждаемой ниже. После этого, измененная матрица Грама приближается с помощью неполного собственного разложения, $\tilde{G}^{(k)} \approx x^{|k\rangle} \operatorname{diag}(\tilde{\lambda}) \left(x^{|k\rangle}\right)^*$, и соответствующие (ортогональные) собственные вектора берутся в качестве k-го TT блока решения.

Легко видеть, что, хотя $\mathrm{rank}(G^{(k)}) = r_k$, и любая фильтрация собственных значений в $G^{(k)} = U \operatorname{diag}(\lambda) U^*$ с положительным порогом даст $r'_k = r_k$, для $\tilde{G}^{(k)}$ это уже не так, и ε-фильтрация возмущенного спектра $\tilde{\lambda}$ может дать другое значение для ранга r'_k.

Для задачи малоранговой аппроксимации, собственное разложение матрицы Грама эквивалентно сингулярному разложению *объединенных* столбцов u и p:

$$\begin{bmatrix} u^{|k\rangle} & \sqrt{a} p^{|k\rangle} \end{bmatrix} \approx x^{|k\rangle} \operatorname{diag}(\sigma) V^*, \quad \sigma^2 = \tilde{\lambda}. \tag{4.39}$$

Это выглядит очень похоже на формулу расширения (4.13). Отличие в том, что (4.39) выполняется приближенно, в то время как расширение (4.13) использует точную ортогонализацию

$$\begin{bmatrix} u^{|k\rangle} & s^{|k\rangle} \end{bmatrix} \begin{bmatrix} t^{\langle k+1|} \\ 0 \end{bmatrix} = x^{|k\rangle} \cdot R \begin{bmatrix} t^{\langle k+1|} \\ 0 \end{bmatrix} = x^{|k\rangle} \begin{bmatrix} R_u t^{\langle k+1|} \\ 0 \end{bmatrix}, \quad \left(x^{|k\rangle}\right)^* x^{|k\rangle} = I.$$

Вспоминая обсуждение в начале этой секции, можно сформулировать первое различие между AMEn и корректированным одноблочным DMRG алгоритмами: AMEn не влияет на компоненты решения в процессе расширения, а DMRG *усредняет* информацию из решения и добавочных векторов расширения посредством вычисления общих сингулярных векторов. Поэтому аккуратный подбор веса a имеет решающее значение: если вес мал, коррекция будет выброшена за счет ε-фильтрации

в приближенном сингулярном разложении; если a является слишком большим, то решение будет заметно возмущено.

В качестве вектора поправки, в [252] предлагается использовать эвристический аналог Крыловского вектора. Обозначим $p = A_{\geqslant k} u_k = \tau(p^{(k)}, \ldots, p^{(d)})$, где в соответствии с (2.12), $p^{(k)} = \left[p^{(k)}_{\eta_k}(\overline{\alpha_{k-1} i_k}) \right] \in \mathbb{C}^{r_{k-1} n_k \times r_k r_k(A)}$, и $\eta_k = \overline{\alpha_k, \gamma_k}$. Для поправки в корректированном DMRG используется непосредственно $p^{|k\rangle} = p^{(k)}$, без предварительной правой ортогонализации блоков $p^{(k+1)}, \ldots, p^{(d)}$. Хотя это и оказывается достаточным в практических расчетах основных состояний спиновых цепочек, можно предложить пример такой матрицы $A_{\geqslant k}$, что $p^{(k)}$ будет давать сильно искаженную информацию о p. Более подробную дискуссию см. в [58].

Наконец, рассмотрим корректированный одноблочный DMRG как частный случай двухблочного метода. Сравним разложение (4.39) с двухблочным разделением (4.12). Левая часть (4.39) имеет размеры $r_{k-1} n_k \times (1 + r_k(A)) r_k$, и мы можем ввести дополнительную нумерующую переменную $b = 1, \ldots, 1 + r_k(A)$, так что

$$\begin{bmatrix} u^{|k\rangle} & \sqrt{a} p^{|k\rangle} \end{bmatrix} = \left[\hat{x}^{(k)}(i_k, b)_{\alpha_{k-1}, \alpha_k} \right]$$

может рассматриваться как суперблок, объединяющий две переменные, i_k и b. Представим, что мы ищем решение в следующем виде:

$$\hat{x} = [\hat{x}(i_1, \ldots, i_k, b, i_{k+1}, \ldots, i_d)] = \tau(x^{(1)}, \ldots, x^{(k-1)}, \hat{x}^{(k)}, x^{(k+1)}, \ldots, x^{(d)}). \qquad (4.40)$$

Соответствующий алгоритм можно назвать *"полутораблочным"* DMRG, или DMRG с дополнительной переменной, поскольку при больших n размер $\hat{x}^{(k)}$ больше, чем размер $x^{(k)}$, но меньше, чем размера двухблочного суперблока $x^{(k,k+1)}$.

Представление (4.40) называется *блочным* TT форматом [41], так как мы можем сказать, что $\hat{x}$ включает в себя несколько векторов (сравним с понятием *блочности* в задаче на несколько собственных значений), перечисляемых индексом b,

$$\hat{x} = [\hat{x}_b]_{b=1}^{1+r_k(A)}, \quad \text{где} \quad \hat{x}_{b+1} = \sqrt{a} p_b, \quad b = 1, \ldots, r_k(A)$$

обозначают векторы поправки, и $\hat{x}_1 = x$, собственно искомое решение.

С помощью сингулярного разложения (4.39), нумератор b может быть перенесен из TT блока $x^{(k)}$ в $x^{(k+1)}$ или обратно, то есть перемещаться в текущий вычисляемый TT блок, с изменением порядка расположения переменных b и $i_1, \ldots, i_d$. Этот метод был использован в [41] для одновременного хранения и вычисления нескольких крайних собственных векторов в едином TT формате, откуда и возникло название "блочный TT". Такая же процедура имеет место и в корректированном одноблочном DMRG из [252]. Однако, когда мы применяем (4.39) к набору крайних собственных векторов, они либо нормированы, и тогда ошибка аппроксимации распределяется равномерно, либо масштабируются с помощью физически обоснованных весов. В корректированном одноблочном DMRG методе идея блочного TT имеет гораздо более слабую аргументацию: во-первых, векторы p не несут никакого особого смысла кроме технических нужд изменения ранга; во-вторых, не существует разумной процедуры для определения веса a в конкретной задаче.

4.4 Практические особенности реализации алгоритмов DMRG и AMEn

В алгоритмах 8, 9 и 10 представлены наиболее важные шаги, необходимые для анализа и показа общей идеи. В этом разделе мы сосредоточимся на деталях реализации, которые улучшают производительность и делают практические методы эффективными. Важным моментом является последовательный проход по размерностям, $k = 1, 2, \ldots, d$, так как он обеспечивает линейную сложность в зависимости от размерности задачи d путем кэширования некоторых данных и использования только локальных (одно- и двухблочных) операций.

Поскольку многие операции будут включать в себя четырехмерные тензоры, введем пару новых перенумераций их элементов.

Определение 4.4.1. Пусть дан тензор $X^{(k)} = \left[X^{(k)}(\alpha, i, j, \beta)\right] \in \mathbb{C}^{p\times m\times n\times q}$. Будем группировать его элементы в виде следующих матриц:

- внешняя развертка $X^{\langle k\rangle} \in \mathbb{C}^{pm\times nq}$, $\quad X^{\langle k\rangle}(\overline{\alpha i}, \overline{j\beta}) = X^{(k)}(\alpha, i, j, \beta)$, и
- внутренняя развертка $X^{\rangle k\langle} \in C^{mn\times qp}$, $\quad X^{\rangle k\langle}(\overline{ij}, \overline{\beta\alpha}) = X^{(k)}(\alpha, i, j, \beta)$.

4.4.1 Вычисления в локальных системах

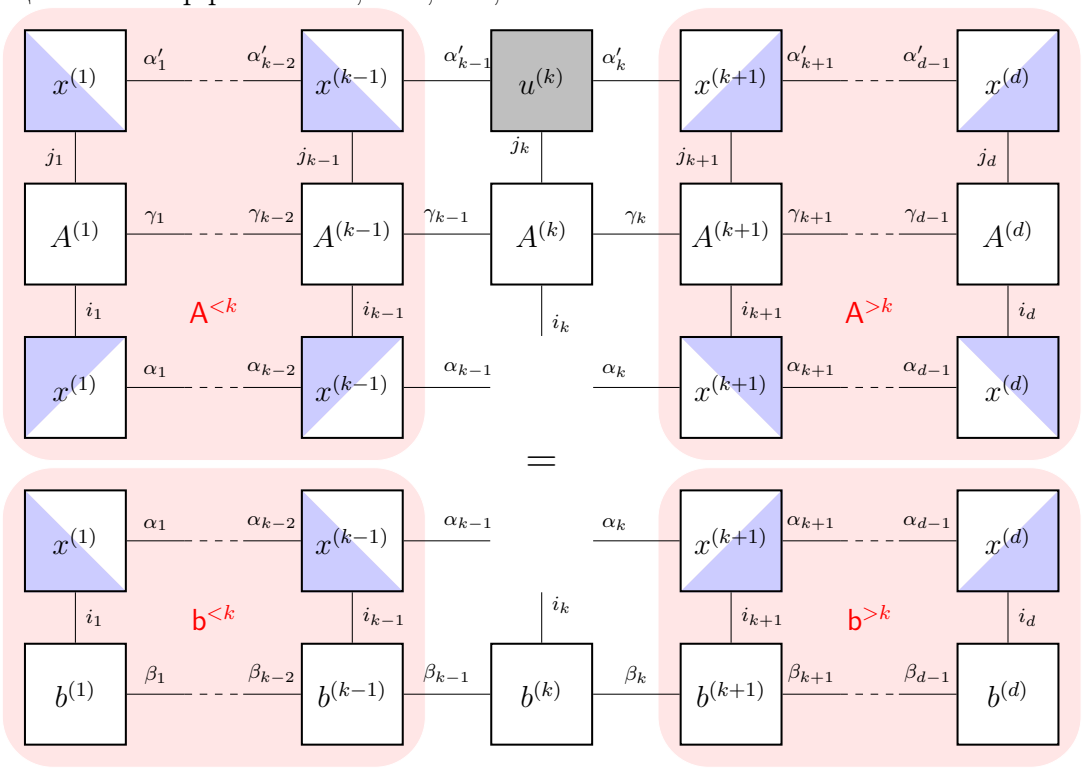
На каждом шаге оптимизации мы строим и решаем локальную систему $A_k u^{(k)} = b_k$ в соответствии с (4.7). Как показано на рис 4.3, она может быть собрана из TT ядер A, x и b за $\mathcal{O}(d)$ операций для каждого k, что, казалось бы, дает сложность полного прохода $\mathcal{O}(d^2)$.

Однако, для пересчета локальных систем между микрошагами $k = 1, 2, \ldots, d$ требуются лишь небольшие поправки (по аналогии с (4.35)). Действительно, как мы видели в (4.28), в одном шаге редукции эффективно участвует только один TT блок. Левые редукции $\mathsf{A}^{<k}$ можно поэтому вычислить следующим образом. Пусть нам дана матричная редукция на интерфейс $\mathsf{A}^{<k} \in \mathbb{C}^{r_{k-1}\times r_{k-1}\times r_{k-1}(A)}$, ее можно рассматривать с помощью трех- или четырехмерных группировок, $\mathsf{A}^{\langle<k|} = \mathsf{A}^{\langle<k\rangle} \in \mathbb{C}^{r_{k-1}\times r_{k-1}r_{k-1}(A)}$. Тогда, пользуясь стандартными матричными произведениями, вычисляем:

$$\begin{array}{lllll} 1. & p^{\langle k\rangle} & = \left(x^{\langle k|}\right)^* \mathsf{A}^{\langle<k|} & \in & \mathbb{C}^{n_k r_k\times r_{k-1}r_{k-1}(A)}; \\ 2. & q^{\langle k\rangle} & = p^{\rangle k\langle} A^{\langle k\rangle} & \in & \mathbb{C}^{r_k r_{k-1}\times n_k r_k(A)}; \\ 3. & \mathsf{A}^{\rangle<k+1\langle} & = \left(x^{|k\rangle}\right)^\top q^{\rangle k\langle} & \in & \mathbb{C}^{1\cdot r_k\times r_k(A) r_k}. \end{array} \tag{4.41}$$

После этого, еще одна "обратная" перегруппировка $\mathsf{A}^{\langle<k+1\rangle} \in \mathbb{C}^{r_k\cdot 1\times r_k r_k(A)}$ (не требует вычислений) переставляет размерности в изначальном порядке, соответствующем $\mathsf{A}^{<k}$. Обратите внимание, что $A^{\langle k\rangle}$ во второй строке является матричной группировкой элементов TT блока $A^{(k)}$ в соответствии с определением 4.4.1. В третьей строке, фиктивный размер 1 пишется явно, чтобы применение четырехмерной перестановки было однозначно определено.

Рис. 4.6: Линейная система $A_k u^{(k)} = b_k$ (4.7) собирается из ТТ блоков $A^{(k)}$, $b^{(k)}$, и редукций на интерфейсы $\mathsf{A}^{<k}$, $\mathsf{A}^{>k}$, $\mathsf{b}^{<k}$, $\mathsf{b}^{>k}$.



Такая же рекурсия выполняется для правых редукций $\mathsf{A}^{>k} \in \mathbb{C}^{r_k(A) \times r_k \times r_k}$.

$$\begin{aligned}
&1.\quad p^{\langle k\rangle} &&= x^{|k\rangle} \left(\mathsf{A}^{|>k\rangle}\right)^\top &&\in \mathbb{C}^{r_{k-1}n_k \times r_k(A) r_k};\\
&2.\quad q^{\langle k\rangle} &&= A^{\langle k\rangle} p^{\rangle k\langle} &&\in \mathbb{C}^{r_{k-1}(A)n_k \times r_k r_{k-1}};\\
&3.\quad \mathsf{A}^{\rangle>k-1\langle} &&= \bar{x}^{\langle k|} q^{\rangle k\langle} &&\in \mathbb{C}^{r_{k-1}\cdot 1 \times r_{k-1} r_{k-1}(A)},
\end{aligned} \tag{4.42}$$

и $\mathsf{A}^{\langle>k-1\rangle} \in \mathbb{C}^{r_{k-1}(A) r_{k-1} \times 1 \cdot r_{k-1}}$ восстанавливает правильный порядок размерностей.

Процедура инициализируется соглашением, что $\mathsf{A}^{<1} = \mathsf{A}^{>d} = 1$. Теперь одноблочная локальная матрица (4.7) собирается как трехмерный матричный ТТ формат: $A_k = \tau(\mathsf{A}^{<k}, A^{(k)}, \mathsf{A}^{>k})$, а двумерная система (4.10) — как четырехмерный матричный ТТ $A_{k,k+1} = \tau(\mathsf{A}^{<k}, A^{(k)}, A^{(k+1)}, \mathsf{A}^{>k+1})$. Рис. 4.3, показывающий построение одноблочной системы, может быть заменен более детальным рис. 4.6.

Кроме того, в процедуре итерационного метода решения локальной системы, произведение $w^{(k)} = A_k v^{(k)}$ для любого вектора $v^{(k)} \in \mathbb{C}^{r_{k-1} n_k r_k}$ может быть эффективно вычислено в следующем структурированном виде, см. также [55]:

$$\begin{aligned}
&1.\quad p^{\langle k\rangle} &&= v^{|k\rangle} \left(\mathsf{A}^{|>k\rangle}\right)^\top &&\in \mathbb{C}^{r_{k-1}n_k \times r_k(A) r_k};\\
&2.\quad q^{\langle k\rangle} &&= A^{\langle k\rangle} p^{\rangle k\langle} &&\in \mathbb{C}^{r_{k-1}(A)n_k \times r_k r_{k-1}};\\
&3.\quad w^{\langle k|} &&= \mathsf{A}^{\langle<k|} \left(q^{\rangle k\langle}\right)^\top &&\in \mathbb{C}^{r_{k-1} \times n_k r_k}.
\end{aligned} \tag{4.43}$$

Сложность этой процедуры, а также редукций (4.41),(4.42), можно легко оце-

нить из размеров возникающих матриц:

$$\texttt{work}\,[(4.41),(4.42),(4.43)] = \mathcal{O}(nr^3r(A)) + \mathcal{O}(n^2r^2r(A)^2). \tag{4.44}$$

Второй член возникает из умножения на $A^{\langle k\rangle}$. Оно может быть выполнено за $\mathcal{O}(nr^2r(A)^2)$ операций, если TT блок $A^{(k)}(i_k, j_k)$ разреженный относительно модовых индексов i_k, j_k. Таким образом, общая сложность становится *линейной* относительно модового размера n.

Для расчетов редукций правой части $\mathsf{b}^{<k} \in \mathbb{C}^{r_{k-1}\times r_{k-1}(b)}$ и $\mathsf{b}^{>k} \in \mathbb{C}^{r_k(b)\times r_k}$ (см. (4.28)), можно также использовать уравнения (4.41) и (4.42), заменяя x на b в первом шаге, и пропуская второй шаг. На самом деле, $\mathsf{b}^{<k+1}$ представляют собой ни что иное, как промежуточные матрицы s_k в алгоритме скалярного произведения в TT формате 1, примененного для (x, b).

4.4.2 Аппроксимация решения

Алгоритм AMEn увеличивает TT ранги решения на каждом шаге. Иногда, однако, разумно уменьшать их, так как расширенный градиентный спуск может давать неоптимальное решение для данных рангов. Для этого мы можем после каждого микрошага применить алгоритм округления TT-SVD 4 к решению. Так как матрицы интерфейсов ортогональны, шаг сингулярного разложения вычислительно эффективен: мы просто добавляем аппроксимацию $u^{|1\rangle}$ между строками 1 и 2 алгоритма 10. Более подробное описание приводится ниже в нерекуррентных версиях алгоритма.

Анализ сходимости в разделе 4.3.3 не зависит от этого шага. Действительно, так как невязка вычисляется после аппроксимации решения, мы можем внести соответствующее возмущение в начальное приближение, подобно тому, как это было сделано в лемме 4.1.1. Фактор роста ошибки $\Omega/(1-\Omega)$, возникающий в лемме 4.1.1 и теореме 4.3.7, может выглядеть пессимистично, однако на практике сходимость метода оказывается гораздо быстрее, чем оценки Ω или $\tilde{\Omega}$, и эффектом аппроксимации решения можно пренебречь.

4.4.3 Аппроксимация невязки: сингулярное разложение

В строке 3 алгоритма AMEn 10, формальная сложность процедуры TT округления составляет $\mathcal{O}((d-k+1)n(rr(A)+r(b))^3)$. Однако, для расширения TT блока решения нам нужен только один TT блок приближенной невязки. Давайте посмотрим, как можно написать алгоритм так, чтобы сложность каждого микрошага не зависела от d.

Согласно секции 2.1.5, первый TT блок *точной* невязки $z_k = b_{\geqslant k} - A_{\geqslant k}u_k$ вычисляется так:

$$(\hat{z}_k^{(k)})_{\eta_k} = \begin{cases} \tau(\mathsf{b}^{<k}, b^{(k)}_{\beta_k}), & \eta_k = 1,\ldots,r_k(b), \\ -\tau(\mathsf{A}^{<k}, A^{(k)}_{\gamma_k})u^{|k\rangle}_{\alpha_k}, & \eta_k = r_k(b) + \overline{\alpha_k\gamma_k} = r_k(b)+1,\ldots,r_k(b)+r_kr_k(A), \end{cases} \tag{4.45}$$

поэтому $z_k^{(k)} \in \mathbb{C}^{r_{k-1}n_k \times (r_k(b)+r_k r_k(A))}$, а остальные блоки вычисляются по общему правилу

$$\begin{aligned} \hat{z}_k^{(d)}(i_d) &= \begin{bmatrix} b^{(d)}(i_d) \\ \sum_{j_d} A^{(d)}(i_d, j_d) \otimes t^{(d)}(j_d) \end{bmatrix}, \\ \hat{z}_k^{(p)}(i_k) &= \begin{bmatrix} b^{(p)}(i_k) & \\ & \sum_{j_k} A^{(p)}(i_k, j_k) \otimes t^{(p)}(j_k) \end{bmatrix}, \quad \text{для} \quad p = k+1, \ldots, d-1. \end{aligned} \tag{4.46}$$

Мы видим, что изменение решения происходит только в k-м блоке, а все остальные блоки невязки (4.46) зависят от предыдущего приближения t. Поэтому, блоки $\hat{z}_k^{(p)} = \hat{z}^{(p)}$ являются одинаковыми для всех $p > k$, и могут быть насчитаны заранее перед AMEn итерацией. То же самое касается и правых ортогонализаций $\hat{z}^{(p)}$, которые выполняются до начала итерации, что составляет первую часть TT-SVD алгоритма 4. Мы сохраняем все LQ факторы, возникающие в процессе ортогонализации, и потом умножаем их на $\hat{z}_k^{(k)}$ в основном проходе. После этого, для получения блока приближенной невязки $z_k^{(k)}$, требуется вычислить единственное сингулярное разложение.

Всю процедуру можно записать в нерекуррентном варианте схемы 10, которую мы назвали AMEn_{svd} алгоритмом 11. Сложность каждого LQ или SVD шага для невязки составляет $\mathcal{O}(n(rr(A) + r(b))^3)$, т.е. $\mathcal{O}(nr^6)$ если TT ранги матрицы и решения сопоставимы.

4.4.4 Аппроксимация невязки: ALS метод

Чтобы уменьшить сложность по отношению к TT рангам, мы можем заменить TT-SVD на одноблочный DMRG (или ALS) алгоритм 8, примененный к задаче аппроксимации $\mathsf{J}_{I,z}(\tilde{z}) = \|z - \tilde{z}\|^2 \to \min$.

