

Институт вычислительной математики
Российская академия наук
119333, Москва, Губкина 8
www.inm.ras.ru

$\left[\begin{array}{c} \text{публикации} \\ \text{И В М} \\ \text{Р А Н} \end{array} \right]$
pub.inm.ras.ru

Д. В. Савостьянов, Е. Е. Тыртыхников

ПРИВЛИЖЕННОЕ УМНОЖЕНИЕ ТЕНЗОРНЫХ
МАТРИЦ НА ОСНОВЕ ИНДИВИДУАЛЬНОЙ
ФИЛЬТРАЦИИ ФАКТОРОВ

Препринт №09-06

2009-03-10



Москва 2010

ПРИБЛИЖЕННОЕ УМНОЖЕНИЕ ТЕНЗОРНЫХ МАТРИЦ НА ОСНОВЕ ИНДИВИДУАЛЬНОЙ ФИЛЬТРАЦИИ ФАКТОРОВ

Д. В. Савостьянов, Е. Е. Тыртышников

*Институт вычислительной математики РАН, 119333 Москва, Губкина 8,
dmitry.savostyanov@gmail.com, tee@inm.ras.ru*

10 марта 2009 г.

Аннотация. Предлагаются алгоритмы приближенного вычисления произведения матриц $\tilde{\mathbf{C}} \approx \mathbf{C} = \mathbf{A} \cdot \mathbf{B}$, где матрицы \mathbf{A} и \mathbf{B} заданы тензорным разложением в каноническом формате или в формате Таккера ранга r . Матрица \mathbf{C} в виде полного массива не вычисляется. Вместо этого она представляется сначала аналогичным разложением с избыточным значением ранга, а затем переаппроксимируется (сжимается) с целью уменьшения ранга в рамках заданной точности.

Известные алгоритмы переаппроксимации в данном случае требуют хранения массива из r^{2d} элементов, где d — размерность пространства. Из-за ограничений по памяти и быстродействию они неприменимы уже для типичных значений $d = 3$ и $r \sim 30$. В данной работе предлагаются методы, основанные на аппроксимации модовых факторов для \mathbf{C} по индивидуально выбранным критериям точности. В качестве приложения рассматривается вычисление трехмерного потенциала Кулона. Показано, что предложенные методы эффективны, когда значение r достигает нескольких сотен, а сложность операций по переаппроксимации (сжатию) \mathbf{C} невелика по сравнению с предварительным вычислением факторов тензорного разложения \mathbf{C} с избыточным значением ранга.

Ключевые слова: многомерные массивы, многомерные операторы, малопараметрические представления, каноническое разложение, разложение Таккера, скелетонная аппроксимация, малоранговые матрицы, сжатие данных, быстрая рекомпрессия, потенциал Кулона.

[§]Работа поддержана грантом РФФИ 08-01-00115, грантом РФФИ/DFG 09-01-91332 и программой Приоритетных исследований Отделения математических наук РАН.

1. Введение

Поскольку d -мерный массив размера n по каждому направлению содержит n^d элементов, для эффективного решения d -мерных задач необходимы специальные форматы представления данных и вычислительные схемы, основанные на данных форматах. Уход от экспоненциальной зависимости от d является серьезной открытой проблемой. Тем не менее, имеется ряд работ, в которых предлагаются форматы данных, в которых число параметров зависит от n линейно или почти линейно [1, 2, 3, 4, 5, 6, 7, 8, 9].

В этой работе мы решаем следующую задачу: для матриц \mathbf{A} и \mathbf{B} , заданных некоторым тензорным разложением, требуется найти приближенное тензорное разложение их произведения $\tilde{\mathbf{C}} \approx \mathbf{C} = \mathbf{A} \cdot \mathbf{B}$, число определяющих параметров которого меньше, чем у точного разложения \mathbf{C} . Это дает основу для построения приближенных пространств Крылова или приближенного обращения матрицы методом Ньютона [1, 6, 8, 10, 11].

В [12] мы рассмотрели эту задачу для основных тензорных форматов и предложили метод решения, обладающий сублинейной сложностью по полному числу данных. Однако, при этом выполняется *глобальная фильтрация* массива, размер которого экспоненциально растет по d . В данной работе мы предлагаем методы, основанные на предварительной *индивидуальной фильтрации* модовых массивов, размер которых не зависит от d . Существенно то, что малая погрешность в отдельной моде может сильно изменить элементы всего тензора. Поэтому независимая фильтрация модовых факторов требует аккуратного выбора индивидуальных критериев точности аппроксимации.

Общая схема умножения с независимой фильтрацией описана в разделе 2. В разделе 3 приведен метод расчета индивидуальных критериев, гарантирующих требуемую точность ответа. В разделе 4 обсуждаются различные методы мало-ранговой аппроксимации. Там же предложен быстрый алгоритм сжатия на основе неполных крестовых аппроксимаций. В разделе 5 разработанные методы прилагаются к вычислению трехмерного потенциала Кулона. В приложении А мы приводим алгоритмы вычисления индивидуальных критериев в трехмерном случае, выраженные через вызовы оптимизированных процедур библиотеки BLAS/LAPACK.

Матрицы и векторы рассматриваются как многомерные массивы (тензоры). Под *тензором* размерности d мы понимаем массив $\mathbf{B} = [b_{j_1 j_2 \dots j_d}]$ с d индексами. Отдельные индексы и соответствующие им направления называются *модами*, а число значений индексов — *модовыми размерами*. В случае модовых размеров n_1, \dots, n_d говорят, что тензор имеет размер $n_1 \times \dots \times n_d$.

Тензор может возникать, например, при дискретизации функции на тензорной сетке. Оператор, действующий на подобную функцию, будет иметь дискретный аналог в виде матрицы $\mathbf{A} = [a_{(i_1 i_2 \dots i_d)(j_1 j_2 \dots j_d)}]$. Для простоты будем считать, что все модовые размеры равны n , а также откажемся от двухуровневой индексации и будем писать $\mathbf{B} = [b_{ij \dots k}]$ и $\mathbf{A} = [a_{(i'j' \dots k')(ij \dots k)}]$ (предлагаемые методы выглядят единообразно в случае $d = 3$ и в пространствах большей размерности).

Для компактного представления тензоров используются *канонический формат* и *формат Таккера*. Канонический формат [13] для вектора \mathbf{B} имеет вид

$$\mathbf{B} = \sum_{s=1}^R \mathbf{u}_s \otimes \mathbf{v}_s \otimes \dots \otimes \mathbf{w}_s, \quad \text{или} \quad b_{ij\dots k} = \sum_{s=1}^R u_{is} v_{js} \dots w_{ks}, \quad (1)$$

где символом « \otimes » обозначено прямое (внешнее, кронекерово) произведение векторов. Аналогичное представление для матрицы \mathbf{A} записывается в виде

$$\mathbf{A} = \sum_{s=1}^R \mathbf{a}_s \otimes \mathbf{b}_s \otimes \dots \otimes \mathbf{c}_s. \quad (2)$$

При этом предполагается выполнение обозначенного ниже символом « \rightsquigarrow » *перематрирования*

$$a_{(i'j'\dots k')(ij\dots k)} \rightsquigarrow a_{(i'i)(j'j)\dots(k'k)} = \sum_{s=1}^R a_{(ii')s} b_{(jj')s} \dots c_{(kk')s},$$

которое переставляет элементы и приводит к d -мерному массиву с парными мультииндексами $(i'i), (j'j), \dots, (k'k)$, объединяющими в один «длинный индекс» индексы для соответствующих строчных и столбцовых мод.

Минимальное число слагаемых в возможных представлениях (1),(2) называют *тензорным* или *каноническим рангом*. Если указанные равенства справедливы с некоторой погрешностью, то говорят о *канонической аппроксимации* тензоров.

Разложение Таккера [14] для вектора \mathbf{B} имеет вид

$$\mathbf{B} = \mathbf{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \dots \times_d \mathbf{W} = \sum_{a=1}^{r_1} \sum_{b=1}^{r_2} \dots \sum_{c=1}^{r_d} g_{ab\dots c} \mathbf{u}_a \otimes \mathbf{v}_b \otimes \dots \otimes \mathbf{w}_c, \quad (3)$$

где тензор коэффициентов $\mathbf{G} = [g_{ab\dots c}]$ называется *ядром разложения*, а матрицы $\mathbf{U} = [\mathbf{u}_a]$, $\mathbf{V} = [\mathbf{v}_b]$, \dots , $\mathbf{W} = [\mathbf{w}_c]$ — *модовыми факторами*. Для матриц разложение Таккера выписывается аналогично (при той же схеме группировки индексов, как и в случае канонического разложения). Символом « \times_k » в (3) обозначено умножение тензора на матрицу по k -му направлению. В частности, запись $\mathbf{C} = \mathbf{B} \times_2 \mathbf{M}$ означает $c_{ij\dots k} = \sum_{j'} m_{jj'} b_{ij'\dots k}$. Пределы суммирования r_1, \dots, r_d называются *модовыми рангами* тензора. Для простоты можно считать, что $r_1 = \dots = r_d = r$ и $R = r$, но предлагаемые ниже алгоритмы применимы, конечно, и в тех случаях, когда все модовые ранги и модовые размеры различны.

Для краткости канонический формат будем называть *C-форматом*, а формат Таккера — *T-форматом*. Для матриц \mathbf{A} и \mathbf{B} допустимы всевозможные сочетания форматов C и T, но для \mathbf{C} применяется только T-формат. Мы налагаем это ограничение для того, чтобы уйти от сложной задачи переаппроксимации в C-формате. Среди множества предложенных для нее алгоритмов [13, 15, 16, 17, 5, 18, 19, 20] нет ни одного достаточно надежного. Напротив, задача сжатия в T-формате решается с помощью сингулярного разложения [21, 22], причем существуют и методы

сублинейной по количеству данных сложности [23, 24] (об их развитии см. [10, 25]). Таким образом, мы будем изучать умножение матриц в случаях, которые символически можно обозначить как $CT \rightarrow T$ и $TT \rightarrow T$. В каждом случае есть алгоритмы с одной и той же асимптотикой по n [12]. Но практические вычисления требуют более тонкого анализа, включающего сравнение этапов, сложность которых определяется только рангом r .

