

# О визуализации ансамбля параметрических численных решений с помощью тензорного разложения

А.К. Алексеев<sup>1,2</sup>, А.Е. Бондарев<sup>1</sup>, Ю.С. Пятакова<sup>2</sup>

<sup>1</sup> ИИПМ им М.В. Келдыша РАН, Миусская пл., 4, Москва, 125047, Россия.

<sup>2</sup> РКК Энергия им. С.П. Королева, Ленина 4а, Королев, 141700, Россия

## Аннотация

Аппроксимация тензора, получаемого при дискретизации многомерной функции на равномерной сетке, рассмотрена с точки зрения хранения и обработки результатов параметрических расчетов в задачах вычислительной аэрогазодинамики. Для этой цели использованы тензорные разложения. Описан новый алгоритм расчета канонического разложения с помощью градиентного спуска для приближенно разложимого целевого функционала. Этот алгоритм использует набор точек в гиперплоскости, ортогональной рассчитываемому ядру канонического разложения (“зонтик”), что позволяет его гибкое применение для аппроксимации тензоров с априорно неизвестным рангом. Этот алгоритм естественным образом переносится и на такое тензорное разложение, как тензорный поезд. Представлены результаты численных экспериментов для модельных шестимерных функций и для ансамбля численных решений двумерных уравнений Эйлера. Эти уравнения описывают течение сжимаемого газа с двумя пересекающимися скачками уплотнения. В качестве параметров течения служат число Маха и углы отклонения потока.

## Ключевые слова

Тензорное разложение, каноническое разложение, параметрические расчеты, вычислительная аэрогазодинамика, градиентная оптимизация

# On the Visualization of the Ensemble of Parametric Numerical Solutions Using Tensor Decomposition

A.K. Alekseev<sup>1,2</sup>, A.E. Bondarev<sup>1</sup>, Yu.S. Pyatakova<sup>2</sup>

<sup>1</sup> Keldysh Institute of Applied mathematics of RAS, Miusskaya sq. 4, Moscow, 125047, Russia

<sup>2</sup> RSC Energia, Lenina 4a, Korolev, 141700, Russia

## Abstract

An approximation of the tensor obtained at a discretization of the multidimensional function on a uniform grid is addressed from the viewpoint of the storage and treating the result of the parametric computations in the CFD problems. The tensor decompositions are considered for this purpose. The new algorithm of the calculation of the canonical decomposition using the gradient descent for an approximately decomposable goal functional. This algorithm applies an ensemble of points on the hyperplane orthogonal to the computed core of the canonical decomposition (“umbrella”) that enables its flexible application for the approximation of the tensors with a priori unknown rank. This algorithm is naturally transferred on such tensor decomposition as the tensor train. The results of the numerical tests are presented for model six-dimensional functions and for the ensemble of the numerical solutions for two-dimensional Euler equations. These equations describe the flow of the compressible gas with two crossing shock waves. Mach number and the the flow deflection angles are considered as the flow parameters.

## Keywords

ГрафиКон 2023: 33-я Международная конференция по компьютерной графике и машинному зрению, 19-21 сентября 2023 г., Институт проблем управления им. В.А. Трапезникова Российской академии наук, г. Москва, Россия

EMAIL: aleksey.k.alekseev@gmail.com (А.К. Алексеев); bond@keldysh.ru, (А.Е. Бондарев); yuliya.pyatakova@rsce.ru (Ю.С. Пятакова)  
ORCID: 0000-0001-8317-8688 (А.К. Алексеев); 0000-0003-3681-5212 (А.Е. Бондарев); 0000-0002-8055-7807 (Ю.С. Пятакова)



© 2023 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Tensor decomposition, canonical decomposition, parametric computation, Computational Fluid Dynamics, gradient descent.

## 1. Введение

Аппроксимация многомерных (в пространствах с размерностью более 3) функций представляет интерес во многих приложениях, таких как квантовая механика, решение кинетических уравнений типа Больцмана, решение задач аэрогазодинамики в пространстве параметров.

В качестве первого примера рассмотрим функцию  $f(x, y, z, u, v, w)$  определенную в области  $\Omega \subset R^6$  и соответствующую плотности распределения Больцмана. Для сетки, содержащей 100 узлов по каждой координате в шестимерном случае необходимо  $10^{12}$  узлов памяти, содержащих числа. Для 64-значного числа это соответствует оперативной памяти 64 ТБ, что соответствует параметрам мощнейших современных суперкомпьютеров (Чебышев, 5.5 терабайт ОЗУ, Червоненкис, 199 ТБ). Это значительно осложняет практические приложения, использующие уравнения Больцмана, как с точки стоимости расчетов, так и оперативности их применения.