Алгоритм ALS, настроенный для конкретного вида $z = b - Au$ (4.46), может быть выполнен с $\mathcal{O}(d)$ сложностью с введением дополнительных величин, подобно (4.41), (4.42). Будем считать, что начальное приближение для невязки дано в TT формате, $\tilde{z} = \tau(\{z^{(k)}\})$, с рангами $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_{d-1})$, также, как и матрица $A = \tau(\{A^{(k)}\})$, и текущее решение $x = \tau(\{x^{(k)}\})$. Введем $\mathsf{A}_z^{<k} \in \mathbb{C}^{\rho_{k-1} \times r_{k-1} \times r_{k-1}(A)}$, составляющие части проекции $Z_{\neq k}^* A X_{\neq k}$, и будем вычислять их рекуррентно следующим образом:

$$\begin{aligned} &1. \quad p^{\langle k \rangle} &&= \left(z^{\langle k|}\right)^* \mathsf{A}_z^{\langle <k|} &&\in \mathbb{C}^{n_k \rho_k \times r_{k-1} r_{k-1}(A)}; \\ &2. \quad q^{\langle k \rangle} &&= p^{\rangle k \langle} A^{\langle k \rangle} &&\in \mathbb{C}^{\rho_k r_{k-1} \times n_k r_k(A)}; \\ &3. \quad \mathsf{A}_z^{\rangle <k+1 \langle} &&= \left(x^{|k\rangle}\right)^\top q^{\rangle k \langle} &&\in \mathbb{C}^{1 \cdot r_k \times r_k(A) \rho_k}, \end{aligned} \tag{4.47}$$

и "обратная" перестановка $\mathsf{A}_z^{\langle <k+1 \rangle} \in \mathbb{C}^{\rho_k \cdot 1 \times r_k r_k(A)}$ упорядочивает размерности в том же порядке, как и в $\mathsf{A}_z^{<k}$.

Правые редукции $\mathsf{A}_z^{>k} \in \mathbb{C}^{r_k(A) \times \rho_k \times r_k}$ вычисляются аналогично.

$$\begin{aligned} &1. \quad p^{\langle k \rangle} &&= x^{|k\rangle} \left(\mathsf{A}_z^{|>k\rangle}\right)^\top &&\in \mathbb{C}^{r_{k-1} n_k \times r_k(A) \rho_k}; \\ &2. \quad q^{\langle k \rangle} &&= A^{\langle k \rangle} p^{\rangle k \langle} &&\in \mathbb{C}^{r_{k-1}(A) n_k \times \rho_k r_{k-1}}; \\ &3. \quad \mathsf{A}_z^{\rangle >k-1 \langle} &&= \bar{z}^{\langle k|} q^{\rangle k \langle} &&\in \mathbb{C}^{\rho_{k-1} \cdot 1 \times r_{k-1} r_{k-1}(A)}, \end{aligned} \tag{4.48}$$

и завершает процедуру “обратная” перестановка $\mathsf{A}_z^{\langle >k-1\rangle} \in \mathbb{C}^{r_{k-1}(A)\rho_{k-1}\times 1\cdot r_{k-1}}$. Заменяя x на b и опуская вторые шаги в (4.47), (4.48), получаем также формулы для $\mathsf{b}_z^{<k} \in \mathbb{C}^{\rho_{k-1}\times r_{k-1}(b)}$ и $\mathsf{b}_z^{>k} \in \mathbb{C}^{r_k(b)\times\rho_k}$.

Выполнение одного микрошага в алгоритме 8 (строка 8) сводится к копированию $z^{(k)} = b_{z_k}$, где b_{z_k} вычисляется с учетом (4.47), (4.48) и блочной структуры (4.46) таким образом:

$$z^{(k)} = \tau(\mathsf{b}_z^{<k}, b^{(k)}, \mathsf{b}_z^{>k}) - \tau(\mathsf{A}_z^{<k}, A^{(k)}, \mathsf{A}_z^{>k})u^{(k)}. \tag{4.49}$$

Для последнего произведения матрицы на $u^{(k)}$ в правой части можно использовать структурированное вычисление (4.43). Оценивая размеры $\mathsf{A}_z, \mathsf{b}_z$, получаем оценку сложности, аналогичную (4.44):

$$\texttt{work}\,[(4.47),(4.48)] = \mathcal{O}(nr^2\rho r(A)) + \mathcal{O}(nr\rho^2 r(A)) + \mathcal{O}(n^2 r\rho r(A)^2).$$

Если TT ранги невязки ρ_k выбираются значительно меньшими, чем ранги матрицы и решения, сложность такого шага ALS является кубической относительно характерного ранга, $\mathcal{O}(n^2r^3)$, что существенно меньше, чем величина $\mathcal{O}(nr^6)$ для алгоритма AMEn_{svd}.

Оказывается, что даже одного микрошага (4.49) после изменения в x достаточно, чтобы поддерживать удовлетворительное приближение $\tilde{z}$. Это позволяет выполнять алгоритмы AMEn для решения $Ax = b$ и ALS для приближения $\tilde{z} \approx z$ одновременно, синхронизируя шаги в обоих методах, как показано в алгоритме AMEn_{als} 12.

Следует отметить, что аппроксимация редуцированной невязки $\tilde{z}_k \approx z_k$, в частности блока расширения $z_k^{(k)}$ в строке 16, вычисляется с использованием интерфейсных блоков от *глобальной* невязки, $z^{(p)}$, $p = k+1, \ldots, d$. Это похоже на кэширование LQ факторов в алгоритме AMEn_{svd} 11. Однако, поскольку пока не доказана глобальная сходимость одноблочного DMRG, невозможно строго оценить качество $z^{(p)}$ для аппроксимации z и z_k. Тем не менее, на практике значительное ускорение перевешивает отсутствие теоретической гарантии для этой эвристики.

4.4.5 AMEn алгоритм для быстрой аппроксимации матричного произведения

Подставляя в алгоритме 12 из предыдущей секции матрицу $A = I$ и правую часть $b = My$, получаем AMEn метод для быстрой аппроксимации произведения матрицы на вектор, например, в явной схеме интегрирования по времени уравнения Власова, см. раздел 3.2. Разница заключается в том, что для шага 12 не нужно реально решать линейную систему, поскольку $A_k = I$, и мы просто копируем $u^{(k)} = b_k$. А вот эффективное вычисление правой части требует структурированного локального произведения наподобие (4.43).

Итак, обозначим за $\mathsf{M}^{<k}$ редукции My на левый интерфейс x, которые вычис-

ляются аналогично (4.41):

$$\begin{array}{lrcl} 1. & p^{\langle k\rangle} = \left(x^{\langle k|}\right)^* \mathsf{M}^{\langle <k|} & \in & \mathbb{C}^{n_k r_k \times r_{k-1}(y) r_{k-1}(M)}; \\ 2. & q^{\langle k\rangle} = p^{\rangle k\langle} M^{\langle k\rangle} & \in & \mathbb{C}^{r_k r_{k-1}(y) \times n_k r_k(A)}; \\ 3. & \mathsf{M}^{\rangle <k+1\langle} = \left(y^{|k\rangle}\right)^\top q^{\rangle k\langle} & \in & \mathbb{C}^{1\cdot r_k(y) \times r_k(A) r_k}. \end{array} \tag{4.50}$$

Здесь, $M^{(k)}$ обозначает TT блок матрицы M, а $y^{(k)}$ TT блок вектора y.

Для правых редукций $\mathsf{M}^{>k} \in \mathbb{C}^{r_k(A)\times r_k \times r_k(y)}$ имеем:

$$\begin{array}{lrcl} 1. & p^{\langle k\rangle} = y^{|k\rangle} \left(\mathsf{M}^{|>k\rangle}\right)^\top & \in & \mathbb{C}^{r_{k-1}(y) n_k \times r_k(A) r_k}; \\ 2. & q^{\langle k\rangle} = M^{\langle k\rangle} p^{\rangle k\langle} & \in & \mathbb{C}^{r_{k-1}(A) n_k \times r_k r_{k-1}(y)}; \\ 3. & \mathsf{M}^{\rangle >k-1\langle} = \bar{x}^{\langle k|} q^{\rangle k\langle} & \in & \mathbb{C}^{r_{k-1}\cdot 1 \times r_{k-1}(y) r_{k-1}(A)}. \end{array} \tag{4.51}$$

После этого вычисляем матрично-векторное произведение:

$$\begin{array}{lrcl} 1. & p^{\langle k\rangle} = y^{|k\rangle} \left(\mathsf{M}^{|>k\rangle}\right)^\top & \in & \mathbb{C}^{r_{k-1}(y) n_k \times r_k(A) r_k}; \\ 2. & q^{\langle k\rangle} = M^{\langle k\rangle} p^{\rangle k\langle} & \in & \mathbb{C}^{r_{k-1}(A) n_k \times r_k r_{k-1}(y)}; \\ 3. & u^{\langle k|} = \mathsf{M}^{\langle <k|} \left(q^{\rangle k\langle}\right)^\top & \in & \mathbb{C}^{r_{k-1} \times n_k r_k}, \end{array} \tag{4.52}$$

составляющее локальный шаг $u^{(k)} = \tau(\mathsf{M}^{<k}, M^{(k)}, \mathsf{M}^{>k}) y^{(k)}$.

Роль невязки в данном методе играет ошибка, $z = My - x$. Следовательно, нам еще потребуются редукции My и x на интерфейсы $\tilde{z}$, с учетом которых вычисление нового TT блока ошибки пишется как

$$z^{(k)} = \tau(\mathsf{M}_z^{<k}, M^{(k)}, \mathsf{M}_z^{>k}) y^{(k)} - \tau(\mathsf{x}_z^{<k}, u^{(k)}, \mathsf{x}_z^{>k}).$$

Вычисление $\mathsf{M}_z^{<k}, \mathsf{M}_z^{>k}$ в точности повторяет соответственно (4.47) и (4.48), с заменой A на M и x на y. Пропуская вторые шаги в (4.47) и (4.48), получаем и редукции для x, т.е. $\mathsf{x}_z^{<k}, \mathsf{x}_z^{>k}$. Полная процедура приведена в Алгоритме 13.

Обратите внимание, что AMEn метод для задачи аппроксимации по своей природе эвристический. Аналог реализации AMEn_{svd} не имеет смысла, потому что SVD аппроксимация ошибки не проще, чем SVD аппроксимация произведения My напрямую.

4.4.6 AMEn и DMRG для формата QTT-Tucker

Таким же образом, как и в разделе про округление в формате QTT-Tucker 2.2.5, методы переменных направлений для этого формата можно написать с использованием их TT версий для ядра Таккера и расширенных факторов, см. алгоритм 5. Единственная модификация, требующаяся в исходных DMRG алгоритмах 8,9, а также AMEn алгоритмах 11,12, это возможность возврата помимо решения, еще и редукций $\mathsf{A}^{<k}, \mathsf{A}^{>k}, \mathsf{b}^{<k}, \mathsf{b}^{>k}$. Действительно, пусть мы провели оптимизацию k-го расширенного фактора (2.21). Тогда последняя редукция $\mathsf{A}_{\gamma_k}^{f(k,<1)} = (x^{f(k)})^* A_{\gamma_k}^{f(k)} x^{f(k)} \in \mathbb{C}^{R_k \times R_k}$, $\gamma_k = 1, \ldots, R_k(A)$, является Таккеровским фактором

для матрицы, служащей для оптимизации Таккеровского ядра. И наоборот, редукции, вычисленные во время оптимизации ядра, например $\mathsf{A}^{c(<k)} \in \mathbb{C}^{r_{k-1} \times r_{k-1} \times r_{k-1}(A)}$, дают последние TT блоки матриц, требующихся для оптимизации расширенных факторов. Соответствующие диаграммы тензорных сетей приведены в [50]. Здесь мы однако предпочтем опустить детальное описание наподобие алгоритмов 11, 12, поскольку оно было бы слишком длинным.

Алгоритм 11 $\mathrm{AMEn}_{\mathrm{svd}}$, одна итерация

Ввод: Начальное приближение $t = \tau(\{t^{(k)}\})$ в TT формате (2.9), точность ε или ранг r для решения, точность ϵ или ранг ρ для невязки.

Вывод: Новое приближение $x = \tau(\{x^{(k)}\})$ с условием $r' \leqslant r + \rho$.

Скопировать $x^{(k)} = t^{(k)}$, $k = 1, \ldots, d$.
Вычислить невязку $z = \tau(\{\hat{z}^{(p)}\})$ используя (4.46).
Положить $\mathsf{A}^{>d} = \mathsf{A}^{<1} = \mathsf{b}^{>d} = \mathsf{b}^{<1} = 1$.
for $k = d, \ldots, 2$ **do** {Ортогонализация и редукция}
    Вычислить LQ разложение $x^{\langle k|} = LQ$, $QQ^* = I$.
    Заменить $x^{\langle k|} := Q$, и $x^{|k-1\rangle} := x^{|k-1\rangle}L$.
    Вычислить правые редукции $\mathsf{A}^{>k-1}$, $\mathsf{b}^{>k-1}$ по (4.42).
    Вычислить LQ разложение $\hat{z}^{\langle k|} = L_{k-1}Q$, $QQ^* = I$.
    Заменить $\hat{z}^{|k-1\rangle} := \hat{z}^{|k-1\rangle}L_{k-1}$, сохранить L_{k-1}.
end for
for $k = 1, \ldots, d$ **do** {Оптимизация TT блоков}
    Построить $b_k = \tau(\mathsf{b}^{<k}, b^{(k)}, \mathsf{b}^{>k})$, и {опционально} $A_k = \tau(\mathsf{A}^{<k}, A^{(k)}, \mathsf{A}^{>k})$.
    Решить $A_k u^{(k)} = b_k$. {Взять $x^{(k)}$ в качестве начального приближения, использовать (4.43)}
    Вычислить SVD $u^{|k\rangle} \approx U\Sigma V^*$ так что $\left\|u^{|k\rangle} - U\Sigma V^*\right\| \leqslant \varepsilon \left\|u^{|k\rangle}\right\|$ или $\operatorname{rank}(\Sigma) \leqslant r$, заменить $u^{|k\rangle} := U\Sigma V^*$.
    if $k \neq d$ **then** {Расширение, ортогонализация и редукция}
        Вычислить (4.45) и его SVD $\hat{z}_k^{|k\rangle}L_k \approx z_k^{|k\rangle}\Sigma_z V_z^*$, так что $\left\|\hat{z}_k^{|k\rangle}L_k - z_k^{|k\rangle}\Sigma_z V_z^*\right\| \leqslant \epsilon \left\|\hat{z}_k^{|k\rangle}L_k\right\|$ или $\operatorname{rank}(\Sigma_z) \leqslant \rho$.
        Объединить $x^{|k\rangle} := \begin{bmatrix} U & z_k^{|k\rangle} \end{bmatrix}$, $x^{\langle k+1|} := \begin{bmatrix} \Sigma V^* x^{\langle k+1|} \\ 0 \end{bmatrix}$.
        Вычислить QR разложение $x^{|k\rangle} = QR$, $Q^*Q = I$.
        Заменить $x^{|k\rangle} := Q$, и $x^{\langle k+1|} := Rx^{\langle k+1|}$.
        Вычислить левые редукции $\mathsf{A}^{<k+1}$, $\mathsf{b}^{<k+1}$ по (4.41).
    end if
end for
return $x = \tau(x^{(1)}, \ldots, x^{(d)})$.

Алгоритм 12 $\mathrm{AMEn}_{\mathrm{als}}$, одна итерация

Ввод: Начальное приближение $t = \tau(\{t^{(k)}\})$ в TT формате (2.9), точность ε или ранг r для решения, начальное приближение $\tilde{z} = \tau(\{z^{(k)}\})$ для невязки.

Вывод: Новые приближения $x = \tau(\{x^{(k)}\})$, с рангами $r' \leqslant r + \rho$, и $\tilde{z} = \tau(\{z^{(k)}\})$.

Скопировать $x^{(k)} = t^{(k)}$, $k = 1, \ldots, d$.
Положить $\mathsf{A}^{>d} = \mathsf{A}^{<1} = \mathsf{b}^{>d} = \mathsf{b}^{<1} = \mathsf{A}_z^{>d} = \mathsf{A}_z^{<1} = \mathsf{b}_z^{>d} = \mathsf{b}_z^{<1} = 1$.
for $k = d, \ldots, 2$ **do** {Ортогонализации и редукции}
    Вычислить LQ разложение $x^{\langle k|} = LQ$, $QQ^* = I$.
    Заменить $x^{\langle k|} := Q$, и $x^{|k-1\rangle} := x^{|k-1\rangle}L$.
    Вычислить LQ разложение $z^{\langle k|} = LQ$, $QQ^* = I$.
    Заменить $z^{\langle k|} := Q$, и $z^{|k-1\rangle} := z^{|k-1\rangle}L$.
    Вычислить редукции $\mathsf{A}^{>k-1}$, $\mathsf{b}^{>k-1}$ по (4.42), и $\mathsf{A}_z^{>k-1}$, $\mathsf{b}_z^{>k-1}$ по (4.48).
end for
for $k = 1, \ldots, d$ **do** {Оптимизация TT блоков}
    Построить $b_k = \tau(\mathsf{b}^{<k}, b^{(k)}, \mathsf{b}^{>k})$, и {опционально} $A_k = \tau(\mathsf{A}^{<k}, A^{(k)}, \mathsf{A}^{>k})$.
    Решить $A_k u^{(k)} = b_k$. {Взять $x^{(k)}$ как начальное приближение, использовать (4.43)}
    Вычислить SVD $u^{|k\rangle} \approx U\Sigma V^*$ так что $\left\|u^{|k\rangle} - U\Sigma V^*\right\| \leqslant \varepsilon \left\|u^{|k\rangle}\right\|$ или $\operatorname{rank}(\Sigma) \leqslant r$, заменить $u^{|k\rangle} := U\Sigma V^*$.
    if $k \neq d$ **then** {Расширение, ортогонализации и редукции}
        Вычислить $z^{(k)} = \tau(\mathsf{b}_z^{<k}, b^{(k)}, \mathsf{b}_z^{>k}) - \tau(\mathsf{A}_z^{<k}, A^{(k)}, \mathsf{A}_z^{>k})u^{(k)}$. {Использовать (4.43)}
        Вычислить $z_k^{(k)} = \tau(\mathsf{b}^{<k}, b^{(k)}, \mathsf{b}_z^{>k}) - \tau(\mathsf{A}^{<k}, A^{(k)}, \mathsf{A}_z^{>k})u^{(k)}$. {Использовать (4.43)}
        Объединить $x^{|k\rangle} := \begin{bmatrix} U & z_k^{|k\rangle} \end{bmatrix}$, $x^{\langle k+1|} := \begin{bmatrix} \Sigma V^* x^{\langle k+1|} \\ 0 \end{bmatrix}$.
        Вычислить QR разложение $x^{|k\rangle} = QR$, $Q^*Q = I$.
        Заменить $x^{|k\rangle} := Q$, и $x^{\langle k+1|} := Rx^{\langle k+1|}$.
        Вычислить QR разложение $z^{|k\rangle} = QR$, $Q^*Q = I$.
        Заменить $z^{|k\rangle} := Q$, и $z^{\langle k+1|} := Rz^{\langle k+1|}$.
        Вычислить редукции $\mathsf{A}^{<k+1}$, $\mathsf{b}^{<k+1}$ по (4.41), и $\mathsf{A}_z^{<k+1}$, $\mathsf{b}_z^{<k+1}$ по (4.47).
    end if
end for
return $x = \tau(x^{(1)}, \ldots, x^{(d)})$, $\tilde{z} = \tau(z^{(1)}, \ldots, z^{(d)})$.

Алгоритм 13 $\mathrm{AMEn}_{\mathrm{mv}}$, одна итерация

Ввод: Начальное приближение $t = \tau(\{t^{(k)}\})$ в TT формате (2.9), точность ε или ранг r для решения, начальное приближение $\tilde{z} = \tau(\{z^{(k)}\})$ для невязки.

Вывод: Новые приближения $x = \tau(\{x^{(k)}\})$, с рангами $r' \leqslant r + \rho$, и $\tilde{z} = \tau(\{z^{(k)}\})$.

Скопировать $x^{(k)} = t^{(k)}$, $k = 1, \ldots, d$.
Положить $\mathsf{M}^{>d} = \mathsf{M}^{<1} = \mathsf{M}_z^{>d} = \mathsf{M}_z^{<1} = \mathsf{x}_z^{>d} = \mathsf{x}_z^{<1} = 1$.
for $k = d, \ldots, 2$ **do** {Ортогонализации и редукции}
  Вычислить LQ разложение $x^{\langle k|} = LQ$, $QQ^* = I$.
  Заменить $x^{\langle k|} := Q$, и $x^{|k-1\rangle} := x^{|k-1\rangle} L$.
  Вычислить LQ разложение $z^{\langle k|} = LQ$, $QQ^* = I$.
  Заменить $z^{\langle k|} := Q$, и $z^{|k-1\rangle} := z^{|k-1\rangle} L$.
  Вычислить редукции $\mathsf{M}^{>k-1}$ по (4.51), и $\mathsf{M}_z^{>k-1}$, $\mathsf{x}_z^{>k-1}$ аналогично (4.48).
end for
for $k = 1, \ldots, d$ **do** {Оптимизация TT блоков}
  Вычислить $u^{(k)} = \tau(\mathsf{M}^{<k}, M^{(k)}, \mathsf{M}^{>k}) y^{(k)}$ с помощью (4.52).
  Вычислить SVD $u^{|k\rangle} \approx U\Sigma V^*$ так что $\left\|u^{|k\rangle} - U\Sigma V^*\right\| \leqslant \varepsilon \left\|u^{|k\rangle}\right\|$ или $\operatorname{rank}(\Sigma) \leqslant r$, заменить $u^{|k\rangle} := U\Sigma V^*$.
  if $k \neq d$ **then** {Расширение, ортогонализации и редукции}
    Вычислить $z^{(k)} = \tau(\mathsf{M}_z^{<k}, M^{(k)}, \mathsf{M}_z^{>k}) y^{(k)} - \tau(\mathsf{x}_z^{<k}, u^{(k)}, \mathsf{x}_z^{>k})$. {Аналогично (4.52)}
    Вычислить $z_k^{(k)} = \tau(\mathsf{M}^{<k}, M^{(k)}, \mathsf{M}_z^{>k}) y^{(k)} - \tau(u^{|k\rangle}, \mathsf{x}_z^{>k})$. {Аналогично (4.52)}
    Объединить $x^{|k\rangle} := \begin{bmatrix} U & z_k^{|k\rangle} \end{bmatrix}$, $x^{\langle k+1|} := \begin{bmatrix} \Sigma V^* x^{\langle k+1|} \\ 0 \end{bmatrix}$.
    Вычислить QR разложение $x^{|k\rangle} = QR$, $Q^*Q = I$.
    Заменить $x^{|k\rangle} := Q$, и $x^{\langle k+1|} := Rx^{\langle k+1|}$.
    Вычислить QR разложение $z^{|k\rangle} = QR$, $Q^*Q = I$.
    Заменить $z^{|k\rangle} := Q$, и $z^{\langle k+1|} := Rz^{\langle k+1|}$.
    Вычислить редукции $\mathsf{M}^{<k+1}$ по (4.50), и $\mathsf{M}_z^{<k+1}$, $\mathsf{x}_z^{<k+1}$ аналогично (4.47).
  end if
end for
return $x = \tau(x^{(1)}, \ldots, x^{(d)})$, $\tilde{z} = \tau(z^{(1)}, \ldots, z^{(d)})$.

