

ISSN 2305-9052 (Print)
ISSN 2410-7034 (Online)

ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА

BULLETIN

OF THE SOUTH URAL
STATE UNIVERSITY

СЕРИЯ

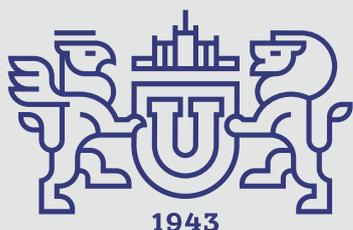
**ВЫЧИСЛИТЕЛЬНАЯ
МАТЕМАТИКА
И ИНФОРМАТИКА**

2025, том 14, № 2

SERIES

**COMPUTATIONAL
MATHEMATICS
AND SOFTWARE ENGINEERING**

2025, volume 14, no. 2



ВЕСТНИК



ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА

2025
Т. 14, № 2

ISSN 2305-9052 (Print)
ISSN 2410-7034 (Online)

СЕРИЯ

«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

Решением ВАК включен в Перечень научных изданий,
в которых должны быть опубликованы результаты диссертаций
на соискание ученых степеней кандидата и доктора наук

Учредитель — Федеральное государственное автономное образовательное учреждение
высшего образования «Южно-Уральский государственный университет
(национальный исследовательский университет)»

Тематика журнала:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Искусственный интеллект и машинное обучение
- Системное программирование
- Перспективные многопроцессорные архитектуры
- Облачные вычисления
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

Редакционная коллегия

Л.Б. Соколинский, д.ф.-м.н., проф., *гл. редактор*
М.Л. Цымблер, д.ф.-м.н., доц., *зам. гл. редактора*
Я.А. Краева, к.ф.-м.н., *отв. секретарь*
А.И. Гоглачев, *техн. редактор*

Редакционный совет

С.М. Абдуллаев, д.г.н., профессор
А. Андреяк, PhD, профессор (Германия)
В.И. Бердышев, д.ф.-м.н., акад. РАН, *председатель*
В.В. Воеводин, д.ф.-м.н., чл.-кор. РАН

Дж. Донгарра, PhD, профессор (США)
С.В. Зыкин, д.т.н., профессор
И.М. Куликов, д.ф.-м.н.
Д. Маллманн, PhD, профессор (Германия)
А.В. Панюков, д.ф.-м.н., профессор
Р. Продан, PhD, профессор (Австрия)
Г.И. Радченко, к.ф.-м.н., доцент (Австрия)
В.Н. Ушаков, д.ф.-м.н., чл.-кор. РАН
М.Ю. Хачай, д.ф.-м.н., чл.-кор. РАН
А. Черных, PhD, профессор (Мексика)
П. Шумяцкий, PhD, профессор (Бразилия)



BULLETIN

OF THE SOUTH URAL
STATE UNIVERSITY

2025

Vol. 14, no. 2

SERIES

“COMPUTATIONAL
MATHEMATICS AND SOFTWARE
ENGINEERING”

ISSN 2305-9052 (Print)
ISSN 2410-7034 (Online)

Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Universiteta.
Seriya “Vychislitel'naya Matematika i Informatika”

South Ural State University

The scope of the journal:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- Artificial intelligence and machine learning
- System software
- Advanced multiprocessor architectures
- Cloud computing
- Software engineering
- Computer graphics
- Internet technologies
- E-learning
- Database processing
- Data mining

Editorial Board

L.B. Sokolinsky, South Ural State University (Chelyabinsk, Russia)
M.L. Zymbler, South Ural State University (Chelyabinsk, Russia)
Ya.A. Kraeva, South Ural State University (Chelyabinsk, Russia)
A.I. Goglachev, South Ural State University (Chelyabinsk, Russia)

Editorial Council

S.M. Abdullaev, South Ural State University (Chelyabinsk, Russia)
A. Andrzejak, Heidelberg University (Germany)
V.I. Berdyshev, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
J. Dongarra, University of Tennessee (USA)
M.Yu. Khachay, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
I.M. Kulikov, Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch of RAS (Novosibirsk, Russia)
D. Mallmann, Julich Supercomputing Centre (Germany)
A.V. Panyukov, South Ural State University (Chelyabinsk, Russia)
R. Prodan, Alpen-Adria-Universität Klagenfurt (Austria)
G.I. Radchenko, Silicon Austria Labs (Graz, Austria)
P. Shumyatsky, University of Brasilia (Brazil)
A. Tchernykh, CICESE Research Center (Mexico)
V.N. Ushakov, Institute of Mathematics and Mechanics, Ural Branch of the RAS (Yekaterinburg, Russia)
V.V. Voevodin, Lomonosov Moscow State University (Moscow, Russia)
S.V. Zykin, Sobolev Institute of Mathematics, Siberian Branch of the RAS (Omsk, Russia)

Содержание

УЛУЧШЕНИЕ ОЦЕНКИ СТОИМОСТНОЙ МЕРЫ РИСКА МНОГОМЕРНЫХ ПОРТФЕЛЕЙ С ПОМОЩЬЮ СМЕСИ ВЕРОЯТНОСТНЫХ АНАЛИЗАТОРОВ ГЛАВНЫХ КОМПОНЕНТ Н.В. Волков	5
ПОИСК ОПТИМАЛЬНЫХ ВЕСОВ ДЛЯ ФУНКЦИИ ЯДРА АКУШСКОГО В.В. Луценко, Д.Е. Горлачев, Н.М. Мирный, М.Г. Бабенко	26
ВЫПОЛНЕНИЕ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ НА MLOPS ПЛАТФОРМЕ НИУ ВШЭ А.С. Хританков, В.А. Полежаев, Г.А. Жуликов, М.С. Галынчик, Н.А. Климин, К.Е. Сахаров, В.О. Минченков, И.В. Спирин, И.И. Крупнов, С.Ф. Якушева, А.С. Маратканова, В.И. Козырев, П.С. Костенецкий, Х.М. Салех	42
СУПЕРКОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СВЕРХЗВУКОВОГО ТЕЧЕНИЯ ВОКРУГ ТЕЛ ВРАЩЕНИЯ РАЗЛИЧНОЙ ФОРМЫ С.В. Поляков, В.О. Подрыга, Н.И. Тарасов	67

Contents

IMPROVING VALUE-AT-RISK ESTIMATION FOR MULTIVARIATE PORTFOLIOS USING A MIXTURE OF PROBABILISTIC PRINCIPAL COMPONENT ANALYZERS N.V. Volkov	5
SEARCHING OF OPTIMAL WEIGHTS FOR THE AKUSHSKY CORE FUNCTION V.V. Lutsenko, D.E. Gorlancev, N.M. Mirny, M.G. Babenko	26
DISTRIBUTED COMPUTATIONAL EXPERIMENTS IN THE MLOPS PLATFORM OF HSE UNIVERSITY A.S. Khritankov, V.A. Polezhaev, G.A. Zhulikov, M.S. Halynchik, N.A. Klimin, K.E. Sakharov, V.O. Minchenkov, I.V. Spirin, I.I. Krupnov, S.F. Yakusheva, A.S. Maratkanova, V.I. Kozyrev, P.S. Kostenetskiy, H.M. Salekh	42
SUPERCOMPUTER MODELING OF SUPERSONIC FLOW AROUND ROTATION BODIES OF DIFFERENT SHAPES S.V. Polyakov, V.O. Podryga, N.I. Tarasov	67



This issue is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

УЛУЧШЕНИЕ ОЦЕНКИ СТОИМОСТНОЙ МЕРЫ РИСКА МНОГОМЕРНЫХ ПОРТФЕЛЕЙ С ПОМОЩЬЮ СМЕСИ ВЕРОЯТНОСТНЫХ АНАЛИЗАТОРОВ ГЛАВНЫХ КОМПОНЕНТ

© 2025 Н.В. Волков

Московский физико-технический институт

(141701 Московская область, Долгопрудный, Институтский пер., д. 9)

E-mail: nikita.v.volkov@phystech.edu

Поступила в редакцию: 23.03.2025

В работе предлагается новый подход для оценки стоимостной меры риска (VaR) многомерных портфелей, основанный на смеси вероятностных анализаторов главных компонент (mPPCA) и информационном критерии Акаике. Проверяется эффективность рассматриваемого подхода на основе исторических данных с учетом различного количества компонент смесей в методе mPPCA. Исследование проводится на 100 сильно и 100 слабо диверсифицированных портфелях акций индекса S&P 500 за период 2009–2023 гг., используя скользящие окна размером 350 торговых дней. Вероятностный метод главных компонент (PPCA) позволяет моделировать сложные зависимости между активами и учитывать «тяжелые» хвосты распределений. Благодаря этому метод mPPCA превосходит классический метод главных компонент (PCA) в точности оценки. Помимо этого, за счет понижения размерности модель оказывается вычислительно существенно легче и стабильнее, чем смесь гауссовских распределений (GMM). В работе показывается зависимость волатильности и «тяжести» хвостов распределений лог-приростов стоимости портфеля как от оптимального количества компонент в методе mPPCA, так и от минимального достаточного количества основных компонент в методах PCA и PPCA для объяснения 80 % дисперсии в данных. Новый подход с оптимизацией количества компонент методом mPPCA показывает более высокие результаты, чем подходы с методами GMM, PCA и PPCA, особенно на слабо диверсифицированных портфелях. В работе описаны подходы по оптимизации обучения метода mPPCA и проведена обширная оценка эффективности на основе исторических данных (бэктестинг). Использование JIT-компиляции, «теплого старта» обучения метода mPPCA на каждом новом положении окна и трехступенчатый алгоритм поиска меры VaR позволяют существенно ускорить эксперименты по сравнению с обычной реализацией.

Ключевые слова: Value at Risk, VaR, PCA, PPCA, mPPCA, бэктестинг, понижение размерности.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Волков Н.В. Улучшение оценки стоимостной меры риска многомерных портфелей с помощью смеси вероятностных анализаторов главных компонент // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2025. Т. 14, № 2. С. 5–25. DOI: 10.14529/cmse250201.

Введение

Каждой компании и каждому банку необходимы методы измерения и контроля своих финансовых рисков для различных будущих сценариев, чтобы поддерживать целевой рейтинг. В качестве классической меры риска используется стоимостная мера риска (VaR, Value At Risk). Если в портфеле инвестора или компании есть только один актив, то оценка параметров одномерного распределения и вычисление меры VaR являются стандартной задачей [1].

Оптимизация финансового портфеля при высоких размерностях значительно сложнее. Это связано с необходимостью оценивать многомерное совместное распределение сильно скоррелированных активов [2–4]. Как правило, для решения этой проблемы предлагается

использовать методы понижения размерности, в частности метод главных компонент (РСА, Principal Component Analysis), введенный в работе [5]. Применение метода РСА к оценке финансовых рисков описано во многих работах (см., например, [6–8]). В работе [9] авторы с применением метода РСА выделили некоррелированные компоненты, которые являются линейными комбинациями исходных активов и объясняют достаточную долю дисперсии портфеля, а затем, основываясь на них, упростили построение границы эффективных портфелей в задаче Марковитца, введенной в [10]. На основе полученного подхода в работе [11] авторы применили метод выделения главных компонент для определения диверсифицированности портфеля. В [12, 13] данный подход обобщен так, что максимально диверсифицированные портфели должны включать в себя активы в практически одинаковых долях, минимизируя при этом дисперсию доходности портфеля. В [8] метод РСА и ядерный метод главных компонент (Kernel PCA) применяются для оптимизации долгосрочного инвестирования на основе диверсификации Марковица.

Также для оптимизации многомерных портфелей и оценки их риска часто применяется смесь гауссовских распределений (GMM, Gaussian Mixture Models), которая позволяет моделировать «тяжелые» хвосты (высокую вероятность по сравнению с гауссовским распределением сильных отклонений случайной величины от своего среднего) за счет использования нескольких гауссовских компонент [14, 15]. Однако для портфелей большой размерности практическое применение метода GMM затруднено, так как требуется оценка полной матрицы ковариаций для каждой смеси, что существенно увеличивает вычислительную сложность, а также может привести к неустойчивости оценок. Чтобы преодолеть эти проблемы, в ряде исследований предлагается использовать регуляризацию и штрафные подходы для стабилизации процесса оценивания и уменьшения эффективной размерности задачи. В частности, в работе [16] предложен подход на основе максимизации правдоподобия со штрафом и с разреживанием (sparsity) обратных ковариационных матриц, что уменьшает количество параметров и стабилизирует оценки. Также для повышения точности оценивания ковариационных матриц компонент в работе [17] предложена аналитическая регуляризация типа Ледуа–Вольфа (Ledoit–Wolf shrinkage), встроенная в алгоритм ожидания-максимизации (EM, Expectation-Maximization, [18]).

В данной работе рассматривается альтернативный подход, сочетающий в себе преимущества методов РСА и GMM в приложении к оценке риска многомерных портфелей, основанный на смеси вероятностных анализаторов главных компонент (mPPCA, Mixture of Probabilistic Principal Component Analyzers), введенной в работах [19, 20]. Предложенный метод активно исследуется, например, в приложении к мониторингу мультимодальных промышленных процессов и поиску в них неисправностей [21–25], при этом он крайне мало рассмотрен в литературе в применении к задачам финансов [26, 27] и вовсе не рассматривался применительно к задаче оценки финансового риска. В отличие от РСА, этот метод предполагает вероятностную модель данных и оптимизирует не долю объясненной дисперсии, а функцию правдоподобия. Таким образом, благодаря вероятностной модели, отдельной оценке шума и совмещению нескольких РРСА, становится возможным точнее учитывать «тяжелые» хвосты распределений и лучше оценивать меру VaR. При этом метод РСА, напротив, крайне неустойчив к выбросам, которые часто встречаются в финансовых данных. Одновременно метод mPPCA обладает преимуществом смеси GMM в использовании смеси нормальных распределений, но при этом за счет понижения размерности не страдает проблемой оценки полноразмерных матриц ковариаций.

Кроме того, разработан и протестирован гибридный подход для оценки меры $\text{VaR}_{5\%}$ с адаптивным выбором нужного количества компонент в методе mPPCA с использованием информационного критерия Акаике (AIC, Akaike Information Criterion) [28]. Отметим, что при оптимизации портфеля крайне важно быстро оценивать риск без повторного обучения модели для каждого случая. Именно этот принцип лежит в основе предложенного метода.

Для корректной оценки качества метода расчета меры VaR необходимо проведение оценки эффективности на основе исторических данных (бэктестинга) [15]. В работе проводится масштабный бэктестинг результатов оценки меры $\text{VaR}_{5\%}$ с помощью метода mPPCA на различных 100 сильно и 100 слабо диверсифицированных портфелях из акций индекса S&P 500 за период 2005–2023 гг. для mPPCA с разным количеством смесей и адаптивным подбором смесей на основе критерия AIC. Обучение алгоритмов проходит на скользящих окнах размером в 350 торговых дней [29, 30], после чего делается оценка VaR на следующий за окном день. Оценка меры VaR для каждого портфеля бэктестируется на прохождении классического Купик-теста [31]. Рассмотрение разных видов портфелей важно, так как все они могут возникать при решении задачи оптимизации портфеля и критично, чтобы алгоритм оценки риска корректно на них работал.