В качестве другого примера представляет интерес набор численных решений задач аэрогазодинамики при аппроксимации в пространстве параметров. Это  $p$  набор функций  $f_i(x, y, z, \mathcal{Q}_1 \dots \mathcal{Q}_q)$ , где  $i = 1, \dots, p$  соответствует переменным течения (плотность, компоненты скорости, энергия), а  $(\mathcal{Q}_1 \dots \mathcal{Q}_q) \subset \Omega_q \subset R^q$  соответствует параметрам задачи (число Маха, Рейнольдса, угол атаки и т.д.). В распространенных практических приложениях эти функции распределены в пространстве размерностью от четырех до шести, что требует ресурсов, сравнимых с указанными в предыдущем примере.

С точки зрения практических приложений было бы крайне желательно данные задачи решать с вычислительными затратами на уровне ресурсов стандартного персонального компьютера.

Аппроксимация функций в многомерном пространстве представляет собой сложную задачу из-за экспоненциального роста требуемой оперативной памяти при увеличении размерности задачи ("проклятия размерности"). Одним из путей преодоления этих трудностей является использование тензорной формы многомерных задач и их аппроксимация с помощью тензорных разложений [1,2,3]. Этому вопросу и посвящена данная работа.

## 2. Каноническое разложение

Каноническое разложение  $A = \sum_{\alpha=1}^r Q_{\alpha}^1 \otimes Q_{\alpha}^2 \otimes \dots \otimes Q_{\alpha}^d$  является самым фундаментальным тензорным разложением (на его основе определяется ранг тензора). Здесь  $\otimes$  – внешнее (тензорное) произведение векторов, в индексной форме имеющее вид  $A_{ij} = a_i \otimes b_j = a_i b_j$ .

В индексном виде каноническое разложение записывается как

$$A_{i_1, i_2 \dots i_d} = \sum_{\alpha=1}^r \prod_{c=1}^{c=d} Q_{\alpha, i_c}^c \quad (1)$$

Это выражение единственно с точностью до перестановки членов и масштабирования, если слои ядер ортогональны [4,10].

Память, требуемая для хранения этой аппроксимации тензора  $\sim rnd$  (где  $d$  - размерность пространства,  $n$  число узлов по одному направлению,  $r$  ранг тензора) (вместо полных затрат на хранение тензора  $n^d$ ). Время для расчета одного узла  $\sim rd$ . Каноническое разложение обеспечивает гигантское сжатие данных (числа узлов сетки, используемых для аппроксимации функции в  $R^d$ ), а именно  $n^d \rightarrow d \cdot n \cdot r$ , где  $d$  - размерность пространства,  $n$  число узлов по одному направлению,  $r$  ранг тензора. Ценой за этот успех служит большое количество вычислительных сложностей, сопутствующих применению этого разложения. Имеются данные

[1,4], согласно которым использование канонического разложения страдает от неустойчивостей и нуждается в регуляризации.

Задача определения набора ядер  $\{Q_{\alpha,i}^1, \dots, Q_{\alpha,k}^d\}$  в вариационной форме имеет вид

$$Q_{\beta}^n = \arg \min_{Q_{\beta}^n} \left\| A - \sum_1^r Q_{\alpha}^1 \otimes Q_{\alpha}^2 \otimes \dots \otimes Q_{\alpha}^d \right\| \quad (2)$$

Ключевой величиной при использовании канонического разложения является ранг тензора  $r$ . По данным [1,4] она не вычислима из-за некорректности постановки задачи

$$r = \arg \min_r \left\| A - \sum_{\alpha=1}^r Q_{\alpha}^1 \otimes Q_{\alpha}^2 \otimes \dots \otimes Q_{\alpha}^d \right\| \quad (3)$$

### 3. Методы расчета тензорных разложений

Методы расчета тензорных разложений можно с некоторой условностью разделить на два подкласса: методы, основанные на матризации тензора (линейной алгебре), например [3], и вариационные методы [11] (которые делятся на прямые и итерационные).

Методы, основанные на линейной алгебре, в существенной степени опираются на матризацию тензоров, сингулярное разложение и содержат много интересных оригинальных алгоритмов, позволяющих проводить операции прямо над ядрами, не обращаясь к аппроксимируемым функциям.

Вариационные формулировки, как правило, опираются на метод наименьших квадратов по переменным направлениям (alternating least squares (ALS)) [6,7], однако чаще всего тоже используют матризацию тензоров. Метод ALS [6, 7] широко используется при поиске тензорных разложений и позволяет оптимизировать один из параметров при фиксированных остальных. Как правило, для канонического разложения ALS реализуется с помощью произведения Хатри-Рао (Khatri Rao product) [8].