Глава 5

Численные эксперименты

В этой главе мы рассмотрим различные численные примеры, демонстрирующие особенности методов тензорных приближений, и проверяющие концепции на конкретных задачах, связанных с основным кинетическим уравнением и гибридной моделью Фарлей-Бунемановской неустойчивости, предложенными в первой главе.

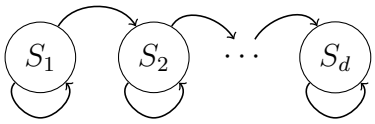
Большинство экспериментов взяты из работ автора и в соавторстве, например, [50, 51, 57, 59]. **Все алгоритмы и численные эксперименты были реализованы автором.** Некоторые тесты, которые ранее были представлены в сокращенном виде (например, из-за ограничения на число страниц в журнале), в данной работе расширены, с целью исследовать некоторые эффекты второстепенного значения.

Методы и тестовые программы были разработаны на основе MATLAB пакета TT-Toolbox `http://github.com/oseledets/TT-Toolbox`, поддерживаемого автором в сотрудничестве с И. Оселедцем. TT-Toolbox представляет собой объектно-ориентированную систему для вычислений в форматах TT, QTT и QTT-Tucker, и содержит как простейшие алгебраические операции (сложения, произведения и округления, см. секцию 2.1.5), так и продвинутые алгоритмы DMRG и AMEn, в частности:

- `dmrg_solve3.m` двухблочный DMRG 9 для линейных систем;
- `amen_solve2.m` AMEn метод для линейных систем (заключает в себе оба алгоритма 11 и 12);
- `amen_mv.m` AMEn метод для аппроксимации матричного произведения (алгоритм 13, см. также замечание 4.2.3).

Некоторые модули, показывающие низкую эффективность в MATLAB (например, метод GMRES для решения локальных систем (4.7)), были переписаны в виде MEX библиотек на языках Fortran90 и C.

Рис. 5.1: Каскадная сеть реакций



5.1 Основное кинетическое уравнение для сетей биологических реакций

5.1.1 Каскад реакций на коротком промежутке времени: сравнение методов

Мы начнем с проверки методов AMEn, и их сравнения с DMRG на примере основного кинетического уравнения для d-мерной каскадной сети генной регуляции, следуя [57]. Каскадный процесс происходит в том случае, когда гены производят белки, влияющие на экспрессию следующего гена, см. рис. 5.1. Это типичная модель в генных сетях; примером является литическая фаза жизненного цикла λ-фага [208]. На рис. 5.1, стрелки обозначают обратные связи между реагентами, соответствующие функциям скоростей w^m: стрелка, идущая от i-го к j-му компоненту означает, что скорость реакции с участием j-го гена зависит от числа копий i-го. Число реакций в каскадной системе составляет $M = 2d$ в соответствии с двумя классами процессов: автономное *разрушение* (что моделирует, например, диффузию молекул через мембрану клетки), и индуцированная предыдущим реагентом *генерация* (с некоторой насыщающейся скоростью, например, вида Michaelis-Menten).

Стехиометрия предполагает *мономолекулярные* реакции: все реакции разрушения, соответствующие $m = 1, \ldots, d$, уменьшают число копий m-го вещества на единицу, т.е. $\boldsymbol{z}^m = -\boldsymbol{e}_m$, отрицательный m-й единичный вектор, а реакции генерации, соответствующие $m = d+1, \ldots, 2d$, увеличивают число копий на единицу, тем самым, обладают положительным единичным стехиометрическим вектором, $\boldsymbol{z}^m = \boldsymbol{e}_{m-d}$.

Как было отмечено, функция скорости m-й разрушающей реакции зависит только от i_m, так что она представляется в виде прямого тензорного произведения (ранга 1):

$$w^m(\boldsymbol{i}) = e(i_1) \cdots e(i_{m-1}) \cdot \check{w}^m(i_m) \cdot e(i_{m+1}) \cdots e(i_d),$$

где $e(i_k) = 1 \; \forall i_k = 0, \ldots, n_k - 1$. Соответствующая часть матрицы ОКУ имеет вид оператора Лапласа:

$$A_1 = D_1 \otimes J^0 \cdots \otimes J^0 + \cdots + J^0 \otimes \cdots \otimes D_d, \quad D_m = (J^{-1} - J^0)\,\mathrm{diag}(\check{w}^m), \qquad (5.1)$$

где J^z являются матрицами сдвига в соответствии с (1.16). Как было уже отмечено, правая часть (5.1) представляется в TT формате со всеми рангами, равными 2.

В генерирующих реакциях, функции скорости зависят от числа копий предыдущего вещества, т.е. $w^m = \hat{w}^m(i_{m-d-1})$ для $m = d+2, \ldots, 2d$. Таким образом, вторая часть оператора является суммой попарных членов:

$$A_2 = D_1^1 \otimes J^0 \cdots \otimes J^0 + D_1^2 \otimes D_2^2 \otimes J^0 \cdots \otimes J^0 + \cdots + J^0 \cdots \otimes D_{d-1}^d \otimes D_d^d, \qquad (5.2)$$

где $D^m_{m-1} = \mathrm{diag}(\hat{w}^{m+d})$, $D^m_m = (J^1 - J^0)$, $m = 1, \ldots, d$. Заметим, что (5.2) обладает ТТ представлением (3.4) со всеми ТТ рангами 3. Поэтому, полный оператор ОКУ $A = A_1 + A_2$ в (1.17) является матричным ТТ форматом с ТТ рангами не более 5.

Конкретные параметры модели были выбраны в соответствии с [227, 12, 51]:

- скорости разрушения $\check{w}^m(i_m) = 0.07 \cdot i_m$, $m = 1, \ldots, d$,
- скорости генерации: $\hat{w}^{d+1} = 0.7$, и $\hat{w}^m(i_{m-1-d}) = \frac{i_{m-1-d}}{5+i_{m-1-d}}$ для $m = d{+}2, \ldots, 2d$,
- количество веществ (размерность) $d = 20$,
- границы области FSP (т.е. модовые размеры тензоров) $n_k = n = 64$.

Несимметричная линейная система $B\psi = f$ возникает после дискретизации по времени неявным методом Эйлера, $\psi^{p+1} = (I - \delta t A)^{-1}\psi^p$, $\psi(t_{p+1}) = \psi(t_p + \delta t) \approx \psi^{p+1}$, где $\psi^p = \psi(t_p)$ является $n \times \ldots \times n$ тензором (или n^d вектором) решения ОКУ (1.17). Следуя разделу 1.3, все временные слои $\psi(t_p)$, $p = 1, \ldots, N_t$ хранятся одновременно в виде $n \times \ldots \times n \times N_t$ тензора $\psi = \left[\psi(p\delta t)\right]_{p=1}^{N_t}$ размерности $d{+}1$. Мы используем предобуславливатель по времени в виде обратной матрицы конечных разностей (Эйлеровый аналог (1.26)), поэтому правая часть пишется в виде $f = \psi(0) \otimes e$, где $e = (1, \ldots, 1)$, и матрица представляется в виде

$$B = I_{n^d} \otimes I_{N_t} - A \otimes \delta t G_t^{-1},$$

где $G_t = \mathrm{tridiag}(-1, 1, 0) \in \mathbb{R}^{N_t \times N_t}$ суть дискретизированный методом конечных разностей оператор градиента, а A это матрица (оператор) ОКУ, исследованная выше.

В этом сравнительном эксперименте мы выбираем небольшой временной интервал $T = 10$, но достаточно много временных шагов $N_t = 2^{12}$, чтобы $\delta t = T/N_t = \mathcal{O}(10^{-3})$ было достаточно малым. Все данные, и по переменным состояний, и по времени, сжимаем в QTT формате (см. секцию 2.2.1). Начальным состоянием является первый единичный вектор, $\psi(0) = (1, 0, \ldots, 0)$, что соответствует тому, что числа копий всех реагентов равны нулю с вероятностью 1. Это обеспечивает небольшие QTT ранги и для B, и для f.

Полный размер задачи $N_t n^d \sim 10^{40}$ делает ее прямое решение невозможным. Существующие методы либо отказываются от решения ОКУ вообще (метод SSA и основанные на нем), или используют многомерные приближения, например разреженные сетки [227], "жадные" алгоритмы [12, 210], или динамические постановки на тензорных многообразиях [123]. Здесь мы применяем алгоритм AMEn (обе версии, $\mathrm{AMEn}_{\mathrm{svd}}$ 11 и $\mathrm{AMEn}_{\mathrm{als}}$ 12) и сравниваем его с двухблочным DMRG алгоритмом 9, как с дополнительным расширением (4.13) *случайными* элементами ($\mathrm{DMRG}_{\mathrm{rnd}}$) так и без него (стандартный DMRG). Для некоторых систем с небольшими размерностями и мелкими шагами по времени, метод DMRG может работать неплохо, см. например [49]. Однако, мы покажем, что алгоритм AMEn работает лучше для данной более сложной задачи.

Мы устанавливаем относительный порог тензорных аппроксимаций для решения $\varepsilon = 10^{-6}$, и отслеживаем сходимость различных методов к эталонному

Таблица 5.1: ОКУ для каскада реакций, ошибки в конечном временном слое (err) и расчетные времена в секундах (time) различных методов

	AMEn_{svd}		AMEn_{als}		DMRG		KSL
	B	B^*B	B	B^*B	B	B^*B	
err	8.3e-6	2.7e-5	9.2e-6	2.4e-5	9.6e-1	1.8e+0	8.4e-4
time	48.7	343	15.3	47.4	7.30	5.21	226

решению, вычисленному с помощью AMEn_{svd} алгоритма с точностью $\varepsilon = 10^{-9}$. Результаты представлены на рис. 5.2. Поскольку $B\psi = f$ не является симметричной положительно определенной системой, мы можем применить традиционную симметризацию $B^*B\psi = B^*f$. Симметризация возводит в квадрат число обусловленности и TT ранги матрицы, что замедляет вычисления. Поэтому мы также применяем все алгоритмы непосредственно к $B\psi = f$, хотя для такого метода пока не существует теоретической гарантии сходимости.

Мы видим, что ни исходная система, ни симметризованная формулировка не решаются стандартным DMRG методом (без расширения). Так как он использует только локальную информацию о системе, он возвращает приближение со значительно заниженными TT рангами, что также отражается малыми расчетными временами. DMRG со случайным расширением также не справляется с несимметричной системой, хотя и демонстрирует некоторую медленную сходимость в симметризованной формулировке. Это до некоторой степени соответствует теории неточного градиентного спуска (4.18): DMRG сходится, если поправка не ортогональна невязке, что с большой вероятностью выполняется для случайных векторов. Однако скорость такой сходимости далека от оптимальной.

Все AMEn алгоритмы возвращают удовлетворительные решения. Интересно, что несимметричные версии оказываются быстрее и точнее, чем симметризованные. Это свидетельствует о том, что практические скорости сходимости AMEn методов намного лучше, чем теоретические оценки, гарантируемые теорией градиентного спуска. Рассматривая детальнее две реализации AMEn, можно заметить, что AMEn_{als} существенно ускоряет вычисления по сравнению с алгоритмом AMEn_{svd} (особенно когда TT ранги матрицы большие, например, $r_k \simeq 40$ для B^*B), в то время как сходимость не ухудшается.

Поскольку исходная задача является обыкновенным дифференциальным уравнением, мы заинтересованы в последнем временном слое решения $\psi(N_t\delta t)$. Так как мы приближаем все слои в общем TT формате, стоит проверить достоверность отдельных компонентов. В таблице 5.1 показаны относительные ошибки во фробениусовой норме для $\psi(N_t\delta t)$ в сравнении с эталонным решением. Мы видим, что действительно, ошибки находятся на том же уровне, что и точность общего решения ψ на рис. 5.2.

Теперь мы можем сравнить последний слой, вычисленный методом AMEn, с результатом, полученным другим методом интегрирования ОДУ. Динамическая задача может быть перенесена на тензорное многообразие с помощью так называ-

емого принципа *Дирака-Френкеля* (см., например, [153]): мы решаем задачу

$$\min_{x^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}} \left\| \frac{d\tau(\{x^{(k)}\})}{dt} - \frac{dy}{dt} \right\|,$$

проектируя точную скорость dy/dt на тангенциальное пространство TT многообразия. Некоторые теоретические результаты были доказаны в [65], а численный метод расщепления по TT блокам $x^{(k)}$ был предложен в [172] под названием *схема KSL*. Она может быть применена к нашему ОДУ, если dy/dt заменить на $M\tau(\{x^{(k)}\})$.

Схема KSL оказалась эффективна для моделирования квантовых молекулярных колебаний [199]. Однако она обладает тем же недостатком, что и алгоритм ALS 8: TT ранги решения предопределены и фиксированы. Более того, мы можем наблюдать, что KSL дает гораздо большую ошибку, чем другие методы, даже если TT ранги установлены в правильные значения. Когда мы уменьшаем временной шаг, вычислительная сложность KSL растет (226 секунд для $N_t = 50$ шагов) и превышает затраты метода AMEn, но ошибка в KSL решении останавливается на неулучшаемом уровне $8 \cdot 10^{-4}$.

Дополнительно может быть исследовано поведение AMEn в зависимости от TT ранга приближения невязки, который является специфическим параметром. На рис. 5.3, мы отслеживаем сходимость симметризованного метода $\mathrm{AMEn}_{\mathrm{svd}}$, для которого мы имеем теоретический анализ, и наиболее быстрого несимметричного метода $\mathrm{AMEn}_{\mathrm{als}}$ в зависимости от ранга невязки $\boldsymbol{\rho} = (\rho, \ldots, \rho)$. Мы можем заметить, что, действительно, слишком большие TT ранги не требуется; более того, небольшие промежуточные значения оказываются оптимальными в соотношении точность-сложность, например $\rho \sim 5$. Это составляет принципиальную разницу с классическими итерационными методами с TT арифметикой (см. алгоритм 7), которые требуют довольно точных высокоранговых приближений невязки и Крыловских векторов, в противном случае может произойти стагнация в соответствии с утверждением 4.1.3. Методы AMEn, наоборот, надежно работают с приближениями невязки с низкими рангами, а использование расширения высокого ранга может в итоге оказаться даже медленнее, чем аппроксимация рангом 1.

Поведение метода TT-GMRES в применении к задаче $B\psi = f$ можно видеть в таблице 5.2. Алгоритм 7 демонстрирует устойчивую сходимость до запрошенного уровня ошибки. Однако тензорные ранги Крыловских векторов быстро растут с числом итераций, и даже небольшое количество лишних шагов может потребовать значительного процессорного времени. В частности, мы не смогли провести расчеты с методом TT-GMRES до точности $\varepsilon = 10^{-6}$, использованной в AMEn алгоритме, из-за ограничений по памяти.

Сравнение с корректированным одноблочным DMRG (поскольку сравнение было проведено для задачи на собственные значения, оно не приводится в данной работе; см. [58] для подробной информации) показывает, что метод AMEn гораздо менее чувствителен к выбору ранга расширения, по сравнению с выбором веса в корректированном DMRG. AMEn сходится для любого ранга расширения (в худшем случае за неоптимальное время), в то время как корректированный DMRG может значительно терять точность, если вес выбирается неправильно.

Таблица 5.2: Процессорное время в секундах (time), число итераций (it), максимальный TT ранг Крыловских векторов $r(w)$, и относительная погрешность в решении алгоритма TT-GMRES (err) в зависимости от порога аппроксимации и остановки ε.

ε	time	it	$r(w)$	err
10^{-2}	50.456	17	40	2.45e-02
10^{-3}	776.54	27	140	2.62e-03
10^{-4}	9514.4	37	260	2.37e-04
10^{-6}			***	

Рис. 5.2: ОКУ для каскада реакций, относительная ошибка во Фробениусовой норме (сверху) и невязка (снизу) в различных методах в зависимости от номера итерации (слева) и вычислительного времени в секундах (справа). Непрерывная линия: метод применяется непосредственно к несимметричной системе $B\psi = f$, пунктирная линия: метод применяется к симметризованной системе $B^*B\psi = B^*f$. TT ранг приближенной невязки ограничен $\rho = 4$. Решение приближается с относительным порогом во Фробениусовой норме $\varepsilon = 10^{-6}$.

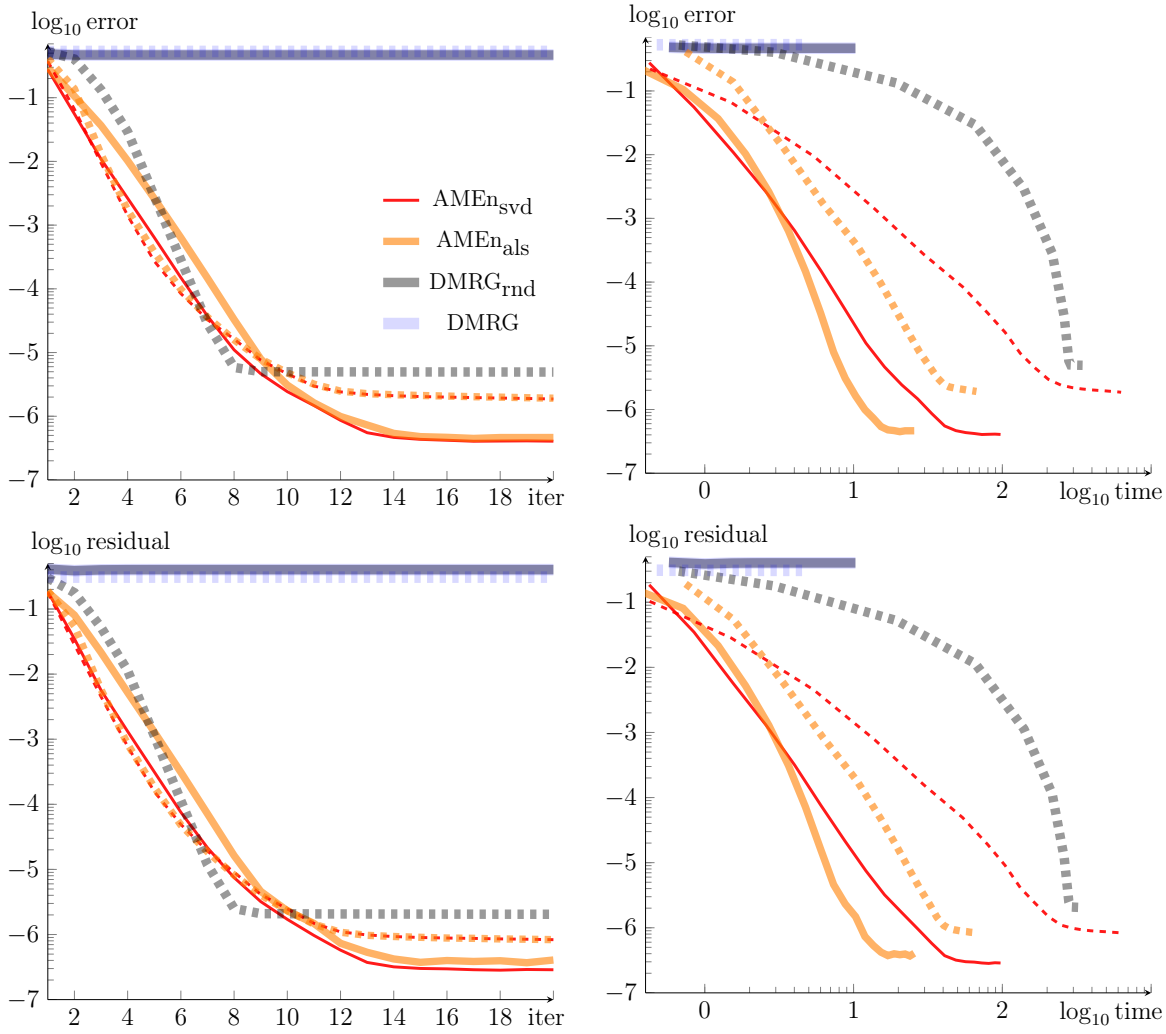
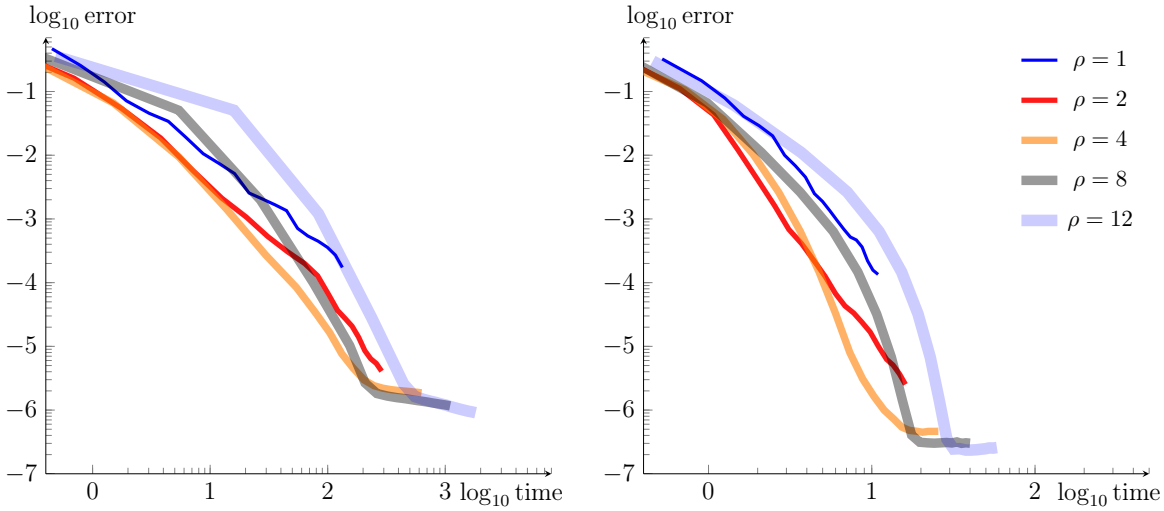


Рис. 5.3: ОКУ для каскада реакций, ошибка во Фробениусовой норме в зависимости от расчетного времени в секундах и TT ранга невязки ρ. Слева: $\mathrm{AMEn}_{\mathrm{svd}}$ примененный к $B^*B\psi = B^*f$, справа: $\mathrm{AMEn}_{\mathrm{als}}$ примененный к $B\psi = f$.