В работе проводится анализ зависимости изменчивости цены (волатильности) и «тяжести» хвостов распределений лог-приростов следующего дня от оптимального количества компонент смеси в методе mPPCA и от минимально достаточного количества основных компонент в методах PPCA и PCA (для объяснения 80 % дисперсии данных). Отдельно предлагается алгоритм расчета доли объясненной дисперсии в случае метода PPCA, его теоретическое обоснование и применение на реальных данных.

В работе также предложены следующие подходы к оптимизации обучения смеси метода PPCA и проведения обширного бэктестинга: компиляция «точно в нужное время» (JIT, Just In Time) Python-кода за счет использования библиотеки Numba [32] и адаптации всего кода под эту библиотеку; присвоение стартовой точке при обучении метода mPPCA в новом положении окна значения оптимума в предыдущем положении окна; трехэтапный поиск численного решения уравнения при финальной оценке VaR.

В разделе 1 представлены основные понятия и подходы, используемые в работе, обосновывается расчет доли объясненной дисперсии в случае метода PPCA. Раздел 2 содержит описание алгоритмов и процесса их обучения. Результаты экспериментов рассмотрены в разделе 3. В заключении приводятся резюме работы, краткая сводка полученных результатов и итоговые выводы.

1. Теоретическая часть

1.1. Probabilistic PCA и смесь PPCA

В отличие от классического метода PCA, вероятностный метод главных компонент (PPCA, Probabilistic PCA) [19] предполагает вероятностную модель данных, которая обеспечивает большую устойчивость к выбросам и пропущенным данным. Пусть имеется выборка из n объектов

$$X = \{x_i\}_{i=1}^n, \quad x_i \in \mathbb{R}^d. \quad (1)$$

Предполагается существование латентного пространства размерности $q < d$, и связь наблюдений x_i с латентными переменными $y_i \in \mathbb{R}^q$ задается линейной моделью с гауссовским

шумом:

$$x_i = \mu + Wy_i + \varepsilon_i, \quad y_i \sim \mathcal{N}(0, I_q), \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2 I_d), \quad (2)$$

где $\mu \in \mathbb{R}^d$ — вектор средних, W — матрица вложения размерности $d \times q$, а $\sigma^2 I_d$ — ковариационная матрица шума. Тогда безусловное распределение x_i имеет вид

$$x_i \sim \mathcal{N}(\mu, C), \quad C = WW^\top + \sigma^2 I_d, \quad (3)$$

где C — ковариационная матрица, а апостериорное распределение латентных переменных при наблюдении x_i задается выражением

$$y_i | x_i \sim \mathcal{N}\left(M^{-1}W^\top(x_i - \mu), \sigma^2 M^{-1}\right), \quad M = W^\top W + \sigma^2 I_q. \quad (4)$$

Оценка параметров (W, μ, σ^2) производится максимизацией логарифма правдоподобия

$$\mathcal{L}(W, \mu, \sigma^2) = -\frac{nd}{2} \log(2\pi) - \frac{n}{2} \log \det(C) - \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^\top C^{-1} (x_i - \mu) \quad (5)$$

с использованием EM-алгоритма (см. [19]).

Для моделирования более сложных распределений метод РРСА естественно обобщается на случай смеси из K компонент [20] — mPPCA с K компонентами. Каждая компонента имеет собственный набор параметров (μ_k, W_k, σ_k^2) , и модель смеси записывается в виде

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, W_k W_k^\top + \sigma_k^2 I_d), \quad (6)$$

где $\{\pi_k\}_{k=1}^K$ — веса смеси, удовлетворяющие условиям $\pi_k \geq 0$, $\sum_{k=1}^K \pi_k = 1$.

Параметры модели метода mPPCA оцениваются максимизацией логарифма правдоподобия смеси

$$\mathcal{L}_{\text{mPPCA}} = \sum_{i=1}^n \log \left[\sum_{k=1}^K \pi_k \mathcal{N}(x_i | \mu_k, W_k W_k^\top + \sigma_k^2 I_d) \right]. \quad (7)$$

Таким образом, метод mPPCA сочетает преимущества вероятностного подхода и снижения размерности, эффективно описывая многомерные распределения с «тяжелыми» хвостами и сложной структурой, что особенно важно в задачах оценки финансовых рисков.

1.2. Определение доли объясненной дисперсии для РРСА

В классическом методе РСА доля дисперсии, объясненной первыми q главными компонентами, определяется как отношение суммы первых q собственных значений к сумме всех собственных значений ковариационной матрицы данных:

$$\frac{\lambda_1 + \lambda_2 + \dots + \lambda_q}{\lambda_1 + \lambda_2 + \dots + \lambda_d}. \quad (8)$$

Однако метод РРСА оперирует вероятностной моделью, в которой, в том числе, дополнительно оценивается и шум с дисперсией σ^2 , поэтому подход для определения доли дисперсии требует модификации. Ниже предлагается и теоретически обосновывается подход для расчета доли объясненной дисперсии в случае метода РРСА.

Утверждение 1. Пусть в модели РРСА размерность наблюдаемого пространства равна d , а размерность латентного пространства — $q < d$. Согласно [19] оценка максимального

правдоподобия матрицы вложений W имеет вид

$$W = U_q(\Lambda_q - \sigma^2 I_q)^{1/2} R, \quad (9)$$

где

- $U_q \in \mathbb{R}^{d \times q}$ — матрица, столбцами которой являются первые q собственных векторов выборочной ковариационной матрицы;
- $\Lambda_q = \text{diag}(\lambda_1, \dots, \lambda_q)$ — диагональная матрица, содержащая первые q собственных значений ковариационной матрицы данных;
- σ^2 — оценка максимального правдоподобия дисперсии шума, вычисляемая как среднее оставшихся $(d - q)$ собственных значений;
- $R \in \mathbb{R}^{q \times q}$ — произвольная ортогональная матрица ($RR^\top = I_q$).

В таком случае аналог доли объясненной дисперсии для модели РРСА определяется следующим образом:

$$\text{coverage}(q) = \frac{\text{trace}(WW^\top)}{\text{trace}(WW^\top) + d\sigma^2} = \frac{\sum_{i=1}^q (\lambda_i - \sigma^2)}{\sum_{i=1}^q (\lambda_i - \sigma^2) + d\sigma^2}. \quad (10)$$

Доказательство. Из оригинальной работы по методу РРСА [19] следует, что оценка максимального правдоподобия для матрицы вложений W представима в виде (9). Рассмотрим выражение для WW^\top :

$$WW^\top = U_q(\Lambda_q - \sigma^2 I_q)^{1/2} R (U_q(\Lambda_q - \sigma^2 I_q)^{1/2} R)^\top. \quad (11)$$

Учитывая ортогональность матрицы R и матрицы собственных векторов U_q , имеем

$$WW^\top = U_q(\Lambda_q - \sigma^2 I_q)U_q^\top. \quad (12)$$

Следовательно, след матрицы WW^\top выражается через собственные значения как

$$\text{trace}(WW^\top) = \text{trace}(U_q(\Lambda_q - \sigma^2 I_q)U_q^\top) = \sum_{i=1}^q (\lambda_i - \sigma^2). \quad (13)$$

Полная дисперсия в методе РРСА состоит из дисперсии, объясняемой латентными переменными, и дисперсии шума. Так как дисперсия шума равна σ^2 для каждой из d компонент, суммарный вклад шума равен $d\sigma^2$. Таким образом, итоговое выражение для аналога доли объясненной дисперсии имеет вид

$$\text{coverage}(q) = \frac{\sum_{i=1}^q (\lambda_i - \sigma^2)}{\sum_{i=1}^q (\lambda_i - \sigma^2) + d\sigma^2}. \quad (14)$$

□

Полученная формула (10) позволяет аналитически рассчитывать аналог доли объясненной дисперсии для метода РРСА, обеспечивая корректное сопоставление с классическим методом РСА и удобство при выборе оптимальной латентной размерности.

1.3. Информационный критерий Акаике для mPPCA

Для автоматического выбора оптимального числа компонент смеси mPPCA в каждом скользящем окне используется информационный критерий Акаике [28], что позволяет балансировать между качеством аппроксимации данных и сложностью модели.

Обозначим набор параметров смеси mPPCA с K компонентами и размерностью латентного пространства q следующим образом:

$$\theta = \{\pi_k, \mu_k, W_k, \sigma_k^2\}_{k=1}^K, \quad (15)$$

где π_k — веса компонент смеси, μ_k — векторы средних компонент, W_k — матрицы вложений, а σ_k^2 — дисперсии шума.

Для модели mPPCA с K компонентами, размерностью данных d и размерностью латентного пространства q число p параметров вычисляется по формуле

$$p = \underbrace{Kdq}_{\text{матрицы } W_k} + \underbrace{Kd}_{\text{векторы средних } \mu_k} + \underbrace{K}_{\text{дисперсии } \sigma_k^2} + \underbrace{(K-1)}_{\text{веса смеси } \{\pi_k\}}, \quad (16)$$

где

- Kdq — количество параметров матриц W_k (каждая размером $d \times q$);
- Kd — число параметров средних векторов μ_k (каждый вектор размерности d);
- K — число дисперсий σ_k^2 (одна на каждую компоненту смеси);
- $(K-1)$ — число свободных параметров для весов смеси $\{\pi_k\}$, так как выполняется условие нормировки $\sum_{k=1}^K \pi_k = 1$.

Таким образом, полная формула для числа параметров модели записывается следующим образом:

$$p = K(dq + d + 1) + (K - 1). \quad (17)$$

Формула для критерия AIC смеси mPPCA задается выражением

$$\text{AIC}_{\text{mPPCA}} = 2p - 2\mathcal{L}_{\text{mPPCA}}(\hat{\theta}), \quad (18)$$

где $\hat{\theta}$ — оценка параметров, полученная после обучения модели на окне, а $\mathcal{L}_{\text{mPPCA}}(\hat{\theta})$ — значение логарифма правдоподобия mPPCA при параметрах $\hat{\theta}$.

1.4. Оценка меры VaR

Для вычисления меры VaR заданного уровня α для конкретного портфеля с весами активов w полученные многомерные распределения проецируются на одномерное распределение доходностей портфеля r_p следующим образом:

$$r_p \sim \sum_{k=1}^K \pi_k \mathcal{N}(w^T \mu_k, w^T C_k w). \quad (19)$$

Затем численно решается нелинейное уравнение, определяющее квантиль уровня α :

$$F_{r_p}(\text{VaR}_\alpha) = \alpha, \quad (20)$$

где $F_{r_p}(\cdot)$ — функция распределения смеси нормальных распределений доходностей портфеля r_p . Решение уравнения проводится в три этапа:

- определяются границы поиска в виде минимального и максимального квантилей среди всех смесей;
- границы поиска уточняются с помощью метода бисекции;
- производится уточнение корня методом Ньютона—Рафсона (история появления метода [33]).

Таким образом, предложенный подход на основе метода mPPCA учитывает вероятностную структуру многомерных данных и эффективно вычисляет меру VaR, оставаясь более устойчивым и менее вычислительно затратным, чем классические подходы PCA и GMM.

1.5. Бэктестинг меры VaR

Пусть $\{X_t\}_{t=1}^n$ обозначает ряд доходностей, а $\text{VaR}_\alpha(X_t)$ — оценка меры VaR на уровне α . Для каждого дня определим индикатор исключения:

$$I_t = \begin{cases} 1, & \text{если } X_t < -\text{VaR}_\alpha(X_t), \\ 0 & \text{иначе.} \end{cases} \quad (21)$$

Если модель корректна, то количество исключений $n_1 = \sum_{t=1}^n I_t$ можно рассматривать как реализацию биномиальной случайной величины с параметром α . Для проверки гипотезы о корректности модели в работе используется тест Купика [31], основанный на асимптотическом поведении отношения правдоподобия.

Тест Купика [31] оценивает, совпадает ли общее количество нарушений с ожидаемым. Пусть

$$n_1 = \sum_{t=1}^n I_t, \quad n_0 = n - n_1, \quad \hat{p} = \frac{n_1}{n}. \quad (22)$$

Тогда статистика теста безусловного покрытия (LR_{uc}) определяется как

$$\text{LR}_{\text{uc}} = -2 \ln \left[\frac{(1 - \alpha)^{n_0} \alpha^{n_1}}{(1 - \hat{p})^{n_0} \hat{p}^{n_1}} \right]. \quad (23)$$

При справедливости нулевой гипотезы « $p = \alpha$ » (т. е. корректном безусловном покрытии) эта статистика асимптотически имеет распределение χ^2 с 1 степенью свободы. Если значение LR_{uc} велико и его p -value меньше 5%, то гипотеза о совпадении средней доли нарушений с α отвергается.

2. Описание алгоритмов

Код основных методов выложен в открытом доступе в репозитории [34].

2.1. Реализация алгоритма mPPCA и процедура обучения

Пусть на текущем скользящем окне размером Δ торговых дней имеются наблюдения доходностей активов в виде матрицы $X \in \mathbb{R}^{\Delta \times d}$, элементы которой обозначим через $x_{i,j}$, где i — индекс наблюдения (день), j — индекс актива, Δ — длина окна, d — количество активов.

Z-нормализация данных. Перед обучением модели данные нормализуются по каждому признаку (активу):

$$x_{i,j}^{\text{norm}} = \frac{x_{i,j} - \bar{x}_j}{s_j}, \quad \text{где } \bar{x}_j = \frac{1}{\Delta} \sum_{i=1}^{\Delta} x_{i,j}, \quad s_j = \sqrt{\frac{1}{\Delta} \sum_{i=1}^{\Delta} (x_{i,j} - \bar{x}_j)^2}. \quad (24)$$

Инициализация параметров модели. Перед началом скольжения окна параметры модели инициализируются следующим образом.

- Нормализованные данные разделяются на K кластеров методом K-means (см. [35, 36]).
- К каждому из полученных кластеров применяется метод PCA, из результатов которого получают начальные приближения параметров смеси mPPCA:
 - векторы средних μ_k ;
 - матрицы вложений W_k ;
 - дисперсии шума σ_k^2 ;
 - веса компонент смеси π_k (на основании количества точек в каждом из кластеров).

Обучение на основе применения EM-алгоритма. На каждом временном окне параметры метода mPPCA оцениваются с использованием EM-алгоритма [18], включающего E-шаг (Expectation step) и M-шаг (Maximization step), которые повторяются, пока разница между новыми оценками параметров и предыдущими не станет незначительной.

E-шаг. Вычисляются вероятности принадлежности (responsibilities) каждого наблюдения x_i каждой компоненте смеси:

$$r_{i,k} = \frac{\pi_k \mathcal{N}(x_i | \mu_k, W_k W_k^\top + \sigma_k^2 I_d)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_i | \mu_j, W_j W_j^\top + \sigma_j^2 I_d)}, \quad (25)$$

где $r_{i,k}$ — вероятность принадлежности наблюдения x_i компоненте k .