Подход к расчету канонического разложения с помощью произведения Хатри-Рао на данный момент доминирует. К сожалению, для наших целей он не пригоден, поскольку использует матризацию тензора, требующую для хранения ту же величину памяти, что и сам тензор. Для задач рассматриваемого нами класса необходимая оперативная память превосходит возможности стандартной вычислительной техники (в данной работе мы оперируем с массивами, формально содержащими до  $10^{12}$  чисел), что исключает использование матризации тензора.

Здесь мы рассмотрим оптимизационный метод, включающий в себя элементы метода наименьших квадратов по переменным направлениям и градиентный спуск. Его отличительными особенностями является использование специального множества (“зонтик”), на котором происходит оптимизация. Это множество позволяет разложить целевой функционал на сумму практически независимых функционалов. Другой особенностью является использование тихоновской регуляризации ядер нулевого порядка в скользящем по рангу варианте. При этой регуляризации компоненты ядер испытывают тем большее демпфирование, чем выше их ранг.

### 4. Градиентная оптимизация на случайном зонтике

Формально качество аппроксимации функции каноническим разложением на каждом шаге глобальной итерации может быть оценено с помощью следующей невязки (записанной для 6D):

$$\tilde{\varepsilon}_{total} = \sum_{i,j,k,l,m,p} \left\{ \sum_{\alpha} Q^x(\alpha,i) \cdot Q^y(\alpha,j) \cdot Q^z(\alpha,k) \cdot Q^u(\alpha,l) \cdot Q^v(\alpha,m) \cdot Q^w(\alpha,p) - \tilde{f}_{ijklmp} \right\}^2 / 2 \quad (4)$$

Здесь используются все узлы тензора, как и в алгоритмах с использованием произведения Хатри-Рао. К сожалению, для рассматриваемых размерностей этот функционал напрямую невычислим (по крайней мере на обычных персональных компьютерах) из-за огромных затрат компьютерного времени. Мы его значение численно оценивали с помощью метода Монте-Карло в форме:

$$\varepsilon_{MC} = \frac{1}{2Mc} \sum_{s=1}^{s=MC} \left\{ \sum_{\alpha} Q^x(\alpha, i) \cdot Q^y(\alpha, j) \cdot Q^z(\alpha, k) \cdot Q^u(\alpha, l) \cdot Q^v(\alpha, m) \cdot Q^w(\alpha, p) - \tilde{f}_{ijklmp} \right\}^2 \quad (5)$$

где на каждом шаге суммирования  $s$  каждый индекс из  $i, j, k, l, m, p$  выбирался случайным и равномерно распределенным числом. В результате мы вычисляли среднюю по ансамблю сумму квадратов ошибки аппроксимации. Число попыток в ансамбле находилось в интервале  $MC = 1000 \div 100000$ , где результат довольно слабо менялся при изменении числа попыток.

К сожалению, функционал (4) практически не вычислим, а функционал (5) может быть применен к расчету ядер только в рамках метода Монте-Карло. Поэтому определение ядер разложения  $\{Q_{\alpha, i}^1, \dots, Q_{\alpha, k}^d\}$  с помощью минимизации этих функционалов крайне затруднено.

В качестве альтернативы здесь мы используем специальную форму целевого функционала, не требующую больших вычислительных усилий и позволяющую получать выражения для градиента целевого функционала по компонентам ядер, пригодные для градиентной оптимизации. В отличие от стандартного подхода, основанного на произведении Хатри-Рао, предлагаемый алгоритм не ограничен размерностью при использовании (не использует матризацию тензора или полный набор узлов тензора) и существенно проще алгоритмически.

Основой этого алгоритма является специальный выбор ансамбля точек, на которых производится оптимизация. Эти точки случайным образом выбираются на гиперплоскости, ортогональной ядру (на “зонтике”). Ядра  $Q^x(\alpha, i), Q^y(\alpha, j), Q^z(\alpha, k) \dots$  получаются последовательным перебором. Рассмотрим алгоритм на примере  $Q^x(\alpha, i)$ . Определим элементы ядра  $Q^x(\alpha, i)$  последовательно по  $i$ , используя последовательность невязок  $\varepsilon_{x, i}(Q^x(*, i))$  на случайном ансамбле (“зонтике”)

$$\varepsilon_{x, i}(Q^x) = 1/2 \sum_{e=1 \dots Lens} \left\{ \left( \sum_{\alpha} C(\alpha) \cdot Q^x(\alpha, i) \cdot Q^y(\alpha, j_e^i) \cdot Q^z(\alpha, k_e^i) \dots - \tilde{f}_{ij_e^i k_e^i l_e^i m_e^i p_e^i} \right) \cdot \right. \\ \left. \cdot \left( \sum_{\beta} C(\beta) \cdot Q^x(\beta, i) \cdot Q^y(\beta, j_e^i) \cdot Q^z(\beta, k_e^i) \dots - \tilde{f}_{ij_e^i k_e^i l_e^i m_e^i p_e^i} \right) \right\}. \quad (6)$$