5.1.2 Каскад реакций на большом промежутке времени

В предыдущем примере мы подтвердили эффективность алгоритма AMEn для нетривиальной многомерной задачи. Однако, интервал времени $T = 10$ слишком мал, чтобы вычислить стационарное решение ОКУ. В этом и следующих экспериментах (разделы 5.1.2, 5.1.3 и 5.1.4), мы покажем три примера динамической эволюции на длительном времени (так что решение с высокой точностью сходится к стационарному состоянию) из [51], которые являются более актуальными для практики.

В первом тесте, мы продлим моделирование каскада реакций до времени $\hat{T} = 400$. Параметры модели такие же, как и в предыдущей секции. Помимо интервала по времени, пара заметных отличий состоят в следующем.

- ОДУ решается с помощью одновременной схемы пространственно-временной дискретизации из раздела 1.3 с рестартами, см. замечание 1.3.3. Мы проводим дополнительной тест для определения оптимальной длины интервала T.
- Порог тензорных аппроксимаций $\varepsilon = 10^{-5}$.

В качестве выходной величины (см. рис. 5.4, слева), мы вычисляем средние числа копий всех веществ во времени,

$$\langle i_k \rangle(t) = \frac{\sum_{\boldsymbol{i}} i_k \psi(\boldsymbol{i}, t)}{\sum_{\boldsymbol{i}} \psi(\boldsymbol{i}, t)} = \frac{(\mathtt{i}_k, \psi(t))}{(\mathtt{e}, \psi(t))}, \quad k = 1, \ldots, d, \tag{5.3}$$

где $e = (1, \ldots, 1) \in \mathbb{R}^n$, $\mathtt{e} = e \otimes \cdots \otimes e$ это тензор из всех единиц, и $\mathtt{i}_k = e \otimes \cdots \otimes e \otimes \{i_k\} \otimes e \otimes \cdots \otimes e$ является тензором, заполненным всеми значениями i_k. Скалярные произведения в TT формате в (5.3) легко вычисляются алгоритмом 1.

Одной из интересных особенностей каскадной системы является временной сдвиг между равными уровнями концентраций различных веществ, который можно наблюдать на рис. 5.4 (слева). Достаточно подробная история эволюции решения во времени важна для измерения таких сдвигов.

Кроме того, сходимость решения к стационарному состоянию показана на рис. 5.4 (справа), что подтверждает, что выбранный интервал $\hat{T}$ выбран достаточно большим, чтобы приблизить стационарное решение с удовлетворительным уровнем точности.

Чтобы продемонстрировать эффективность QTT схемы дискретизации во времени, сравним общие расчетные времена при различных числах временных шагов N_t в каждом подынтервале $[(q-1)T, qT]$, $q = 1, \ldots, \hat{T}/T$ (рис. 5.5, слева), и длинах интервалов T (рис. 5.5, справа).

Мы видим, что QTT формат действительно дает логарифмический рост вычислительного времени с числом временных шагов. Оптимальная длина временных интервалов,при которой решение вычисляется наиболее быстро, приходится на промежуточное значение $T = 10$. При меньших T, решение на каждом интервале вычисляется быстро, но количество интервалов требуется большое. Наоборот, при больших T числа обусловленности матриц и TT ранги решений в линейных системах высоки, поэтому для получения решения требуется больше времени.

Рис. 5.4: ОКУ для каскада реакций. Слева: средние числа копий $\langle i_k \rangle(t)$. Справа: невязка $\|A\psi(t)\|/\|\psi(t)\|$

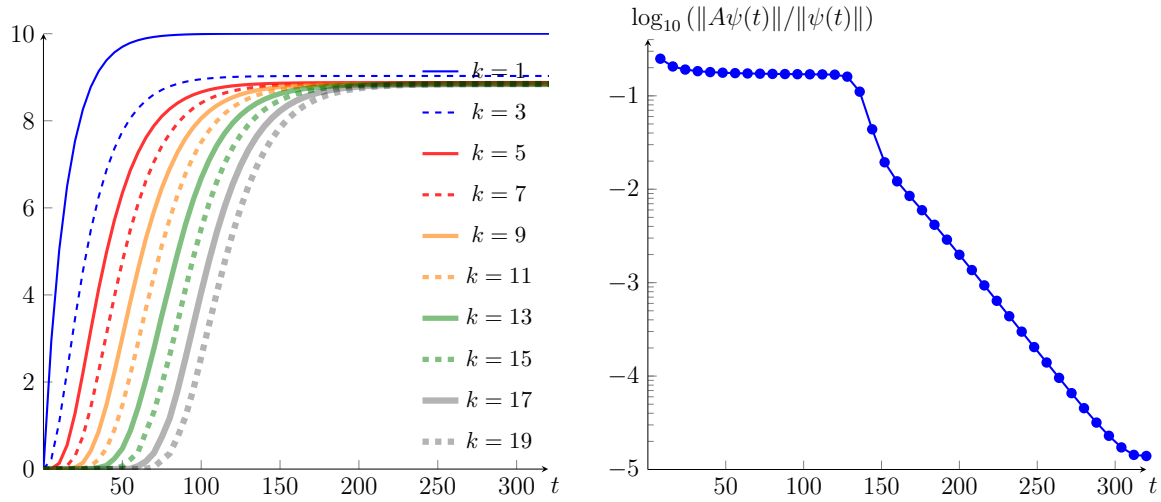


Рис. 5.5: ОКУ для каскада реакций, расчетное время (секунды) в зависимости от параметров дискретизации. Слева: время в зависимости от $\log_2(N_t)$ при $T = 15$. Справа: время в зависимости от T при $N_t = 2^{14}$.

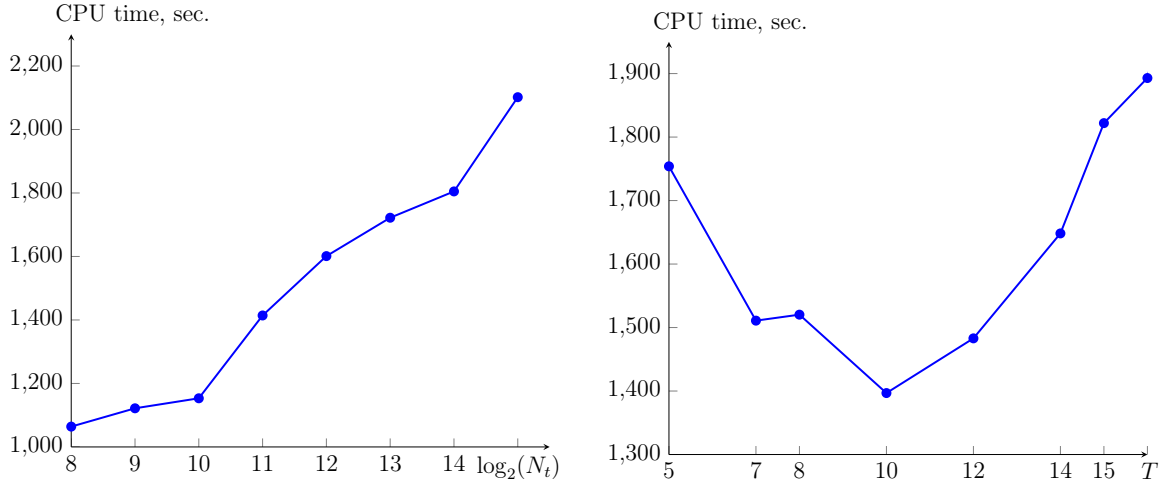
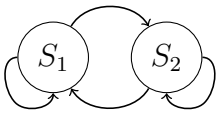


Рис. 5.6: Генетический переключатель



5.1.3 Генетический переключатель с параметром

В этом тесте мы моделируем синтетический генетический переключатель с двумя метастабильными состояниями (см. рис 5.6), выращенный в *Escherichia coli* [76], с коэффициентами модели, зависящими от параметра. Модель содержит $d = 2$ генов и $M = 4$ следующих реакций.

$w^1(\boldsymbol{i}) = \dfrac{\alpha_1}{1+i_2^\beta}$,	$\boldsymbol{z}^1 = (1,0)$:	генерация S_1;	$\alpha_1 = 156.25$, $\beta = 2.5$.
$w^2(\boldsymbol{i}) = i_1$,	$\boldsymbol{z}^2 = (-1,0)$:	разрушение S_1.	
$w^3(\boldsymbol{i}) = \dfrac{\alpha_2}{1+\dfrac{i_1}{(1+y/K)^\eta}}$,	$\boldsymbol{z}^3 = (0,1)$:	генерация S_2;	$\alpha_2 = 15.6$, $\eta = 2.0015$, $K = 2.9618 \cdot 10^{-5}$.
$w^4(\boldsymbol{i}) = i_2$,	$\boldsymbol{z}^4 = (0,-1)$:	разрушение S_2.	

В соответствии с самой большой скоростью реакции w^1, мы ограничиваем пространство состояний до квадрата $n_1 = n_2 = n = 256$. Параметр y задает концентрацию катализатора IPTG, и варьируется от 10^{-6} до 10^{-2}. Главной особенностью этой системы является наличие двух метастабильных состояний: так называемого *низкого* (низкое количество молекул i_2) и, в обратном случае, *высокого*. Вероятность найти систему том или ином состоянии зависит от концентрации y, см. рис. 5.8.

Отметим, что y не регулируется ОКУ, но только входит в его коэффициенты в качестве параметра, $w^m = w^m(\boldsymbol{i}, y)$, т.е. задачу можно поставить так: решить

$$\frac{d\psi(y,t)}{dt} = A(y)\psi(y,t) = \sum_{m=1}^{M} (\boldsymbol{J}^{\boldsymbol{z}^m} - \boldsymbol{J}^0)\,\mathrm{diag}(w^m(y))\psi(y,t) \quad \text{для всех } y.$$

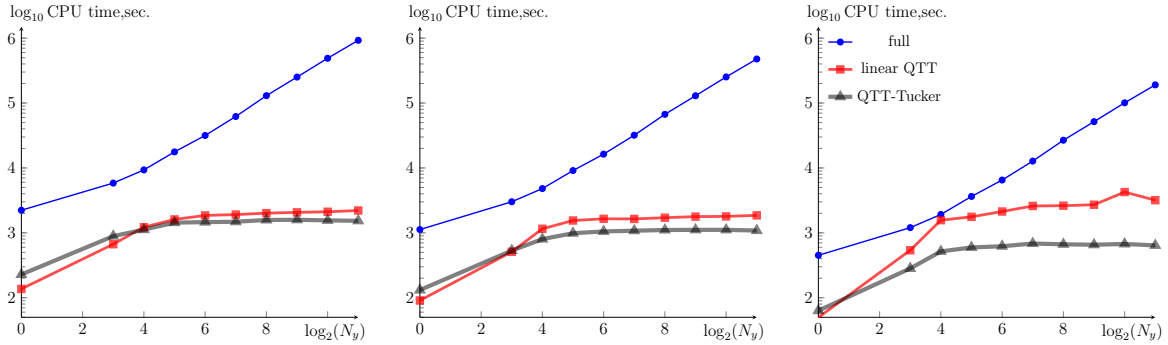
Вводя дискретизацию по параметру (в данном примере используется коллокация на экспоненциально-равномерной сетке), мы приходим к блочно-диагональной линейной системе:

$$\frac{\partial\psi(t)}{\partial t} = \mathcal{A}\psi(t) = \begin{bmatrix} A(y_1) & & \\ & \ddots & \\ & & A(y_{N_y}) \end{bmatrix} \psi(t), \qquad \psi(t) = \begin{bmatrix} \psi(y_1,t) \\ \vdots \\ \psi(y_{N_y},t) \end{bmatrix},$$

где $\{y_1, \ldots, y_{N_y}\}$ задает сетку значений параметра, а $A(y_j)$ отвечает исходной матрице в ОКУ (1.17) при фиксированном $y = y_j$, $j = 1, \ldots, N_y$.

Если общее количество параметрических точек N_y невелико, простейший подход состоит в раздельном решении независимых ОКУ для каждого значения y_j. Однако если N_y большое (y может представлять собой на самом деле вектор из *многих* параметров длины d_y, в результате чего мы получаем в итоге $N_y = n^{d_y}$

Рис. 5.7: Генетический переключатель, расчетное время в зависимости от $\log_2(N_y)$. Слева: $\delta t = 1$, в середине: $\delta t = 2$, справа: $\delta t = 5$.



степеней свободы), может иметь смысл воспользоваться тензорным сжатием данных и решать сразу глобальную систему, без учета ее диагональности. Случаи многих параметров естественно возникают в стохастических уравнениях (см. например, [176, 66, 137]), когда некоторые коэффициенты не могут быть заранее известны точно, а указаны только их допустимые диапазоны, или обратных задачах, таких как калибровка моделей и анализ чувствительности [229, 209]. Данный пример можно отнести к последнему классу, хотя с вычислительной точки зрения разница несущественная.

В противоположность предыдущему примеру каскадной сети, моделирование переключателя может быть проведено в полном формате без разделения переменных, так как сравнительно небольшой размер задачи $n^2 N_y$ это позволяет. Из-за диагональности матрицы системы по y, и дополнительно разреженности по i_1, i_2, прямые алгоритмы для неявных схем интегрирования по времени работают весьма эффективно. Поэтому особенно интересно сравнить их с методами тензорных произведений.

Мы ищем стационарное решение с использованием итераций Эйлера (см. раздел 1.3.2) во временном интервале $\hat{T} = 1000$, что обеспечивает падение нормы невязки ниже порога тензорного округления $\varepsilon = 10^{-5}$. Шаг по времени δt варьируется от 1 до 5.

В первом тесте, мы отслеживаем вычислительные времена для различных размеров параметрической сетки и временных шагов, см. рис. 5.7. Как и ожидалось, процессорное время в схеме с полным форматом демонстрирует линейный рост с N_y. Для обоих тензорных форматов на основе QTT сложность является логарифмической. Вообще говоря, не самые низкие TT ранги решения (до 40) могли бы дать большой вклад в вычислительную сложность. На удивление, тензорный метод (AMEn_{als} в применении к неявной системе Эйлера (1.28)) и в QTT, и в QTT-Tucker форматах оказывается быстрее схемы в полном формате даже для одной параметрической точки, то есть на двумерной задаче размера 256^2.

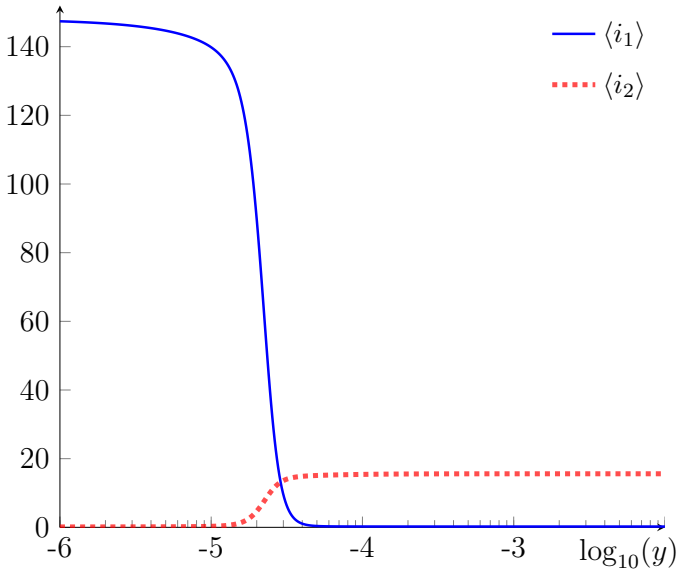
С увеличением δt и N_y, так же растут и обусловленность матрицы системы, и TT ранги решения. В этом случае, QTT-Tucker формат обеспечивает дополнительное ускорение примерно в 4 раза по сравнению с линейным QTT представлением.

Поскольку тензорное округление вносит возмущение порядка $\varepsilon = 10^{-5}$, нуж-

Таблица 5.3: Генетический переключатель, средние и максимальные относительные погрешности (5.4) при $y = 3 \cdot 10^{-5}$

	линейный QTT		QTT-Tucker	
	$\mathrm{E}(i_1)$	$\mathrm{E}(i_2)$	$\mathrm{E}(i_1)$	$\mathrm{E}(i_2)$
сред.	8.8e-4	7.8e-5	5.5e-4	4.9e-5
макс.	2.5e-3	2.2e-4	2.1e-3	1.8e-4
$\delta t^{\max}$	1	1	1	1
$N_y^{\max}$	2^7	2^7	2^6	2^6

Рис. 5.8: Генетический переключатель, средние числа копий $\langle i_1 \rangle$, $\langle i_2 \rangle$ в зависимости от y



но проверить фактические ошибки в наблюдаемых величинах. Мы отслеживаем точку $y = 3 \cdot 10^{-5}$ (она находится в интересной переходной области, см. рис. 5.8), добавляя ее в сетку явно, и проверяем, как ошибка во всем решении влияет на средние количества молекул в точке y, то есть вычисляем

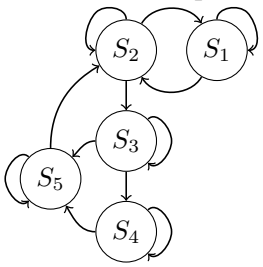
$$\mathrm{E}(i_k) = |\langle i_k \rangle - \langle i_k \rangle_{ex}| / \langle i_k \rangle_{ex}, \quad k = 1, 2, \tag{5.4}$$

где $\langle i_k \rangle_{ex}$ берется из моделирования в полном формате, см. таблицу 5.3. Для краткости, мы приводим только средние и максимальные ошибки при разных N_y и δt. Мы видим, что рост погрешности может быть довольно большим (на фактор до 250), но тем не менее, уровень точности $\mathcal{O}(10^{-3})$, как правило, достаточный для феноменологических моделей, поддерживается во всех экспериментах, в то время как сложность может быть значительно уменьшена.

Наконец, мы рассматриваем средние числа копий обоих реагирующих генов в зависимости от концентрации катализатора, см. рис. 5.8. Величину $\langle i_2 \rangle$ можно также использовать в качестве меры доли клеток, находящихся в высоком (или, наоборот, низком) состоянии. В самом деле, $\langle i_2 \rangle$ демонстрирует асимптотику: при стремлении y к бесконечности, клетки стремятся оставаться в высоком состоянии.

Таким образом, долю клеток в высоком состоянии можно оценить как нормированную величину $\langle i_2 \rangle$. Можно заметить хорошее сходство с экспериментальными результатами, приведенными в [76, рис. 5(b)].

Рис. 5.9: λ-фаг



5.1.4 λ-фаг

В последнем примере мы моделируем жизненный цикл бактериофага λ [227, 123]. В первой работе [227] использовался подход разреженных сеток. Вторая из них более связана с аппроксимациями тензорными произведениями, и использует принцип Дирака-Френкеля для динамического приближения низкого ранга в формате Таккера (Dynamical Low Rank Approximation, DLRA). Моделирование проводилось на относительно небольшой промежуток времени ($T = 10$), который слишком мал для достижения стационарного решения. Численное решение осложняется тем, что и число копий второго белка, и время релаксации весьма велики ($\sim 10^4$). С помощью QTT формата вместе с методом AMEn для решения линейных систем, мы смогли провести эффективный расчет стационарного решения на очень больших сетках.