При записи формул числа обозначаются строчными буквами, векторы строчными жирными, матрицы заглавными, а тензоры жирными заглавными. Запись $A =: QR$ определяет правую часть как некоторое разложение матрицы A . Если индексы тензора сгруппированы в два мультииндекса, то он естественным образом определяет матрицу и может применяться матричное обозначение. Пределы суммирования опускаются в предположении, что индексы i, j, \dots, k пробегает значения от 1 до n , а индексы a, b, c, p, q, s, t от 1 до r . При суммировании по паре индексов полагаем, что каждый индекс независимо пробегает свой интервал. Фробениусова норма тензора и скалярное произведение тензоров определяются следующим образом:

$$\|A\|_F^2 \equiv \sum_{ij\dots k} |a_{ij\dots k}|^2, \quad \langle B, C \rangle \equiv \sum_{ij\dots k} b_{ij\dots k} \bar{c}_{ij\dots k}.$$

2. Общая схема умножения тензорных матриц

Пусть A и B заданы в Т-формате (С-формат можно рассматривать как Т-формат с полностью диагональным ядром):

$$\begin{aligned} A &= G \times_1 U^{(A)} \times_2 V^{(A)} \times_3 \dots \times_d W^{(A)} = \sum_{pq\dots s} g_{pq\dots s} u_p^{(A)} \otimes v_q^{(A)} \otimes \dots \otimes w_s^{(A)}, \\ B &= H \times_1 U^{(B)} \times_2 V^{(B)} \times_3 \dots \times_d W^{(B)} = \sum_{ab\dots c} h_{ab\dots c} u_a^{(B)} \otimes v_b^{(B)} \otimes \dots \otimes w_c^{(B)}. \end{aligned} \quad (4)$$

Тогда для $C = A \cdot B$ получаем представление в Т-формате

$$\begin{aligned} C &= \sum_{pq\dots s} \sum_{ab\dots c} g_{pq\dots s} h_{ab\dots c} (u_p^{(A)} \cdot u_a^{(B)}) \otimes (v_q^{(A)} \cdot v_b^{(B)}) \otimes \dots \otimes (w_s^{(A)} \cdot w_c^{(B)}) = \\ &= \sum_{pq\dots s} \sum_{ab\dots c} g_{pq\dots s} h_{ab\dots c} u_{pa} \otimes v_{qb} \otimes \dots \otimes w_{sc} = F(G, H) \times_1 U \times_2 V \times_3 \dots \times_d W \end{aligned} \quad (5)$$

с модовыми факторами $U = [u_{pa}]$, $V = [v_{qb}]$, \dots , $W = [w_{sc}]$ и ядром $F(G, H)$, имеющим размер $r^2 \times r^2 \times \dots \times r^2$ и некоторую специальную структуру. Для получения и сжатия Т-формата для C в [12] предложен следующий алгоритм.

Алгоритм 1. Умножение с глобальной фильтрацией

1: Вычислить модовые факторы

$$u_{pa} := u_p^{(A)} \cdot u_a^{(B)}, \quad v_{qb} := v_q^{(A)} \cdot v_b^{(B)}, \quad \dots \quad w_{sc} := w_s^{(A)} \cdot w_c^{(B)}.$$

Сложность этого шага пропорциональна r^2 , зависит от n и выполняемой операции. В табл. 1 приведены оценки сложности для трех случаев: (CV) матрица \mathbf{A} является d -уровневой теплицевой, \mathbf{B} — вектор (свертка); (MV) умножение матрицы на вектор; (MM) умножение матрицы на матрицу. Этот шаг может быть очень дорогим, но обладает большим внутренним параллелизмом, так как все умножения могут быть произведены на разных процессорах независимо.

- 2: Провести ортогонализацию модовых факторов (например, с помощью QR-разложения)

$$\mathbf{U} =: \hat{\mathbf{U}} \mathbf{U}', \quad \mathbf{V} =: \hat{\mathbf{V}} \mathbf{V}', \quad \dots, \quad \mathbf{W} =: \hat{\mathbf{W}} \mathbf{W}', \quad (6)$$

где матрицы $\mathbf{U}', \mathbf{V}', \dots, \mathbf{W}'$ имеют размер $r^2 \times r^2$, а $\hat{\mathbf{U}}, \hat{\mathbf{V}}, \dots, \hat{\mathbf{W}}$ содержат r^2 ортогональных столбцов размера N , (см. табл. 1). Сложность этого шага $O(Nr^2)$, что также весьма дорого, особенно в (MM)-случае, когда размер модового вектора $N = n^2$ велик, поскольку каждым «вектором» является матрица $n \times n$. Возможности параллелизации QR-разложения достаточно ограничены.

- 3: Сформировать новое ядро размера $r^2 \times r^2 \times \dots \times r^2$ по формуле

$$\hat{\mathbf{F}} := \mathbf{F} \times_1 \mathbf{U}' \times_2 \mathbf{V}' \times_3 \dots \times_d \mathbf{W}', \quad (7)$$

что требует $O(dr^{2d+2})$ операций. Теперь произведение представляется в виде

$$\mathbf{C} = \hat{\mathbf{F}} \times_1 \hat{\mathbf{U}} \times_2 \hat{\mathbf{V}} \times_3 \dots \times_d \hat{\mathbf{W}}. \quad (8)$$

- 4: Для тензора $\hat{\mathbf{F}}$ найти аппроксимацию в T-формате

$$\hat{\mathbf{F}} \approx \mathbf{F}_\clubsuit \times_1 \mathbf{U}_\clubsuit \times_2 \mathbf{V}_\clubsuit \times_3 \dots \times_d \mathbf{W}_\clubsuit,$$

например, с помощью сингулярных разложений *разверток* тензора по каждой моде — матриц, составленных из всех векторов данной моды (см. [21]). Сложность этого метода составляет $O(dr^{2d+2})$.

После того, как (8) получено, ответ представляется с той же точностью в виде

$$\mathbf{C} \approx \tilde{\mathbf{C}} = \mathbf{F}_\clubsuit \times_1 (\hat{\mathbf{U}} \mathbf{U}_\clubsuit) \times_2 (\hat{\mathbf{V}} \mathbf{V}_\clubsuit) \times_3 \dots \times_d (\hat{\mathbf{W}} \mathbf{W}_\clubsuit).$$

Таблица 1. Стоимость шагов в алгоритме умножения с глобальной фильтрацией

	Модовое умножение	Ортогонализация факторов	размер вектора N
CV	$O(n \log n) r^2$	$O(nr^4)$	n
MV	$O(n^2) r^2$	$O(nr^4)$	n
MM	$O(n^3) r^2$	$O(n^2 r^4)$	n^2

Таблица 2. Память на хранение массива размером r^d , МБ

	$d = 3$	$d = 4$	$d = 5$	$d = 6$
$r = 15$	0.026	0.4	5.8	87
$r = 30$	0.2	6.2	185	5560
$r = 50$	0.95	47	2384	119210
$r = 100$	7.7	763	76300	много

Постановка задачи предполагает, что ядра \mathbf{G}, \mathbf{H} размера r^d помещаются в доступной памяти, однако для тензоров \mathbf{F} и $\hat{\mathbf{F}}$ размера r^{2d} памяти может оказаться недостаточно — см. табл. 2. В этом случае сжатие с помощью глобальной фильтрации невыполнимо. Ниже предлагается алгоритм индивидуальной фильтрации модовых факторов, требующий существенно меньших затрат.

Идея индивидуальной фильтрации проста: в представлении (5) предлагается аппроксимировать факторы $\mathbf{U}, \mathbf{V}, \dots, \mathbf{W}$ матрицами ранга $\rho \leq r^2$, заменив полную ортогонализацию (6) приближенным разложением с матрицами $\mathbf{U}', \mathbf{V}', \dots, \mathbf{W}'$ размера $r^2 \times \rho$ и факторами $\hat{\mathbf{U}}, \hat{\mathbf{V}}, \dots, \hat{\mathbf{W}}$, содержащими ρ ортогональных столбцов. В результате промежуточное ядро $\hat{\mathbf{F}}$ будет иметь ρ^d элементов, и при условии $\rho \ll r^2$ затраты на его вычисление, хранение и сжатие существенно снизятся. Ключевым здесь является выбор *индивидуальных критериев* точности аппроксимации модовых факторов. Для их расчета применим следующую процедуру.

Ошибка, внесенная в фактор \mathbf{U} , приводит к возмущению тензора

$$\Delta_{\mathbf{u}} \mathbf{C} = \mathbf{F}(\mathbf{G}, \mathbf{H}) \times_1 \Delta \mathbf{U} \times_2 \mathbf{V} \times_3 \dots \times_d \mathbf{W}, \quad (9)$$

которое можно записать в виде произведения матриц

$$\Delta_{\mathbf{u}} \mathbf{C} = \Delta \mathbf{U} \mathbf{C}_{\mathbf{u}}^T, \quad \mathbf{C}_{\mathbf{u}} = \mathbf{C}_{\mathbf{u}}(\mathbf{G}, \mathbf{H}, \mathbf{V}, \dots, \mathbf{W}).$$

После этого для нормы $\|\Delta_{\mathbf{u}} \mathbf{C}\|_{\mathbf{F}}$ можно предложить оценки

$$\|\Delta_{\mathbf{u}} \mathbf{C}\|_{\mathbf{F}} \leq \|\Delta \mathbf{U}\|_{\mathbf{F}} \|\mathbf{C}_{\mathbf{u}}\|_{\mathbf{F}}, \quad \|\Delta_{\mathbf{u}} \mathbf{C}\|_{\mathbf{F}} \leq \|\Delta \mathbf{U}\|_{\mathbf{F}} \|\mathbf{C}_{\mathbf{u}}\|_2,$$

из которых первая проще вычисляется, а вторая позволяет получить меньшее значение ρ . Матрицу $\mathbf{C}_{\mathbf{u}}$ назовем *дополнением* к \mathbf{U} , а ее норму — *числом обусловленности* по \mathbf{U} .