Точки ансамбля  $j_e^i, k_e^i, l_e^i, m_e^i, p_e^i$  нумеруем индексом  $e = 1 \dots Lens$  ( $i$  соответствует узлу раскрытия “зонтика”). Они расположены в гиперплоскости, ортогональной  $x$  (в точке  $i$ ) и выбираются случайным образом. Здесь  $\tilde{f}_{ij_e^i k_e^i l_e^i m_e^i p_e^i}$  точное значение функции в точке  $i, j_e^i, k_e^i, l_e^i, m_e^i, p_e^i$ .

В расчетах размер ансамбля выбирается большим предварительной оценки ранга тензора  $Lens \geq r$ . Увеличение размера ансамбля увеличивает вычислительные затраты на расчет невязки и градиента.

Возмутим это ядро на величину  $\Delta Q^x(\beta, i)$ . Соответствующее возмущение невязки имеет вид:

$$\Delta \varepsilon_{x, i}(Q^x) = \sum_{e=1 \dots Lens} \left\{ \left( \sum_{\alpha} C(\alpha) \cdot Q^x(\alpha, i) \cdot Q^y(\alpha, j_e^i) \cdot Q^z(\alpha, k_e^i) \dots - \tilde{f}_{ij_e^i k_e^i l_e^i m_e^i p_e^i} \right) \cdot \right. \\ \left. \cdot \left( \sum_{\beta} C(\beta) \cdot \Delta Q^x(\beta, i) \cdot Q^y(\beta, j_e^i) \cdot Q^z(\beta, k_e^i) \dots - \tilde{f}_{ij_e^i k_e^i l_e^i m_e^i p_e^i} \right) \right\}. \quad (7)$$

Соответственно градиент невязки имеет вид:



$$\partial \varepsilon_{x,i} / \partial Q^x(\beta, i) = \sum_{e=1 \dots L_{ens}} \{ (\sum_{\alpha} C(\alpha) \cdot Q^x(\alpha, i) \cdot Q^y(\alpha, j_e^i) \cdot Q^z(\alpha, k_e^i) \dots - \tilde{f}_{ij_e^i k_e^i l_e^i m_e^i p_e^i}) \cdot C(\beta) \cdot Q^y(\beta, j_e^i) \cdot Q^z(\beta, k_e^i) \dots \}. \quad (8)$$

Для перекрестных производных (по другим ядрам, здесь для примера, по  $Q^y(\beta, j)$ ) в точках ансамбля  $e$  соответствующее выражение имеет вид:

$$\partial \varepsilon_{x,i} / \partial Q^y(\beta, j) = \{ (\sum_{\alpha} C(\alpha) \cdot Q^x(\alpha, i) \cdot Q^y(\alpha, j_e^i) \cdot Q^z(\alpha, k_e^i) \dots - \tilde{f}_{ij_e^i k_e^i l_e^i m_e^i p_e^i}) \cdot C(\beta) \cdot Q^x(\beta, i) \cdot Q^z(\beta, k_e^i) \dots \}. \quad (9)$$

при  $j = j_e^i$  (иначе – 0). Важно, что суммирование по ансамблю пропадает. Поэтому такие члены примерно в  $1/L_{ens}$  раз меньше основного (8). На этом основании мы ими пренебрегаем, что позволяет проводить оптимизацию только по  $Q^x(\beta, i)$ .

Таким образом, вместо одной глобальной невязки (используемой в расчетах для контроля)

$$\varepsilon_{\Sigma} = \{ \sum_i \varepsilon_{x,i} + \sum_j \varepsilon_{y,j} + \dots \} / (6N \cdot L_{ens}) \quad (10)$$

мы оптимизируем  $6N$  отдельных невязок (считаем, что по каждой координате у нас  $N$  узлов)

$$\varepsilon_{x,i}; \varepsilon_{y,j} \dots \quad (11)$$

Компоненты ядер ищутся последовательной оптимизацией невязок по номеру ядра и номеру узла на ядре

$$Q^c(\beta, i) = \arg \min \varepsilon_{c,i}(Q^c(\beta, i)) \quad (12)$$

Для оптимизации мы используем градиентный спуск, для  $Q^x(\beta, i)$  имеющий вид:

$$\{Q^x(\beta, i)\}^{n+1} = \{Q^x(\beta, i)\}^n - \tau \nabla_{x,i,\beta} \varepsilon. \quad (13)$$