Модель содержит $d = 5$ видов веществ и $M = 10$ реакций, определенных следующим образом ($\boldsymbol{e}_1, \ldots, \boldsymbol{e}_5$ обозначают единичные векторы).

$$
\begin{array}{llll}
w^1(\boldsymbol{i}) = \dfrac{a_1 b_1}{b_1 + i_2}, & \boldsymbol{z}^1 = \boldsymbol{e}_1: & \text{генерация } S_1; & a_1 = 0.5,\ b_1 = 0.12. \\
w^2(\boldsymbol{i}) = c_1 \cdot i_1, & \boldsymbol{z}^2 = -\boldsymbol{e}_1: & \text{разрушение } S_1; & c_1 = 0.0025. \\
w^3(\boldsymbol{i}) = \dfrac{(a_2 + i_5) b_2}{b_2 + i_1}, & \boldsymbol{z}^3 = \boldsymbol{e}_2: & \text{генерация } S_2; & a_2 = 1,\ b_2 = 0.6. \\
w^4(\boldsymbol{i}) = c_2 \cdot i_2, & \boldsymbol{z}^4 = -\boldsymbol{e}_2: & \text{разрушение } S_2; & c_2 = 0.0007. \\
w^5(\boldsymbol{i}) = \dfrac{a_3 b_3 i_2}{b_3 \cdot i_2 + 1}, & \boldsymbol{z}^5 = \boldsymbol{e}_3: & \text{генерация } S_3; & a_3 = 0.15,\ b_3 = 1. \\
w^6(\boldsymbol{i}) = c_3 \cdot i_3, & \boldsymbol{z}^6 = -\boldsymbol{e}_3: & \text{разрушение } S_3; & c_3 = 0.0231. \\
w^7(\boldsymbol{i}) = \dfrac{a_4 b_4 i_3}{b_4 \cdot i_3 + 1}, & \boldsymbol{z}^7 = \boldsymbol{e}_4: & \text{генерация } S_4; & a_4 = 0.3,\ b_4 = 1. \\
w^8(\boldsymbol{i}) = c_4 \cdot i_4, & \boldsymbol{z}^8 = -\boldsymbol{e}_4: & \text{разрушение } S_4; & c_4 = 0.01. \\
w^9(\boldsymbol{i}) = \dfrac{a_5 b_5 i_3}{b_5 \cdot i_3 + 1}, & \boldsymbol{z}^9 = \boldsymbol{e}_5: & \text{генерация } S_5; & a_5 = 0.3,\ b_5 = 1. \\
w^{10}(\boldsymbol{i}) = c_5 \cdot i_5, & \boldsymbol{z}^{10} = -\boldsymbol{e}_5: & \text{разрушение } S_5; & c_5 = 0.01.
\end{array}
$$

В первом тесте, мы исследуем короткий диапазон времени, $\hat{T} = 10$, и сравниваем различные методы. Пространственная сетка $16 \times 64 \times 16 \times 16 \times 16$ дает достаточную точность FSP ограничения на данном интервале времени. Начальное состояние ОКУ выбирается в соответствии с [123] в виде следующего полиномиального распределения:

$$\psi(\boldsymbol{i}, 0) = \frac{3!}{i_1! \cdots i_5! \cdot (3 - |i|)!} 0.05^{|i|} (1 - 5 \cdot 0.05)^{3 - |i|} \cdot \theta(3 - |i|),$$

где $|i| = i_1 + \cdots + i_5$, и $\theta(s)$ это функция Хевисайда. Это распределение может быть построено в качестве тензора $4 \times 4 \times 4 \times 4 \times 4$ в полном формате, так как функция Хевисайда равна нулю, если любой из индексов i_k больше 3. После этого, TT разложение (с рангами 4) вычисляется с использованием алгоритма сжатия 4, и каждый фактор расширяется нулями до нужного размера сетки (что уже является одномерной операцией). Наконец, представление TT переаппроксимируется в QTT формат.

Рис. 5.11, 5.10 и таблица 5.4 показывают поведение четырех методов. Первый предложенный метод это AMEn алгоритм 12, примененный к пространственно-временной схеме (1.22). Порог тензорного приближения зафиксирован в $\varepsilon = 10^{-5}$, но мы меняем интервал во времени T и количество внутренних временных шагов N_t.

Второй подход это алгоритм DLRA (мы ссылаемся на результаты, представленные в [123]), и схема KSL, уже рассмотренная в секции 5.1.1. Поскольку схема KSL не позволяет адаптировать тензорные ранги, мы выбираем их априори по следующей стратегии: QTT блоки начального состояния ОКУ дополняются нулями до выбранного значения ранга (например, $r = 20$ на рис. 5.11). Эталонное решение вычисляется в стандартном векторном формате (без аппроксимаций), с использованием схемы Кранка-Николсон с временным шагом $\delta t = 10/4096$. Матрицы перехода размера 2^{22} хранятся в Matlab в разреженном `sparse` формате, и для неявного шага Кранка-Николсон используется метод минимальных невязок.

Качественную правильность решения можно видеть из частичных распределений, показанных на рис. 5.10. Мы видим, что графики совпадают с ранее полученными результатами, например, в [123].

Ошибки AMEn и KSL решений во Фробениусовой норме по сравнению с эталонным решением, а также процессорные времена представлены на рис. 5.11. Горизонтальная ось на рис. 5.11 показывает эффективный шаг по времени δt. Заметим, что одно и то же значение δt может соответствовать разным параметрам, в силу формулы $\delta t = T/N_t$ в (1.22). Например, график, лежащий левее и ниже всех на рис. 5.11, разделен на две части: в области сплошной линии, мы фиксируем $N_t = 64$ и меняем $T \in \{1.25, 2.5, 5, 10\}$, а в пунктирной части зафиксирован $T = 1.25$ но N_t меняется от 2^6 до 2^{11} (обе части относятся к методу AMEn). По области достаточно большого временного шага δt мы можем построить линейную регрессию и убедиться, что имеет место второй порядок аппроксимации схемы Кранка-Николсон. На меньших δt, ошибка падает до уровня тензорного округления $\mathcal{O}(\varepsilon)$. Такой же эффект наблюдается и при других параметрах (см. $T = 5$). Уровень ошибки при $T = 5$ несколько выше, поскольку линейные системы имеют худшую обусловленность.

Ошибка в схеме KSL зависит главным образом от тензорных рангов решения, а не временного шага. Хотя начальное состояние точно представимо в QTT формате с рангами 4, это становится неверно в конце временного промежутка ($t = 10$). Из рис. 5.11 и таблицы 5.4 можно видеть, что даже если мы увеличим TT ранги до 20, уровень ошибки остается выше, чем в методе AMEn.

Если мы посмотрим на вычислительные сложности, в правой части рис. 5.11 мы наблюдаем логарифмическую зависимость процессорного времени в схеме AMEn

Рис. 5.10: λ-фаг, частичные распределения в $t = 10$.

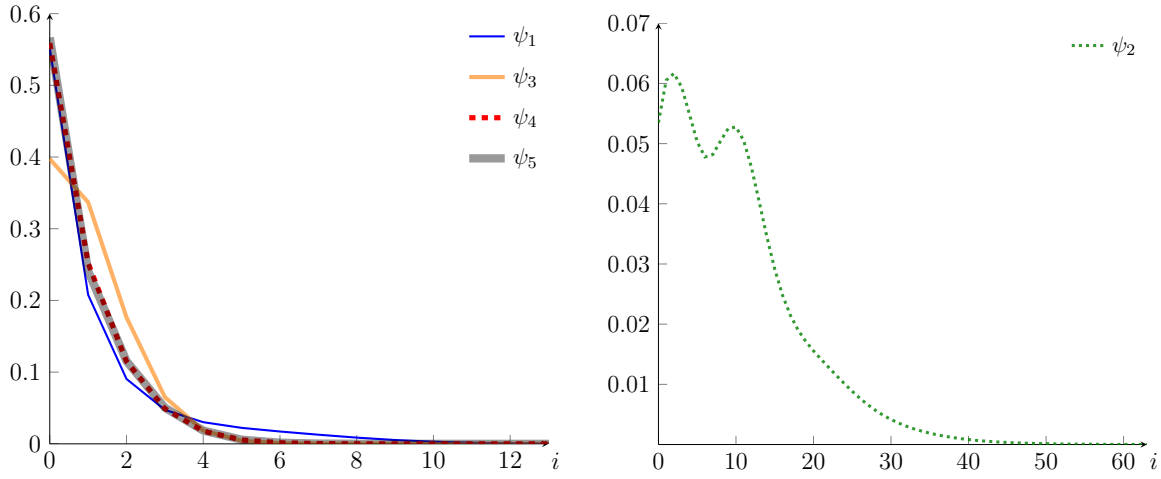


Таблица 5.4: λ-фаг, $\hat{T} = 10$. Процессорное время (сек.) и ошибки в различных методах.

	full	KSL, $\delta t = \frac{10}{128}$			DLRA	AMEn	
	$\delta t = \frac{10}{4096}$	$r = 4$	$r = 8$	$r = 20$	из [123]	$\delta t = \frac{10}{64}$	$\delta t = \frac{1.25}{2048}$
время	6840	21.7	24.6	37.6	~ 300	22.9	32.0
$\frac{\|\psi - \psi_{full}\|}{\|\psi_{full}\|}$		1.59e-1	1.92e-2	3.92e-4	$\sim$ 3e-3	9.14e-4	2.74e-5

от временного шага. Число операций в методе KSL растет линейно с увеличением числа временных шагов, что также подтверждается рис. 5.11.

Наконец, отметим, что вычисления в полном представлении ($\sim$ 2 часа в соответствии с таблицей 5.4), так же как и методом SSA ($\sim$ 3 часа в соответствии с [123]) выполняются заметно медленнее. Это показывает очевидное преимущество метода AMEn даже для такой относительно небольшой задачи.

Для сходимости эволюции к стационарному состоянию требуется гораздо большее время, $\hat{T} = \exp(10) \sim 2 \cdot 10^4$. Более того, поскольку число молекул второго вещества достигает $4 \cdot 10^4$ (см. рис. 5.12, справа), используется FSP область с размерами $128 \times 65536 \times 64 \times 64 \times 64$, что делает задачу особенно сложной. Временной интервал разбит посредством экспоненциальной сетки,

$$t_q = \exp(0.05 \cdot q), \quad q = 1, \ldots, 200.$$

В каждом подинтервале $[t_{q-1}, t_q]$ мы решаем глобальную временную задачу (1.22) в QTT формате с числом временных шагов $N_t = 1024$. История сходимости и итоговое процессорное время показаны на рисунке 5.12 (слева) и в таблице 5.5 (слева). Мы видим, что требуется около часа расчетного времени, чтобы вычислить всю динамику системы до уровня невязки 10^{-7}. Несмотря на большие сетки, в этом случае формат QTT-Tucker не дает более быстрого расчета по сравнению с линейным QTT из-за больших TT рангов в ядре Таккера, которые доминируют в кубической асимптотике сложности QTT-Tucker формата. Так, схема на основе QTT-Tucker требует около 4000 секунд.

Рис. 5.11: λ-фаг, $\hat{T} = 10$. Ошибка $\dfrac{\|\psi_{qtt} - \psi_{full}\|}{\|\psi_{full}\|}$ (слева) и процессорное время (справа) в зависимости от временного интервала δt.

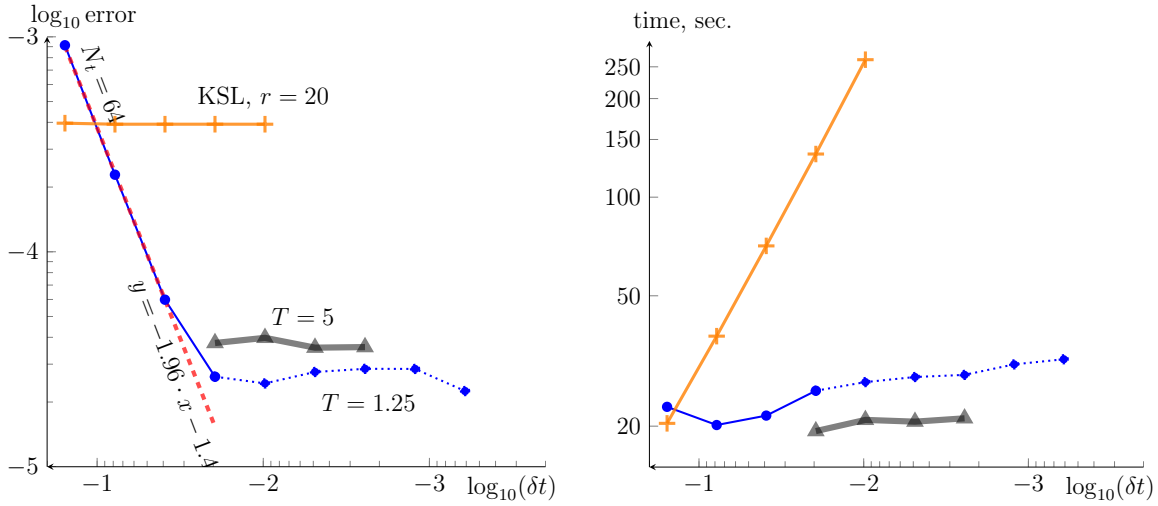


Таблица 5.5: λ-фаг, большой интервал времени $\hat{T} = \exp(10)$. Слева: расчетные времена (сек.). Справа: точности стационарных средних количеств молекул

	linear QTT		QTT-Tucker	
N_t	2^{10}	1	2^{10}	1
time	3500	670	4000	410

S_1	S_2	S_3	S_4	S_5
4.6e-5	1.7e-6	6.2e-8	3.1e-7	1.7e-7.

Если мы не заинтересованы в исследовании переходных процессов, мы можем использовать неявные итерации Эйлера (1.28) на временной сетке t_q. Суммарные расчетные времена приведены в таблице 5.5 (слева), и относительные точности средних чисел копий в конечной точке времени, по сравнению с более мелкой сеткой по времени и схемой Кранка-Николсон, приведены в таблице 5.5 справа. Для этого моделирования, формат QTT-Tucker оказывается предпочтительным.

Насколько нам известно, это первый пример прямого моделирования многомерного основного кинетического уравнения с существенно различными временными и пространственными масштабами. Подводя итог, можно утверждать, что представленные примеры открывают многообещающий потенциал методов тензорных приближений для быстрого и высокоточного моделирования в системной биологии.

Рис. 5.12: λ-фаг. Слева: невязка $\|A\psi(t)\|/\|\psi(t)\|$ и итоговое расчетное время (сек.). Справа: средние числа копий, $\log_{10}\langle i_k\rangle(t)$

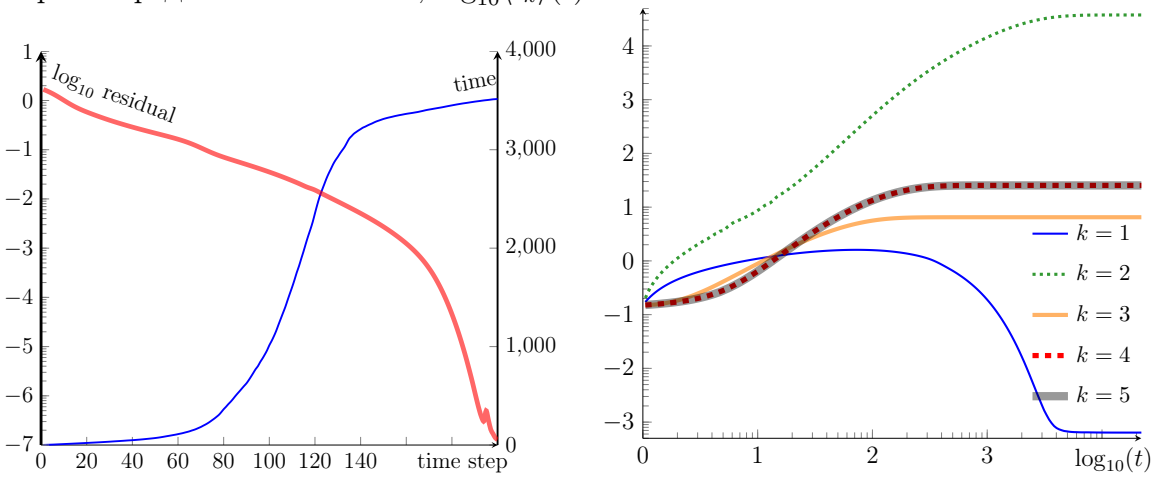


Таблица 5.6: Параметры моделирования, соответствующие модифицированной электронной массе

Физические параметры				Параметры дискретизации	
T_i	300·1.3806505e-23	[Дж]		L	50
T_e	1			v_{max}	6
E_0	0.05	[В/м]		n_x	250
B_0	5e-5	[Т]		n_v	31
n_0	1e+10	[м$^{-3}$]		δt	0.01
m_i	4.9936722e-26	[кг]		N_{ext}	40
ν_{in}	1800	[1/с]		ε	$0.05 \cdot \frac{\lvert\psi(t)-\psi(t-\delta t)\rvert}{\lvert\psi(t-\delta t)\rvert}$
Масштабные величины				Физические константы	
γ	0.1575			e	1.60217653e-19 [К]
θ	0.03528			ε_0	8.85418781e-12 $\left[\frac{\Phi}{\text{м}}\right]$
l	0.16	[м]		m_e	3.97950489e-29 [кг]

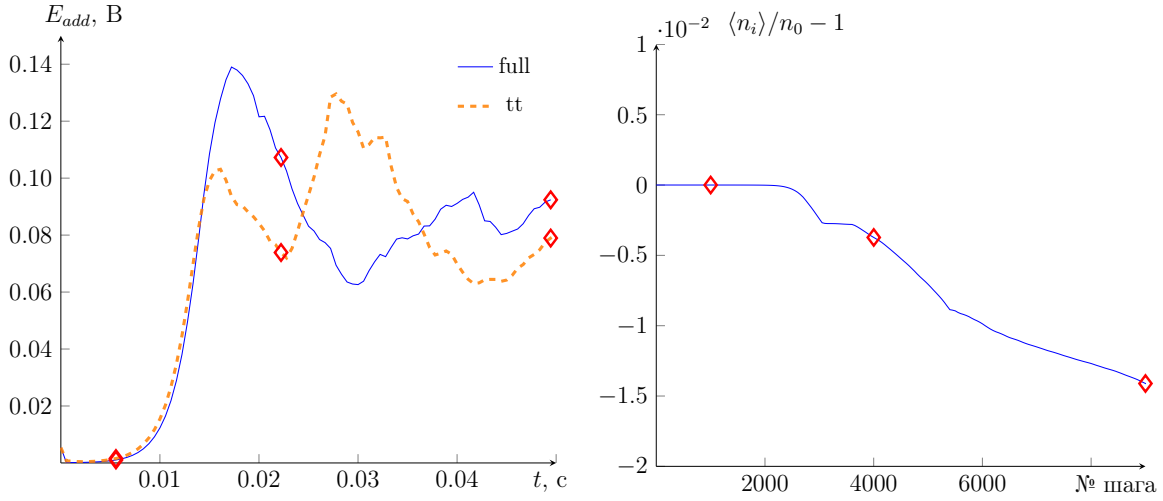
5.2 Моделирование Фарлей-Бунемановской неустойчивости

В отличие от предыдущих работ [158, 5], мы реализовали вычислительный программный код в MATLAB, и проводили эксперименты в последовательном режиме на одном ядре процессора Intel Xeon E5504, поскольку механизмы распараллеливания в MATLAB далеки от оптимальных. В частности, мы сравнивали версию численной схемы в TT формате с полным (*full*) форматом, при котором ионное распределение ψ хранится как обычный четырехмерный массив. Поскольку объем памяти в нашем компьютере составляет 70Gb, мы были вынуждены ограничится сравнительно грубыми сетками (см. Таблицу 5.6), для того чтобы полное решение поместилось в память. В свою очередь, чтобы уменьшить влияние доминирующей конвекции на таких сетках, мы модифицируем массу электрона в соответствии с [155, 5]. Это позволяет ускорить вычисления, не затрагивая качественного поведения системы, и следовательно сфокусироваться на особенностях тензорных аппроксимаций а не сложностях турбулентной природы движения электронов. В конце данной главы мы обсудим возможные пути решения этой проблемы. Модельные параметры приведены в Таблице 5.6. Обратите внимание, что параметры даны уже в обезразмеренных в соответствии с таблицей 1.1 величинах.

При безразмерном шаге по времени $\delta t = 0.01$ (это удовлетворяет условиям Куранта в Утверждении 1.1.1), мы проводим 9000 шагов, что соответствует размерному времени $t = 0.05$ с. Благодаря качественной природе модели (BGK оператор столкновений, модифицированная электронная масса), мы можем допустить довольно высокий порог тензорного округления ε, а именно, мы требуем аппроксимации изменения ψ на каждом временном шаге с 5% точностью (см. таблицу 5.6). Таким образом, для выходных величин можно ожидать итоговую точность около 10%.

Рассмотрим усредненное дополнительное поле E_{add} (рис. 5.13, слева), и сред-

Рис. 5.13: Дополнительное электрическое поле при моделировании в полном (full) и TT форматах (слева). Потеря средней концентрации ионов в TT формате (справа). Красные маркеры отмечают временные шаги: 1000, 4000 и 9000.



нюю концентрацию ионов (рис. 5.13, справа). Мы можем заключить, что уровень точности порядка 10% действительно имеет место. Так как тензорное округление является серией ортогональных проекций, норма решения в TT формате уменьшается от шага к шагу, что хорошо видно по убыванию средней концентрации ионов $\langle n_i \rangle$.

Рассматривая электрическое поле E_{add} (рис. 5.13), а также концентрации электронов (рис. 5.14) в различные моменты времени более подробно, можно увидеть, что во время начальной стадии развития процесса Фарлей-Бунемана, решения в полном формате и в TT приближении совпадают с высокой точностью (номер шага по времени $\leqslant 2000$). То же самое справедливо для временных масштабов нелинейного насыщения (таблица 5.7, последняя строка): считается, что система входит в существенно нелинейную стадию, если дополнительное электрическое поле начинает уменьшаться после линейного роста (не принимается во внимание небольшая область осцилляций поля в самом начале процесса).

Тем не менее, нелинейная система становится более чувствительной к возмущениям, возникающим из тензорных приближений, и решения развиваются значительно разными путями во время дальнейших переходных процессов (шаги по времени ~ 4000).

Наконец, полностью насыщенная нелинейная система выходит на стационарный участок (шаги по времени до 9000), где дополнительное поле колеблется вокруг своего среднего значения. Несмотря на значительно отличающиеся распределения концентраций (рис. 5.14), статистические величины гораздо менее чувствительны к ошибкам в решении, см. таблицу 5.7. Анализируя пространственные гармоники электрического поля (рис. 5.15), мы наблюдаем, что в начале процесса, спектр является почти изотропным относительно оси x, тогда как после насыщения появляются анизотропные компоненты. Таким образом, модель также правильно предсказывает поворот вектора дрейфа.

Таблица 5.7: Статистические величины по электрическому полю

	Полный формат	TT формат	Ошибка
$\frac{1}{0.05-0.03}\int\limits_{0.03}^{0.05} E_{add}(t)dt$	8.3083e-02	8.0523e-02	3.08%
$\max\limits_{0.015\leqslant t\leqslant 0.03} E_{add}(t)$	1.3903e-01	1.3200e-01	5.05%
$\min\limits_{t>0.01} t: \ \frac{dE_{add}(t)}{dt} < 0$	1.7267e-02	1.5989e-02	7.40%

Убедившись в корректности решения, даваемого нашей моделью, рассмотрим теперь вычислительную сложность (рис. 5.16). TT ранги растут в ходе развития Фарлей-Бунемановского процесса, и стабилизируется на уровне максимального значения 55 после насыщения системы (вместо рангов, на рис. 5.16 мы показываем непосредственно количество ячеек памяти, необходимых для хранения каждого $\psi(t)$). Наибольшие вычислительные затраты возникают при работе с первым TT блоком $\psi^{(1)}(i,j)$ в (3.1), который содержит $n_x^2 r$ элементов. Таким образом, $r = 55$ нужно сравнивать с $n_v^2 = 961$ в полном формате, что дает фактор редукции памяти более чем 17.