M-шаг. На основе полученных на E-шаге вероятностей $r_{i,k}$ параметры смеси обновляются следующим образом.

- Веса компонент смеси:

$$\pi_k \leftarrow \frac{1}{\Delta} \sum_{i=1}^{\Delta} r_{i,k}. \quad (26)$$

- Векторы средних компонент:

$$\mu_k \leftarrow \frac{\sum_{i=1}^{\Delta} r_{i,k} x_i}{\sum_{i=1}^{\Delta} r_{i,k}}. \quad (27)$$

- Выборочные ковариационные матрицы компонент:

$$S_k = \frac{1}{\Delta \pi_k} \sum_{i=1}^{\Delta} r_{i,k} (x_i - \mu_k)(x_i - \mu_k)^\top. \quad (28)$$

- Матрицы вложений:

$$W_k \leftarrow S_k W_k \left(\sigma_k^2 I_q + M_k^{-1} W_k^\top S_k W_k \right)^{-1}, \quad (29)$$

где матрица M_k определяется выражением

$$M_k = W_k^\top W_k + \sigma_k^2 I_q. \quad (30)$$

- Дисперсии шума:

$$\sigma_k^2 \leftarrow \frac{1}{d} \text{trace} \left[S_k - S_k W_k M_k^{-1} W_k^\top \right]. \quad (31)$$

Важно отметить, что в ходе этих операций приходится обращаться матрицы не исходной размерности $d \times d$, а только матрицы размерности $q \times q$, где q — латентная размерность. Для повышения численной устойчивости при обращении матриц в любом случае используется регуляризация.

Обучение метода mPPCA на новом положении окна производится с так называемым «теплым стартом», при котором в качестве начальных приближений параметров модели берутся оценки, полученные на предыдущем положении окна. Это позволяет значительно сократить число итераций до достижения сходимости EM-алгоритма.

Оптимальное количество компонент смеси на каждом окне выбирается на основе минимизации информационного критерия Акаике, что позволяет автоматически балансировать между сложностью модели и точностью аппроксимации данных.

Алгоритм выбора числа компонент K по критерию АИС

1. Задаем диапазон значений K от K_{\min} до K_{\max} (в работе от 1 до 4).
2. Для каждого значения $K = K_{\min}, K_{\min} + 1, \dots, K_{\max}$:
 - оцениваем параметры модели θ с помощью EM-алгоритма;
 - вычисляем соответствующее значение $\text{AIC}_{\text{mPPCA}}(K)$.
3. Оптимальным числом компонент K^* считается то, при котором достигается минимальное значение критерия АИС:

$$K^* = \arg \min_K [\text{AIC}_{\text{mPPCA}}(K)]. \quad (32)$$

Таким образом, в каждом скользящем окне автоматически определяется оптимальное число компонент смеси mPPCA, исходя из компромисса между точностью аппроксимации данных и сложностью модели.

Описанная реализация алгоритма mPPCA с пошаговой нормализацией данных, «теплым стартом» и автоматическим выбором оптимального числа компонент по критерию АИС значительно повышает эффективность оценки VaR многомерных финансовых портфелей.

2.2. Численное решение для меры VaR

В реализации вычисления VaR_α для смеси одномерных нормальных распределений используется следующая схема.

1. Начальные границы поиска $[L, U]$ берутся на основе α -квантилей для каждой гауссовской компоненты смеси:

$$L = \min_k F^{-1}(\alpha; \mu_k, \sigma_k), \quad U = \max_k F^{-1}(\alpha; \mu_k, \sigma_k), \quad (33)$$

где F — функция распределения для нормального распределения $\mathcal{N}(\mu_k, \sigma_k^2)$. Благодаря этому отрезок $[L, U]$ наверняка содержит искомым квантиль смеси.

2. На отрезке $[L, U]$ выполняется несколько шагов классического метода бисекции, чтобы быстро получить аппроксимацию корня уравнения $F_{\text{mix}}(x) = \alpha$ с точностью порядка 10^{-2} по аргументу.
3. Полученный результат из бисекции выступает начальными данными для метода Ньютона—Рафсона, с помощью которого выполняется поиск решения с точностью уже порядка 10^{-6} . На каждом шаге для текущего приближения x вычисляется

$$x_{\text{new}} = x_{\text{old}} - \frac{F_{\text{mix}}(x_{\text{old}}) - \alpha}{f_{\text{mix}}(x_{\text{old}})}, \quad (34)$$

где $f_{\text{mix}}(\cdot)$ — смесь плотностей.

Симуляция на десяти миллионах различных нормальных смесей показывает ускорение подсчета квантилей с нужным уровнем точности более чем в три раза по сравнению с каждым из методов отдельно. Например, метод бисекции занимает 161 секунду, а трехшаговый — 45 секунд. Все эксперименты здесь и далее выполнялись в бесплатном Google Colab на центральном процессоре (CPU, Central Processing Unit), частота 2.2–2.8 ГГц, без использования графического процессора (GPU, Graphics Processing Unit).

2.3. JIT-компиляция и отказ от внешних библиотек

Для повышения скорости вычислений используется `@jit(nopython=True)` (или `@njit`) из библиотеки **Numba**, что дает перевод вычислительно интенсивных участков в машинный код. Это позволяет:

- избегать медленных Python-циклов в критических местах (все операции выполняются над NumPy-массивами или в простых `for`-циклах, которые Numba оптимизирует);
- минимизировать вызовы «тяжелых» библиотек (например, `scipy.stats`). Вместо этой функции CDF, PDF, PPF нормального распределения реализованы вручную, а вся линейная алгебра (например, умножение матриц) делается через NumPy-функции, совместимые с Numba;
- Данный подход вместе с использованием «теплого» старта дает ускорением в более чем 100 раз. К примеру обучение mPPCA со всеми оптимизациями с двумя смесями на 650 точках на 221 активе занимает 72 секунды, а без оптимизаций — более 4 часов.

3. Вычислительные эксперименты

3.1. Результаты бэкстестинга

Был проведен масштабный бэкстестинг результатов оценки меры $VaR_{5\%}$ с помощью метода mPPCA на различных 100 сильно и 100 слабо диверсифицированных портфелях на акциях из индекса S&P 500 за период 2009–2023 гг. (тестирование; обучение начинается ранее). Под слабо диверсифицированными портфелями понимаются портфели, в которых 90 % весов сосредоточены в 1–5 ведущих акциях, а под сильно диверсифицированными — портфели, где веса случайным образом распределены между всеми акциями. Рассматривались методы PCA, mPPCA и GMM с разным количеством компонент. Оценки меры VaR проверялись на прохождение классического Купик-теста. В табл. 1 приведены доли портфелей, на которых тест пройден успешно.

Таблица 1. Сравнение результатов бэктестирования оценок $VaR_{5\%}$

Метод	Слабо диверс.	Сильно диверс.
PCA	0.00	1.00
GMM 1	0.37	0.00
GMM 2	0.46	0.00
GMM 3	0.51	0.00
mPPCA 1	0.78	1.00
mPPCA 2	0.83	1.00
mPPCA 3	0.86	1.00
mPPCA 4	0.88	1.00
mPPCA AIC	0.89	1.00

Из полученных результатов следует, что для слабо диверсифицированных портфелей классический метод РСА не справляется с задачей, демонстрируя нулевую долю успешных прохождений теста Купика. Метод GMM существенно улучшает результаты по сравнению с методом РСА, однако даже при использовании трех компонент смеси доля успешно пройденных тестов составляет лишь 51 %. Однако при этом метод GMM крайне плохо показывает себя на диверсифицированных портфелях.

Наилучшие результаты достигаются при использовании метода mPPCA: уже при одной компоненте доля успешных прохождений значительно превышает результаты методов GMM и РСА, достигая 78 %. Дальнейшее увеличение числа компонент смеси до четырех последовательно повышает качество оценки, доходя до 88 %. Адаптивный выбор количества компонент по критерию AIC дополнительно улучшает результаты (до 89 %). Стоит отметить, что дальнейшее увеличение числа компонент смеси для метода mPPCA свыше четырех уже не приносит улучшения.

Таким образом, метод mPPCA демонстрирует явные преимущества перед методами РСА и GMM в оценке меры VaR, особенно для слабо диверсифицированных портфелей, совмещая эффективность понижения размерности и гибкость вероятностной модели смеси.

3.2. Влияние оптимального по критерию AIC числа компонент в mPPCA на лог-приросты следующего дня

Был также проведен анализ зависимости абсолютных значений лог-приростов следующего дня от оптимального по критерию AIC количества компонент в методе mPPCA (далее «группа»). Соответствующая ящичная диаграмма (box plot) изображена на рис. 1. Такие показатели, как среднее, стандартное отклонение, асимметрия и коэффициент эксцесса лог-приростов для каждой из групп приведены в табл. 2. Здесь и далее важно отметить, что речь идет о значениях лог-приростов каждого из активов на следующий день за текущим окном, на котором обучается модель и происходит оценка оптимального количества смесей в методе mPPCA. Из таблицы видно существенное изменение стандартных отклонений и коэффициентов асимметрии в зависимости от количества компонент в смеси. В группе с двумя смесями самые маленькие стандартное отклонение и коэффициент асимметрии, в группе с тремя смесями эти показатели средние, а в группе с четырьмя — самые высокие.

Таким образом, использование метода mPPCA дает возможность выявления скрытых рыночных режимов и учета потенциального уровня волатильности и экстремальных значений для оценки риска на следующий день.

Таблица 2. Статистики лог-приростов следующего дня в зависимости от оптимального количества смесей по критерию AIC

Группа	Среднее	СКО	Асимметрия	Эксцесс
2	0.0007	0.015	0.027	13.461
3	0.0008	0.020	0.170	20.516
4	0.0005	0.027	0.534	17.942

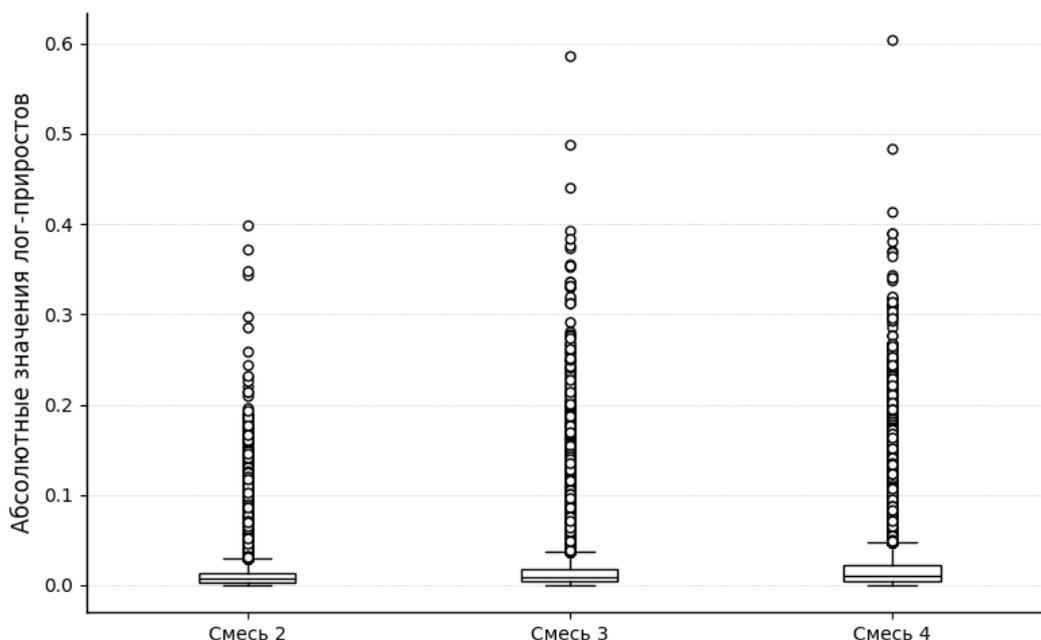


Рис. 1. Ящичная диаграмма (box plot) абсолютных значений лог-приростов и оптимального количества смесей в mPPCA по критерию AIC

3.3. Влияние минимально достаточного числа главных компонент в методах PCA и PPCA на абсолютные лог-приросты следующего дня

Была также рассмотрена зависимость абсолютных лог-приростов следующего дня от минимально достаточного количества основных компонент в методах PCA и PPCA для объяснения α процентов дисперсии данных. В качестве α бралось 80 % из эмпирических соображений. На рис. 2 изображены графики минимально достаточного количества основных компонент в методах PCA и PPCA, из которого видно, что количества требуемых компонент у методов PCA и PPCA крайне близки.

На рис. 3 приведены ящичные диаграммы абсолютных лог-приростов следующего дня, сгруппированных по числу необходимых главных компонент в методах PCA и PPCA, достаточных для объяснения 80 % дисперсии данных. Из графиков видно, что в группе (0, 40] наблюдаются наиболее крупные выбросы и более высокая медиана абсолютных значений, что указывает на повышенную волатильность.

В табл. 3 приведены параметры распределений абсолютных значений лог-приростов для тех же групп. Сравнение показывает, что в группе (0, 40] абсолютные лог-приросты характеризуются максимальным стандартным отклонением, наиболее высокими значениями асимметрии и эксцесса. Это согласуется с большим разбросом абсолютных лог-приростов на рис. 3. В группах с большим количеством необходимых компонент (более 40) волатильность и «толщина» хвостов распределения заметно снижаются, что отражается и в уменьшении выбросов на диаграмме абсолютных значений, и в относительно меньших значениях стандартного отклонения, асимметрии и эксцесса в таблице.

Дополнительно было проведено исследование связи между минимальным достаточным количеством главных компонент и средней корреляцией активов внутри скользящего окна.