Такую примитивную структуру алгоритма можно реализовать из-за того, что градиенты приближенно разложимы в соответствии с (14):

$$\nabla \varepsilon_{\Sigma} = \{ \nabla_{Q_i^x} \varepsilon_{x,i} + \nabla_{Q_j^y} \varepsilon_{y,j} + \dots + \nabla_{Q_j^y} \varepsilon_{x,i} + \nabla_{Q_j^y} \varepsilon_{y,j} + \dots \} = \{ \nabla_{Q_i^x} \varepsilon_{x,i} + \nabla_{Q_j^y} \varepsilon_{y,j} + \dots \} + O(1/L_{ens}). \quad (14)$$

## 5. Результаты численных экспериментов по интерполяции многомерных функций

Рассмотрим аппроксимацию шестимерной функции  $\tilde{f}$  (точнее говоря тензора  $\tilde{f}_{ijklmp}$ , соответствующего значениям функции в узлах регулярной сетки) с помощью канонического разложения и соответствующего набора ядер  $Q^x(\alpha, i) \cdot Q^y(\alpha, j) \cdot Q^z(\alpha, k) \cdot Q^u(\alpha, l) \cdot Q^v(\alpha, m) \cdot Q^w(\alpha, p)$ .

В численных экспериментах использовалась сетка, содержащая 100 узлов по каждой координате. С формальной точки зрения при такой сетке хранение  $\tilde{f}_{ijklmp}$  потребует  $10^{12}$  ячеек памяти, что совершенно нереалистично ни с точки зрения хранения, ни с точки зрения расчетов. В качестве иллюстрации, отметим, что в этом случае память, занимаемая ядрами с рангом 100 (достаточно большим) составляет  $6 \cdot 100 \cdot 100 = 60000$  ячеек, что иллюстрирует сверхвысокую степень сжатия информации ( $\sim 10^7$ ) при использовании канонического разложения.

В процессе отладки проводилось сравнение численных (полученных прямым численным дифференцированием) и аналитических градиентов (полученных с помощью выражений типа (8)), показавшее их практически полное совпадение.

Проведено тестирование аппроксимации различных функций с помощью канонического разложения. Первой рассмотрена функция

$$f = x + y. \quad (15)$$

Это предельно простая функция, но ее априорный ранг не известен. Использовано 30 итераций, ранг варьировался от одного до 10. Зависимость величины невязки (5) от величины ранга представлена в таблице 1.

**Таблица 1** – Зависимость невязки (5) от ранга для функции (15)

Ранг	1	2	3	4	5	10
Невязка	$1.46 \cdot 10^{-2}$	$7.13410^{-8}$	$1.68 \cdot 10^{-7}$	$2.22 \cdot 10^{-7}$	$4.39 \cdot 10^{-7}$	$2.00 \cdot 10^{-5}$

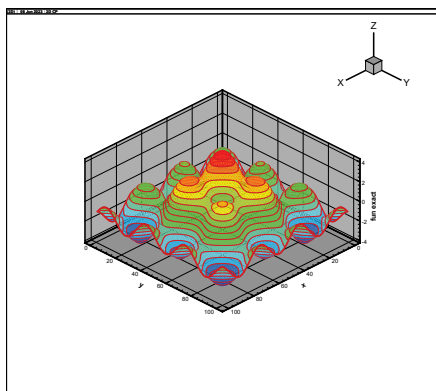
Характерно быстрое падение невязки при достижении истинного значения ранга (здесь- 2) , и медленный рост при дальнейшем увеличении ранга. Из этих результатов видно, что, в целом, величина невязки может служить индикатором истинного ранга тензора. По мере увеличения ранга увеличивается зашумленность результата, по всей видимости, это отражает неустойчивости, возникающие вследствие некорректности для канонического разложения [1,12].

Функция (1) является двумерной, рассмотрим более сложный истинно шестимерный случай, задаваемый гауссианой и суммой синусов по всем направлениям:

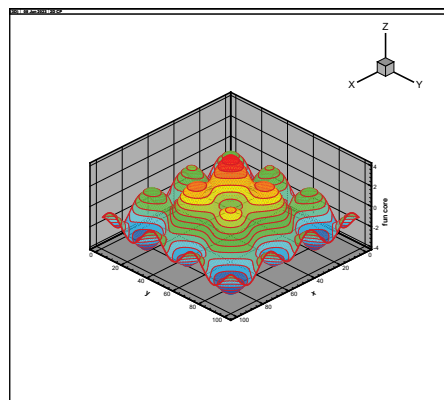
$$f = 5 \cdot \exp(-rad^2) + \sin(x/5) + \sin(y/5) + \sin(z/5) + \sin(u/5) + \sin(v/5) + \sin(w/5) \quad (16)$$

$$(rad = 0.001 \cdot ((ix - 50) + (iy - 50) + (iz - 50) + (iu - 50) + (iv - 50) + (iw - 50))) .$$