Суммируя все погрешности, внесенные в \mathbf{C} при возмущении факторов, и пренебрегая старшими степенями, получим

$$\begin{aligned} \|\Delta \mathbf{C}\|_{\mathbf{F}} &\lesssim \|\Delta \mathbf{U} \mathbf{C}_{\mathbf{u}}^T\|_{\mathbf{F}} + \|\Delta \mathbf{V} \mathbf{C}_{\mathbf{v}}^T\|_{\mathbf{F}} + \dots + \|\Delta \mathbf{W} \mathbf{C}_{\mathbf{w}}^T\|_{\mathbf{F}} \leq \\ &\leq \|\Delta \mathbf{U}\|_{\mathbf{F}} \|\mathbf{C}_{\mathbf{u}}\|_{2\mathbf{F}} + \|\Delta \mathbf{V}\|_{\mathbf{F}} \|\mathbf{C}_{\mathbf{v}}\|_{2\mathbf{F}} + \dots + \|\Delta \mathbf{W}\|_{\mathbf{F}} \|\mathbf{C}_{\mathbf{w}}\|_{2\mathbf{F}}, \end{aligned} \quad (10)$$

где $\|\cdot\|_{2\mathbf{F}}$ означает, что для оценки может быть выбрана как фробениусова, так и спектральная норма. Теперь мы можем указать границы погрешности для модовых

факторов

$$\|\Delta U\|_F \leq \varepsilon_{\text{abs}}/(d\|C_u\|_{2|F}), \quad \dots \quad \|\Delta W\|_F \leq \varepsilon_{\text{abs}}/(d\|C_w\|_{2|F}),$$

выполнение которых с учетом (10) гарантирует, что $\|\Delta C\|_F \leq \varepsilon_{\text{abs}}$. На практике часто удобнее работать с относительной погрешностью; соответствующие критерии требуют вычисления $\|C\|_F$ и имеют вид

$$\varepsilon_u = \frac{\varepsilon\|C\|_F}{d\|C_u\|_{2|F}\|U\|_F}, \quad \dots \quad \varepsilon_w = \frac{\varepsilon\|C\|_F}{d\|C_w\|_{2|F}\|W\|_F}. \quad (11)$$

Теорема 1. При внесении в модовые факторы тензора (5) погрешностей

$$\|\Delta U\|_F \leq \varepsilon_u\|U\|_F, \quad \|\Delta V\|_F \leq \varepsilon_v\|V\|_F, \quad \dots \quad \|\Delta W\|_F \leq \varepsilon_w\|W\|_F, \quad (12)$$

в которых индивидуальные границы фильтрации $\varepsilon_u, \varepsilon_v, \dots, \varepsilon_w$ определены согласно (11), гарантируется $\|\Delta C\|_F \leq \varepsilon\|C\|_F$.

Доказательство теоремы состоит в подстановке (11) в (12) и затем в (10). На ней основан следующий алгоритм.

Алгоритм 2. Умножение с индивидуальной фильтрацией

- 1: Модовое умножение выполняется так же, как в алгоритме 1.
- 2: Вычислить матрицы Грама $\langle u_{a_1 b_1}, u_{a'_1 b'_1} \rangle, \langle v_{a_2 b_2}, v_{a'_2 b'_2} \rangle, \dots, \langle w_{a_d b_d}, w_{a'_d b'_d} \rangle$. Сложность этого шага $d\tau^4 N$ (см. табл. 1), что совпадает по порядку с затратами на ортогонализацию модовых факторов (шаг 2 алгоритма 1). Ресурсы по параллелизации этого шага значительно выше, чем у QR-разложения.
- 3: а) Вычислить норму тензора $\|C\|_F$.
 б) Вычислить нормы дополнений $\|C_x\|_{2|F}$ для всех факторов $X = U, V, \dots, W$.
 в) Получить индивидуальные границы фильтрации по формуле (11).

Сложность этих вычислений зависит от структуры ядра F (то есть форматов сомножителей), но определяется только рангами, а не модовыми размерами. Алгоритмы расчета величин $\|C\|_F$ и $\|C_x\|_{2|F}$ для СС-, СТ- и ТТ-случаев мы приведем в разделе 3.

- 4: Найти малоранговое приближение (индивидуальную фильтрацию) факторов

$$U =: \hat{U} U' + \Delta U, \quad V =: \hat{V} V' + \Delta V, \quad \dots \quad W =: \hat{W} W' + \Delta W, \quad (13)$$

где факторы $\hat{U}, \hat{V}, \dots, \hat{W}$ содержат по ρ ортонормированных столбцов, матрицы перехода U', V', \dots, W' имеют размер $\rho \times r^2$, а погрешности $\Delta U, \Delta V, \dots, \Delta W$ ограничены согласно (12). Сложность этого шага зависит от выбора метода аппроксимации. Детальное обсуждение мы проведем в разделе 4.

5: Сформировать промежуточное ядро $\hat{\mathbf{F}}$ размера $\rho \times \rho \times \dots \times \rho$ по формуле (7) и сжать его, как указано в последнем шаге алгоритма 1.

Для иллюстрации метода индивидуальной фильтрации в случае $d = 2$ рассмотрим матрицу ранга 3

$$\mathbf{A} = \mathbf{Y}\mathbf{Z}^T = [\mathbf{y}_1 \quad h^2\mathbf{y}_2 \quad h^{1.5}\mathbf{y}_3] [\mathbf{h}\mathbf{z}_1 \quad \mathbf{z}_2 \quad h^{1.5}\mathbf{z}_3]^T = h\mathbf{y}_1\mathbf{z}_1^T + h^2\mathbf{y}_2\mathbf{z}_2^T + h^3\mathbf{y}_3\mathbf{z}_3^T$$

с ортонормированными системами векторов $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ и $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$. Попытка индивидуальной фильтрации факторов \mathbf{Y}, \mathbf{Z} без аккуратного выбора критериев точности приведет при достаточно малом h к неверной аппроксимации матрицы

$$\mathbf{Y} = [\mathbf{y}_1 \quad 0 \quad 0] + \Delta\mathbf{Y}, \quad \mathbf{Z} = [0 \quad \mathbf{z}_2 \quad 0] + \Delta\mathbf{Z}, \quad \|\Delta\mathbf{Y}\|_F \sim h^{1.5}, \quad \|\Delta\mathbf{Z}\|_F \sim h,$$

$$\tilde{\mathbf{A}} = [\mathbf{y}_1 \quad 0 \quad 0] [0 \quad \mathbf{z}_2 \quad 0]^T = 0.$$

Поступим теперь согласно алгоритму 2. Найдем норму $\|\mathbf{A}\|_F$:

$$\|\mathbf{A}\|_F^2 = \langle \mathbf{A}, \mathbf{A} \rangle = \left\langle \sum_s \mathbf{y}_s \otimes \mathbf{z}_s, \sum_t \mathbf{y}_t \otimes \mathbf{z}_t \right\rangle = \sum_{st} ((\mathbf{Y}^T \mathbf{Y}) \odot (\mathbf{Z}^T \mathbf{Z}))_{st},$$

где символом « \odot » обозначено поэлементное (адамарово) произведение матриц Грама

$$\mathbf{Y}^T \mathbf{Y} = \text{diag}(1, h^4, h^3), \quad \mathbf{Z}^T \mathbf{Z} = \text{diag}(h^2, 1, h^3).$$

Отсюда $\|\mathbf{A}\|_F = h\sqrt{1 + h^2 + h^4}$. Дополнения $\mathbf{C}_y \stackrel{\text{def}}{=} \mathbf{Z}$, $\mathbf{C}_z \stackrel{\text{def}}{=} \mathbf{Y}$ имеют единичные спектральные нормы, и индивидуальные критерии (11) принимают вид

$$\varepsilon_u^{\text{abs}} = \varepsilon_v^{\text{abs}} = \varepsilon h \sqrt{1 + h^2 + h^4}/2.$$

При выборе относительной точности аппроксимации $\varepsilon = h^{0.4}$ факторы фильтруются в виде

$$\mathbf{Y} = [\mathbf{y}_1 \quad 0 \quad 0] + \Delta\mathbf{Y}, \quad \mathbf{Z} = [\mathbf{h}\mathbf{z}_1 \quad \mathbf{z}_2 \quad 0] + \Delta\mathbf{Z}, \quad \|\Delta\mathbf{Y}\|_F \sim h^{1.5}, \quad \|\Delta\mathbf{Z}\|_F \sim h^{1.5},$$

$$\tilde{\mathbf{A}} = [\mathbf{y}_1 \quad 0 \quad 0] [\mathbf{h}\mathbf{z}_1 \quad \mathbf{z}_2 \quad 0]^T = h\mathbf{y}_1\mathbf{z}_1^T,$$

что является наилучшей аппроксимацией в пределах заданной точности. При выборе относительной точности аппроксимации $\varepsilon = h^{0.6}$ сжатие факторов меньше:

$$\mathbf{Y} = [\mathbf{y}_1 \quad 0 \quad h^{1.5}\mathbf{y}_3] + \Delta\mathbf{Y}, \quad \mathbf{Z} = [\mathbf{h}\mathbf{z}_1 \quad \mathbf{z}_2 \quad h^{1.5}\mathbf{z}_3] + \Delta\mathbf{Z}, \quad \|\Delta\mathbf{Y}\|_F \sim h^2, \quad \|\Delta\mathbf{Z}\|_F = 0,$$

$$\tilde{\mathbf{A}} = [\mathbf{y}_1 \quad 0 \quad h^{1.5}\mathbf{y}_3] [\mathbf{h}\mathbf{z}_1 \quad \mathbf{z}_2 \quad h^{1.5}\mathbf{z}_3]^T = h\mathbf{y}_1\mathbf{z}_1^T + h^3\mathbf{y}_3\mathbf{z}_3^T,$$

и аппроксимация содержит «лишний» член порядка h^3 . В данном случае фильтрация модовых факторов позволила снизить ранг, но наилучшая аппроксимация в пределах заданной точности должна осуществляться с помощью глобальной фильтрации ядра размером 2×2 (а не 3×3 , как было бы без фильтрации факторов).