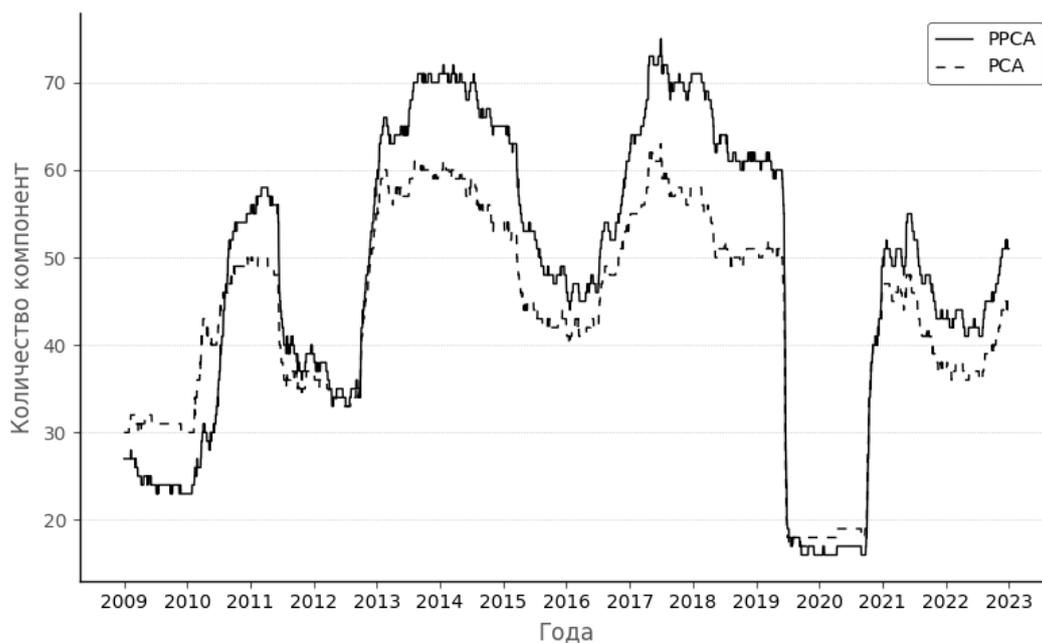


Рис. 2. Графики минимального достаточного количества основных компонент для объяснения 80 % дисперсии в методах PCA и PPCA

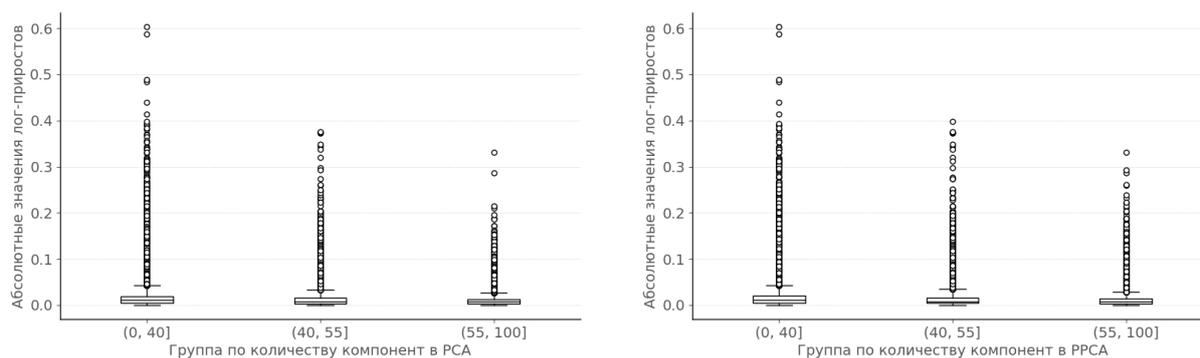


Рис. 3. Ящичные диаграммы (box plot) абсолютных значений лог-приростов и минимального достаточного количества основных компонент для методов PCA и PPCA

Таблица 3. Параметры распределений абсолютных значений лог-приростов на следующий день в зависимости от группы для методов PCA и PPCA

Группа	Среднее		СКО		Асимметрия		Эксцесс	
	PCA	PPCA	PCA	PPCA	PCA	PPCA	PCA	PPCA
(0, 40]	0.015	0.016	0.018	0.019	4.656	4.613	46.877	45.665
(40, 55]	0.011	0.012	0.012	0.013	3.689	3.914	34.699	38.115
55 и более	0.010	0.010	0.010	0.011	3.562	3.533	34.619	31.096

На каждом положении окна вычислялось среднее значение всех попарных корреляций активов, после чего оно сопоставлялось с минимально необходимым количеством компонент.

Полученный коэффициент корреляции между этими двумя величинами составил -78% и -79% для методов PCA и PPCA соответственно. Это демонстрирует тесную обратную зависимость между рыночной корреляцией и числом необходимых компонент. Из рис. 4 видно,

что периоды высоких средних корреляций совпадают с уменьшением количества главных компонент. Это наблюдение подтверждается и ящичной диаграммой на рис. 5: чем меньше необходимое количество компонент, тем выше средний уровень корреляций активов. С экономической точки зрения это естественный результат — во время кризиса корреляция активов значительно увеличивается, а значит, уменьшается количество основных компонент, с помощью которых можно достаточно точно описать общее движение портфеля. Однако в случае отсутствия кризиса все активы ведут себя по-разному, корреляции уменьшаются и, соответственно, требуется большее количество компонент для описания портфеля.

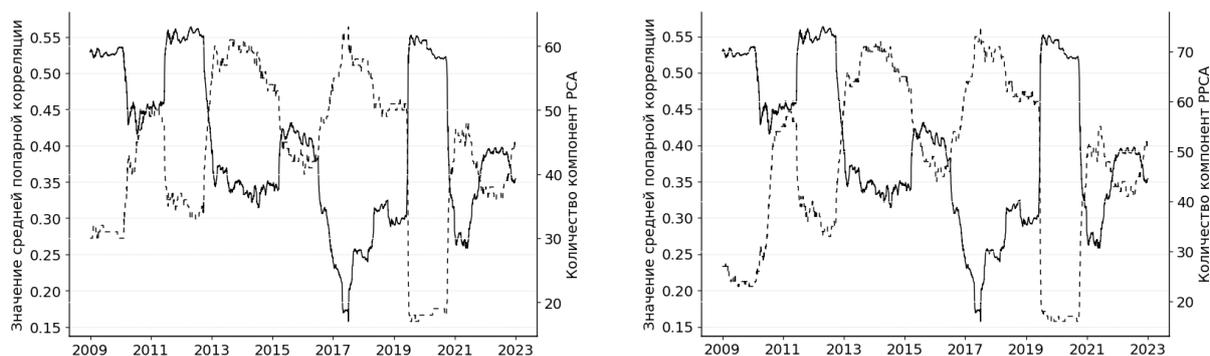


Рис. 4. Средняя попарная корреляция активов на окне и минимальное достаточное количество компонент для объяснения нужной доли дисперсии в PCA и PPCA

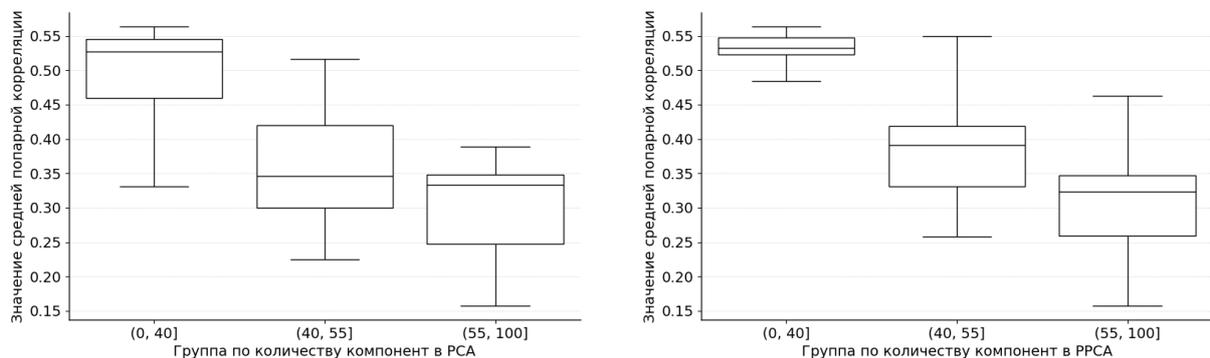


Рис. 5. Ящичные диаграммы (box plot) средней попарной корреляции активов внутри групп по количеству минимально достаточных компонент в PCA и PPCA

Заключение

В работе предложен подход к оценке меры $\text{VaR}_{5\%}$ для многомерных портфелей на основе смеси PPCA с адаптивным выбором оптимального количества компонент смеси по критерию AIC. Данный подход позволяет учитывать нелинейные зависимости и «тяжелые» хвосты распределений доходностей активов, что существенно улучшает точность оценки меры VaR в сравнении с традиционными методами PCA и GMM, особенно для слабо диверсифицированных портфелей. Масштабный бэктестинг на основе теста Купика, проведенный на 100 сильно и 100 слабо диверсифицированных портфелях из акций индекса S&P 500 за период 2009–2023 гг. на скользящих окнах размером в 350 торговых дней, подтвердил преимущество предложенного метода, где доля успешно пройденных тестов увеличилась до

89 % по сравнению с методом mPPCA при фиксированном числе компонент смеси, классических методов PCA и GMM.

В ходе работы выявлено, что оптимальное по критерию AIC количество компонент смеси в mPPCA демонстрирует прямую связь с характеристиками оцененного распределения лог-приростов следующего дня. Увеличение числа компонент соответствует росту волатильности и степени «тяжести» хвостов распределения. Предложен подход для расчета объясненной дисперсии в случае метода PPCA и показано, что минимальное достаточное число главных компонент (для объяснения 80 % дисперсии) в методах PCA и PPCA обратно коррелирует (-78 % и -79 %) со средней попарной корреляцией активов и также служит индикатором повышения волатильности и риска следующего торгового дня. Таким образом, предложенные статистики могут быть использованы как индикаторы стрессовых состояний финансовых рынков.

Важным результатом работы стало достижение высокой вычислительной эффективности алгоритма. Использование JIT-компиляции (Numba), «теплого старта» при перемещении окна и трехшагового алгоритма нахождения VaR позволило существенно ускорить обучение и бэктестинг предложенного подхода по сравнению с прямой реализацией.

Литература

1. Duffie D., Pan J. An overview of value at risk // *Journal of Derivatives*. 1997. Vol. 4, no. 3. P. 7–49. DOI: 10.3905/jod.1997.407971.
2. Ledoit O., Wolf M. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection // *Journal of Empirical Finance*. 2003. Vol. 10, no. 5. P. 603–621. DOI: 10.1016/S0927-5398(03)00007-0.
3. Fan J., Liao Y., Liu H. An overview of the estimation of large covariance and precision matrices // *The Econometrics Journal*. 2016. Mar. Vol. 19, no. 1. P. C1–C32. DOI: 10.1111/ectj.12061.
4. Pafka S., Kondor I. Noisy covariance matrices and portfolio optimization II // *Physica A: Statistical Mechanics and its Applications*. 2003. Vol. 319. P. 487–494. DOI: 10.1016/S0378-4371(02)01499-1.
5. Pearson K. LIII. On lines and planes of closest fit to systems of points in space // *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. 1901. Vol. 2, no. 11. P. 559–572. DOI: 10.1080/14786440109462720.
6. Kelly B., Malamud S., Pedersen L.H. Principal Portfolios // *The Journal of Finance*. 2023. Vol. 78, no. 1. P. 347–387. DOI: 10.1111/jofi.13199.
7. Mavungu M. Computation of financial risk using principal component analysis // *Algorithmic Finance*. 2023. Vol. 10, no. 1-2. P. 1–20. DOI: 10.3233/af-220339.
8. Kulikov A.V., Polozov D.S., Volkov N.V. Long-term investment optimization based on Markowitz diversification // *Business Informatics*. 2024. Vol. 18, no. 3. P. 56–69. DOI: 10.17323/2587-814X.2024.3.56.69.
9. Partovi M.H., Caputo M. Principal Portfolios: Recasting the Efficient Frontier // *Economics Bulletin*. 2004. Jan. Vol. 7. P. 1–10.
10. Markowitz H. Portfolio Selection // *The Journal of Finance*. 1952. Vol. 7, no. 1. P. 77–91. DOI: 10.2307/2975974.

11. Meucci A. Managing diversification // Risk. 2009. P. 74–79.
12. Lohre H., Neugebauer U., Zimmer C. Diversified Risk Parity Strategies for Equity Portfolio Selection // The Journal of Investing. 2012. Aug. Vol. 21. P. 111–128. DOI: 10.3905/joi.2012.21.3.111.
13. Lohre H., Opfer H., Ország G. Diversifying risk parity // The Journal of Risk. 2014. June. Vol. 16. P. 53–79. DOI: 10.21314/JOR.2014.284.
14. Luxenberg E., Boyd S. Portfolio construction with Gaussian mixture returns and exponential utility via convex optimization // Optimization and Engineering. 2024. Vol. 25, no. 1. P. 555–574. DOI: 10.1007/s11081-023-09814-y.
15. Indrè Morkūnaitė D.C., Leipus R. Evaluation of Value-at-Risk (VaR) using the Gaussian Mixture Models // Research in Statistics. 2024. Vol. 2, no. 1. P. 2346075. DOI: 10.1080/27684520.2024.2346075.
16. Ruan L., Yuan M., Zou H. Regularized Parameter Estimation in High-Dimensional Gaussian Mixture Models // Neural Computation. 2011. Vol. 23. P. 1605–1622. DOI: 10.1162/NECO_a_00128.
17. Halbe Z., Bortman M., Aladjem M. Regularized Mixture Density Estimation with an Analytical Setting of Shrinkage Intensities // IEEE Transactions on Neural Networks and Learning Systems. 2013. Vol. 24. P. 460–470. DOI: 10.1109/TNNLS.2012.2234477.
18. Dempster A.P., Laird N.M., Rubin D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm // Journal of the Royal Statistical Society. Series B (Methodological). 1977. Vol. 39, no. 1. P. 1–38. DOI: 10.1111/j.2517-6161.1977.tb01600.x.
19. Tipping M.E., Bishop C.M. Probabilistic Principal Component Analysis // Journal of the Royal Statistical Society Series B. 1999. Vol. 61, no. 3. P. 611–622. DOI: 10.1111/1467-9868.00196.
20. Tipping M.E., Bishop C.M. Mixtures of Probabilistic Principal Component Analyzers // Neural Computation. 1999. Feb. Vol. 11, no. 2. P. 443–482. DOI: 10.1162/089976699300016728.
21. Lyu Y., Zhou L., Cong Y., *et al.* Multirate mixture probability principal component analysis for process monitoring in multimode processes // IEEE Transactions on Automation Science and Engineering. 2023. Vol. 21, no. 2. P. 2027–2038. DOI: 10.1109/TASE.2023.3253285.
22. Tra V., Amayri M., Bouguila N. Unsupervised fault detection for building air handling unit systems using deep variational mixture of principal component analyzers // IEEE Transactions on Automation Science and Engineering. 2024. Vol. 21, no. 4. P. 6787–6803. DOI: 10.1109/tase.2023.3331347.
23. Zhang J., Chen M., Hong X. Nonlinear process monitoring using a mixture of probabilistic PCA with clusterings // Neurocomputing. 2021. Vol. 458. P. 319–326. DOI: 10.1016/j.neucom.2021.06.039.
24. Zhao B., Xiao X., Zhang W., *et al.* Self-Paced Probabilistic Principal Component Analysis for Data with Outliers // ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020. P. 3737–3741. DOI: 10.1109/icassp40776.2020.9054487.