Вычислительная сложность растет с TT рангами быстрее, чем объем памяти, так что разумно ожидать менее заметного ускорения. Тем не менее, в TT формате моделирование выполняется по крайней мере в 2 раза быстрее, чем в полной модели (рис. 5.16, слева). Кроме того, каждый шаг в TT выполняется быстрее, чем в полном формате, т.е. при дальнейшей эволюции системы разница будет только накапливаться.

То же самое можно ожидать по отношению к измельчению сеток. В нашем случае, все данные, необходимые для решения в полном формате, не помещаются в памяти при $n_x \gtrsim 300,\ n_v \gtrsim 30$. Однако, так как TT ранги в большинстве случаев остаются почти постоянными с изменением размеров сетки, разделение переменных должно быть более эффективным при большем числе точек дискретизации.

Рис. 5.14: Распределения концентрации электронов в полном формате (слева) и TT формате (справа) на разных временных шагах: 1000 (сверху), 4000 (в середине) и 9000 (снизу). Величины на осях: x и y [м].

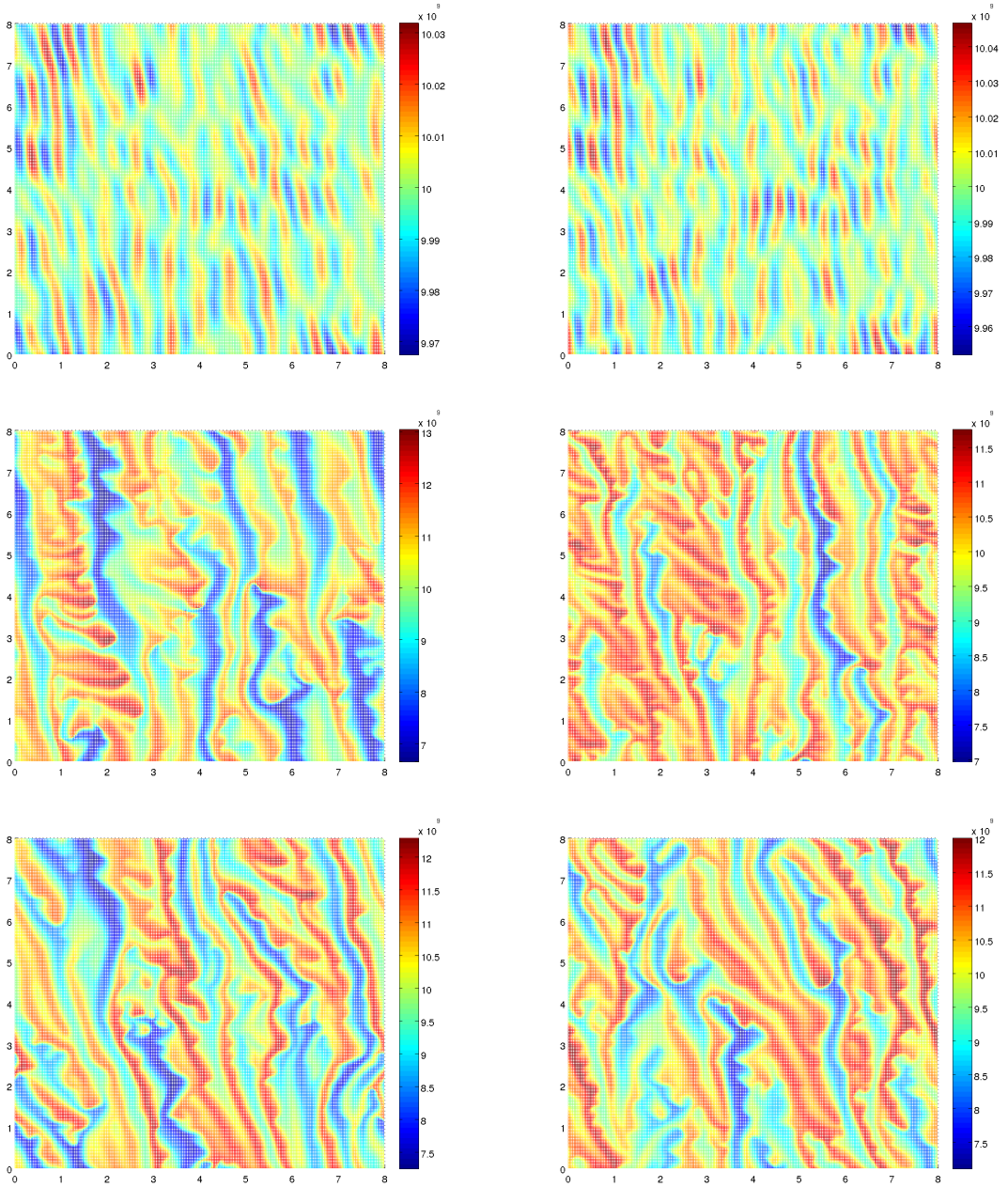


Рис. 5.15: Спектральные интенсивности электрического поля $\hat{E}^2$ в TT формате на временных шагах 1000 (слева), и 9000 (справа). Оси: k_x и k_y [1/м].

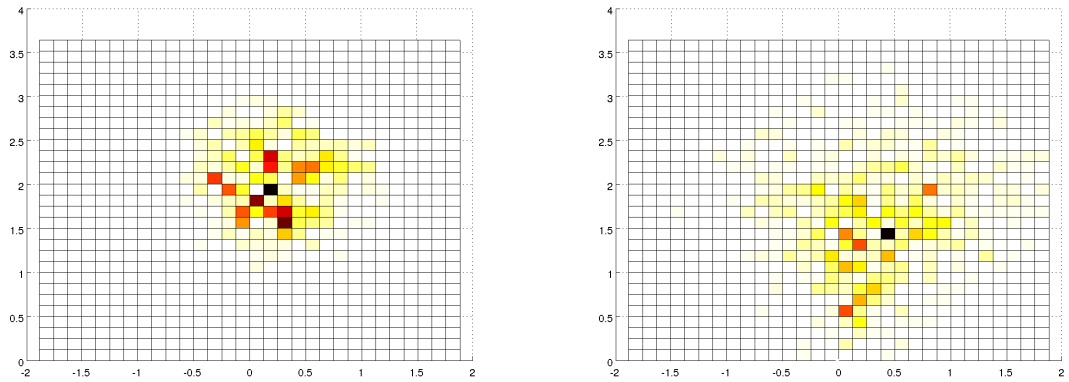
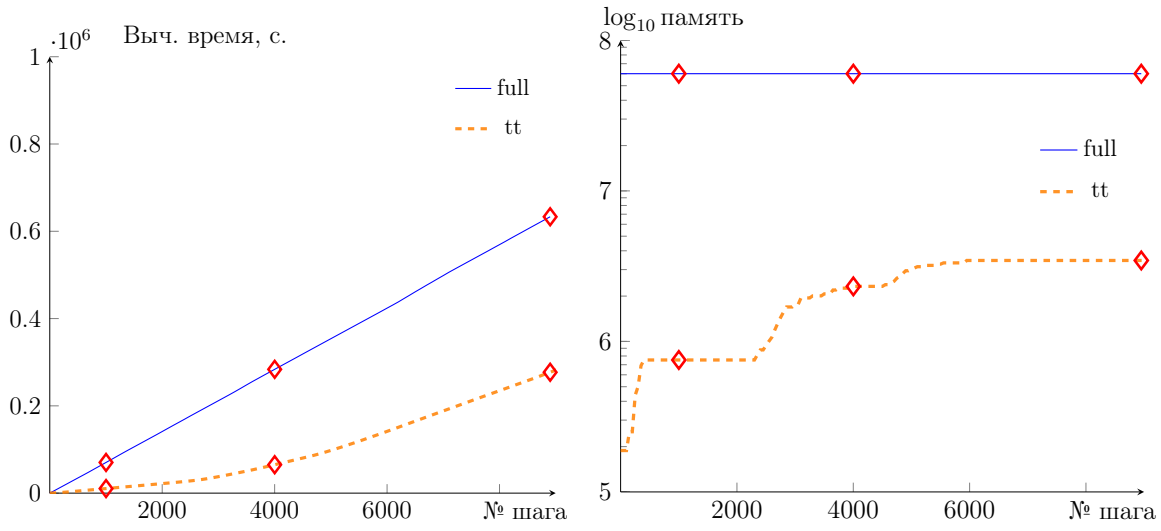


Рис. 5.16: Общее процессорное время, сек. (слева) и объем памяти для ψ (справа) в полном и TT форматах.



Заключение

В данной диссертации, мы предложили и исследовали представления и алгоритмы для аппроксимаций функций и операторов, и решения многомерных линейных систем в форматах тензорных произведений. Особенно интересные многомерные модели возникают в виде нестационарных дифференциальных уравнений для описания стохастических динамических систем и химической кинетики. В течение долгого времени, прямой расчет функции плотности вероятности уравнениями Власова, Фоккера-Планка или основным кинетическим (управляющим) избегался в связи с проклятия размерности, за исключением случаев малого числа переменных. Форматы разреженного представления данных общего вида, косвенно хранящие все n^d элементов, открывают путь к быстрым и точным методами стохастического моделирования в машиностроении, биологии, химии и физики.

Центральным результатом диссертации является новый вычислительный алгоритм (AMEn) для решения больших систем линейных уравнений и аппроксимации данных в форматах тензорных произведений, см. [56, 57, и раздел 4.3.3]. Этот метод был получен путем объединения DMRG схемы оптимизации в переменных направлениях, и классического метода градиентного спуска, и позволяет вычислять приближенные решения систем уравнений значительно быстрее и точнее, чем каждая из используемых схем в отдельности.

В частности, быстрая сходимость метода AMEn позволяет решать и системы с сильно несимметричными матрицами без изменения алгоритма, хотя его формулировка и анализ приводятся для симметричных положительно определенных матриц. Это находит особенно важное применение в решение нестационарных уравнений, где неявные схемы дискретизации по времени приводят к несимметричным задачам. Так, отдельным результатом диссертации является блочная версия стандартных схем Кранка-Николсон и Эйлера, которая, с использованием аппроксимаций тензорными произведениями, позволяет достичь логарифмической сложности в зависимости от числа временных шагов. Мы показали, что этот подход действительно эффективен для практических применений, и позволяет быстро вычислять эволюцию сложных систем с высокой точностью [51, 53, и раздел 1.3].

Мы убедились, что достижение прогресса в ускорении расчетов или повышении точности стало возможным только после усовершенствования вычислительных методов: ни DMRG алгоритмы переменных направлений, ни классические итерации сами по себе не способны справиться с предложенными задачами. Но в новых комбинированных схемах они демонстрируют замечательную кооперацию, позволяя вычислять решения эффективно и с высокой точностью даже в существенно многомерных примерах. Это достаточно редкий случай, когда инструмент

для многомерных задач одновременно является эффективным на практике, и обладает теоретическим анализом глобальной сходимости.

Численные эксперименты, представленные в диссертации, включают в себя примеры стохастических моделей в биологии и в физике плазмы. Методика, предложенная автором, успешно применяется и в других задачах, не вошедших в рамки данной работы. Совсем недавно было успешно проведено квантовое моделирование спиновой динамики для ядерного магнитного резонанса [69] – использование AMEn метода, предложенного в данной работе, позволяет рассчитывать структуры сложных белков на одном процессоре рабочей станции при весьма общих начальных знаниях о системе. До сих пор, единственной применимой альтернативой был метод SSR [165, 228], существенно использующий интуитивные химические соображения для построения редуцированной модели, и тем не менее требующий до терабайта общей памяти.

Направлений будущей работы можно наметить несколько. Сочетание одноблочных методов переменных направлений и шагов расширения базиса может быть применено и к другим задачам – в первую очередь, естественно, к многомерной задаче на собственные значения. Схемы DMRG/MPS первоначально были разработаны для решения этой задачи для квантовой системы многих тел, и в настоящее время реализованы в нескольких широко применяемых численных пакетах для квантовых физических расчетов. Поэтому имеется много возможностей для сравнения новых методов с профессиональными инструментами, разработанными в сообществе DMRG/MPS. Эта работа была начата в [41, 58, 159, 243].

Теоретическое понимание тензорных методов по-прежнему далеко от полного. Например, очевидно определенное рассогласование между теоретической оценкой сходимости для алгоритма AMEn, которая находится на уровне одношагового алгоритма градиентного спуска, и практическими результатами, которые показывают намного более быструю суперлинейную сходимость. Это может вдохновить нас на поиск возможных связей с итерационными методами наподобие Крыловских и Ньютона, и более тщательный анализ самих проекционных схем с переменными направлениями. Также стоит обсудить использование предобуславливателей, см. например [141, 137, 162, 159]. Все еще являются под вопросом свойства методов интегрирования по времени на многообразиях, порожденных тензорными форматами. Последние достижения в этой области тоже появились совсем недавно [65, 172], так что можно ожидать появления новых результатов.

Список обозначений

Индексы, размеры и диапазоны	
d	размерность тензора, кол-во координат в уравнении.
$n_k \leqslant n$	модовый размер, диапазон k-го индекса, $k = 1, \ldots, d$.
i_k, j_k	k-й индекс в исходном d-мерном тензоре.
$\boldsymbol{i}$, $\boldsymbol{j}$	Мультииндекс исходного тензора, $\boldsymbol{i} = (i_1, \ldots, i_d)$.
$\overline{i_1, \ldots, i_d}$	Эквивалентная запись мультииндекса с векторизацией, $\overline{i_1, \ldots, i_d} = i_1 + (i_2 - 1)n_1 + \cdots + (i_d - 1)n_1 \cdots n_{d-1}$.
$A \otimes B$	Кронекеровское произведение: $C = A \otimes B = \left[C_{\overline{ik},\overline{jm}}\right] = [AB_{k,m}] = [A_{i,j}B_{k,m}]$.
α_k, $\boldsymbol{\alpha}$	ранговый индекс в k-м и $(k+1)$-м TT блоках. $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{d-1})$.
$r_k \leqslant r$	TT ранг, диапазон α_k. Принадлежность тензору x обозначается как $r_k(x)$.
γ_k, $\boldsymbol{\gamma}$	Индекс ранга Таккера, или ранговый индекс в матричном TT.
$R_k \leqslant R$	Ранг Таккера, диапазон γ_k.
Типичные контексты и обозначения тензоров	
x	тензор, эквивалентный вектору, дискретная функция, решение.
A	тензор, эквивалентный матрице, дискретный оператор.
b	правая часть линейной системы, "векторный" тензор.
z, $\tilde{z}$	(приближенная) невязка, $\tilde{z} \approx z = b - Ax$.
ψ	тензор решения основного кинетического уравнения, уравнения Власова.
Блоки (ядра) тензорных форматов	
$x^{(k)}$, $A^{(k)}$	TT блоки вектора x и матрицы A, тензоры вида: $\left[x^{(k)}_{\alpha_{k-1},\alpha_k}(i_k)\right] \in \mathbb{C}^{r_{k-1}\times n_k \times r_k}$, $\left[A^{(k)}_{\alpha_{k-1},\alpha_k}(i_k, j_k)\right] \in \mathbb{C}^{r_{k-1}\times n_k \times m_k \times r_k}$. В произведениях $Mx^{(k)}$ выступают как векторы, $x^{(k)} \in \mathbb{C}^{r_{k-1}n_k r_k}$.
$x^{(c)}$, $x^{c(k)}$	Ядро Таккера, $x^{(c)} \in \mathbb{C}^{R_1\times\cdots\times R_d}$, и его TT блок, $x^{c(k)} \in \mathbb{C}^{r_{k-1}\times R_k \times r_k}$.
$x^{f(k,l)}$	Факторы Таккера или QTT-Tucker, $x^{f(k)} \in \mathbb{C}^{n_k \times R_k}$, $x^{f(k,l)} \in \mathbb{C}^{r_{k,l}\times n_{k,l}\times r_{k,l-1}}$.
Отображения и перестановки блоков	
$\tau(\{x^{(k)}\})$	TT отображение, $\tau(x^{(p)}, \ldots, x^{(q)}) \in \mathbb{C}^{r_{p-1}\times n_p \cdots n_q \times r_q}$. (Опр. 2.1.11)
$x^{(\leqslant k)}$, $x^{(\geqslant k)}$	TT интерфейс, $x^{(\leqslant k)} = \tau(x^{(1)}, \ldots, x^{(k)})$, $x^{(\geqslant k)} = \tau(x^{(k)}, \ldots, x^{(d)})$.
$X_{<k}$, $X_{\neq k}$	фрейм-матрицы, $X_{<k} = x^{(<k)} \otimes I_{n_k \cdots n_d}$, $X_{\neq k} = x^{(<k)} \otimes I_{n_k} \otimes \left(x^{(>k)}\right)^\top$.
$x^{\|k\rangle}$	левая развертка 3-мерного тензора, $x^{\|k\rangle} \in \mathbb{C}^{r_{k-1}n_k \times r_k}$. (Опр. 2.1.12)
$x^{\langle k\|}$	правая развертка 3-мерного тензора, $x^{\langle k\|} \in \mathbb{C}^{r_{k-1}\times n_k r_k}$. (Опр. 2.1.12)
$x^{\|k\|}$	центральная развертка 3-мерного тензора, $x^{\|k\|} \in \mathbb{C}^{n_k \times r_{k-1} r_k}$. (Опр. 2.1.12)
$A^{\langle k\rangle}$	внешняя развертка 4-мерного тензора, $A^{\langle k\rangle} \in \mathbb{C}^{r_{k-1}n_k \times m_k r_k}$. (Опр. 4.4.1)
$A^{\rangle k\langle}$	внутренняя развертка 4-мерного тензора, $A^{\rangle k\langle} \in \mathbb{C}^{n_k m_k \times r_k r_{k-1}}$. (Опр. 4.4.1)
$\mathsf{A}^{<k}$, $\mathsf{A}^{>k}$	проекции матриц на интерфейсы, $\mathsf{A}^{<k}_{\gamma_{k-1}} = \left(x^{(<k)}\right)^* A^{(<k)}_{\gamma_{k-1}} x^{(<k)}$.
$\mathsf{b}^{<k}$, $\mathsf{b}^{>k}$	проекции векторов на интерфейсы, $\mathsf{b}^{<k} = \left(x^{(<k)}\right)^* b^{(<k)}$. (Секция 4.4)
Величины, связанные с численными моделями	
x, y	пространственные координаты.
v, w	скоростные координаты.
t	время.
δt	эффективный временной шаг в численной схеме.
N_t	число временных шагов.
T	"большой" временной шаг (интервал) в блочной схеме.
$\hat{T}$	временной диапазон всей динамики системы: $t \in [0, \hat{T}]$.

Литература

[1] Горейнов С. А., Замарашкин Н. Л., Тыртышников Е. Е. Псевдоскелетные аппроксимации при помощи подматриц наибольшего объема // Матем. заметки. — 1997. — Vol. 62, no. 4. — P. 619–623. — URL: http://www.mathnet.ru/php/getFT.phtml?jrnid=mzm&paperid=1644&what=fullt&option_lang=rus.

[2] Горейнов С. А., Тыртышников Е. Е., Замарашкин Н. Л. Псевдоскелетная аппроксимация матриц // Докл. РАН. — 1995. — Т. 342, № 2. — С. 151–152.

[3] Казеев В. А., Тыртышников Е. Е. Структура гессиана и экономичная реализация метода Ньютона в задаче канонической аппроксимации тензоров // Ж. вычисл. матем. и матем. физ. — 2010. — Vol. 50, no. 6. — P. 979–998. — URL: http://www.mathnet.ru/php/getFT.phtml?jrnid=zvmmf&paperid=4884&what=fullt&option_lang=rus.

[4] Канторович Л. В. Функциональный анализ и прикладная математика // УМН. — 1948. — Vol. 3, no. 28. — P. 89–185. — URL: http://www.mathnet.ru/php/getFT.phtml?jrnid=rm&paperid=8775&what=fullt&option_lang=rus.

[5] Ковалёв Д. В. Численое моделирование Фарлей-Бунемановской неустойчивости в ионосфере Земли : Дисс... кандидата наук / Д. В. Ковалёв ; МГУ, ВМК. — Москва, 2009.

[6] Марчук Г. И. Методы расщепления для решения нестационарных задач // Ж. вычисл. матем. и матем. физ. — 1995. — Т. 35, № 6. — С. 843–849. — URL: http://www.mathnet.ru/php/getFT.phtml?jrnid=zvmmf&paperid=2383&what=fullt&option_lang=rus.

[7] Оселедец И. В. Оценки снизу для сепарабельных аппроксимаций ядра Гильберта // Матем. сб. — 2007. — Т. 198, № 3. — С. 137–144.

[8] Савостьянов Д. В. Полилинейная аппроксимация матриц и интегральные уравнения. — Дисс. ... канд. физ.-матем. наук — М.: ИВМ РАН. — 2006. — URL: http://www.inm.ras.ru/library/Tyrtyshnikov/savostyanov_disser.pdf.

[9] Смоляк С. А. Квадратурные и интерполяционные формулы на тензорных произведениях некоторых классов функций // Докл. АН СССР. — 1963. — Т. 148, № 5. — С. 1042–1053.

[10] Тыртышников Е. Е. Тензорные аппроксимации матриц, порожденных асимптотически гладкими функциями // Матем. сб. — 2003. — Т. 194, № 5. — С. 147–160.

[11] Ю. Н. Днестровский, Д. П. Костомаров. Математическое моделирование плазмы. — Физматлит, 1993.

[12] Ammar A., Cueto E., Chinesta F. Reduction of the chemical master equation for gene regulatory networks using proper generalized decompositions // Int. J. Numer. Meth. Biomed. Engng. — 2012. — Vol. 28, no. 9. — P. 960–973.

[13] Multilevel preconditioning and low rank tensor iteration for space-time simultaneous discretizations of parabolic PDEs : Tech. Rep. : 16 / SAM, ETH Zürich ; Executor: R. Andreev, C. Tobler : 2012. — URL: `http://sma.epfl.ch/~anchpcommon/publications/bpx.pdf`.