3. Нормы тензора и дополнений

Приведем методы расчета величин $\|\mathbf{C}\|_F$ и $\|C_u\|_F$ или $\|C_u\|_2$ (для факторов $X = V, \dots, W$, нормы дополнений выписываются аналогично), необходимых для вычисления индивидуальных порогов фильтрации модовых факторов согласно (11) и оценим их сложность при $d = 3$. Для этого используются матрицы Грама модовых факторов и легко проверяемое тождество

$$\langle \mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}, \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \rangle = \langle \mathbf{u}, \mathbf{a} \rangle \langle \mathbf{v}, \mathbf{b} \rangle \langle \mathbf{w}, \mathbf{c} \rangle.$$

3.1. Случай СС

$$\mathbf{A} = \sum_s \mathbf{u}_s^{(A)} \otimes \mathbf{v}_s^{(A)} \otimes \mathbf{w}_s^{(A)}, \quad \mathbf{B} = \sum_t \mathbf{u}_t^{(B)} \otimes \mathbf{v}_t^{(B)} \otimes \mathbf{w}_t^{(B)}, \quad (14)$$

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} = \sum_{st} \left(\mathbf{u}_s^{(A)} \cdot \mathbf{u}_t^{(B)} \right) \otimes \left(\mathbf{v}_s^{(A)} \cdot \mathbf{v}_t^{(B)} \right) \otimes \left(\mathbf{w}_s^{(A)} \cdot \mathbf{w}_t^{(B)} \right) = \sum_{st} \mathbf{u}_{st} \otimes \mathbf{v}_{st} \otimes \mathbf{w}_{st}. \quad (15)$$

Квадрат нормы результирующего тензора определяется из равенства

$$\begin{aligned} \|\mathbf{C}\|_F^2 = \langle \mathbf{C}, \mathbf{C} \rangle &= \left\langle \sum_{st} \mathbf{u}_{st} \otimes \mathbf{v}_{st} \otimes \mathbf{w}_{st}, \sum_{s't'} \mathbf{u}_{s't'} \otimes \mathbf{v}_{s't'} \otimes \mathbf{w}_{s't'} \right\rangle = \\ &= \sum_{st, s't'} \langle \mathbf{u}_{st}, \mathbf{u}_{s't'} \rangle \langle \mathbf{v}_{st}, \mathbf{v}_{s't'} \rangle \langle \mathbf{w}_{st}, \mathbf{w}_{s't'} \rangle \end{aligned}$$

как сумма всех элементов поэлементного произведения матриц Грама. Сложность этой операции составляет $O(r^4)$. Далее согласно алгоритму 2

$$\begin{aligned} (\Delta_u \mathbf{C})_{ijk} &= \sum_{st} \Delta u_{ist} v_{jst} w_{kst}, \quad \Delta_u \mathbf{C} = \Delta \mathbf{U} C_u^T, \quad (C_u)_{jk, st} = v_{jst} w_{kst}, \\ \|C_u\|_F^2 &= \sum_{jk, st} |v_{jst} w_{kst}|^2 = \sum_{st} \|\mathbf{v}_{st}\|_F^2 \|\mathbf{w}_{st}\|_F^2. \end{aligned} \quad (16)$$

Чтобы получить $\|C_u\|_F$, требуется $O(r^2)$ операций. Чтобы найти $\|C_u\|_2$, достаточно вычислить старшее собственное значение матрицы

$$C_u^T C_u = (V^T V) \odot (W^T W), \quad (17)$$

что требует $O(r^4)$ действий.

3.2. Случай СТ

$$\mathbf{A} = \sum_s \mathbf{u}_s^{(A)} \otimes \mathbf{v}_s^{(A)} \otimes \mathbf{w}_s^{(A)}, \quad \mathbf{B} = \sum_{abc} g_{abc} \mathbf{u}_a^{(B)} \otimes \mathbf{v}_b^{(B)} \otimes \mathbf{w}_c^{(B)}, \quad (18)$$

$$\begin{aligned} \mathbf{C} = \mathbf{A} \cdot \mathbf{B} &= \sum_s \sum_{abc} g_{abc} \left(\mathbf{u}_s^{(A)} \cdot \mathbf{u}_a^{(B)} \right) \otimes \left(\mathbf{v}_s^{(A)} \cdot \mathbf{v}_b^{(B)} \right) \otimes \left(\mathbf{w}_s^{(A)} \cdot \mathbf{w}_c^{(B)} \right) = \\ &= \sum_{s, abc} g_{abc} \mathbf{u}_{sa} \otimes \mathbf{v}_{sb} \otimes \mathbf{w}_{sc} = \sum_s \mathbf{G} \times_1 \mathbf{U}_s \times_2 \mathbf{V}_s \times_3 \dots \times_d \mathbf{W}_s. \end{aligned} \quad (19)$$

Норма результата выписывается в виде

$$\begin{aligned}\|\mathbf{C}\|_F^2 &= \left\langle \sum_{s,abc} g_{abc} \mathbf{u}_{sa} \otimes \mathbf{v}_{sb} \otimes \mathbf{w}_{sc}, \sum_{s',a'b'c'} g_{a'b'c'} \mathbf{u}_{s'a'} \otimes \mathbf{v}_{s'b'} \otimes \mathbf{w}_{s'c'} \right\rangle = \\ &= \sum_{ss'} \sum_{abc} \sum_{a'b'c'} g_{abc} g_{a'b'c'} \langle \mathbf{u}_{sa}, \mathbf{u}_{s'a'} \rangle \langle \mathbf{v}_{sb}, \mathbf{v}_{s'b'} \rangle \langle \mathbf{w}_{sc}, \mathbf{w}_{s'c'} \rangle\end{aligned}\quad (20)$$

и может быть вычислена за $O(r^6)$ операций (см. алгоритм **CTnorm** из приложения **A**). Норма дополнения \mathbf{C}_u , возникающего при оценке возмущения тензора

$$\begin{aligned}(\Delta_u \mathbf{C})_{ijk} &= \sum_{s,abc} g_{abc} \Delta u_{isa} v_{jsb} w_{ksc} = \sum_{sa} \Delta u_{isa} \left(\sum_{bc} g_{abc} v_{jsb} w_{ksc} \right), \\ \Delta_u \mathbf{C} &= \Delta U \mathbf{C}_u^T, \quad (\mathbf{C}_u)_{jk,as} = \sum_{bc} g_{abc} v_{jsb} w_{ksc},\end{aligned}$$

определяется формулой

$$\begin{aligned}\|\mathbf{C}_u\|_F^2 &= \sum_{jk,as} \left| \sum_{bc} g_{abc} v_{jsb} w_{ksc} \right|^2 = \\ &= \sum_s \sum_{bc,b'c'} \langle g_{:,b,c}, g_{:,b',c'} \rangle \langle \mathbf{v}_{sb}, \mathbf{v}_{sb'} \rangle \langle \mathbf{w}_{sc}, \mathbf{w}_{sc'} \rangle,\end{aligned}\quad (21)$$

и вычисляется за $O(r^5)$ операций (см. алгоритм **CTcdf**). Величина $\|\mathbf{C}_u\|_2$ определяется старшим собственным значением матрицы $\mathbf{C}_u^T \mathbf{C}_u$, где

$$(\mathbf{C}_u^T \mathbf{C}_u)_{as,a's'} = \sum_{bc,b'c'} g_{abc} g_{a'b'c'} \langle \mathbf{v}_{sb}, \mathbf{v}_{s'b'} \rangle \langle \mathbf{w}_{sc}, \mathbf{w}_{s'c'} \rangle. \quad (22)$$

Для ее вычисления необходимо $O(r^6)$ операций (см. алгоритм **CTcnd2**).

3.3. Случай ТТ

Согласно (4) и (5) норма произведения

$$\|\mathbf{C}\|_F^2 = \sum_{pqs} \sum_{p'q's'} \sum_{abc} \sum_{a'b'c'} g_{pqs} g_{p'q's'} h_{abc} h_{a'b'c'} \langle \mathbf{u}_{pa}, \mathbf{u}_{p'a'} \rangle \langle \mathbf{v}_{qb}, \mathbf{v}_{q'b'} \rangle \langle \mathbf{w}_{sc}, \mathbf{w}_{s'c'} \rangle \quad (23)$$

вычисляется за $O(r^8)$ действий (см. алгоритм **TTnorm**), а дополнение имеет вид

$$(\mathbf{C}_u)_{jk,pa} = \sum_{qs,bc} g_{pqs} h_{abc} v_{jqb} w_{ksc}.$$

Число обусловленности $\|\mathbf{C}_u\|_F$ вычисляется по формуле

$$\begin{aligned}\|\mathbf{C}_u\|_F^2 &= \sum_{jk,pa} \left| \sum_{qs,bc} g_{pqs} h_{abc} v_{jqb} w_{ksc} \right|^2 = \\ &= \sum_{qs,bc} \sum_{q's',b'c'} \langle g_{:,q,s}, g_{:,q',s'} \rangle \langle h_{:,b,c}, h_{:,b',c'} \rangle \langle \mathbf{v}_{qb}, \mathbf{v}_{q'b'} \rangle \langle \mathbf{w}_{sc}, \mathbf{w}_{s'c'} \rangle.\end{aligned}\quad (24)$$

за $O(r^6)$ операций (см. алгоритм **TTcndf**). Для расчета $\|C_u\|_2$ ищется старшее собственное значение матрицы $C_u^T C_u$, где

$$(C_u^T C_u)_{pa, p'a'} = \sum_{qs, bc} \sum_{q's', b'c'} g_{pqs} h_{abc} g_{p'q's'} h_{a'b'c'} \langle \mathbf{v}_{qb}, \mathbf{v}_{q'b'} \rangle \langle \mathbf{w}_{sc}, \mathbf{w}_{s'c'} \rangle, \quad (25)$$

и затраты увеличиваются до $O(r^8)$ (см. алгоритм **TTcnd2**).