25. Chao Han S.L., House L. Covariance-Guided Mixture Probabilistic Principal Component Analysis (C-MPPCA) // *Journal of Computational and Graphical Statistics*. 2015. Vol. 24, no. 1. P. 66–83. DOI: 10.1080/10618600.2014.891460.
26. Tzagkarakis G., Caicedo-Llano J., Dionysopoulos T. Exploiting market integration for pure alpha investments via probabilistic principal factors analysis // *Journal of Mathematical Finance*. 2013. Vol. 3, no. 1. P. 192–200. DOI: 10.4236/jmf.2013.31A018.
27. Begušić S., Kostanjčar Z. Cluster-Specific Latent Factor Estimation in High-Dimensional Financial Time Series // *IEEE Access*. 2020. Vol. 8. P. 164365–164379. DOI: 10.1109/ACCESS.2020.3021898.
28. Akaike H. A new look at the statistical model identification // *IEEE Transactions on Automatic Control*. 1974. Dec. Vol. 19, no. 6. P. 716–723. DOI: 10.1109/TAC.1974.1100705.
29. Brutti Righi M., Sergio Ceretta P. On the existence of an optimal estimation window for risk measures // *Applied Finance Letters*. 2015. Nov. Vol. 4, no. 1-2. P. 28–32. DOI: 10.24135/af1.v4i1and2.30.
30. Волков Н.В. Оценка VaR портфелей с применением методов понижения размерности PCA и PPCA // *Труды МФТИ*. 2025. Т. 17, № 1. С. 127–141.
31. Kupiec P.H. Techniques for verifying the accuracy of risk measurement models // *Journal of Derivatives*. 1995. Vol. 3, no. 2. P. 73–84. DOI: 10.3905/jod.1995.407942.
32. Lam S.K., Pitrou A., Seibert S. Numba: A LLVM-based Python JIT Compiler // *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM 2015, Austin, TX, USA, November 15, 2015*. ACM, 2015. P. 1–6. DOI: 10.1145/2833157.2833162.
33. Ypma T.J. Historical development of the Newton–Raphson method // *SIAM Review*. 1995. Vol. 37, no. 4. P. 531–551. DOI: <https://doi.org/10.1137/1037125>.
34. Volkov N. MPPCA for VaR Estimation: Source Code Repository. 2025. URL: <https://gitlab.com/n.volkovsky/mppca-for-var-estimation> (дата обращения: 20.03.2025).
35. MacQueen J. Some methods for classification and analysis of multivariate observations // *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1, Statistics*. Vol. 5. University of California Press. 1967. P. 281–298.
36. Lloyd S. Least squares quantization in PCM // *IEEE Transactions on Information Theory*. 1982. Vol. 28, no. 2. P. 129–137. DOI: 10.1109/TIT.1982.1056489.

Волков Никита Васильевич, кафедра дискретной математики, Московский физико-технический институт (национальный исследовательский университет) (Долгопрудный, Российская Федерация)

IMPROVING VALUE-AT-RISK ESTIMATION FOR MULTIVARIATE PORTFOLIOS USING A MIXTURE OF PROBABILISTIC PRINCIPAL COMPONENT ANALYZERS

© 2025 N.V. Volkov

Moscow Institute of Physics and Technology

(Institutskiy Pereulok 9, Dolgoprudny, Moscow Region, 141701 Russia)

E-mail: nikita.v.volkov@phystech.edu

Received: 23.03.2025

This paper proposes a novel approach for estimating the Value-at-Risk (VaR) of multidimensional portfolios, based on a mixture of probabilistic principal component analyzers (mPPCA) and the Akaike information criterion. The effectiveness of the proposed approach is evaluated on historical data, accounting for various numbers of mixture components in the mPPCA method. The study is performed on 100 highly diversified and 100 weakly diversified stock portfolios of the S&P 500 index over the 2009–2023 period, using rolling windows of 350 trading days. Probabilistic principal component analysis (PPCA) can model complex dependencies among assets while capturing the “heavy” tails of return distributions. As a result, the mPPCA method surpasses conventional principal component analysis (PCA) in the accuracy of VaR estimation. In addition, by reducing dimensionality, the model is computationally much more efficient and stable than a mixture of Gaussian distributions (GMM). The paper demonstrates how portfolio return volatility and tail heaviness depend both on the optimal number of components in mPPCA and on the minimal sufficient number of principal components in PCA and PPCA needed to explain 80 % of the variance in the data. The new approach, which optimizes the number of components in mPPCA, consistently achieves higher performance than GMM, PCA, or PPCA, especially for less diversified portfolios. The paper describes methods for optimizing mPPCA training and provides an extensive historical performance evaluation (backtesting). By employing just-in-time compilation, a “warm start” for mPPCA with each new window position, and a three-step algorithm for VaR estimation, the experiments are significantly accelerated compared with the standard implementation.

Keywords: Value at Risk, VaR, PCA, PPCA, mPPCA, backtesting, dimensionality reduction.

FOR CITATION

Vokov N.V. Improving Value-at-Risk Estimation for Multivariate Portfolios Using a Mixture of Probabilistic Principal Component Analyzers. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2025. Vol. 14, no. 2. P. 5–25. (in Russian) DOI: 10.14529/cmse250201.

This paper is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Duffie D., Pan J. An overview of value at risk. Journal of Derivatives. 1997. Vol. 4, no. 3. P. 7–49. DOI: 10.3905/jod.1997.407971.
2. Ledoit O., Wolf M. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. Journal of Empirical Finance. 2003. Vol. 10, no. 5. P. 603–621. DOI: 10.1016/S0927-5398(03)00007-0.

3. Fan J., Liao Y., Liu H. An overview of the estimation of large covariance and precision matrices. *The Econometrics Journal*. 2016. Mar. Vol. 19, no. 1. P. C1–C32. DOI: 10.1111/ectj.12061.
4. Pafka S., Kondor I. Noisy covariance matrices and portfolio optimization II. *Physica A: Statistical Mechanics and its Applications*. 2003. Vol. 319. P. 487–494. DOI: 10.1016/S0378-4371(02)01499-1.
5. Pearson K. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. 1901. Vol. 2, no. 11. P. 559–572. DOI: 10.1080/14786440109462720.
6. Kelly B., Malamud S., Pedersen L.H. Principal Portfolios. *The Journal of Finance*. 2023. Vol. 78, no. 1. P. 347–387. DOI: 10.1111/jofi.13199.
7. Mavungu M. Computation of financial risk using principal component analysis. *Algorithmic Finance*. 2023. Vol. 10, no. 1-2. P. 1–20. DOI: 10.3233/af-220339.
8. Kulikov A.V., Polozov D.S., Volkov N.V. Long-term investment optimization based on Markowitz diversification. *Business Informatics*. 2024. Vol. 18, no. 3. P. 56–69. DOI: 10.17323/2587-814X.2024.3.56.69.
9. Partovi M.H., Caputo M. Principal Portfolios: Recasting the Efficient Frontier. *Economics Bulletin*. 2004. Jan. Vol. 7. P. 1–10.
10. Markowitz H. Portfolio Selection. *The Journal of Finance*. 1952. Vol. 7, no. 1. P. 77–91. DOI: 10.2307/2975974.
11. Meucci A. Managing diversification. *Risk*. 2009. P. 74–79.
12. Lohre H., Neugebauer U., Zimmer C. Diversified Risk Parity Strategies for Equity Portfolio Selection. *The Journal of Investing*. 2012. Aug. Vol. 21. P. 111–128. DOI: 10.3905/joi.2012.21.3.111.
13. Lohre H., Opfer H., Ország G. Diversifying risk parity. *The Journal of Risk*. 2014. June. Vol. 16. P. 53–79. DOI: 10.21314/JOR.2014.284.
14. Luxenberg E., Boyd S. Portfolio construction with Gaussian mixture returns and exponential utility via convex optimization. *Optimization and Engineering*. 2024. Vol. 25, no. 1. P. 555–574. DOI: 10.1007/s11081-023-09814-y.
15. Indrė Morkūnaitė D.C., Leipus R. Evaluation of Value-at-Risk (VaR) using the Gaussian Mixture Models. *Research in Statistics*. 2024. Vol. 2, no. 1. P. 2346075. DOI: 10.1080/27684520.2024.2346075.
16. Ruan L., Yuan M., Zou H. Regularized Parameter Estimation in High-Dimensional Gaussian Mixture Models. *Neural Computation*. 2011. Vol. 23. P. 1605–1622. DOI: 10.1162/NECO_a_00128.
17. Halbe Z., Bortman M., Aladjem M. Regularized Mixture Density Estimation with an Analytical Setting of Shrinkage Intensities. *IEEE Transactions on Neural Networks and Learning Systems*. 2013. Vol. 24. P. 460–470. DOI: 10.1109/TNNLS.2012.2234477.
18. Moon T. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*. 1996. Vol. 13, no. 6. P. 47–60. DOI: 10.1109/79.543975.

19. Tipping M.E., Bishop C.M. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society Series B*. 1999. Vol. 61, no. 3. P. 611–622. DOI: 10.1111/1467-9868.00196.
20. Tipping M.E., Bishop C.M. Mixtures of Probabilistic Principal Component Analyzers. *Neural Computation*. 1999. Feb. Vol. 11, no. 2. P. 443–482. DOI: 10.1162/089976699300016728.
21. Lyu Y., Zhou L., Cong Y., *et al.* Multirate mixture probability principal component analysis for process monitoring in multimode processes. *IEEE Transactions on Automation Science and Engineering*. 2023. Vol. 21, no. 2. P. 2027–2038. DOI: 10.1109/TASE.2023.3253285.
22. Tra V., Amayri M., Bouguila N. Unsupervised fault detection for building air handling unit systems using deep variational mixture of principal component analyzers. *IEEE Transactions on Automation Science and Engineering*. 2024. Vol. 21, no. 4. P. 6787–6803. DOI: 10.1109/tase.2023.3331347.
23. Zhang J., Chen M., Hong X. Nonlinear process monitoring using a mixture of probabilistic PCA with clusterings. *Neurocomputing*. 2021. Vol. 458. P. 319–326. DOI: 10.1016/j.neucom.2021.06.039.
24. Zhao B., Xiao X., Zhang W., *et al.* Self-Paced Probabilistic Principal Component Analysis for Data with Outliers. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020. P. 3737–3741. DOI: 10.1109/icassp40776.2020.9054487.
25. Chao Han S.L., House L. Covariance-Guided Mixture Probabilistic Principal Component Analysis (C-MPPCA). *Journal of Computational and Graphical Statistics*. 2015. Vol. 24, no. 1. P. 66–83. DOI: 10.1080/10618600.2014.891460.
26. Tzagkarakis G., Caicedo-Llano J., Dionysopoulos T. Exploiting market integration for pure alpha investments via probabilistic principal factors analysis. *Journal of Mathematical Finance*. 2013. Vol. 3, no. 1. P. 192–200. DOI: 10.4236/jmf.2013.31A018.
27. Begušić S., Kostanjčar Z. Cluster-Specific Latent Factor Estimation in High-Dimensional Financial Time Series. *IEEE Access*. 2020. Vol. 8. P. 164365–164379. DOI: 10.1109/ACCESS.2020.3021898.
28. Akaike H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*. 1974. Dec. Vol. 19, no. 6. P. 716–723. DOI: 10.1109/TAC.1974.1100705.
29. Brutti Righi M., Sergio Ceretta P. On the existence of an optimal estimation window for risk measures. *Applied Finance Letters*. 2015. Nov. Vol. 4, no. 1-2. P. 28–32. DOI: 10.24135/af1.v4i1and2.30.
30. Volkov N.V. Estimating VaR of diversified portfolios using PCA and PPCA dimensionality reduction methods. *Proceedings of MIPT*. 2025. Vol. 17, no. 1. P. 127–141. (in Russian).
31. Kupiec P.H. Techniques for verifying the accuracy of risk measurement models. *Journal of Derivatives*. 1995. Vol. 3, no. 2. P. 73–84. DOI: 10.3905/jod.1995.407942.
32. Lam S.K., Pitrou A., Seibert S. Numba: A LLVM-based Python JIT Compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM 2015, Austin, TX, USA, November 15, 2015. ACM, 2015*. P. 1–6. DOI: 10.1145/2833157.2833162.

33. Ypma T.J. Historical development of the Newton–Raphson method. *SIAM Review*. 1995. Vol. 37, no. 4. P. 531–551. DOI: <https://doi.org/10.1137/1037125>.
34. Volkov N. MPPCA for VaR Estimation: Source Code Repository. 2025. URL: <https://gitlab.com/n.volkovsky/mppca-for-var-estimation> (accessed: 20.03.2025).
35. MacQueen J. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1, Statistics*. Vol. 5. University of California Press. 1967. P. 281–298.
36. Lloyd S. Least squares quantization in PCM. *IEEE Transactions on Information Theory*. 1982. Vol. 28, no. 2. P. 129–137. DOI: [10.1109/TIT.1982.1056489](https://doi.org/10.1109/TIT.1982.1056489).

ПОИСК ОПТИМАЛЬНЫХ ВЕСОВ ДЛЯ ФУНКЦИИ ЯДРА АКУШСКОГО

© 2025 В.В. Луценко, Д.Е. Горлачев, Н.М. Мирный, М.Г. Бабенко

Северо-Кавказский федеральный университет

(355017 Ставрополь, ул. Пушкина, д. 1)

E-mail: officialvladlutsenko@gmail.com, dimagorlacev12@gmail.com,

mirnyjn@gmail.com, mgbabenko@ncfu.ru

Поступила в редакцию: 07.03.2025

Современные вычислительные задачи, связанные с обработкой больших чисел, требуют не только высокой точности, но и значительной скорости. В этом контексте использование системы остаточных классов предлагает подход к параллельной обработке больших данных, применяемый в криптографии, обработке сигналов и искусственных нейронных сетях. Несмотря на преимущества системы остаточных классов, ее распространение замедленно в связи с вычислительной сложностью так называемых немодульных операций системы остаточных классов. Одним из универсальных инструментов для реализации немодульных операций является функция ядра Акушского. В работе исследуется функция ядра Акушского как инструмент для определения позиционной характеристики числа в системе остаточных классов. Для поиска оптимальных весов функции ядра предложено применение метода Монте-Карло и генетического алгоритма. Экспериментальные результаты демонстрируют, что генетический алгоритм обеспечивает более стабильные результаты при увеличении количества модулей, в то время как метод Монте-Карло эффективен на малых размерностях. Генетический алгоритм в среднем на 38% быстрее метода Монте-Карло, что делает его предпочтительным выбором. Дополнительно проведено сравнение времени вычисления функции ядра с оптимальными весами и функции Пирло. Результаты показали, что функция ядра с оптимальными весами в среднем на 14% быстрее функции Пирло.

Ключевые слова: система остаточных классов, высокопроизводительные вычисления, функция ядра Акушского, метод Монте-Карло, генетический алгоритм.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Луценко В.В., Горлачев Д.Е., Мирный Н.М., Бабенко М.Г. Поиск оптимальных весов для функции ядра Акушского // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2025. Т. 14, № 2. С. 26–41. DOI: 10.14529/cmse250202.

Введение

Система остаточных классов (СОК) представляет собой математический аппарат, который находит широкое применение в различных областях, таких как криптография [1], обработка сигналов [2], вычислительная техника [3] и теория чисел [4]. Основная идея СОК заключается в представлении чисел в виде набора остатков от деления на взаимно простые модули, что позволяет упростить арифметические операции и повысить эффективность вычислений. В отличие от традиционных позиционных систем счисления, СОК предлагает параллельный подход к обработке данных, что особенно полезно в задачах, требующих высокой скорости и точности.