На рисунках 1 и 2 представлены результаты для ранга 10 и ансамбля из 1000 точек в плоскости  $x, y$ , остальные переменные соответствуют серединам интервалов на сетке 100. Результаты аппроксимации практически неотличимы от точной функции. При уменьшении размеров ансамбля качество аппроксимации уменьшается. В целом при расчетах достаточно сложных функций ансамбль должен быть существенно больше ожидаемого ранга. К счастью, использование Тихоновской регуляризации нулевого порядка позволяет минимизировать текущий ранг и избавляет от необходимости его априорного знания (или перебора рангов).

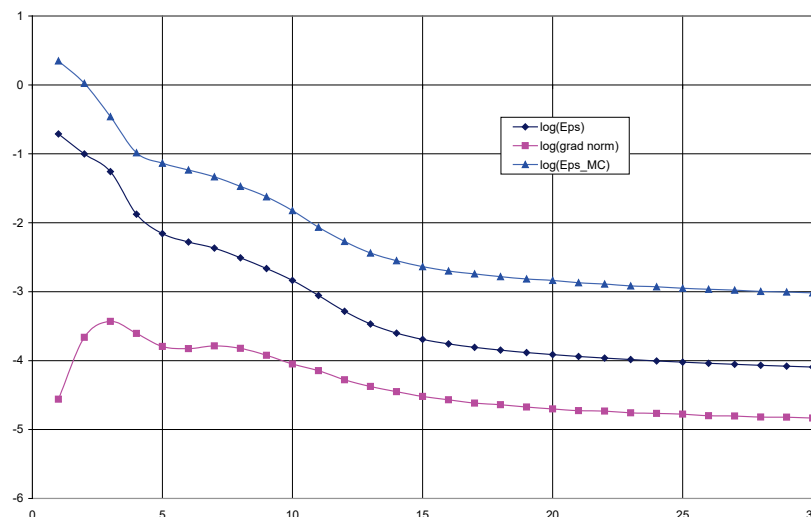


**Рисунок 1** – Точная функция (16)



**Рисунок 2** – Аппроксимация (16) с помощью канонического разложения

На рисунке 3 представлено поведение разных критериев сходимости в зависимости от числа итераций. В логарифмической форме представлены невязки  $\mathcal{E}_\Sigma$  (10) (Eps), глобальная невязка, оцененная с помощью Монте-Карло  $\mathcal{E}_{MC}$  (5) (Eps\_MC) и норма градиента невязки (grad norm).

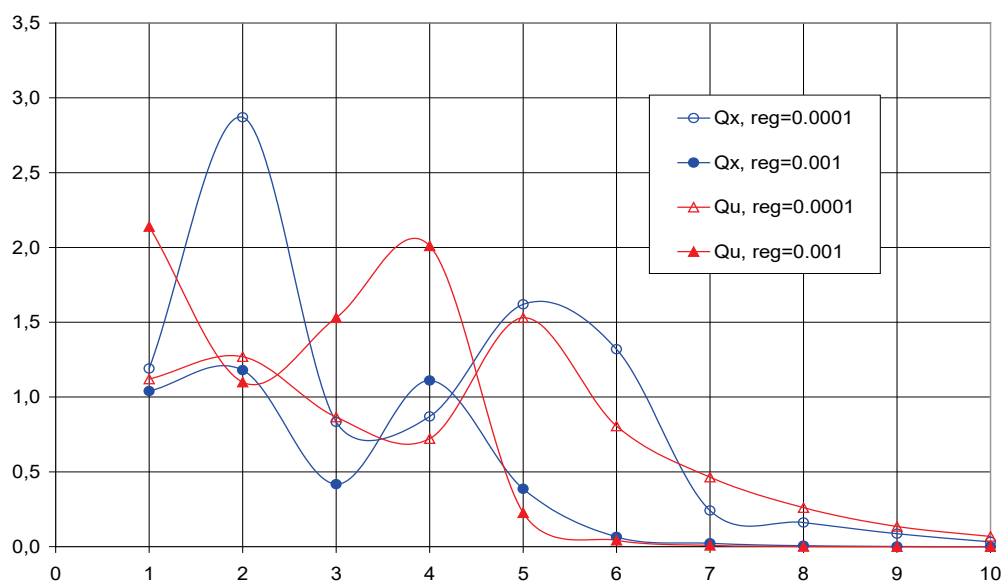


**Рисунок 3** – Разные критерии сходимости в зависимости от числа итераций

Невязки  $\varepsilon_\Sigma$  (10) (Eps) и глобальная невязка, оцененная с помощью Монте-Карло  $\varepsilon_{MC}$  (5) (Eps\_MC), демонстрируют качественно близкое поведение, пригодное для контроля сходимости. Норма градиента невязки (grad norm) на начальных итерациях ведет себя немонотонно, что ограничивает ее применимость.