[14] Arkin A., Ross J., McAdams H.H. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells // Genetics. — 1998. — Vol. 149, no. 4. — P. 1633–1648.

[15] Bader B. W., Kolda T. G. Efficient MATLAB computations with sparse and factored tensors // SIAM J. Sci. Comput. — 2007. — Vol. 30, no. 1. — P. 205–231.

[16] Ballani J., Grasedyck L. A projection method to solve linear systems in tensor format // Numerical Linear Algebra with Applications. — 2013. — Vol. 20, no. 1. — P. 27–43.

[17] Tree Adaptive Approximation in the Hierarchical Tensor Format : Preprint : 141 ; Executor: Jonas Ballani, Lars Grasedyck : 2013.

[18] Ballani Jonas, Grasedyck Lars, Kluge Melanie. Black box approximation of tensors in hierarchical Tucker format // Linear Alg. Appl. — 2013. — Vol. 428. — P. 639–657.

[19] Barth Andrea, Schwab Christoph, Zollinger Nathaniel. Multi-level Monte Carlo Finite Element method for elliptic PDEs with stochastic coefficients // Numerische Mathematik. — 2011. — Vol. 119. — P. 123–161.

[20] Bartlett R. J., Musiał M. Coupled-cluster theory in quantum chemistry // Reviews of Modern Physics. — 2007. — Vol. 79, no. 1. — P. 291.

[21] Bauer F. L., Householder A. S. Some inequalities involving the euclidian condition of a matrix // Numerische Mathematik. — 1960. — Vol. 2, no. 1. — P. 308–311.

[22] Bebendorf M. Adaptive cross approximation of multivariate functions // Constructive approximation. — 2011. — Vol. 34, no. 2. — P. 149–179.

[23] Bellman R. E. Dynamic programming. — Princeton University Press, 1957.

[24] Benner P., Breiten T. Low rank methods for a class of generalized Lyapunov equations and related issues // Numerische Mathematik. — 2013. — Vol. 124, no. 3. — P. 441–470.

[25] Self-Generating and Efficient Shift Parameters in ADI Methods for Large Lyapunov and Sylvester Equations : MPI Magdeburg Preprint : 13-18 ; Executor: P. Benner, P. Kürschner, J. Saak : 2013. — URL: http://www2.mpi-magdeburg.mpg.de/preprints/2013/MPIMD13-18.pdf.

[26] Low Rank Solution of Unsteady Diffusion Equations with Stochastic Coefficients : MPI Magdeburg Preprint : 13-13 ; Executor: P. Benner, A. Onwunta, M. Stoll : 2013. — URL: http://www2.mpi-magdeburg.mpg.de/preprints/2013/MPIMD13-13.pdf.

[27] Bernstein S.N. Lecons sur les propriétés extrémales et la meilleure approximation des fonctions analytiques d´une variable réelle. — Paris: Gauthier-Villars, 1926.

[28] Bertoglio C., Khoromskij B. N. Low-rank quadrature-based tensor approximation of the Galerkin projected Newton/Yukawa kernels // Computer Phys. Comm. — 2012. — Vol. 183, no. 4. — P. 904–912.

[29] Beylkin G., Mohlenkamp M. J. Numerical operator calculus in higher dimensions // Proc. Nat. Acad. Sci. USA. — 2002. — Vol. 99, no. 16. — P. 10246–10251.

[30] Beylkin G., Mohlenkamp M. J. Algorithms for numerical analysis in high dimensions // SIAM J. Sci. Comput. — 2005. — Vol. 26, no. 6. — P. 2133–2159.

[31] Fast iterative solvers for fractional differential equations : MPI Magdeburg Preprint : 14-02 ; Executor: T. Breiten, V. Simoncini, M. Stoll : 2014. — URL: http://www2.mpi-magdeburg.mpg.de/preprints/2014/MPIMD14-02.pdf.

[32] Bro Richard. PARAFAC: Tutorial and applications // Chemometrics and Intelligent Lab. Syst. — 1997. — Vol. 38, no. 2. — P. 149–171.

[33] Buhmann M.D. Multivariate cardinal interpolation with radial-basis functions // Constr. Approx. — 1990. — Vol. 6, no. 3. — P. 225–255.

[34] Buhmann M.D. Radial basis functions // Acta Numerica. — 2000. — Vol. 9, no. 1. — P. 1–38.

[35] Buneman O. Excitation of Field Aligned Sound Waves by Electron Streams // Phys. Rev. Lett. — 1963. — Vol. 10. — P. 285–287.

[36] Bungatrz Hans-Joachim, Griebel Michael. Sparse grids // Acta Numerica. — 2004. — Vol. 13, no. 1. — P. 147–269.

[37] Cancés Eric, Ehrlacher Virginie, Leliévre Tony. Convergence of a greedy algorithm for high-dimensional convex nonlinear problems // Mathematical Models and Methods in Applied Sciences. — 2011. — Vol. 21, no. 12. — P. 2433–2467.

[38] Caroll J. D., Chang J. J. Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart–Young decomposition // Psychometrika. — 1970. — Vol. 35. — P. 283–319.

[39] Cohen A, DeVore R, Schwab Christoph. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs // Found. Comput. Math. — 2010. — Vol. 10. — P. 615–646.

[40] Comon P. Tensor decomposition: state of the art and applications // IMA Conf. Math. in Sig. Proc., Warwick, UK. — 2000.

[41] 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.

[42] Convergence rates for greedy algorithms in reduced basis methods / P. Binev, A. Cohen, W. Dahmen et al. // SIAM J. Math. Anal. — 2011. — Vol. 43, no. 3. — P. 1457–1472.

[43] de Lathauwer L., de Moor B., Vandewalle J. A multilinear singular value decomposition // SIAM J. Matrix Anal. Appl. — 2000. — Vol. 21. — P. 1253–1278.

[44] 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.

[45] De Lathauwer L., Vandewalle J. Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, ..., R_N)$ reduction in multilinear algebra // Linear Algebra Appl. — 2004. — Vol. 391. — P. 31–55.

[46] 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.

[47] Differential-geometric Newton method for the best rank-(r_1, r_2, r_3) approximation of tensors / M. Ishteva, L. de Lathauwer, P. A. Absil, S. van Huffel // Numerical Algorithms. — 2009. — Vol. 51, no. 2. — P. 179–194.

[48] Dimant Y.S., Oppenheim M.M. Ion thermal effects on E-region instabilities: linear theory // Journal of Atmospheric and Solar-Terrestrial Physics. — 2004. — Vol. 66, no. 17. — P. 1639 – 1654.

[49] Direct solution of the Chemical Master Equation using Quantized Tensor Trains / Vladimir Kazeev, Mustafa Khammash, Michael Nip, Christoph Schwab // PLOS Computational Biology. — 2014. — March.

[50] Dolgov S., Khoromskij B. Two-Level QTT-Tucker Format for Optimized Tensor Calculus // SIAM J. on Matrix An. Appl. — 2013. — Vol. 34, no. 2. — P. 593–623.

[51] Dolgov S., Khoromskij B. Simultaneous state-time approximation of the chemical master equation using tensor product formats // Numerical Linear Algebra with Applications. — 2014. — P. n/a–n/a.

[52] Dolgov S. V. TT-GMRES: solution to a linear system in the structured tensor format // Russ. J. Numer. Anal. Math. Modelling. — 2013. — Vol. 28, no. 2. — P. 149–172.

[53] Dolgov S. V., Khoromskij Boris N., Oseledets Ivan V. Fast solution of multi-dimensional parabolic problems in the tensor train/quantized tensor train–format with initial application to the Fokker-Planck equation // SIAM J. Sci. Comput. — 2012. — Vol. 34, no. 6. — P. A3016–A3038.

[54] 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.

[55] Dolgov S. V., Oseledets I. V. Solution of linear systems and matrix inversion in the TT-format // SIAM J. Sci. Comput. — 2012. — Vol. 34, no. 5. — P. A2718–A2739.

[56] 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.

[57] 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.

[58] Dolgov S. V., Savostyanov D. V. Corrected one-site density matrix renormalization group and alternating minimal energy algorithm // Proc. ENUMATH 2013, accepted. — 2014. — URL: http://arxiv.org/abs/1312.6542.

[59] Dolgov S. V., Smirnov A. P., Tyrtyshnikov E. E. Low-rank approximation in the numerical modeling of the Farley-Buneman instability in ionospheric plasma // J. Comp. Phys. — 2014. — Vol. 263. — P. 268–282.

[60] Domain Decomposition Solution of Elliptic Boundary-Value Problems via Monte Carlo and Quasi-Monte Carlo Methods / Juan A. Acebrón, Maria Pia Busico, Piero Lanucara, Renato Spigler // SIAM J. Sci. Comput. — 2005. — Vol. 27. — P. 440–457. — URL: http://portal.acm.org/citation.cfm?id=1093655.1093684.

[61] Domanov I. Study of Canonical Polyadic Decomposition of Higher-Order Tensors : Ph. D. thesis / I. Domanov. — 2013.

[62] Drake G. W. F. High Precision Theory of Atomic Helium // Physica Scripta. — 1999. — Vol. 1999, no. T83. — P. 83.

[63] Drineas P., Kannan R., Mahoney M. W. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition // SIAM J Comput. — 2006. — Vol. 36, no. 1. — P. 184–206.

[64] Dunning Jr Thom H. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen // The Journal of Chemical Physics. — 1989. — Vol. 90. — P. 1007.

[65] Dynamical Approximation by Hierarchical Tucker and Tensor-Train Tensors / C. Lubich, T. Rohwedder, R. Schneider, B. Vandereycken // SIAM J. Matrix. Anal. Appl. — 2013. — Vol. 34, no. 2. — P. 470–494.

[66] Efficient analysis of high dimensional data in tensor formats / M. Espig, W. Hackbusch, A. Litvinenko et al. // Sparse Grids and Applications. — Springer, 2013. — P. 31–56.

[67] Espig M., Grasedyck L., Hackbusch W. Black box low tensor rank approximation using fibre-crosses // Constr. Appr. — 2009. — Vol. 30, no. 3. — P. 557–597.

[68] Espig Mike, Hackbusch Wolfgang. A regularized Newton method for the efficient approximation of tensors represented in the canonical tensor format // Numer. Math. — 2012. — Vol. 122, no. 3. — P. 489–525.

[69] Exact NMR simulation of protein-size spin systems using tensor train formalism / D. V. Savostyanov, S. V. Dolgov, J. M. Werner, Ilya Kuprov // Phys. Rev. B. — 2014.

[70] Fannes M., Nachtergaele B., Werner R.F. Finitely correlated states on quantum spin chains // Comm. Math. Phys. — 1992. — Vol. 144, no. 3. — P. 443–490.

[71] Fannes M., Nachtergaele B., Werner R. F. Ground states of VBS models on Cayley trees // J. Stat. Phys. — 1992. — Vol. 66. — P. 939–973. — URL: `http://dx.doi.org/10.1007/BF01055710`.

[72] Farley D. T. A plasma instability resulting in field-aligned irregularities in the ionosphere // Journal of Geophysical Research. — 1963. — Vol. 68, no. 22. — P. 6083–6097.

[73] Figueroa L. E., Süli E. Greedy approximation of high-dimensional Ornstein–Uhlenbeck operators // Foundations of Computational Mathematics. — 2012. — Vol. 12, no. 5. — P. 573–623.

[74] Fishman G. S. Monte Carlo: concepts, algorithms, and applications. — Springer New York, 1996. — Vol. 1196.

[75] Garcke J., Griebel M., Thess M. Data mining with sparse grids // Computing. — 2001. — Vol. 67, no. 3. — P. 225–253.

[76] Gardner T.S., Cantor C.R., Collins J.J. Construction of a genetic toggle switch in Escherichia coli // Nature. — 2000. — Vol. 403. — P. 339–342.

[77] Regularity and approximability of the solutions to the chemical master equation : Matheon Preprint : 1010 ; Executor: L. Gauckler, H. Yserentant : 2013. — URL: http://opus4.kobv.de/opus4-matheon/frontdoor/index/index/docId/1214.

[78] Gavrilyuk I. P., Hackbusch W., Khoromskij B. N. $\mathcal{H}$-Matrix approximation for the operator exponential with applications // Numerische Mathematik. — 2002. — Vol. 92, no. 1. — P. 83–111.

[79] Gavrilyuk I. P., Hackbusch W., Khoromskij B. N. Tensor-product approximation to the inverse and related operators in high-dimensional elliptic problems // Computing. — 2005. — no. 74. — P. 131–157.

[80] Gavrilyuk I. P., Khoromskij B. N. Quantized-TT-Cayley transform for computing the dynamics and the spectrum of high-dimensional Hamiltonians // Comput. Methods in Appl. Math. — 2011. — Vol. 11, no. 3. — P. 273–290.

[81] Gillespie D.T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions // J Comput. Phys. — 1976. — Vol. 22, no. 4. — P. 403–434.

[82] Gillespie D.T. Approximate accelerated stochastic simulation of chemically reacting systems // The Journal of Chemical Physics. — 2001. — Vol. 115. — P. 1716.

[83] Gillespie D.T. The chemical Langevin and Fokker-Planck equations for the reversible isomerization reaction // The Journal of Physical Chemistry A. — 2002. — Vol. 106, no. 20. — P. 5063–5071.

[84] Gillespie Daniel T. A rigorous derivation of the chemical master equation // Physica A: Statistical Mechanics and its Applications. — 1992. — Vol. 188, no. 1-3. — P. 404 – 425. — URL: http://www.sciencedirect.com/science/article/pii/037843719290283V.

[85] Gillespie D. T. The chemical Langevin equation // The Journal of Chemical Physics. — 2000. — Vol. 113, no. 1. — P. 297–306.

[86] Low-rank approximate inverse for preconditioning tensor-structured linear systems : arXiv preprint : 1304.6004 ; Executor: L. Giraldi, A. Nouy, G. Legrain : 2013. — URL: http://arxiv.org/abs/1304.6004.

[87] Golub G., Kahan W. Calculating the singular values and pseudo-inverse of a matrix // SIAM J. Numer. Anal. — 1965. — Vol. 2, no. 2. — P. 205–224.

[88] Golub G.H., Van Loan C.F. Matrix computations. — Johns Hopkins University Press, Baltimore, MD, 1996.

[89] 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.

[90] Goreinov S. A., Tyrtyshnikov E. E. The maximal-volume concept in approximation by low-rank matrices // Contemporary Mathematics. — 2001. — Vol. 208. — P. 47–51.

[91] Goreinov S. A., Tyrtyshnikov E. E., Zamarashkin N. L. A theory of pseudo–skeleton approximations // Linear Algebra Appl. — 1997. — Vol. 261. — P. 1–21.

[92] Goutsias J. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems // The Journal of chemical physics. — 2005. — Vol. 122. — P. 184102.

[93] Grasedyck L. Existence and computation of low Kronecker-rank approximations for large systems in tensor product structure // Computing. — 2004. — Vol. 72. — P. 247–265.

[94] Grasedyck L. Hierarchical singular value decomposition of tensors // SIAM J. Matrix Anal. Appl. — 2010. — Vol. 31, no. 4. — P. 2029–2054.

[95] Polynomial approximation in hierarchical Tucker format by vector-tensorization : DFG-SPP1324 Preprint : 43 / Philipps-Univ. ; Executor: L. Grasedyck. — Marburg : 2010. — URL: `http://www.dfg-spp1324.de/download/preprints/preprint043.pdf`.

[96] Grasedyck L., Hackbusch W. An introduction to hierarchical ($\mathcal{H}$-) and TT-rank of tensors with examples // Comput. Meth. Appl. Math. — 2011. — Vol. 3. — P. 291–304.

[97] Grasedyck L., Kressner D., Tobler C. A literature survey of low-rank tensor approximation techniques // GAMM-Mitteilungen. — 2013. — Vol. 36, no. 1. — P. 53–78.

[98] Griebel M. Sparse grids and related approximation schemes for higher dimensional problems. — SFB 611, 2005.

[99] Griebel M., Harbrecht H. Approximation of bi-variate functions: singular value decomposition versus sparse grids // IMA Journal of Numerical Analysis. — 2013.

[100] Griebel M., Oeltz D. A sparse grid space-time discretization scheme for parabolic problems // Computing. — 2007. — Vol. 81. — P. 1–34. — URL: `http://dx.doi.org/10.1007/s00607-007-0241-3`.

[101] Determining the long-term behavior of cell populations: A new procedure for detecting ergodicity in large stochastic reaction networks : arXiv : 1312.2879 ; Executor: A. Gupta, M. Khammash : 2013.

[102] Hackbusch W. Tensor spaces and numerical tensor calculus. — Springer–Verlag, Berlin, 2012. — ISBN: 978-3642280269.

[103] Hackbusch W., Braess D. Approximation of $\frac{1}{x}$ by exponential sums in $[1, \infty]$ // IMA J. Numer. Anal. — 2005. — Vol. 25, no. 4. — P. 685–697.

[104] Hackbusch W., Khoromskij B. N. Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. I. Separable approximation of multi-variate functions // Computing. — 2006. — Vol. 76, no. 3-4. — P. 177–202.

[105] 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.

[106] Hackbusch W., Khoromskij B. N., Tyrtyshnikov E. E. Hierarchical Kronecker tensor-product approximations // J. Numer. Math. — 2005. — Vol. 13. — P. 119–156.

[107] Hackbusch W., Khoromskij B. N, Tyrtyshnikov E. E. Approximate iterations for structured matrices // Numer. Mathematik. — 2008. — Vol. 109, no. 3. — P. 365–383.

[108] Hackbusch W., Kühn S. A new scheme for the tensor representation // J. Fourier Anal. Appl. — 2009. — Vol. 15, no. 5. — P. 706–722.

[109] Hamza A. M., St.-Maurice J.-P. A fully self-consistent fluid theory of anomalous transport in Farley-Buneman turbulence // Journal of Geophysical Research: Space Physics. — 1995. — Vol. 100, no. A6. — P. 9653–9668.

[110] Harshman R. A. Foundations of the PARAFAC procedure: models and conditions for an explanatory multimodal factor analysis // UCLA Working Papers in Phonetics. — 1970. — Vol. 16. — P. 1–84.

[111] Hastings W. K. Monte Carlo sampling methods using Markov chains and their applications // Biometrika. — 1970. — Vol. 57, no. 1. — P. 97–109.

[112] Hegland M., Garcke J. On the numerical solution of the chemical master equation with sums of rank one tensors // ANZIAM. — 2011. — Vol. 52. — P. C628–C643.

[113] Hellander A., Lötstedt P. Hybrid method for the chemical master equation // Journal of Computational Physics. — 2007. — Vol. 227, no. 1. — P. 100–122.

[114] Hemberg M., Barahona M. Perfect sampling of the master equation for gene regulatory networks // Biophysical journal. — 2007. — Vol. 93, no. 2. — P. 401–410.

[115] Hitchcock F. L. The expression of a tensor or a polyadic as a sum of products // J. Math. Phys. — 1927. — Vol. 6, no. 1. — P. 164–189.

[116] 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.

[117] 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.

[118] Huckle T., Waldherr K. Subspace iteration method in terms of matrix product states // Proc. Appl. Math. Mech. — 2012. — Vol. 12. — P. 641–642.

[119] Huckle T., Waldherr K., Schulte-Herbrüggen T. Computations in quantum tensor networks // Linear Algebra Appl. — 2013. — Vol. 438. — P. 750–781.

[120] Ibraghimov I. Application of the three-way decomposition for matrix compression // Numer. Linear Algebra Appl. — 2002. — Vol. 9, no. 6-7. — P. 551–565.

[121] Jahnke T. An adaptive wavelet method for the chemical master equation // SIAM J. Sci. Comput. — 2010. — Vol. 31, no. 6. — P. 4373.

[122] Jahnke T. On Reduced Models for the Chemical Master Equation // Multiscale Modeling and Simulation. — 2011. — Vol. 9, no. 4. — P. 1646–1676.

[123] Jahnke Tobias, Huisinga Wilhelm. A Dynamical Low-Rank Approach to the Chemical Master Equation // Bulletin of Mathematical Biology. — 2008. — Vol. 70. — P. 2283–2302. — URL: http://dx.doi.org/10.1007/s11538-008-9346-x.

[124] Jahnke T., Kreim M. Error Bound for Piecewise Deterministic Processes Modeling Stochastic Reaction Systems // Multiscale Modeling and Simulation. — 2012. — Vol. 10, no. 4. — P. 1119–1147.

[125] Janhunen P. Perpendicular particle simulation of the E region Farley-Buneman instability // Journal of Geophysical Research: Space Physics. — 1994. — Vol. 99, no. A6. — P. 11461–11473.

[126] Jeckelmann E. Dynamical density–matrix renormalization–group method // Phys. Rev. B. — 2002. — Vol. 66. — P. 045114.

[127] Kazeev V., Khoromskij B., Tyrtyshnikov E. Multilevel Toeplitz Matrices Generated by Tensor-Structured Vectors and Convolution with Logarithmic Complexity // SIAM J. Sci. Comp. — 2013. — Vol. 35, no. 3. — P. A1511–A1536.

[128] hp-DG-QTT solution of high-dimensional degenerate diffusion equations : Tech. Report : 2012-11 / ETH SAM, Zürich ; Executor: V Kazeev, O Reichmann, Ch Schwab : 2012. — URL: ftp://magellan-03.math.ethz.ch/hg/pub/sam-reports/reports/reports2012/2012-11.pdf.

[129] Kazeev V. A., Khoromskij B. N. Low-rank explicit QTT representation of the Laplace operator and its inverse // SIAM J. Matrix Anal. Appl. — 2012. — Vol. 33, no. 3. — P. 742–758.

[130] The tensor structure of a class of adaptive algebraic wavelet transforms : Preprint : 2013-28 / ETH SAM, Zürich ; Executor: Vladimir A. Kazeev, Ivan V. Oseledets : 2013. — URL: http://www.sam.math.ethz.ch/sam_reports/reports_final/reports2013/2013-28.pdf.