Исторически система остаточных классов восходит к Китайской теореме об остатках (КТО). В последние десятилетия интерес к СОК возрос благодаря ее применению в квантовых вычислениях [5], нейронных сетях [6, 7].

Несмотря на преимущества СОК у нее существуют недостатки в виде так называемых немодульных операций. К данным операциям относятся: обратное преобразование числа,

Таблица 1. Представление чисел для СОК с базисом $\{3, 4\}$

$0 \xrightarrow{\text{СОК}} (0, 0)$	$1 \xrightarrow{\text{СОК}} (1, 1)$	$2 \xrightarrow{\text{СОК}} (2, 2)$	$3 \xrightarrow{\text{СОК}} (0, 3)$
$4 \xrightarrow{\text{СОК}} (1, 0)$	$5 \xrightarrow{\text{СОК}} (2, 1)$	$6 \xrightarrow{\text{СОК}} (0, 2)$	$7 \xrightarrow{\text{СОК}} (1, 3)$
$8 \xrightarrow{\text{СОК}} (2, 0)$	$9 \xrightarrow{\text{СОК}} (0, 1)$	$10 \xrightarrow{\text{СОК}} (1, 2)$	$11 \xrightarrow{\text{СОК}} (2, 3)$

расширение базиса, сравнение чисел, определение знака числа, обнаружение переполнения, общее деление, масштабирование и другие. Для выполнения немодульных операций необходимо знать так называемую позиционную характеристику (ПХ) числа в СОК. Универсальным инструментом для определения ПХ является так называемая функция ядра Акушского [8]. Функция ядра Акушского может эффективно применяться для обратного преобразования в позиционную систему счисления [9], определения знака числа [10], определения четности числа [11], общего деления в СОК [12].

Однако несмотря на все исследования, как и все остальные методы поиска ПХ, вычисление значения функции ядра остается вычислительно сложной задачей. С целью снизить вычислительную сложность нахождения значения функции ядра мы предлагаем использовать оптимальные веса для функции ядра поиск которых осуществлять с использованием метода Монте-Карло или генетического алгоритма.

Статья имеет следующую структуру. В разделе 1 рассматривается система остаточных классов. В разделе 2 представлены основы функции ядра Акушского. В разделе 3 представлены алгоритмы поиска оптимальных весов для функции ядра Акушского. Наконец, в разделе 4 проведен анализ эффективности предложенных методов. В заключении суммируются полученные результаты.

1. Основы системы остаточных классов

Определение 1. Базисом системы остаточных классов называется множество модулей $\{p_1, p_2, \dots, p_n\}$, где каждый модуль $p_i \geq 2$ ($i = 1, 2, \dots, n$) и $\gcd(p_i, p_j) = 1$. Здесь и далее по тексту \gcd обозначает наибольший общий делитель (англ. greatest common divisor). По умолчанию модули упорядочены по возрастанию, то есть $p_1 < p_2 < \dots < p_n$.

Определение 2. Произведение модулей $P = \prod_{i=1}^n p_i$ называется динамическим диапазоном системы остаточных классов.

Определение 3. Целое число $X \in [0, P)$ представимо в виде n -мерного вектора, составленного из наименьших неотрицательных остатков от деления соответствующего числа на p_i :

$$X = (x_1, x_2, \dots, x_n),$$

где $x_i \equiv X \pmod{p_i}$, что также обозначается как $x_i = |X|_{p_i}$.

В табл. 1 представлены соответствия чисел из позиционной системы счисления и системы остаточных классов для базиса $\{3, 4\}$.

Система остаточных классов обладает особенностью выполнять операции сложения, вычитания и умножения параллельно и независимо для каждого модуля. Для чисел $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_n)$ выразим арифметику следующим образом:

$$X * Y = (x_1 * y_1, x_2 * y_2, \dots, x_n * y_n), \text{ где } * \in \{+, -, \times\}. \quad (1)$$

В примере 1 представлено сложение двух чисел в СОК.

Пример 1. Сложим два числа $X = 5$ и $Y = 4$ в базисе $\{3, 4\}$. Их представление в заданном базисе указано в табл. 1. Воспользуемся уравнением (1) для сложения:

$$X + Y = (|2 + 1|_3, |1 + 0|_4) = (0, 1).$$

Отсутствие переносов при выполнении модульных операций — одно из ключевых преимуществ арифметики в СОК. Это означает, что результаты операций сложения и умножения остаются в пределах заданного диапазона значений для каждого модуля. Это значительно упрощает и ускоряет процесс выполнения операций сложения и умножения по сравнению с позиционными системами счисления.

Преимущество арифметики с остатками особенно ярко проявляется при выполнении вычислительных процедур, состоящих из последовательности модулярных арифметических операций. При этом можно выбрать набор модулей таким образом, чтобы конечные результаты всегда находились в допустимом диапазоне значений для любых входных операндов. Промежуточные результаты могут выходить за границы диапазона, но это не представляет проблемы.

Несмотря на преимущества системы остаточных классов в некоторых аспектах, она имеет ряд существенных ограничений в других. Например, для чисел $(0, 3)$ и $(0, 2)$ из табл. 1, невозможно сразу определить, какое из них больше, что требует использования специальных методов для вычисления позиционной характеристики.

Для того чтобы восстановить позиционное представление числа в СОК, можно воспользоваться Китайской теоремой об остатках [13].

Теорема 1. Пусть $\{p_1, p_2, \dots, p_n\}$ есть некоторые натуральные взаимно простые числа и $P = \prod_{i=1}^n p_i$. Любое число X , такое что $0 \leq X < P$, может быть однозначно представлено в виде последовательности (x_1, x_2, \dots, x_n) , где $x_i = X \bmod p_i$, при этом

$$X = \left| \sum_{i=1}^n P_i \cdot x_i \cdot |P_i^{-1}|_{p_i} \right|_P, \quad (2)$$

где $P_i = \frac{P}{p_i}$, $|P_i^{-1}|_{p_i}$ мультипликативная инверсия P_i по модулю p_i .

Определение 4. Значения $B_i = |P_i^{-1}|_{p_i} \cdot P_i$ называются ортогональными базисами системы остаточных классов.

Пример 2. Возьмем СОК с набором модулей $\{3, 4\}$. Рассмотрим процесс перевода числа $X = (0, 2)$ в позиционную систему счисления.

Найдем значения P_i и $|P_i^{-1}|_{p_i}$:

$$P_1 = \frac{P}{p_1} = 4, \quad P_2 = \frac{P}{p_2} = 3.$$

$$|P_1^{-1}|_{p_1} = 1, \quad |P_2^{-1}|_{p_2} = 3.$$

По формуле 2 найдем X :

$$X = |4 \cdot 1 \cdot 0 + 3 \cdot 3 \cdot 2|_{12} = 6.$$

Таким образом, получено число в позиционной системе.

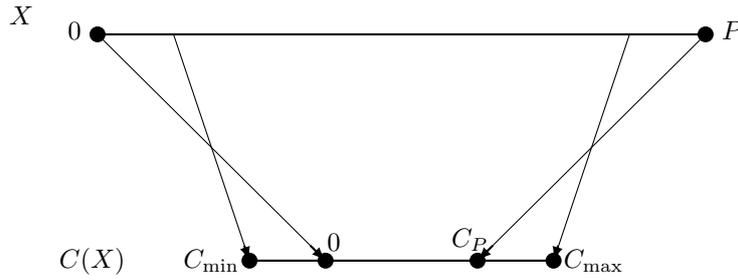


Рис. 1. Отображение $0 \leq X < P$ на $C_{min} \leq C(X) < C_{max}$

Немодульные операции, такие как обратное преобразование числа, общее деление, сравнение чисел, определение знака числа являются вычислительно сложными и требуют большого количества вычислений, что ограничивает применение СОК и ее использование для быстрых вычислений.

2. Функция ядра Акушского

Исследования, проведенные Акушским И.Я., Бурцевым В.М. и Паком И.Т. [14], были направлены на нахождение позиционной характеристики числа в СОК. В результате этих исследований была предложена новая математическая конструкция, получившая название функции ядра Акушского. Функция ядра Акушского определяется следующей формулой:

$$C(X) = \sum_{i=1}^n w_i \cdot \left\lfloor \frac{X}{p_i} \right\rfloor, \quad (3)$$

где целые числа w_i выступают в роли постоянных величин, которые определяются выбором точки для интерполяции. Константы w_i задают вес каждого из частных $\left\lfloor \frac{X}{p_i} \right\rfloor$ в формуле (3), тем самым задавая функцию ядра и придавая ей различные свойства.

Подставив $X = P$ в (3), получим:

$$C(P) = C_P = \sum_{i=1}^n w_i \cdot \left\lfloor \frac{P}{p_i} \right\rfloor = \sum_{i=1}^n w_i \cdot P_i. \quad (4)$$

Разделив $C(P)$ на P , получим:

$$\frac{C(P)}{P} = \sum_{i=1}^n \frac{w_i}{p_i}. \quad (5)$$

Отсюда следует, что некоторые значения w_i должны быть отрицательными, чтобы получить малые значения $C(P)$. Подставив (5) в (3), получим:

$$C(X) = X \cdot \frac{C(P)}{P} - \sum_{i=1}^n \frac{x_i \cdot w_i}{p_i} = X \cdot \sum_{i=1}^n \frac{w_i}{p_i} - \sum_{i=1}^n \frac{x_i \cdot w_i}{p_i}. \quad (6)$$

Используя функцию ядра можно получить информацию о позиционной характеристики числа как показано на рис. 1.

Уравнение (6) указывает, что график $C(X)$ относительно X должен быть прямой линией с наклоном $C(P)/P$ с некоторой нелинейностью. Величина нелинейности определяется

величиной весов, в свою очередь связанной с конкретным значением $C(P)$ для заданного набора модулей СОК.

Учитывая, что $P_j \equiv 0 \pmod{p_i}$ для любого $i \neq j$, веса w_i функции $C(X)$ могут быть определены соотношением:

$$w_i \equiv |C(P) \cdot P_i^{-1}|_{p_i}. \quad (7)$$

Таким образом, веса могут быть определены после выбора $C(P)$, но с условием выполнения (4).

Пример 3. Рассмотрим систему с модулями $p_1 = 3, p_2 = 4$ и весами $w_1 = 1, w_2 = 2$. Динамический диапазон $P = 12$. Функция ядра от динамического диапазона равна $C(P) = 10$. Для данных параметров график функции ядра изображена на рис. 2.

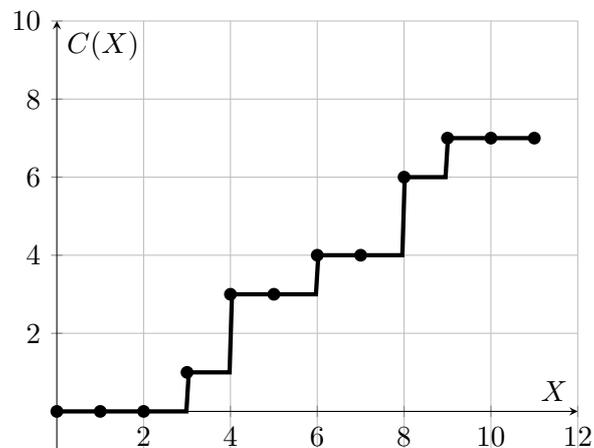


Рис. 2. График функции ядра с $w_1 = 1, w_2 = 2$ для СОК с основаниями $p_1 = 3, p_2 = 4$

Отметим, что (3) плохо подходит для вычисления значений функции ядра в практических приложениях, так как необходимо знать позиционное представление X . Применяя КТО, возможно модифицировать подход к расчету значения функции ядра для X , основываясь на ее остатках.

Значение функции ядра $C(X)$ с весами w_1, w_2, \dots, w_n , при условии $0 \leq C(X) < C_P$, $X \in [0, P)$, можно вычислить с использованием формулы

$$C(X) \equiv \left| \sum_{i=1}^n k_i \cdot x_i \right|_{C_P}, \quad (8)$$

где $k_i = C(B_i)$.

Однако в общем случае при определенных w_i , $C(X)$ может не удовлетворять условию $0 \leq C(X) < C_P$ для $X \in [0, P)$, тогда возникает проблема так называемых критических ядер, которая ограничивает возможность использовать формулу (8). В данном случае функции ядра будут располагаться в некотором промежутке $[-\delta_1, C_P + \delta_2]$ (см. рис. 1).

В статье [8] рассмотрены эффективные методы, которые позволяют определить критические ядра.

В данной статье мы сосредоточимся на поиске оптимальных весов для функции ядра Акушского. Для этого введем следующее определение.

Определение 5. Будем называть функцию ядра Акушского дружественной для программной и аппаратной реализации, если $C(P) = C_P = 2^N$, где $C(P) = \sum_i^n w_i P_i$.

Теорема 2. Для множество модулей $\{p_1, p_2, \dots, p_n\}$ где $p_i \geq 2$ и $\gcd(p_i, p_j) = 1, \forall i \neq j$, существует набор неотрицательных весов $\{w_1, w_2, \dots, w_n\}$, таких, что

$$C(P) = \sum_{i=1}^n w_i \cdot P_i = 2^N.$$

Доказательство. Докажем теорему методом индукции.

1. База индукции ($n = 1$). Если $n = 1$, то у нас есть только один модуль $p_1 \geq 2$, тогда $P = p_1$ и $P_1 = \frac{P}{p_1} = 1$.

Тогда существует единственный вес $w_1 = 2^N$, который удовлетворяет условию $C(P) = 2^a$.

2. Индукционное предположение ($n = k$). Предположим, что теорема верна для $n = k$, то есть для любого множества модулей $\{p_1, p_2, \dots, p_k\}$, тогда с учетом свойств КТО существуют такие неотрицательные веса $\{w_1, w_2, \dots, w_k\}$, что

$$C(D_k) = \sum_{i=1}^k w_i \cdot P_i = 2^N,$$

где $D_k = \prod_{i=1}^k p_i$.

3. Шаг индукции ($n = k + 1$). Рассмотрим множество из $k + 1$ попарно взаимно простых модулей $\{p_1, p_2, \dots, p_{k+1}\}$. Обозначим $D_{k+1} = D_k \cdot p_{k+1}$. Согласно предположению индукции, для множества модулей $\{p_1, p_2, \dots, p_k\}$ существует набор неотрицательных весов $\{w_1, w_2, \dots, w_k\}$, таких что $C(D_k) = 2^N$.

Теперь добавим новый модуль p_{k+1} и определим соответствующий вес w_{k+1} , такой что сумма

$$C(D_{k+1}) = C(D_k) + w_{k+1} \cdot P_{k+1} = 2^N.$$

Заметим, что данное условие выполняется, например при $w_{k+1} = 0$. □

Приведем пример 4 работы с функцией ядра, когда она является дружественной.