Строго говоря, используемый метод не предусматривает монотонной минимизации глобальных критериев качества, так как минимизация осуществляется поточно. Однако, численные эксперименты демонстрируют монотонную сходимость. Данный алгоритм обеспечивает заметно более быструю сходимость в сравнении с использованным в [5].

В реальности в численных экспериментах мы получаем не истинный ранг функции (тензора), а аппроксимацию ранга  $r_\varepsilon$ , обеспечивающую невязку  $\varepsilon$ . Рисунок 4 демонстрирует, что используемый в работе вариант Тихоновской регуляризации [9] позволяет уменьшить эффективный ранг за счет увеличения параметра регуляризации (reg), но при этом качественно меняет ядра (увеличение параметра регуляризации сокращает соответствующий ранг).



**Рисунок 4** – Зависимость невязки от величины ранга для функции (16)

## 6. Результаты численных экспериментов по интерполяции результатов параметрических расчетов

В качестве примера рассмотрим двумерное стационарное поле течения в пространстве параметров размерностью от одного до трех. Оно соответствует случаю взаимодействия двух плоскопараллельных сверхзвуковых потоков воздуха, образующихся при обтекании симметричных клиньев. Структура течения описывается стационарными Маховским и регулярным режимами взаимодействия ударных волн. Параметрами является число Маха  $M \in (2,6)$  и углы отклонения потока  $(\theta_1, \theta_2) \in (15,30)$ .

Набор расчетов на одной пространственной сетке с разными параметрами  $(\theta_1, \theta_2, M)$  можно рассматривать как тензор порядка 6  $T_{p;i,j;k,n,m} = (\rho_{i,j;k,n,m}, u_{i,j;k,n,m}, v_{i,j;k,n,m}, e_{i,j;k,n,m})$  ( $p$  - номер газодинамической переменной,  $i, j$  - индексы узлов по координатам,  $k, n, m$  - номера параметров  $(\theta_1, \theta_2, M)$ ).

Рассмотрим построение канонического разложения на таком ансамбле решений, позволяющего радикальное сжатие данных.

Используя структуру данных типа “зонтик” будем строить каноническое разложение. Один наш расчет поля течения (элементарный “строительный блок”)  $T_{p;i,j;1,1,1}$  - это полностью заполненный тензор порядка 3.

Можно достаточно легко построить тензор порядка четыре используя последовательность тензоров порядка три (переходя с  $T_{p;i,j;1,1,1}$  на  $T_{p;i,j;2,1,1}$  и так далее). Это достаточно трудоемко ( $I_{ens} = 10 \div 100$  отдельных расчетов), но реализуемо. Однако уже при построении тензора порядка пять, возникнет проблема – на каждом слое надо заново набирать тензор ранга 4 снова, что вычислительно крайне трудоемко, тем более это не реализуемо для тензора следующего ранга. Таким образом, возникает дополнительная задача-восстановление всего тензора по неполным данным (Tensor completion) [8,13], которая также рассматривается описанной выше оптимизационной постановке со специальным выбором опорного ансамбля расчетов.

На рисунках 5 и 6 представлены результаты расчета поля течения (плотности) для  $\theta_1 = 15^\circ$  (отклонение верхнего потока),  $\theta_2 = 20^\circ$  (отклонение нижнего потока),  $M = 4$  и соответствующей аппроксимации с помощью канонического разложения (ранг 15, число точек ансамбля-2000).