[131] Kellogg R. B. An Alternating Direction Method for Operator Equations // SIAM. — 1964. — Vol. 12, no. 4. — P. 848–854. — URL: http://dx.doi.org/10.1137/0112072.

[132] Khoromskaia V. Computation of the Hartree-Fock exchange by tensor-structured methods // Comput. Methd. Appl. Math. — 2008. — Vol. 10, no. 2.

[133] 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/.

[134] Khoromskaia V. Black-Box Hartree–Fock Solver by Tensor Numerical Methods // Computational Methods in Applied Mathematics. — 2014. — Vol. 14, no. 1. — P. 89–111.

[135] Grid-based lattice summation of electrostatic potentials by low-rank tensor approximation : Preprint : 116 / MPI MIS ; Executor: V. Khoromskaia, B. N. Khoromskij : 2013. — URL: http://www.mis.mpg.de/preprints/2013/preprint2013_116.pdf.

[136] Khoromskaia Venera, Khoromskij Boris N., Schneider Reinhold. Tensor-structured factorized calculation of two-electron integrals in a general basis // SIAM J. Sci. Comput. — 2013. — Vol. 35, no. 2. — P. A987–A1010.

[137] Khoromskij B.N., Schwab Ch. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs // SIAM J. of Sci. Comp. — 2011. — Vol. 33, no. 1. — P. 1–25.

[138] Khoromskij B. N. Structured rank-$(r_1, \ldots, r_d)$ decomposition of function-related operators in $\mathbb{R}^d$ // Comput. Meth. Appl. Math. — 2006. — Vol. 6, no. 2. — P. 194–220.

[139] Khoromskij B. N. On tensor approximation of Green iterations for Kohn-Sham equations // Computing and visualization in science. — 2008. — Vol. 11, no. 4-6. — P. 259–271.

[140] $\mathcal{O}(d \log N)$-Quantics Approximation of N-d Tensors in High-Dimensional Numerical Modeling : Preprint : 55 / MPI MIS ; Executor: B. N. Khoromskij. — Leipzig : 2009. — URL: http://www.mis.mpg.de/preprints/2009/preprint2009_55.pdf.

[141] Khoromskij B. N. Tensor-structured preconditioners and approximate inverse of elliptic operators in $\mathbb{R}^d$ // Constr. Approx. — 2009. — no. 30. — P. 599–620.

[142] 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.

[143] Introduction to tensor numerical methods in scientific computing : Preprint, Lecture Notes : 06-2011 / University of Zürich ; Executor: B. N. Khoromskij : 2010. — URL: http://www.math.uzh.ch/fileadmin/math/preprints/06_11.pdf.

[144] Khoromskij B. N. $\mathcal{O}(d \log N)$–Quantics approximation of N–d tensors in high-dimensional numerical modeling // Constr. Appr. — 2011. — Vol. 34, no. 2. — P. 257–280.

[145] Khoromskij B. N. Tensor-structured numerical methods in scientific computing: Survey on recent advances // Chemometr. Intell. Lab. Syst. — 2012. — Vol. 110, no. 1. — P. 1–19.

[146] 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.

[147] 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.

[148] 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.

[149] Superfast Wavelet Transform Using QTT Approximation. I: Haar Wavelets : Preprint MPI MIS, Leipzig : 103 ; Executor: B. N. Khoromskij, S. Miao : 2013. — URL: http://www.mis.mpg.de/preprints/2013/preprint2013_103.pdf.

[150] Khoromskij B. N., Oseledets I. V. Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs // Comput. Meth. Appl. Math. — 2010. — Vol. 10, no. 4. — P. 376–394.

[151] Khoromskij B. N., Oseledets I. V. QTT-approximation of elliptic solution operators in higher dimensions // Rus. J. Numer. Anal. Math. Model. — 2011. — Vol. 26, no. 3. — P. 303–322.

[152] Klümper A., Schadschneider A., Zittartz J. Matrix Product Ground States for One-Dimensional Spin-1 Quantum Antiferromagnets // Europhys. Lett. — 1993. — Vol. 24, no. 4. — P. 293–297.

[153] Koch Ottmar, Lubich Christian. Dynamical tensor approximation // SIAM J. Matrix Anal. Appl. — 2010. — Vol. 31, no. 5. — P. 2360–2375.

[154] Kolda T. G., Bader B. W. Tensor decompositions and applications // SIAM Review. — 2009. — Vol. 51, no. 3. — P. 455–500.

[155] Kovalev DV, Smirnov AP, Dimant YS. On the effect of electron-mass variation in numerical simulations of the Farley-Buneman instability // Moscow University Computational Mathematics and Cybernetics. — 2009. — Vol. 33, no. 1. — P. 17–24.

[156] Kovalev DV, Smirnov AP, Dimant Ya S. Simulations of the nonlinear stage of Farley-Buneman instability with allowance for electron thermal effects // Plasma physics reports. — 2009. — Vol. 35, no. 7. — P. 603–610.

[157] Kovalev DV, Smirnov AP, Dimant Ya S. Study of kinetic effects arising in simulations of Farley-Buneman instability // Plasma physics reports. — 2009. — Vol. 35, no. 5. — P. 420–425.

[158] Kovalev D. V., Smirnov A. P., Dimant Y. S. Modeling of the Farley-Buneman instability in the E-region ionosphere: a new hybrid approach // Annales Geophysicae. — 2008. — Vol. 26, no. 9. — P. 2853–2870.

[159] Low-rank tensor methods with subspace correction for symmetric eigenvalue problems : MATHICSE preprint : 40.2013 / EPFL, Lausanne ; Executor: D. Kressner, M. Steinlechner, A. Uschmajew : 2013. — URL: http://mathicse.epfl.ch/files/content/sites/mathicse/files/Mathicsereports2013/40.2013_DK-MS-AU.pdf.

[160] 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.

[161] Kressner D., Tobler C. Low-rank tensor Krylov subspace methods for parametrized linear systems // SIAM J. Matrix Anal. Appl. — 2011. — Vol. 32, no. 4. — P. 273–290.

[162] 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.

[163] Kroonenberg P.M., de Leeuw J. Principal component analysis of three-mode data by means of alternating least squares algorithms // Psychometrika. — 1980. — Vol. 45, no. 1. — P. 69–97.

[164] Kühn S. Hierarchische Tensordarstellung : Ph. D. thesis / S. Kühn ; Uni. Leipzig, Faculty of Mathematics and Informatics. — 2012.

[165] Kuprov Ilya, Wagner-Rundell N., Hore P. J. Polynomially scaling spin dynamics simulation algorithm based on adaptive state-space restriction // J Magn. Reson. — 2007. — Vol. 189, no. 2. — P. 241–250.

[166] Le Bris C., Leliévre T., Maday Y. Results and Questions on a Nonlinear Approximation Approach for Solving High-dimensional Partial Differential Equations // Constr. Approx. — 2009. — Vol. 30. — P. 621–651.

[167] 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.

[168] 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.

[169] Lécot Christian, Khettabi Faysal El. Quasi-Monte Carlo Simulation of Diffusion // Journal of Complexity. — 1999. — Vol. 15, no. 3. — P. 342 – 359. — URL: http://www.sciencedirect.com/science/article/pii/S0885064X99905095.

[170] Low-frequency electrostatic waves in the ionospheric E-region: a comparison of rocket observations and numerical simulations / L. Dyrud, B. Krane, M. Oppenheim et al. // Annales Geophysicae. — 2006. — Vol. 24, no. 11. — P. 2959–2979.

[171] Low-rank tensor structure of solutions to elliptic problems with jumping coefficients / S. V. Dolgov, Boris N. Khoromskij, Ivan V. Oseledets, Eugene E. Tyrtyshnikov // J. Comput. Math. — 2012. — Vol. 30, no. 1. — P. 14–23.

[172] Lubich Christian, Oseledets Ivan V. A projector-splitting integrator for dynamical low-rank approximation // BIT. — 2014. — Vol. 54, no. 1. — P. 171–188.

[173] 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.

[174] Machida S., Goertz C. K. Computer simulation of the Farley-Buneman instability and anomalous electron heating in the auroral ionosphere // Journal of Geophysical Research: Space Physics. — 1988. — Vol. 93, no. A9. — P. 9993–10000.

[175] Matrix product state representations / D. Perez-Garcia, F. Verstraete, M. M. Wolf, J. I. Cirac // Quantum Info. Comput. — 2007. — Vol. 7, no. 5. — P. 401–430.

[176] Matthies H, Keese A. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations // Computer Methods in Applied Mechanics and Engineering. — 2005. — Vol. 194, no. 12-16. — P. 1295–1331.

[177] Metropolis N., Ulam S. The monte carlo method // Journal of the American statistical association. — 1949. — Vol. 44, no. 247. — P. 335–341.

[178] Meyer H.-D., Gatti F., Worth G. A. Multidimensional Quantum Dynamics: MCTDH Theory and Applications. — Weinheim : Wiley-VCH, 2009.

[179] Molecular Electronic-Structure Theory / Trygve Helgaker, Poul Jørgensen, Jeppe Olsen, Mark A Ratner // Physics Today. — 2001. — Vol. 54. — P. 52.

[180] Multiresolution quantum chemistry: basic theory and initial applications / G. Beylkin, G. Fann, Z. Gan et al. // J. Chem. Phys. — 2004. — Vol. 121, no. 23. — P. 11587–11598.

[181] Munsky B., Khammash M. The finite state projection algorithm for the solution of the chemical master equation // The Journal of chemical physics. — 2006. — Vol. 124. — P. 044104.

[182] The Convergence Rate of Inexact Preconditioned Steepest Descent Algorithm for Solving Linear Systems : Numerical Analysis Report : NA-87-04 / Stanford University ; Executor: Hans Munthe-Kaas : 1987. — URL: `http://i.stanford.edu/pub/cstr/reports/na/m/87/04/NA-M-87-04.pdf`.

[183] Nené Nuno R., Zaikin Alexey. Decision making in noisy bistable systems with time-dependent asymmetry // Phys. Rev. E. — 2013. — Vol. 87. — P. 012715.

[184] A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids / A. Ammar, B. Mokdad, F. Chinesta, R. Keunings // Journal of Non-Newtonian Fluid Mechanics. — 2006. — Vol. 139, no. 3. — P. 153 – 176.

[185] Newman Alice L., Ott Edward. Nonlinear simulations of type 1 irregularities in the equatorial electrojet // Journal of Geophysical Research: Space Physics. — 1981. — Vol. 86, no. A8. — P. 6879–6891.

[186] Niederreiter Harald. Quasi–Monte Carlo methods and pseudo–random numbers // Bull. AMS. — 1978. — Vol. 84, no. 6. — P. 957–1041.

[187] Notay Y. Convergence analysis of inexact Rayleigh quotient iteration // SIAM J. on Matrix An. Appl. — 2003. — Vol. 24, no. 3. — P. 627–644.

[188] Nouy A. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations // Computer Methods in Applied Mechanics and Engineering. — 2010. — Vol. 199, no. 23. — P. 1603–1626.

[189] Oppenheim M.M., Dimant Y.S. Ion thermal effects on E-region instabilities: 2D kinetic simulations // Journal of Atmospheric and Solar-Terrestrial Physics. — 2004. — Vol. 66, no. 17. — P. 1655–1668.

[190] Oppenheim Meers, Otani Niels. Spectral characteristics of the Farley-Buneman instability: Simulations versus observations // Journal of Geophysical Research: Space Physics. — 1996. — Vol. 101, no. A11. — P. 24573–24582.

[191] Oppenheim Meers, Otani Niels, Ronchi Corrado. Saturation of the Farley-Buneman instability via nonlinear electron E × B drifts // Journal of Geophysical Research: Space Physics. — 1996. — Vol. 101, no. A8. — P. 17273–17286.

[192] Oppenheim M. M., Dimant Y., Dyrud L. P. Large-scale simulations of 2-D fully kinetic Farley-Buneman turbulence // Annales Geophysicae. — 2008. — Vol. 26, no. 3. — P. 543–553.

[193] Oppenheim M. M., Dimant Y. S. Kinetic simulations of 3-D Farley-Buneman turbulence and anomalous electron heating // Journal of Geophysical Research: Space Physics. — 2013. — Vol. 118, no. 3. — P. 1306–1318.

[194] Compact matrix form of the d-dimensional tensor decomposition : Preprint : 2009-01 / INM RAS ; Executor: I. V. Oseledets. — Moscow : 2009. — URL: http://pub.inm.ras.ru/pub/inmras2009-01.pdf.

[195] 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.

[196] 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.

[197] Oseledets I. V. Tensor-train decomposition // SIAM J. Sci. Comput. — 2011. — Vol. 33, no. 5. — P. 2295–2317.

[198] Oseledets I. V. Constructive representation of functions in low-rank tensor formats // Constr. Appr. — 2013. — Vol. 37, no. 1. — P. 1–18. — URL: http://pub.inm.ras.ru/pub/inmras2010-04.pdf.

[199] Efficient time-stepping scheme for dynamics on TT-manifolds : Preprint : 24 / MPI MIS ; Executor: I. V. Oseledets, B. N. Khoromskij, R. Schneider : 2012. — URL: http://www.mis.mpg.de/preprints/2012/preprint2012_24.pdf.

[200] 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.

[201] Oseledets I. V., Savostyanov D. V., Tyrtyshnikov E. E. Linear algebra for tensor problems // Computing. — 2009. — Vol. 85, no. 3. — P. 169–188.

[202] Oseledets I. V., Savostyanov D. V., Tyrtyshnikov E. E. Cross approximation in tensor electron density computations // Numer. Linear Algebra Appl. — 2010. — Vol. 17, no. 6. — P. 935–952.

[203] 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.

[204] Tensor tree decomposition does not need a tree : Preprint (Submitted to Linear Algebra Appl) : 2009-04 / INM RAS ; Executor: I. V. Oseledets, E. E. Tyrtyshnikov. — Moscow : 2009. — URL: `http://pub.inm.ras.ru/pub/inmras2009-08.pdf`.

[205] Oseledets I. V., Tyrtyshnikov E. E. TT-cross approximation for multidimensional arrays // Linear Algebra Appl. — 2010. — Vol. 432, no. 1. — P. 70–88.

[206] Östlund S., Rommer S. Thermodynamic limit of Density Matrix Renormalization // Phys. Rev. Lett. — 1995. — Vol. 75, no. 19. — P. 3537–3540.

[207] 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.

[208] Ptashne M. A genetic switch: λ-phage and higher organisms. — Wiley-Blackwell, 1992.

[209] Rabitz H, Kramer M, Dacol D. Sensitivity analysis in chemical kinetics // Annual review of physical chemistry. — 1983. — Vol. 34, no. 1. — P. 419–461.

[210] Real-time in silico experiments on gene regulatory networks and surgery simulation on handheld devices / I. Alfaro, D. Gonzalez, F. Bordeu et al. // Journal of Computational Surgery. — 2014. — Vol. 1, no. 1.

[211] Rigorous results on valence-bond ground states in antiferromagnets / Ian Affleck, Tom Kennedy, Elliott H Lieb, Hal Tasaki // Phys. Rev. Lett. — 1987. — Vol. 59, no. 7. — P. 799–802.

[212] 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.

[213] Saad Y. Iterative methods for sparse linear systems. — SIAM, 2003.

[214] Savas B., Eldén L. Krylov-type methods for tensor computations I // Linear Algebra and its Applications. — 2013. — Vol. 438, no. 2. — P. 891–918.

[215] Savostyanov D. V. Fast revealing of mode ranks of tensor in canonical form // Numer. Math. Theor. Meth. Appl. — 2009. — Vol. 2, no. 4. — P. 439–444.

[216] Savostyanov D. V. Quasioptimality of maximum–volume cross interpolation of tensors // Linear Algebra Appl. — 2014. — Vol. 458. — P. 217–244.

[217] 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.

[218] Schlegel K., Thiemann H. Particle-in-cell plasma simulations of the modified two-stream instability // Annales Geophysicae. — 1994. — Vol. 12, no. 10-11. — P. 1091–1100.

[219] Schneider J. Error estimates for two–dimensional cross approximation // J. Approx. Theory. — 2010. — Vol. 162. — P. 1685–1700.

[220] Schneider R., Uschmajew A. Approximation rates for the hierarchical tensor format in periodic Sobolev spaces // Journal of Complexity. — 2013.

[221] Schollwöck U. The density–matrix renormalization group // Rev. Mod. Phys. — 2005. — Vol. 77, no. 1. — P. 259–315.

[222] 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.

[223] Schötzau D. hp-DGFEM for parabolic evolution problems. Applications to diffusion and viscous incompressible fluid flow : Ph. D. thesis / D. Schötzau ; ETH. — Zürich, 1999. — URL: http://dx.doi.org/10.3929/ethz-a-002057769.

[224] Simoncini Valeria, Szyld Daniel B. Theory of Inexact Krylov Subspace Methods and Applications to Scientific Computing // SIAM J. Sci. Comput. — 2003. — Vol. 25. — P. 454–477.

[225] Skadron G., Weinstock J. Nonlinear stabilization of a two-stream plasma instability in the ionosphere // Journal of Geophysical Research. — 1969. — Vol. 74, no. 21. — P. 5113–5126.

[226] Sloan I.H., Wozniakowski H. When are quasi-Monte Carlo algorithms efficient for high dimensional integrals // J. of Complexity. — 1998. — Vol. 14, no. 1. — P. 1–33.

[227] A solver for the stochastic master equation applied to gene regulatory networks / Markus Hegland, Conrad Burden, Lucia Santoso et al. // Journal of Computational and Applied Mathematics. — 2007. — Vol. 205, no. 2. — P. 708 – 724.

[228] Spinach — A software library for simulation of spin dynamics in large spin systems / H. J. Hogben, M. Krzystyniak, G. T. P. Charnock et al. // J Magn. Reson. — 2011. — Vol. 208, no. 2. — P. 179–194.

[229] Sreenath S. N., Kwang-Hyun C., Wellstead P. Modelling the dynamics of signalling pathways // Essays Biochemistry. — 2008. — Vol. 45. — P. 1–28.

[230] Steuer R. Effects of stochasticity in models of the cell cycle: from quantized cycle times to noise-induced oscillations // Journal of theoretical biology. — 2004. — Vol. 228, no. 3. — P. 293–301.

[231] A low-rank in time approach to PDE-constrained optimization : MPI Magdeburg Preprint : 13-08 ; Executor: M. Stoll, T. Breiten : 2013. — URL: http://www2.mpi-magdeburg.mpg.de/preprints/2013/MPIMD13-08.pdf.

[232] Strang Gilbert. On the Construction and Comparison of Difference Schemes // SIAM Journal on Numerical Analysis. — 1968. — Vol. 5, no. 3. — P. 506–517. — URL: http://www.jstor.org/stable/2949700.

[233] Sudan R. N., Akinrimisi J., Farley D. T. Generation of small-scale irregularities in the equatorial electrojet // Journal of Geophysical Research. — 1973. — Vol. 78, no. 1. — P. 240–248.

[234] Tadmor E. The exponential accuracy of Fourier and Chebychev differencing methods // SIAM J. Numer. Anal. — 1986. — Vol. 23. — P. 1–23.

[235] Temlyakov Victor. Greedy Approximation. — Cambridge University Press, 2011.

[236] 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.

[237] Tensor product approximation DMRG and coupled cluster method in quantum chemistry : arXiv preprint : 1310.2736 ; Executor: Ors Legeza, Thorsen Rohwedder, Reinhold Schneider, Szilard Szalay : 2013. — URL: http://arxiv.org/abs/1310.2736.

[238] Tensor product approximation with optimal rank in quantum chemistry / S. R. Chinnamsetty, M. Espig, W. Hackbusch et al. // J. Chem. Phys. — 2007. — Vol. 127. — P. 84–110.

[239] Trefethen Lloyd N. Spectral methods in MATLAB. — Philadelphia : SIAM, 2000.

[240] Tucker L. R. Some mathematical notes on three-mode factor analysis // Psychometrika. — 1966. — Vol. 31. — P. 279–311.

[241] Tyrtyshnikov E. E. Incomplete cross approximation in the mosaic–skeleton method // Computing. — 2000. — Vol. 64, no. 4. — P. 367–380.

[242] Tyrtyshnikov E. E. Kronecker-product approximations for some function-related matrices // Linear Algebra Appl. — 2004. — Vol. 379. — P. 423–437.

[243] 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.

[244] van Kampen N. G. Stochastic processes in physics and chemistry. — North Holland, Amsterdam, 1981.

[245] Variational matrix-product-state approach to quantum impurity models / A. Weichselbaum, F. Verstraete, U. Schollwöck et al. // Phys. Rev. B. — 2009. — Vol. 80. — P. 165117.

[246] Verification of the cross 3D algorithm on quantum chemistry data / H.-J. Flad, B. N. Khoromskij, D. V. Savostyanov, E. E. Tyrtyshnikov // Rus. J. Numer. Anal. Math. Model. — 2008. — Vol. 23, no. 4. — P. 329–344.

[247] Vidal G. Efficient classical simulation of slightly entangled quantum computations // Phys. Rev. Lett. — 2003. — Vol. 91, no. 14. — P. 147902.

[248] von Petersdorff T., Schwab Ch. Numerical solution of parabolic equations in high dimensions // ESAIM: Mathematical Modelling and Numerical Analysis. — 2004. — Vol. 38, no. 01. — P. 93–127. — URL: http://dx.doi.org/10.1051/m2an:2004005.

[249] White Steven R. Density matrix formulation for quantum renormalization groups // Phys. Rev. Lett. — 1992. — Vol. 69, no. 19. — P. 2863–2866.

[250] White Steven R. Density-matrix algorithms for quantum renormalization groups // Phys. Rev. B. — 1993. — Vol. 48, no. 14. — P. 10345–10356.

[251] White Steven R. Spin gaps in a frustrated Heisenberg model for CaV4O9 // Phys. Rev. Lett. — 1996. — Vol. 77, no. 17. — P. 3633–3636.

[252] White Steven R. Density matrix renormalization group algorithms with a single center site // Phys. Rev. B. — 2005. — Vol. 72, no. 18. — P. 180403.

[253] Wilson K. G. The renormalization group: Critical phenomena and the Kondo problem // Rev. Mod. Phys. — 1975. — Vol. 47, no. 4. — P. 773–840.