Пример 4. Пусть дана система модулей $p_1 = 3, p_2 = 4, p_3 = 5$. Объем динамического диапазона $P = 3 \cdot 4 \cdot 5 = 60$. Покажем, что веса $w_1 = 1, w_2 = 0, w_3 = 1$ являются оптимальными для данной системы.

Найдем значения P_i :

$$P_1 = \frac{P}{p_1} = 20, P_2 = \frac{P}{p_2} = 15, P_3 = \frac{P}{p_3} = 12.$$

Далее по формуле (4) найдем диапазон функции ядра:

$$C(P) = 1 \cdot 20 + 0 \cdot 15 + 1 \cdot 12 = 32 = 2^5.$$

Значения ортогональных базисов B_i равны:

$$B_1 = 40, B_2 = 45, B_3 = 36.$$

Далее найдем коэффициенты $k_i = C(B_i)$:

$$k_1 = 21, k_2 = 24, k_3 = 19.$$

Найдем значение $C(X)$ для числа $(2, 3, 1)$:

$$C(X) = |21 \cdot 2 + 24 \cdot 3 + 19 \cdot 1|_{2^5} = |133|_{2^5} = |[10000101]_2|_{2^5} = [101]_2 = 5.$$

Таким образом, вычислительно сложную операцию взятия по модулю можно заменить срезом $N = 5$ старших битов.

В следующем разделе рассмотрим методы поиска оптимальных весов функции ядра, при использовании которых будет выполняться условие $C_P = 2^N$.

3. Методы подбора оптимальных весов для функции ядра

3.1. Метод Монте-Карло

Метод Монте-Карло — это группа численных методов, основанных на случайном моделировании (стохастическом подходе), которые применяются для решения различных математических, статистических и других задач [15]. Метод был разработан в середине двадцатого века, главным образом в контексте исследований, связанных с ядерной физикой, но быстро нашел применение во множестве других дисциплин. Метод основан на использовании случайных или псевдослучайных чисел. Они применяются для имитации поведения сложных систем или случайных процессов. Благодаря своей случайности, метод позволяет быстро находить решения в больших диапазонах. В работе [16] были описаны основные преимущества данного метода и подробное разъяснение в различных сферах его использования.

В статье [17] было предложено использование метода Монте-Карло для поиска весов функции ядра, которые увеличивают точность вычисления приближенного ранга числа в СОК. Данный метод может быть интегрирован в системе остаточных классов, для поиска оптимальных весов функции ядра Акушского, удовлетворяющих условию $C_P = 2^N$. Ниже представлен алгоритм 1, который реализует данный метод.

Метод Монте-Карло эффективен для небольших базисов, однако с увеличением количества модулей вероятность нахождения оптимальных весов падает, из-за этого время выполнения алгоритма может быть значительным. С целью ускорения подбора оптимальных весов предложено использовать генетический алгоритм.

3.2. Генетический алгоритм

Генетический алгоритм — это метод оптимизации и поиска, основанный на принципах естественного отбора и эволюции в биологии [18]. Генетические алгоритмы часто рассматриваются как оптимизаторы функций, хотя спектр задач, к которым применяются генетические алгоритмы, довольно широк. Метод является разновидностью эволюционных вычислений, таких как наследование, мутации, отбор и кроссинговер. Из работы [19] можно выделить следующую структуру:

Алгоритм 1: Метод Монте-Карло для поиска оптимальных весов функции ядра

Input: $\{p_1, p_2, \dots, p_n\}$,
 $max_iterations$,
 $weigh_limit$
Data: $P = \prod_{i=1}^n p_i$,
 $P_i = \frac{P}{p_i}, i = 1, 2, \dots, n$
Output: $\{w_1, w_2, \dots, w_i\}, N$ — если найдены оптимальные веса

```

1 for  $i := 1$  to  $max\_iterations$  do
2    $w_i = random(0, weigh\_limit)$ 
3    $C_P = \sum_i w_i \cdot P_i$ 
4   if  $C_P > 0$  and  $(C_P \wedge (C_P - 1)) = 0$  then
5      $N = \log_2 C_P$ 
6     return  $w_i, N$ 

```

- Ген — это структура данных, представляющая одно возможное решение задачи. Множество генов — популяция.
- Функция приспособленности оценивает качество генов. Она определяет, насколько решение соответствует заданным критериям.
- Селекция (selection) — выбор генов (родителей) для создания следующего поколения. Обычно выбираются гены с высоким значением функции приспособленности.
- Скрещивание (crossover) — объединение генетической информации двух родителей для создания одного или нескольких потомков.
- Мутация (mutation) — внесение случайных изменений в потомков для поддержания разнообразия и предотвращения застревания в локальных максимумах.

Такой алгоритм можно применить для поиска оптимальных весов. Для начала нужно будет задать начальную популяцию, путем генерации случайных весов. Оценивать приспособленность будем через степень двойки функции C_P . Чем меньше значение функции приспособленности, тем лучше данное решение.

Далее выбираются лучшие веса из текущей популяции, происходит этап селекции. Отбирается верхняя половина популяции — веса с наименьшими значениями функции приспособленности. Эти лучшие решения используются для создания потомков.

Создаются новые «потомки» путем рекомбинации генов (весов) двух родителей, выбирается точка скрещивания. Для поддержания генетического разнообразия потомки подвергаются мутации с определенной вероятностью и затем, новая популяция формируется из лучших индивидов текущего поколения и потомков, тем самым получая оптимальные веса. Ниже представлен псевдокод данного алгоритма.

4. Оценка производительности

Моделирование и вычислительные эксперименты проведены на компьютере, оснащенном процессором Intel Core i7-7700HQ с тактовой частотой 2.80 ГГц, 8 ГБ оперативной памяти DDR4 с частотой 1196 МГц и твердотельным накопителем емкостью 512 ГБ, работающий под управлением Windows 10 Home Edition, с использованием языка программирования высокого уровня Python.

Алгоритм 2: Генетический алгоритм для поиска оптимальных весов функции ядра

Input: $\{p_1, p_2, \dots, p_n\}$,
population_size,
weight_limit
Data: $P = \prod_{i=1}^n p_i$,
 $P_i = \frac{P}{p_i}, i = 1, 2, \dots, n$
Output: $\{w_1, w_2, \dots, w_i\}, N$ — если найдены оптимальные веса

- 1 $w_i = \text{random}(0, \text{weight_limit})$
 // Оценка приспособленности:
- 2 $C_P = \sum_i^n w_i \cdot P_i$
- 3 **if** $C_P > 0$ **and** $(C_P \wedge (C_P - 1)) = 0$ **then**
- 4 $N = \log_2 C_P$
- 5 **return** w_i, N
 // Селекция:
- 6 $n = \text{population_size}$
- 7 $f_{w_i} = \frac{C(P)}{\sum_{i=1}^n C(P)}$, где f_{w_i} — вероятность выбора w_i веса
- 8 $\text{crossover_point} = \text{random}(0, \text{len}(p_1, p_2, \dots, p_n))$
- 9 $\text{new_population} := \text{new_population} + w_i$
 // Мутация:
- 10 **for** $i := 1$ **to** new_population **do**
- 11 Замена случайного веса w_i в диапазоне weight_limit
- 12 $\text{population} := \text{population} + \text{new_population}$
- 13 Возврат на оценку приспособленности

Для оценки эффективности предложенных алгоритмов проведен замер времени нахождения оптимальных весов для наборов модулей общего вида. Оптимальные веса для используемых в моделировании наборов модулей представлены в табл. 2.

Таблица 2. Веса, найденные алгоритмом 1 и 2

Набор модулей	Оптимальные веса	
	Метод Монте-Карло	Генетический алгоритм
{23, 29, 31}	{4, 17, 1}	{4, 17, 1}
{23, 29, 31, 37}	{7, 40, 11, 26}	{8, 3, 40, 37}
{23, 29, 31, 37, 41}	{7, 20, 27, 46, 48}	{14, 49, 22, 38, 10}
{23, 29, 31, 37, 41, 43}	{2, 25, 10, 11, 30, 38}	{1, 6, 27, 33, 27, 22}
{23, 29, 31, 37, 41, 43, 47}	{13, 18, 40, 26, 33, 6, 10}	{18, 6, 1, 12, 6, 19, 11}
{23, 29, 31, 37, 41, 43, 47, 53}	{15, 10, 36, 38, 23, 2, 43, 28}	{18, 29, 3, 48, 23, 18, 41, 11}

В табл. 3 представлено время работы алгоритмов.

Для наглядности на рис. 3 отображены результаты времени выполнения алгоритмов из табл. 3.

Таблица 3. Время нахождения оптимальных весов, с

Набор модулей	Метод Монте-Карло	Генетический алгоритм
{23, 29, 31}	0.014	0.006
{23, 29, 31, 37}	2	12
{23, 29, 31, 37, 41}	47	24
{23, 29, 31, 37, 41, 43}	125	31
{23, 29, 31, 37, 41, 43, 47}	189	42
{23, 29, 31, 37, 41, 43, 47, 53}	278	48

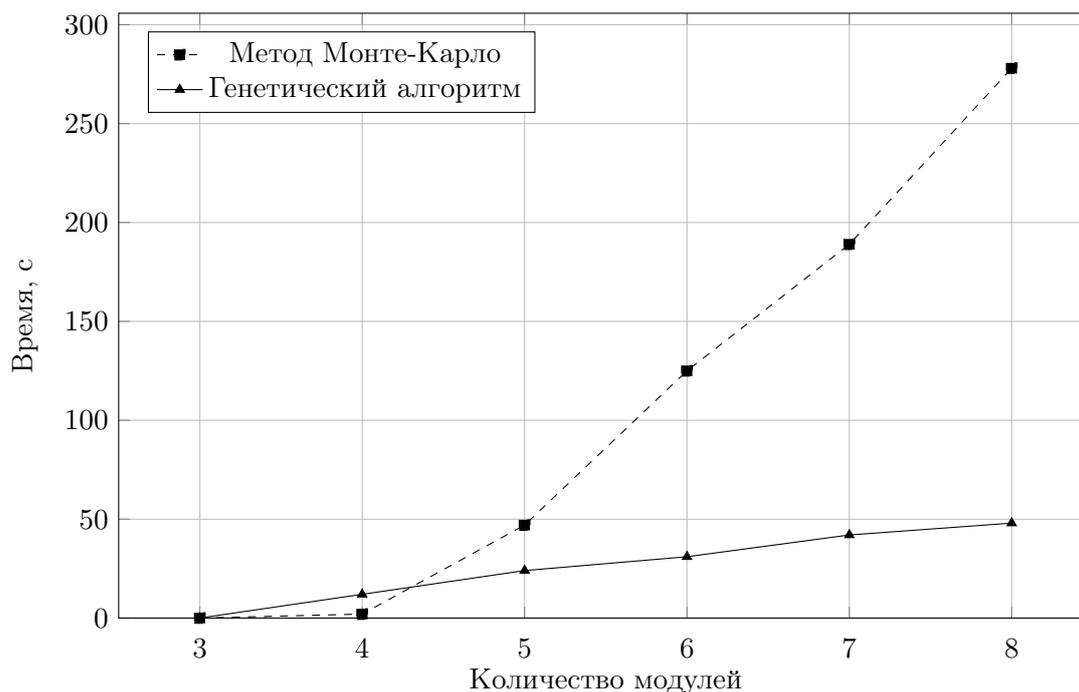


Рис. 3. График зависимости времени выполнения от количества модулей для методов подбора оптимальных весов

С увеличением размера набора модулей наблюдается значительное увеличение времени выполнения для обоих методов. Начиная с набора {23, 29, 31, 37} и далее, генетический алгоритм демонстрирует значительно более высокую производительность по сравнению с методом Монте-Карло. Например, для набора из восьми модулей {23, 29, 31, 37, 41, 43, 47, 53} генетический алгоритм работает примерно в 5.8 раз быстрее.

В среднем, генетический алгоритм быстрее метода Монте-Карло на 38%. Данные результаты связаны с тем, что метод Монте-Карло основан на случайной выборке и статистическом усреднении, он требует большого числа итераций для достижения точности, особенно в задачах высокой размерности [16]. Генетический алгоритм не просто случайно блуждает по пространству решений, а направленно улучшает популяцию решений через селекцию, скрещивание и мутацию. Для задач с большим числом модулей целесообразнее использовать генетический алгоритм, который показывает лучшее время выполнения, несмотря на усложнение задачи.

Для доказательства эффективности работы функции с оптимальными весами как позиционной характеристики, проведем следующее моделирование. Сравним время вычисления дружественных функций с функцией Пирло [20]. Функция Пирло имеет следующий вид

$$Pi(X) = \left| \sum_{i=1}^n k_i^{**} \cdot x_i \right|_{P_n}, \quad (9)$$

где $k_i^{**} = \left\lfloor \frac{|P_i^{-1}|_{p_i} P_i}{p_n} \right\rfloor$.

Рассмотрим пример вычисления функции Пирло.

Пример 5. Для числа $X = (2, 2, 3)$, представленное в СОК с модулями $\{3, 5, 7\}$ найдем позиционную характеристику с использованием функции Пирло. Для начала вычислим необходимые константы:

$$\begin{aligned} Pi(P) &= P_n = P_3 = 15, \\ k_1^{**} &= \left\lfloor \frac{|P_1^{-1}|_{p_1} P_1}{p_3} \right\rfloor = \left\lfloor \frac{2 \cdot 35}{7} \right\rfloor = 10, \\ k_2^{**} &= \left\lfloor \frac{|P_2^{-1}|_{p_2} P_2}{p_3} \right\rfloor = \left\lfloor \frac{1 \cdot 21}{7} \right\rfloor = 3, \\ k_3^{**} &= \left\lfloor \frac{|P_3^{-1}|_{p_3} P_3}{p_3} \right\rfloor = \left\lfloor \frac{1 \cdot 15}{7} \right\rfloor = 2. \end{aligned}$$

Найдем значение функции Пирло

$$Pi(X) = |2 \cdot 10 + 2 \cdot 3 + 3 \cdot 2|_{15} = 2.$$

Отметим, что функция Пирло является частным случаем функции ядра Акушского с весами $\{0, 0, \dots, w_{n-1} = 0, w_n = 1\}$.

Были использованы оптимальные веса, найденные генетическим методом из табл. 2. Для чистоты эксперимента, замер времени проводился только для нахождения $C(X)$ и $Pi(X)$. Остальные данные, необходимые для расчетов вычислены заранее. Позиционные характеристики вычислялись для максимального числа из диапазона $[0, P)$. Результаты замеров времени показаны в табл. 4.

Таблица 4. Время нахождения позиционных характеристик, мс

Набор модулей	Функция Пирло	Функция ядра
{23, 29, 31}	23	19
{23, 29, 31, 37}	25	22
{23, 29, 31, 37, 41}	26	23
{23, 29, 31, 37, 41, 43}	27	25
{23, 29, 31, 37, 41, 43, 47}	36	31
{23, 29, 31, 37, 41, 43, 47, 53}	39	35

В среднем, нахождение функции ядра Акушского с использованием оптимальных весов быстрее расчета функции Пирло на 14%.