3.4. Сравнение

(СС) Затраты на вычисление индивидуальных границ фильтрации минимальны (11) (см. табл. 3, слева). Этот случай рассматривается в [26] при вычислении трехмерных сверток, а также в [7, 11] при обращении матриц методом Ньютона. Однако после сжатия ответ получается в Т-формате, поэтому для продолжения расчета нужно либо отказаться от С-формата для обновляемых векторов, либо решать сложную задачу преобразования из формата Т в формат С. В случае $d = 3$ мы уверенно рекомендуем первое.

(СТ) Важен при итерационном решении интегральных уравнений (см. [23, 27]), когда матрица \mathbf{A} неизменна и записана С-формате (например, с помощью аналитического представления оператора). Тензорный вектор \mathbf{B} меняется на каждой итерации, но задается в Т-формате.

(ТТ) Этот случай возникает при обращении тензорных операторов [12] и при решении интегральных уравнений с операторами, заданными в Т-формате. Отметим, что Т-формат представления оператора может оказаться предпочтительнее С-формата, поскольку модовые ранги обычно меньше тензорных, что приводит к экономии на этапе вычисления одномерных умножений (шаг 1 алгоритмов 1 и 2). С другой стороны, стоимость накладных расходов по вычислению $\|\mathbf{C}\|_F$, $\|C_x\|_{2F}$ в ТТ-случае несколько выше (см. табл. 3, слева). Например, при $d = 3$ они составляют в СТ-случае $O(r^6)$, а в ТТ-случае $O(r^8)$. Заметим, что при $r \sim 70$ значение $r^2 \sim 5000$, что может быть сравнимо с модовым размером задачи. Таким образом, хотя стоимость третьего шага алгоритма 2 не зависит от n , она может быть сравнима с затратами на прочих шагах и даже превосходить их.

Таблица 3. Стоимость вычислений, необходимых для расчета $\varepsilon_u, \dots, \varepsilon_w$

	Точное вычисление			Оценка		
	$\ \mathbf{C}\ _F$	$\ C_x\ _F$	$\ C_x\ _2$	$\ \mathbf{C}\ _F$	$\ C_x\ _2$	
СС	$O(r^4)$	$O(r^2)$	$O(r^4)$			
СТ	$O(r^6)$	$O(r^5)$	$O(r^6)$	$O(r^5)$	$O(r^5)$	СТ
ТТ	$O(r^8)$	$O(r^6)$	$O(r^8)$	$O(r^6)$	$O(r^6)$	ТТ

Высокой точности при оценке чисел обусловленности не требуется. Поэтому очень важно изучить возможность грубой оценки с низкими затратами. Для этой цели ниже применяются методы неполной крестовой аппроксимации, снижающие затраты на вычисление $\|\mathbf{C}\|_F$ и $\|\mathbf{C}_x\|_2$, $x = u, v, w$, до значений, указанных в правой части таблицы 3.

4. Быстрые методы фильтрации

Сингулярное разложение матрицы является надежным инструментом получения малоранговых аппроксимаций, но имеет высокую вычислительную сложность. Более эффективными (и в достаточной степени надежными) оказываются методы на основе *скелетонного разложения* $\tilde{F} = UGV^T$, где U и V^T состоят из столбцов и строк матрицы F , а $G = B^{-1}$ получается обращением $r \times r$ -подматрицы B на пересечении *креста*, образованного выбранными в F столбцами и строками. Точность скелетонной аппроксимации существенно зависит от выбора B . В работах [28, 29, 30] показано, что хорошим выбором является подматрица *наибольшего объема*, то есть обладающая наибольшим модулем определителя среди всех подматриц размера $r \times r$. Известно, что поиск такой подматрицы есть NP-сложная задача, не разрешимая при сколь-либо существенных значениях размера и ранга. В качестве практической альтернативы можно рассматривать различные методы поиска подматриц, «достаточно близких» к искомой. Алгоритм такого рода был предложен в [31], а более детальное его изложение и обсуждение свойств возникающих *доминирующих* подматриц можно найти в [32].

Если в процессе выбора опорного креста на каждом шаге к существующему набору добавляется столбец и строка, на пересечении которых находится максимум модуля невязки, то алгоритм крестовой аппроксимации становится тождественен разложению Гаусса с полным выбором. Однако чтобы ускорить вычисления и достигнуть линейной сложности по максимальному размеру матрицы, можно отказаться от полного выбора и предложить разумный эвристический метод расширения креста. В итоге алгоритм аппроксимации может идти без вычисления всего массива элементов матрицы, принимая решение о следующем шаге или об остановке лишь на основе элементов текущего креста. Сама идея и ее первая реализация впервые представлены в [31], более простые версии затем получены в [33, 34, 2].

Для симметричных положительно определенных матриц алгоритм неполной крестовой аппроксимации существенно упрощается, поскольку максимальный по модулю элемент невязки на каждом шаге исключения находится на диагонали матрицы. В этом случае крестовой метод равносильен незавершенному разложению Холецкого; для достижения линейной по размеру матрицы сложности остается всего лишь предложить разумный критерий остановки, не требующий точного расчета погрешности. Мы будем использовать это разложение для оценки старшего собственного значения $\mathbf{C}_x^T \mathbf{C}_x$, которое очень быстро стабилизируется в ходе итераций. Фактически использование крестового метода позволяет нам оценивать $\|\mathbf{C}_x\|_2$ с точностью 1%, требуя лишь вычисления диагонали и нескольких (не более 5 во

всех экспериментах) столбцов матрицы $C_x^T C_x$. Это снижает стоимость вычислений в r^2 раз по сравнению с полным вычислением матрицы Грама.

Сформулируем теперь быстрый алгоритм умножения с индивидуальной фильтрацией, использующий преимущества крестовых методов. Для СС-случая аналогичный по существу алгоритм предложен в [35] при обсуждении быстрой компрессии из формата С (с относительно большим рангом) в формат Т. Здесь мы адаптируем эту схему для случаев СТ и ТТ.

Алгоритм 3. Быстрое умножение с индивидуальной фильтрацией

- 1: Выполнить одномерные умножения, как в алгоритме 2.
- 2: Аппроксимировать факторы U, V, \dots, W размера $N \times r^2$ матрицами ранга $\rho = sr$, как указано в (13). Параметр s выбирается эмпирически, можно начать с $s = 1$. Оценить погрешности $\|\Delta U\|_F, \|\Delta V\|_F, \dots, \|\Delta W\|_F$.
- 3: Вычислить матрицы Грама «недоопределенных» факторов U', V', \dots, W' размера $\rho \times r^2$. Используя их вместо матриц Грама полных факторов, получить оценки для чисел обусловленности $\|C_x\|_2$.
- 4: Вычислить (19) и (5), заменив факторы U, V, \dots, W «недоопределенными» матрицами U', V', \dots, W' . Тем самым получить тензор $\hat{\mathbf{F}}$ размера $\rho \times \rho \times \dots \times \rho$, представляющий ядро Т-формата для \mathbf{C} с факторами $\hat{U}, \hat{V}, \dots, \hat{W}$, полученными на шаге 2. Оценить норму $\|\mathbf{C}\|_F = \|\hat{\mathbf{F}}\|_F$.
- 5: Оценить погрешность нормы тензора по формуле (10)

$$\|\Delta \mathbf{C}\|_F \leq \sum_{x=U, V, \dots, W} \|C_x\|_2 \|\Delta X\|_F \stackrel{\text{def}}{=} \text{err}.$$

Если $\text{err} \leq 0.1 \|\mathbf{C}\|_F$, продолжить, иначе положить $s := s + 1$ и вернуться на шаг 2 для уточнения аппроксимации факторов.

- 6: Грубая аппроксимация завершена, величины $\|\mathbf{C}\|_F, \|C_x\|_2$ получены. Находим пороги фильтрации согласно (11) и продолжаем вычисления с шага 4 алгоритма 2.

Оценим сложность каждого шага данного алгоритма при $d = 3$ в предположении, что шаги 2 и 3 реализуются на основе метода неполной крестовой аппроксимации.

1. Одномерные умножения требуют от $O(n \log n)r^2$ до $O(n^3)r^2$, см. табл. 1.
2. Применение неполного крестового метода для аппроксимации матриц размера $N \times r^2$ рангом ρ требует $O(N + r^2)\rho^2$ действий. При $\rho \sim r$ эта оценка имеет вид $O(Nr^2 + r^4)$.