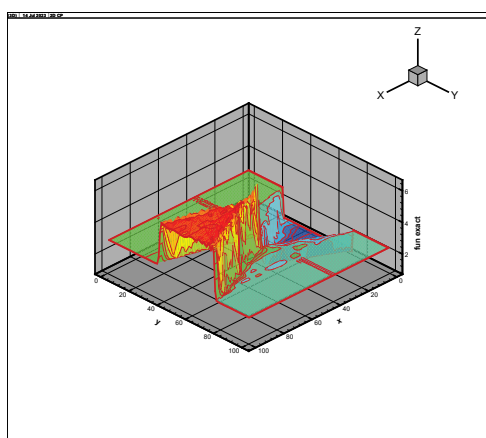


Рисунок 5 – Расчет поля плотности

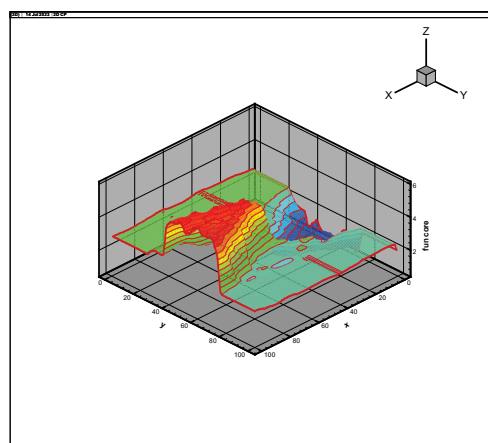


Рисунок 6 – Аппроксимация поля плотности с помощью канонического разложения

## 7. Обсуждение

В связи с некорректностью задачи [4] при определении ядер канонического разложения используется квадратичная регуляризации нулевого порядка по Тихонову [9].

Тензорные разложения достаточно активно используются для целей визуализации, поскольку позволяют построить легко вычислимую модель, аппроксимирующую труднообрабатываемый набор данных в параметрическом пространстве. Иногда для этих целей используется формат тензорного произведения [3] и кросс-аппроксимация [12]. Тензорные разложения (каноническое разложение, тензорный произведение, иерархический Таккер) используются для экономичного решения многомерных задач типа уравнения Больцмана [11].

Каноническое разложение позволяет эффективно аппроксимировать и хранить многомерные функции. Затраты компьютерного времени на работу с функциями в шестимерном пространстве (при использовании 100 узлов по каждой координате, что формально требует хранения и работу с  $10^{12}$  чисел) составляют 1-3 минуты на ПК (процессор Intel I5, 2.66 ГГц) при затратах памяти на хранение ядер в максимальном случае около  $10^5$  чисел.

## 8. Заключение

Предложен оптимизационный алгоритм определения ядер в каноническом разложении, требования по используемой памяти которого, существенно меньше, чем в стандартных методах, использующих матризацию и произведение Хатри-Рао.

Численные эксперименты на модельных шестимерных функциях и ансамбле численных расчетов течения сжимаемой жидкости со скачками уплотнения (описываемом четырехмерным тензором) показывают, что применение данного алгоритма для канонического разложения позволяет хранить и визуализировать функции в многомерном пространстве с очень умеренными затратами памяти и времени счета.

## 9. Список источников

- [1] Hackbusch. W. Tensor Spaces and Numerical Tensor Calculus. Springer, 2012. 500 p.
- [2] Yorick H., Willi-Hans S. Matrix Calculus, Kronecker Product and Tensor Product: A Practical Approach to Linear Algebra, Multilinear Algebra and Tensor Calculus with Software I, WOS 2019 373 p.
- [3] Oseledets I. V., Tensor- train decomposition, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317
- [4] V. D. Silva and L.-H. Lim, Tensor rank and the ill-posedness of the best low rank approximation problem, SIAM J. Matrix Anal. Appl., 30(3) 1084–1127, 2008.
- [5] Алексеев А.К., Бондарев А.Е., Пятакова Ю.С., О визуализации функций в многомерном пространстве с помощью канонического разложения, Научная визуализация, т. 14, N 3, с. 73-91, 2022
- [6] P. Comon, X. Luciani, and A. L.F. De Almeida, Tensor decompositions, alternating least squares and other tales, Journal of Chemometrics, vol. 23, no. 7-8, pp. 393–405, 2009.
- [7] A. Uschmajew, Local convergence of the alternating least squares algorithm for canonical tensor approximation, SIAM J. Matrix Anal. Appl. 33 (2) (2012) 639–652.
- [8] S. Rabanser, O. Shchur, S. Gunnemann, Introduction to Tensor Decompositions and their Applications in Machine Learning, arXiv:1711.10781v1, 2017.
- [9] Тихонов А.Н., Арсенин В.Я., Методы решения некорректных задач, М., Наука, 1979.
- [10] Wim P Krijnen, Theo K Dijkstra, and Alwin Stegeman. 2008. On the non-existence of optimal solutions and the occurrence of "degeneracy" in the CANDECOMP/PARAFAC Model. Psychometrika 73, 3 (2008), 431–439.
- [11] A. M. P. Boelens, D. Venturi, D. M. Tartakovsky, Parallel tensor methods for high-dimensional linear PDEs, J. Computat. Phys. 375 (2018) 519-539.

- [12] Oseledets I., Tyrtysnikov E., TT-cross approximation for multidimensional arrays, *Linear Algebra Appl.*, 432 (2010), pp. 70–88.
- [13] Y. Xu and W. Yin, A block coordinate descent method for regularized multi-convex optimization with applications to nonnegative tensor factorization and completion, *SIAM J. Imaging Sci.*, 6 (2013), pp. 1758–1789.