3. Матрицы Грама факторов вычисляются за $O(r^4\rho) \sim O(r^5)$ операций. Крестовая аппроксимация матрицы Грама $C_x^T C_x$ требует вычисления ее диагонали и нескольких столбцов. Таким образом, сложность вычислений падает в r^2 раз по сравнению с полными алгоритмами **CTcnd2**, **TTcnd2** и составляет $O(r^5)$ и $O(r^6)$ для СТ- и ТТ-случаев, соответственно (табл. 3, справа).
4. Вычисление тензоров вида (19) и (5) производится при $d = 3$ алгоритмами **CTfull** и **TTfull** сложности $O(r^4N + r^3N^2 + r^2N^3)$ и $O(r^4N^2 + r^2N^3)$, соответственно, где N — модовый размер результата. Для «недоопределенных» факторов $N = \rho \sim r$, и оценки принимают вид $O(r^5)$ и $O(r^6)$, соответственно. Это и есть сложность приближенной оценки нормы $\|\mathbf{C}\|_F$, которую мы указываем в правой части табл. 3.
5. Грубая аппроксимация завершена. Вычисление «аккуратной» аппроксимации факторов и дальнейшие действия согласно алгоритму 2 устроены фактически аналогично «грубому» шагу, с единственной разницей, что ранги факторов ρ теперь не устанавливаются равными sr , а определяются в ходе аппроксимации по критериям $\varepsilon_u, \varepsilon_v, \dots, \varepsilon_w$, заданным формулой (11). Значения этих «промежуточных» рангов определяют скорость алгоритма 3. В худшем случае $\rho = r^2$, и мы имеем ту же скорость, что и в алгоритме 1. Для типичного на практике значения $\rho \sim r$ сложность «компрессионных шагов» метода (то есть всех шагов, кроме первого) остается равной

$$\text{time}_{\text{CT}} = O(r^5) + Nr^2, \quad \text{time}_{\text{TT}} = O(r^6) + Nr^2.$$

Сравнив полученную оценку с затратами, указанными в табл. 1, можно утверждать, что стоимость сжатия $\tilde{\mathbf{C}} \approx \mathbf{C}$ имеет более низкую асимптотику по n , чем вычисление всех видов одномерных умножений. На практике, однако, r^2 может быть сравнимо с n , и стоимость этапов, зависящих только от рангов, тоже существенно влияет на эффективность метода. При выполнении операций (MV) и (MM) одномерные умножения достаточно дорогие, и затраты на сжатие результата по сравнению с ними не будут существенны. При вычислении свертки (CV) одномерные умножения производятся за почти линейное по n время с помощью быстрого преобразования Фурье, поэтому сравнение затрат на первом и последующих шагах вызывает особый интерес. Мы сделаем это в следующем разделе по результатам численного эксперимента.

5. Вычисление трехмерного потенциала Кулона

Рассмотрим задачу вычисления трехмерного потенциала Кулона

$$p(y) = \int_{\mathbb{R}^3} \frac{f(x)}{|x - y|} dx, \quad (26)$$

где функцию $f(x)$ будем полагать экспоненциально затухающей на бесконечности. В качестве приложения рассмотрим вычисление потенциала Кулона от электронной плотности, возникающее при решении уравнений Хартри–Фока.

Для вычисления приближенного канонического представления для интегрального оператора (26) воспользуемся соотношением

$$\frac{1}{r} = \frac{1}{\sqrt{\pi}} \int_{\mathbb{R}} \exp(-t^2 r^2) dt, \quad (27)$$

справедливым при всех $r > 0$. Тогда

$$p(y) = \int_{\mathbb{R}} \frac{dt}{\sqrt{\pi}} \int_{\mathbb{R}^3} \exp(|x - y|^2 t^2) f(x) dx,$$

и трехмерный оператор распадается в прямое произведение одномерных. Если для $f(x)$ тоже доступно представление

$$f(x) = \sum_{abc} g_{abc} u_a(x_1) \otimes v_b(x_2) \otimes w_c(x_3),$$

то

$$p(y) = \int_{\mathbb{R}} \frac{dt}{\sqrt{\pi}} \sum_{abc} g_{abc} \hat{u}_a(y_1, t) \otimes \hat{v}_b(y_2, t) \hat{w}_c(y_3, t), \quad (28)$$

где

$$\hat{z}(y, t) = \int_{\mathbb{R}} \exp(|x - y|^2 t^2) z(x) dx.$$

Чтобы завершить приготовления, осталось выбрать квадратуру для приближенного интегрирования (27). Мы используем квадратуры, доступные с сайта www.mis.mpg.de/scicomp и описанные в [36]. Порядок квадратуры $r_q = 35$, заявленная точность при $|x - y| \leq 1000$ в равномерной норме составляет 10^{-10} . В качестве функций $f(x)$ мы берем электронную плотность различных молекул, вычисленную с помощью программы MOLPRO в каноническом формате с большим рангом и сжатую в формат Таккера процедурой, основанной на индивидуальной компрессии факторов [35]. Результаты и время расчета приведены в табл. 4. Сравнение времен выполнения отдельных этапов умножения, указанных в таблице («свертки» — одномерные умножения, «границы» — шаги со второго по шестой, «моды» — индивидуальная фильтрация факторов, «ядро» — вычисление ядра \hat{F} и его сжатие) позволяет сделать вывод, что сложность операций по сжатию $\tilde{C} \approx C$ превосходит затраты на вычисление одномерных сверток не более, чем в 5 раз. Это вполне приемлемое соотношение для операции (CV), поскольку затраты на модовые свертки почти линейны по n .

Общее время вычисления трехмерной свертки методом на основе структурированных тензорных произведений существенно ниже, чем методами на основе трехмерного быстрого преобразования Фурье (FFT3D). Это было замечено также в [26], где свертка посчитана для электронной плотности воды. На сетке размера $n = 512$ время работы алгоритма FFT3D на компьютере SunFire X4600 с процессором 2.6 ГГц равно 582.8 секунды — против 5.3 секунд для предложенного

Таблица 4. Время вычисления потенциала Кулона алгоритмом 3, секунды

ε	ранги В	свертки	границы	моды	ядро	всего	ранги С
10^{-5}	$20 \times 36 \times 30$	6 60%	1 10%	2 20%	1 10%	10	$15 \times 25 \times 22$
10^{-7}	$30 \times 50 \times 44$	9 43%	3 14%	4 19%	5 24%	21	$24 \times 43 \times 39$
10^{-9}	$45 \times 69 \times 59$	12 28%	7 16%	8 19%	16 37%	43	$40 \times 66 \times 55$
10^{-5}	$47 \times 47 \times 47$	10 41%	4 17%	4 17%	6 25%	24	$31 \times 30 \times 31$
10^{-7}	$70 \times 70 \times 71$	15 25%	13 21%	11 18%	22 36%	61	$51 \times 51 \times 53$
10^{-9}	$96 \times 94 \times 98$	20 16%	33 26%	21 16%	53 42%	127	$78 \times 79 \times 79$
10^{-5}	$26 \times 66 \times 67$	12 39%	8 26%	7 22%	4 13%	31	$15 \times 38 \times 38$
10^{-7}	$37 \times 97 \times 100$	17 24%	18 26%	17 24%	18 26%	70	$25 \times 64 \times 66$
10^{-9}	$46 \times 132 \times 139$	23 17%	40 29%	30 22%	45 32%	138	$41 \times 98 \times 101$

Размер сетки $n = 5121$.

Первая секция — этан, вторая — этиловый спирт, третья — глицин.

Время измерено на процессоре Core2Duo T5300 с частотой 1.33 ГГц.

Компилятор GNU Fortran 4.3.3, библиотеки GotoBLAS-1.26, FFTW-3.1.2.

там же алгоритма $Conv_{CC}$, вычисляющего произведение в CC-случае. Для сетки размера $n = 4096$ время работы $Conv_{CC}$ составило 127.1 секунды, а относительная точность аппроксимации 10^{-6} . Предложенный в нашей работе алгоритм вычисляет потенциал для более сложных молекул (спирта, глицина) на более мелкой сетке с большей точностью 10^{-7} за вдвое меньшее время (60 и 70 секунд соответственно, см. табл. 4).

6. Выводы и заключительные замечания

Основным результатом этой работы является быстрый алгоритм 3 умножения матриц в тензорных форматах, основанный на индивидуальной фильтрации модовых факторов и использующий процедуры крестовой аппроксимации. При больших значениях модового размера n и фиксированных рангах сложность сжатия данных (26) много меньше сложности одномерных умножений (см. табл. 1); на практике при сравнимых значениях n и r^2 и «дешевой» выполняемой операции затраты

на сжатие результата могут в несколько раз превосходить время на его вычисление. Общее время вычисления свертки ядра в С-формате ранга 35 и плотности в Т-формате с рангами порядка сотни составило 2 минуты, что превосходит по скорости другие известные нам методы.

Предложенные методы формально применимы в пространстве любой размерности, но их эффективное использование при $d > 10$ в задачах со сколь-либо существенными значениями модовых рангов (даже при $r \sim 10$) невозможно из-за ограничений по памяти, см. табл. 2. Для работы в многомерных пространствах необходимо снять экспоненциальную зависимость от d . Если доступно каноническое представление для исходных данных, то можно хранить ядро Т-формата в С-формате того же ранга (см. [26]). Но для практического снятия «проклятия размерности» нужны другие форматы представления данных [37, 38]. На данный момент наибольшее продвижение в развитии новых тензорных форматов сделано в [38]: в отличие от канонического формата, в предложенном там формате обеспечивается устойчивое и быстрое выполнение основных тензорных операций.

Благодарности

Эта работа поддержана грантом РФФИ 08-01-00115, грантом РФФИ/DFG 09-01-91332 и программой приоритетных исследований Отделения математических наук РАН. Авторы признательны С. А. Горейнову за внимательное чтение черновика статьи и множество ценных советов. В численных экспериментах для молекул этана, этилового спирта и глицина использованы дискретные выражения электронной плотности, любезно предоставленные нам доктором Х.-Ю. Фладом и доктором Р. Чинамсети.

А. Алгоритмы

В этом разделе мы приводим алгоритмы вычисления норм тензора $\|\mathbf{C}\|_F$ и дополнений $\|C_x\|_F$, $\|C_x\|_2$, которые необходимы для расчета индивидуальных границ фильтрации (11) в случае $d = 3$. При записи алгоритмов мы используем следующие обозначения.

« x » Внутренние переменные даны шрифтом печатной машинки: x, y, U, V, \dots

« $:=$ » Запись $x := \text{foo}(\dots)$ означает, что левая часть (пере)определяется по правой.

« $=:$ » Запись $A =: U S V^T$ означает, что правая часть есть разложение A .

« \rightarrow » Запись $\text{foo}(\dots) \rightarrow x$ означает, что выражение слева сохраняется в x .

« \rightsquigarrow » Запись $x_{ijk} \rightsquigarrow x_{jki}$ означает переформатирование тензора.

Большинство из приведенных алгоритмов сводится к выбору оптимального порядка суммирования выражений с большим количеством индексов. Например, каждый элемент произведения (5) в случае $d = 3$ задан шестикратной суммой; однако

вычисление всех N^3 элементов осуществляется не за $N^3 r^6$, как могло бы показаться, а всего лишь за $O(N^3 r^2)$ операций, см. алг. **TTfull**. Для каждого расчетного шага указаны сложность и название соответствующей ему процедуры библиотеки BLAS/LAPACK. Для облегчения чтения индексы, по которым на данном шаге ведется суммирование, подчеркнуты, а индексы, по которым объявлен цикл, взяты в скобки.

A.1. Случай СТ

Алгоритм СТfull: Вычислить произведение СТ по формуле (19)

```

for s do
   $u_{i(s)a} \rightsquigarrow u_{ia}, v_{j(s)b} \rightsquigarrow v_{jb}, w_{k(s)c} \rightsquigarrow w_{kc}$ 
  dgetm:  $\mathbf{C} := \mathbf{C} + \mathbf{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$   $r O(r^3 N + r^2 N^2 + r N^3)$ 
end for
return  $\mathbf{C}$  итого  $O(r^4 N + r^3 N^2 + r^2 N^3)$ 

```

Здесь и далее процедура **dgetm** реализует вычисление тензора, заданного Т-форматом (3). Примечательно, что оно может быть выражено через d матричных умножений и без использования циклов.

Алгоритм dgetm: Вычислить тензор $N \times N \times \dots \times N$, заданный Т-форматом (3)

```

dgemm  $\sum_a g_{ab\dots c} u_{ia} \rightarrow x_{b\dots ci}$   $O(r^d N)$ 
dgemm  $\sum_b x_{b\dots ci} v_{jb} \rightarrow y_{\dots cij}$   $O(r^{d-1} N^2)$ 
...
dgemm  $\sum_c z_{cij\dots} w_{kc} \rightarrow b_{ij\dots k}$   $O(r N^d)$ 
return  $\mathbf{B} = [b_{ij\dots k}]$  итого  $O(r^d N + r^{d-1} N^2 + \dots + r N^d)$ 

```

Этот же алгоритм используется при вычислении нормы в СТ-случае.

Алгоритм СТnorm: Вычислить норму тензора по формуле (20)

```

nrm2 = 0.
for s, s' do
   $\langle u_{(s)a}, u_{(s')a'} \rangle \rightsquigarrow u_{aa'}, \langle v_{(s)b}, v_{(s')b'} \rangle \rightsquigarrow v_{bb'}, \langle w_{(s)c}, w_{(s')c'} \rangle \rightsquigarrow w_{cc'}$ 
  dgetm:  $\mathbf{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \rightarrow \mathbf{T}$   $r^2 O(r^4)$ 
   $nrm^2 := nrm^2 + \langle \mathbf{G}, \mathbf{T} \rangle$   $r^2 O(r^3)$ 
end for
return  $\|\mathbf{C}\|_F = nrm$  итого  $O(r^6)$ 

```

Алгоритм STcndf: Вычислить норму $\|C_u\|_F$ по формуле (21)

$\langle \mathbf{v}_{sb}, \mathbf{v}_{sb'} \rangle \rightsquigarrow v_{bb's}, \langle \mathbf{w}_{sc}, \mathbf{w}_{sc'} \rangle \rightsquigarrow w_{cc's}$
dgemm: $\langle g_{:,b,c}, g_{:,b',c'} \rangle \rightarrow g_{bcb'c'} \rightsquigarrow g_{cc'bb'}$ $O(r^5)$
dgemm: $\sum_{cc'} g_{cc'bb'} w_{cc's} \rightarrow x_{bb's}$ $O(r^5)$
 $\|C_u\|_F^2 := \langle V, X \rangle$ $O(r^3)$
return $\|C_u\|_F$ итого $O(r^5)$

Алгоритм STcnd2: Вычислить норму $\|C_u\|_2$ через разложение матрицы (22)

for $a's'$ **do**
 $\langle \mathbf{v}_{sb}, \mathbf{v}_{(s')b'} \rangle \rightsquigarrow v_{bb's(s')}, \langle \mathbf{w}_{sc}, \mathbf{w}_{(s')c'} \rangle \rightsquigarrow w_{c'cs(s')}$
dgemm: $\sum_{c'} g_{(a')b'\underline{c'}} w_{c'cs(s')} \rightarrow x_{b'cs(a's')}$ $r^2 O(r^4)$
for s **do**
dgemm: $\sum_{b'} v_{bb'(ss')} x_{b'c(sa's')} \rightarrow y_{bc(sa's')}$ $r^3 O(r^3)$
end for
dgemm: $\sum_{bc} g_{abc} y_{bcs(a's')} \rightarrow (C_u^T C_u)_{as(a's')}$ $r^2 O(r^4)$
 {Вычислили очередной столбец матрицы Грама}
end for
 $\|C_u\|_2^2 = \text{dsyev}_{\max}(C_u^T C_u)$ $O(r^4)$
 {Нашли старшее собственное значение $C_u^T C_u$ }
return $\|C_u\|_2$ итого $O(r^6)$

A.2. Случай TT

Алгоритм TTfull: Вычислить произведение по формуле (5)

$u_{ipa} \rightsquigarrow u_{iap}, h_{abc} \rightsquigarrow h_{bca}$
dgemm: $\sum_p u_{iap} g_{pq} \rightarrow x_{iaq} \rightsquigarrow x_{isaq}$ $O(Nr^4)$
dgemm: $\sum_b v_{jqb} h_{bca} \rightarrow y_{jqca} \rightsquigarrow y_{jcaq}$ $O(Nr^4)$
dgemm: $\sum_{aq} x_{isaq} y_{jcaq} \rightarrow z_{isjc} \rightsquigarrow z_{ijsc}$ $O(N^2 r^4)$
dgemm: $\sum_{sc} z_{ijsc} w_{ksc} \rightarrow c_{ijk}$ $O(N^3 r^2)$
return $\mathbf{C} = [c_{ijk}]$ итого $O(r^4 N^2 + r^2 N^3)$

Алгоритм TTnorm: Вычислить норму произведения по формуле (23)

```

 $\langle \mathbf{u}_{pa}, \mathbf{u}_{p'a'} \rangle \rightsquigarrow \mathbf{u}_{pp'aa'}, \langle \mathbf{v}_{qb}, \mathbf{v}_{q'b'} \rangle \rightsquigarrow \mathbf{v}_{bb'qq'}, \langle \mathbf{w}_{sc}, \mathbf{w}_{s'c'} \rangle \rightsquigarrow \mathbf{w}_{cc'ss'}$ 
for s do
  dgemm:  $\sum_p \mathbf{g}_{pq(s)} \mathbf{u}_{pp'aa'} \rightarrow \mathbf{x}_{qp'aa'(s)}$   $r O(r^5)$ 
end for
for s' do
  dgemm:  $\sum_{q'} \mathbf{v}_{bb'qq'} \mathbf{g}_{p'q'(s')} \rightarrow \mathbf{y}_{bb'qp'(s')}$   $r O(r^5)$ 
end for
for ss' do
  dgemm:  $\sum_{qp'} \mathbf{y}_{bb'qp'(s')} \mathbf{x}_{qp'aa'(s)} \rightarrow \mathbf{p}_{bb'aa'(ss')} \rightsquigarrow \mathbf{p}_{a'b'ab(ss')}$   $r^2 O(r^6)$ 
  dgemm:  $\sum_{a'b'} \mathbf{h}_{a'b'c'} \mathbf{p}_{a'b'ab(ss')} \rightarrow \mathbf{q}_{c'ab(ss')}$   $r^2 O(r^5)$ 
  dgemm:  $\sum_{ab} \mathbf{q}_{c'ab(ss')} \mathbf{h}_{abc} \rightarrow \mathbf{z}_{c'c(ss')} = \mathbf{z}_{cc'(ss')}$   $r^2 O(r^4)$ 
end for
 $\|\mathbf{C}\|_F^2 = \langle \mathbf{Z}, \mathbf{W} \rangle$   $O(r^4)$ 
return  $\|\mathbf{C}\|_F$  итого  $O(r^8)$ 

```

Алгоритм TTcndf: Вычислить норму $\|\mathbf{C}_u\|_F$ по формуле (24)

```

 $\langle \mathbf{v}_{qb}, \mathbf{v}_{q'b'} \rangle \rightarrow \mathbf{v}_{qbq'b'} \rightsquigarrow \mathbf{v}_{qq'bb'}, \langle \mathbf{w}_{sc}, \mathbf{w}_{s'c'} \rangle \rightarrow \mathbf{w}_{scs'c'} \rightsquigarrow \mathbf{w}_{ss'cc'}$ 
dgemm:  $\langle \mathbf{g}_{:,q,s}, \mathbf{h}_{:,q',s'} \rangle \rightarrow \mathbf{g}_{qsq's'} \rightsquigarrow \mathbf{g}_{qq'ss'}$   $O(r^5)$ 
dgemm:  $\langle \mathbf{h}_{:,b,c}, \mathbf{h}_{:,b',c'} \rangle \rightarrow \mathbf{h}_{bcb'c'} \rightsquigarrow \mathbf{h}_{bb'cc'}$   $O(r^5)$ 
dgemm:  $\sum_{ss'} \mathbf{g}_{qq'ss'} \mathbf{w}_{ss'cc'} \rightarrow \mathbf{x}_{qq'cc'}$   $O(r^6)$ 
dgemm:  $\sum_{bb'} \mathbf{v}_{qq'bb'} \mathbf{h}_{bb'cc'} \rightarrow \mathbf{y}_{qq'cc'}$   $O(r^6)$ 
 $\|\mathbf{C}_u\|_F^2 := \langle \mathbf{Y}, \mathbf{X} \rangle$   $O(r^4)$ 
return  $\|\mathbf{C}_u\|_F$  итого  $O(r^6)$ 

```

Алгоритм TTcnd2: Вычислить норму $\|C_u\|_2$ через разложение матрицы (25)

```

for a' do
  dgemm:  $\sum_{b'} h_{(a')\underline{b}'c'} \langle \mathbf{v}_{\underline{b}'q'}, \mathbf{v}_{bq} \rangle \rightarrow x_{c'q'bq(a')}$  r O(r5)
end for
for p' do
  dgemm:  $\sum_{s'} \langle \mathbf{w}_{cs}, \mathbf{w}_{c's'} \rangle g_{(p')q's'} \rightarrow y_{csc'q'(p')}$  r O(r5)
end for
for p'a' do
  dgemm:  $\sum_{c'q'} y_{csc'q'(p')} x_{c'q'bq(a')} \rightarrow z_{cscq(p'a')} \rightsquigarrow z_{qsbq(p'a')}$  r2 O(r6)
  dgemm:  $\sum_{qs} g_{pqs} z_{qsbq(p'a')} \rightarrow t_{pbc(p'a')}$  r2 O(r5)
  dgemm:  $\sum_{bc} t_{pbc(p'a')} h_{abc} \rightarrow (C_u^T C_u)_{pa(p'a')}$  r2 O(r4)
  {Вычислили очередной столбец матрицы Грама}
end for
 $\|C_u\|_2^2 := \text{dsyev}_{\max}(C_u^T C_u)$  O(r4)
{Нашли старшее собственное значение  $C_u^T C_u$ }
return  $\|C_u\|_2$  итого O(r8)

```

Список литературы

- [1] *Beylkin G., Mohlenkamp M. J.* Numerical operator calculus in higher dimensions // *Proc. Nat. Acad. Sci. USA*. 2002. V. 99, № 16. P. 10246-10251. doi:10.1073/pnas.112329799.
- [2] *Ford J. M., Tyrtysnikov E. E.* Combining Kronecker product approximation with discrete wavelet transforms to solve dense, function-related systems // *SIAM J. Sci. Comp.* 2003. V. 25, № 3. P. 961-981. doi:10.1137/s1064827503421689.
- [3] *Тыртышников Е. Е.* Тензорные аппроксимации матриц, порожденных асимптотически гладкими функциями // *Матем. сб.* 2003. Т. 194, № 5. С. 147-160.
- [4] *Tyrtysnikov E. E.* Kronecker-product approximations for some function-related matrices // *Linear Algebra Appl.* 2004. V. 379. P. 423-437. doi:10.1016/j.laa.2003.08.013.
- [5] *Grasedyck L.* Existence and computation of low Kronecker-rank approximations for large systems in tensor product structure // *Computing*. 2004. V. 72. P. 247-265.
- [6] *Hackbusch W., Khoromskij B. N., Tyrtysnikov E. E.* Hierarchical Kronecker tensor-product approximations // *J. Numer. Math.* 2005. V. 13. P. 119-156.
- [7] *Beylkin G., Mohlenkamp M. J.* Algorithms for numerical analysis in high dimensions // *SIAM J. Sci. Comput.* 2005. V. 26, № 6. P. 2133-2159.
- [8] *Oseledets I. V., Tyrtysnikov E. E.* Approximate inversion of matrices in the process of solving a hypersingular integral equation // *Comp. Math. and Math. Phys.* 2005. V. 45, № 2. P. 302-313.

- [9] Hackbusch W., Khoromskij B. N., Sauter S. A., Tyrtyshnikov E. E. Use of Tensor Formats in Elliptic Eigenvalue Problems: Preprint 78. — Leipzig: MIS MPI, 2008. www.mis.mpg.de/preprints/2008/preprint2008_78.pdf.
- [10] Flad H.-J., Khoromskij B. N., Savostyanov D. V., Tyrtyshnikov E. E. Verification of the cross 3D algorithm on quantum chemistry data // *Rus. J. Numer. Anal. Math. Model.* 2008. V. 23, № 4. P. 329-344. doi:10.1515/RJNAMM.2008.020.
- [11] Hackbusch W., Khoromskij B. N., Tyrtyshnikov E. E. Approximate iterations for structured matrices // *Numer. Mathematik.* 2008. V. 109, № 3.
- [12] Oseledets I. V., Savostyanov D. V., Tyrtyshnikov E. E. Linear algebra for tensor problems // *Computing.* 2009. V. 85, № 3. P. 169-188. doi:10.1007/s00607-009-0047-6.
- [13] Harshman R. A. Foundations of the Parafac procedure: models and conditions for an explanatory multimodal factor analysis // *UCLA Working Papers in Phonetics.* 1970. V. 16. P. 1-84.
- [14] Tucker L. R. Some mathematical notes on three-mode factor analysis // *Psychometrika.* 1966. V. 31. P. 279-311.
- [15] Carroll J. D., Chang J. J. Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition // *Psychometrika.* 1970. V. 35. P. 283-319.
- [16] Bro R. PARAFAC: Tutorial and applications // *Chemometrics and Intelligent Lab. Syst.* 1997. V. 38, № 2. P. 149-171.
- [17] Comon P. Tensor decomposition: state of the art and applications // IMA Conf. Math. in Sig. Proc., Warwick, UK. — 2000.
- [18] de Lathauwer L., de Moor B., Vandewalle J. Computing of Canonical decomposition by means of a simultaneous generalized Schur decomposition // *SIAM J. Matrix Anal. Appl.* 2004. V. 26. P. 295-327.
- [19] Espig M., Grasedick L., Hackbusch W. Black box low tensor rank approximation using fibre-crosses // *Constr. appr.* 2009. V. 30, № 3. P. 557-597. doi:10.1007/s00365-009-9076-9.
- [20] Oseledets I. V., Savostyanov D. V., Tyrtyshnikov E. E. Fast simultaneous orthogonal reduction to triangular matrices // *SIAM J. Matrix Anal. Appl.* 2009. V. 31, № 2. P. 316-330. doi:10.1137/060650738.
- [21] de Lathauwer L., de Moor B., Vandewalle J. A multilinear singular value decomposition // *SIAM J. Matrix Anal. Appl.* 2000. V. 21. P. 1253-1278. doi:10.1137/s0895479896305696.

- [22] de Lathauwer L., de Moor B., Vandewalle J. On best rank-1 and rank- (R_1, R_2, \dots, R_N) approximation of high-order tensors // *SIAM J. Matrix Anal. Appl.* 2000. V. 21. P. 1324-1342.
- [23] Савостьянов Д. В. Полилинейная аппроксимация матриц и интегральные уравнения. Дисс. ... канд. физ.-матем. наук — М.: ИВМ РАН. 2006. www.inm.ras.ru/library/Tyrtysnikov/savostyanov_disser.pdf.
- [24] Oseledets I. V., Savostyanov D. V., Tyrtysnikov E. E. Tucker dimensionality reduction of three-dimensional arrays in linear time // *SIAM J. Matrix Anal. Appl.* 2008. V. 30, № 3. P. 939-956. doi:10.1137/060655894.
- [25] Горейнов С. А. О крестовой аппроксимации многоиндексного массива // Докл. РАН. 2008. Т. 420, № 4. С. 439-441.
- [26] Khoromskij B. N., Khoromskaia V. Multigrid accelerated tensor approximation of function related multidimensional arrays // *SIAM J. Sci. Comp.* 2009. V. 31, № 4. P. 3002-3026. doi:10.1137/080730408.
- [27] Khoromskij B. N. On tensor approximation of Green iterations for Kohn-Sham equations // *Computing and visualization in science*. 2008. V. 11, №4-6. P. 259-271. doi:10.1007/s00791-008-0097-x.
- [28] Горейнов С. А., ТЫРТЫШНИКОВ Е. Е., ЗАМАРАШКИН Н. Л. Псевдоскелетная аппроксимация матриц // Докл. РАН. 1995. Т. 342, № 2. С. 151-152.
- [29] Goreinov S. A., Tyrtysnikov E. E., Zamarashkin N. L. A theory of pseudo-skeleton approximations // *Lin. Algebra Appl.* 1997. V. 261. P. 1-21.
- [30] Goreinov S. A., Tyrtysnikov E. E. The maximal-volume concept in approximation by low-rank matrices // *Contemporary Mathematics*. 2001. V. 208. P. 47-51.
- [31] Tyrtysnikov E. E. Incomplete cross approximation in the mosaic-skeleton method // *Computing*. 2000. V. 64, № 4. P. 367-380. doi:10.1007/s006070070031.
- [32] Goreinov S. A., Oseledets I. V., Savostyanov D. V. et al. How to find a good submatrix: Research Report 08-10. — Kowloon Tong, Hong Kong: ICM HKBU, 2008. www.math.hkbu.edu.hk/ICM/pdf/08-10.pdf.
- [33] Bebendorf M. Approximation of boundary element matrices // *Numer. Math.* 2000. V. 86, № 4. P. 565-589. doi:10.1007/p100005410.
- [34] Савостьянов Д. В. Мозаично-скелетонные аппроксимации. Работа на степ. бакалавра — М.: ИВМ РАН. 2001.
- [35] Oseledets I. V., Savostyanov D. V., Tyrtysnikov E. E. Cross approximation in tensor electron density computations // *Numer. Lin. Alg. Appl.* 2010. V. 17, № 6. P. 935-952. doi:10.1002/nla.682.

- [36] *Braess D., Hackbusch W.* On the efficient computation of high-dimensional integrals and the approximation by exponential sums: Preprint 3. — Leipzig: MPI MIS, 2009. www.mis.mpg.de/preprints/2009/preprint2009_3.pdf.
- [37] *Hackbusch W., Kühn S.* A new scheme for the tensor representation: Preprint 2. — Leipzig: MPI MIS, 2009. www.mis.mpg.de/preprints/2009/preprint2009_2.pdf.
- [38] *Oseledets I. V., Tyrtshnikov E. E.* Breaking the curse of dimensionality, or how to use SVD in many dimensions // *SIAM J. Sci. Comp.* 2009. V. 31, № 5. P. 3744-3759.