Заключение

В работе рассмотрена функция ядра Акушского как универсальный инструмент для определения позиционной характеристики числа в системе остаточных классов.

С целью поиска оптимальных весов функции ядра Акушского в статье предложено использование метода Монте-Карло и генетического алгоритма. Экспериментальные результаты показали, что генетический алгоритм обеспечивает более стабильные результаты, при увеличении количества модулей, в то время как метод Монте-Карло быстрее на малых размерностях, но менее эффективен при усложнении задачи. В среднем генетический алгоритм быстрее метода Монте-Карло на 38%, что делает его предпочтительным для выбора.

Дополнительно был проведен эксперимент, в котором сравнивалось вычисление позиционной характеристики числа в СОК на основе функции ядра с оптимальными весами и функцией Пирло. Эксперимент показал, что вычисление функции ядра с оптимальными весами в среднем на 14% быстрее чем вычислении функции Пирло. Дальнейшие исследования будут направлены на использование функции ядра Акушского в криптографических алгоритмах.

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00149, <https://rscf.ru/project/24-21-00149/>.

Литература

1. Schoiniakakis D. Residue arithmetic systems in cryptography: a survey on modern security applications // Journal of Cryptographic Engineering. 2020. Vol. 10, no. 3. P. 249–267. DOI: 10.1007/s13389-020-00231-w.
2. Cardarilli G.C., Nunzio L.D., Fazzolari R. An RNS-based initial absolute position estimator for Electrical Encoders // IEEE Access. 2023. Vol. 11. P. 98586–98595. DOI: 10.1109/access.2023.3312619.
3. Mohan P.V.A. Residue number systems: algorithms and architectures. Springer Science & Business Media, 2002.
4. Omondi A.R., Premkumar A.B. Residue number systems: theory and implementation. World Scientific, 2007. Vol. 2.
5. Bajard J.C., Eynard J. RNS Approach in Lattice-Based Cryptography // Embedded Systems Design with Special Arithmetic and Number Systems. 2017. P. 345–368. DOI: 10.1007/978-3-319-49742-6_13.
6. Valueva M. V., Nagornov N.N., Lyakhov P.A., *et al.* Application of the residue number system to reduce hardware costs of the convolutional neural network implementation // Mathematics and Computers in Simulation. 2020. Vol. 177. P. 232–243. DOI: 10.1016/j.matcom.2020.04.031.
7. Roohi A., Taheri M.R., Angizi S., *et al.* Rnsim: Efficient deep neural network accelerator using residue number systems // 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD). IEEE, 2021. P. 1–9. DOI: 10.1109/iccad51958.2021.9643531.
8. Lutsenko V., Babenko M., Deryabin M. Construction of Akushsky Core Functions Without Critical Cores // Mathematics. 2024. Vol. 12, no. 21. P. 3399. DOI: 10.3390/math12213399.

9. Lutsenko V.V., Babenko M.G., Khamidov M.M. High speed method of conversion numbers from residue number system to positional notation // Proceedings of the Institute for System Programming of the RAS. 2024. Vol. 36, no. 4. P. 117–132. DOI: 10.15514/ispras-2024-36(4)-9.
10. Shiriaev E., Kucherov N., Babenko M., *et al.* Fast operation of determining the sign of a number in RNS using the Akushsky core function // Computation. 2023. Vol. 11, no. 7. P. 124. DOI: 10.3390/computation11070124.
11. Lutsenko V., Geryugova A., Babenko M., *et al.* High-Speed Parity Number Detection Algorithm in RNS Based on Akushsky Core Function // International Conference on Communication and Computational Technologies / eds. by S. Kumar, S. Hiranwal, R. Garg, S.D. Purohit. Cham: Springer, Singapore, 2024. P. 491–504. DOI: 10.1007/978-981-97-7423-4_38.
12. Луценко В.В., Бабенко М.Г., Черных А.Н., Лапина М.А. Оптимизация алгоритма деления чисел в системе остаточных классов на основе функции ядра Акушского. Труды ИСП РАН. 2023. Т. 35, № 5. С. 157–168. DOI: 10.15514/ispras-2022-35(5)-11.
13. Wang Y. Residue-to-binary converters based on new Chinese remainder theorems // IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 2000. Vol. 47, no. 3. P. 197–205. DOI: 10.1109/82.826745.
14. Акушский И.Я., Бурцев В.М., Пак И.Т. О новой позиционной характеристике непозиционного кода и ее приложения // Теория кодирования и оптимизация сложных систем. Алма-Ата, Наука, КазССР. 1977. С. 8–16.
15. Metropolis N., Ulam S. The Monte Carlo method // Journal of the American Statistical Association. 1949. Vol. 44, no. 247. P. 335–341.
16. Kroese D.P., Brereton T., Taimre T., *et al.* Why the Monte Carlo method is so important today // Wiley Interdisciplinary Reviews: Computational Statistics. 2014. Vol. 6, no. 6. P. 386–392. DOI: 10.1002/wics.1314.
17. Shiriaev E., Kucherov N., M Babenko M., *et al.* Algorithm for determining the optimal weights for the Akushsky core function with an approximate rank // Applied Sciences. 2023. Vol. 13, no. 18. P. 10495. DOI: 10.3390/app131810495.
18. Курейчик В.М. Генетические алгоритмы // Известия Южного федерального университета. Технические науки. 1998. Т. 8, № 2. С. 4–7.
19. Mirjalili S. Evolutionary algorithms and neural networks // Studies in Computational Intelligence. 2019. Vol. 780, no. 1. P. 43–53. DOI: 10.1007/978-3-319-93025-1.
20. Pirlo G., Impedovo D. A new class of monotone functions of the residue number system // International Journal of Mathematical Models and Methods in Applied Sciences. 2013. Vol. 7, no. 9. P. 802–809.

Луценко Владислав Вячеславович, инженер-исследователь, департамент науки, Северо-Кавказский федеральный университет (Ставрополь, Российская Федерация)

Горлачев Дмитрий Евгеньевич, студент, кафедра вычислительной математики и кибернетики, Северо-Кавказский федеральный университет (Ставрополь, Российская Федерация)

Мирный Никита Михайлович, студент, кафедра вычислительной математики и кибернетики, Северо-Кавказский федеральный университет (Ставрополь, Российская Федерация)

Бабенко Михаил Григорьевич, д.ф.-м.н., заведующий кафедрой вычислительной математики и кибернетики факультета математики и компьютерных наук имени профессора Н.И. Червякова, Северо-Кавказский федеральный университет (Ставрополь, Российская Федерация)

DOI: 10.14529/cmse250202

SEARCHING OF OPTIMAL WEIGHTS FOR THE AKUSHSKY CORE FUNCTION

© 2025 V.V. Lutsenko, D.E. Gorlacev, N.M. Mirny, M.G. Babenko

North-Caucasus Federal University

(Pushkin st. 1, Stavropol, 355017 Russia)

E-mail: officialvladlutsenko@gmail.com, dimagorlacev12@gmail.com,

mirnyjn@gmail.com, mgbabenko@ncfu.ru

Received: 07.03.2025

Modern computational tasks involving the processing of large numbers require not only high accuracy but also significant speed. In this context, the use of the residue number system offers an approach to parallel big data processing used in cryptography, signal processing and artificial neural networks. Despite the advantages of the residue number system, its diffusion has been slow due to the computational complexity of the so-called non-modular operations of the residual class system. One of the universal tools for realising non-modular operations is the Akushsky core function. This paper studies the Akushsky core function as a tool for determining the positional characteristic of a number in the residue number system. The application of Monte Carlo method and genetic algorithm is proposed to find the optimal weights of the core function. Experimental results demonstrate that the genetic algorithm provides more stable results when the number of moduli increases, while the Monte Carlo method is effective on small dimensions. The genetic algorithm is on average 38% faster than the Monte Carlo method, making it the preferred choice. Additionally, the computation time of the core function with optimal weights and the Pirlo function were compared. The results showed that the core function with optimal weights is on average 14% faster than the Pirlo function.

Keywords: residue number system, high-performance computing, Akushsky core function, Monte Carlo method, genetic algorithm.

FOR CITATION

Lutsenko V.V., Gorlacev D.E., Mirny N.M., Babenko M.G. Searching of Optimal Weights for the Akushsky Core Function. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2025. Vol. 14, no. 2. P. 26–41. (in Russian) DOI: 10.14529/cmse250202.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Schoinianakis D. Residue arithmetic systems in cryptography: a survey on modern security applications. Journal of Cryptographic Engineering. 2020. Vol. 10, no. 3. P. 249–267.

DOI: 10.1007/s13389-020-00231-w.

2. Cardarilli G.C., Nunzio L.D., Fazzolari R. An RNS-based initial absolute position estimator for Electrical Encoders. *IEEE Access*. 2023. Vol. 11. P. 98586–98595. DOI: 10.1109/access.2023.3312619.
3. Mohan P.V.A. *Residue number systems: algorithms and architectures*. Springer Science & Business Media, 2002.
4. Omondi A.R., Premkumar A.B. *Residue number systems: theory and implementation*. World Scientific, 2007. Vol. 2.
5. Bajard J.C., Eynard J. RNS Approach in Lattice-Based Cryptography. *Embedded Systems Design with Special Arithmetic and Number Systems*. 2017. P. 345–368. DOI: 10.1007/978-3-319-49742-6_13.
6. Valueva M.V., Nagornov N.N., Lyakhov P.A., *et al.* Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*. 2020. Vol. 177. P. 232–243. DOI: 10.1016/j.matcom.2020.04.031.
7. Roohi A., Taheri M.R., Angizi S., *et al.* Rnsim: Efficient deep neural network accelerator using residue number systems. *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2021. P. 1–9. DOI: 10.1109/iccad51958.2021.9643531.
8. Lutsenko V., Babenko M., Deryabin M. Construction of Akushsky Core Functions Without Critical Cores. *Mathematics*. 2024. Vol. 12, no. 21. P. 3399. DOI: 10.3390/math12213399.
9. Lutsenko V.V., Babenko M.G., Khamidov M.M. High speed method of conversion numbers from residue number system to positional notation. *Proceedings of the Institute for System Programming of the RAS*. 2024. Vol. 36, no. 4. P. 117–132. DOI: 10.15514/ispras-2024-36(4)-9.
10. Shiriaev E., Kucherov N., Babenko M., *et al.* Fast operation of determining the sign of a number in RNS using the Akushsky core function. *Computation*. 2023. Vol. 11, no. 7. P. 124. DOI: 10.3390/computation11070124.
11. Lutsenko V., Geryugova A., Babenko M., *et al.* High-Speed Parity Number Detection Algorithm in RNS Based on Akushsky Core Function. *International Conference on Communication and Computational Technologies / eds. by S. Kumar, S. Hiranwal, R. Garg, S.D. Purohit*. Cham: Springer, Singapore, 2024. P. 491–504. DOI: 10.1007/978-981-97-7423-4_38.
12. Lutsenko V.V., Babenko M.G., Chernykh A.N., Lapina M.A. Optimization of a number division algorithm in the residue number system based on the Akushsky core function. *Proceedings of ISP RAS*. 2023. Vol. 35, no. 5. P. 157–168. (in Russian) DOI: 10.15514/ispras-2022-35(5)-11.
13. Wang Y. Residue-to-binary converters based on new Chinese remainder theorems. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. 2000. Vol. 47, no. 3. P. 197–205. DOI: 10.1109/82.826745.
14. Akushskiy I.Ya., Burtsev V.M., Pak I.T. About new positional characteristic of non-positional code and its application. *Theory of Coding and Optimisation of Complex Systems*. Alma-Ata, Nauka, KazSSR. 1977. P. 8–16. (in Russian)
15. Metropolis N., Ulam S. The Monte Carlo method. *Journal of the American Statistical Association*. 1949. Vol. 44. no. 247. P. 335–341.

16. Kroese D.P., Brereton T., Taimre T., *et al.* Why the Monte Carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*. 2014. Vol. 6, no. 6. P. 386–392. DOI: 10.1002/wics.1314.
17. Shiriaev E., Kucherov N., M Babenko M., *et al.* Algorithm for determining the optimal weights for the Akushsky core function with an approximate rank. *Applied Sciences*. 2023. Vol. 13, no. 18. P. 10495. DOI: 10.3390/app131810495.
18. Kureichik V.M. Genetic algorithms. *Izvestiya Yuzhny Federal University. Technical Sciences*. 1998. Vol. 8, no. 2. P. 4–7. (in Russian)
19. Mirjalili S. Evolutionary algorithms and neural networks. *Studies in Computational Intelligence*. 2019. Vol. 780, no. 1. P. 43–53. DOI: 10.1007/978-3-319-93025-1.
20. Pirlo G., Impedovo D. A new class of monotone functions of the residue number system. *International Journal of Mathematical Models and Methods in Applied Sciences*. 2013. Vol. 7, no. 9. P. 802–809.

ВЫПОЛНЕНИЕ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ НА MLOPS ПЛАТФОРМЕ НИУ ВШЭ*

© 2025 А.С. Хританков, В.А. Полежаев, Г.А. Жуликов, М.С. Галынчик,
Н.А. Климин, К.Е. Сахаров, В.О. Минченков, И.В. Спирин, И.И. Крупнов,
С.Ф. Якушева, А.С. Маратканова, В.И. Козырев, П.С. Костенецкий,
Х.М. Салех

*Национальный исследовательский университет Высшая школа экономики
(109028 Москва, Покровский бул., д. 11)*

E-mail: akhritankov@hse.ru

Поступила в редакцию: 19.05.2025

Несмотря на распространение и успешные применения средств интеллектуального анализа и обработки данных для решения отдельных прикладных задач, все еще не решена проблема разработки технологии создания таких программных средств. В работе в контексте единого процесса MLOps создания технологий машинного обучения рассматриваются возникающие задачи автоматизации и выполнения распределенных вычислительных экспериментов на базе единой вычислительной платформы. Разрабатываемая в НИУ ВШЭ платформа MLOps предназначена для развертывания интеллектуальных веб-сервисов и программных средств анализа данных. Платформа должна управлять доступными локально и в облачной среде разнородными ресурсами и объединять их с ресурсами вычислительного кластера CHARISMa НИУ ВШЭ под управлением Slurm. Таким образом актуальна задача интеграции указанных ресурсов для проведения вычислительных экспериментов, реализации конвейеров настройки моделей машинного обучения, решения задач обработки и анализа данных. Особенности решаемой задачи являются рассмотрение процесса вычислений, как составной части технологии создания интеллектуальных веб-сервисов, обусловленная этой технологией необходимость использования разнородных ресурсов и использование единой гибридной платформы для выполнения вычислений. В работе предложено решение указанной задачи интеграции вычислений и приведены результаты апробации решения для интеллектуальных веб-сервисов. Показана принципиальная возможность такой интеграции разнородных ресурсов в одном вычислительном эксперименте на базе расширяемой пользователем объектной модели эксперимента и предметно-ориентированного языка его спецификации, решены вопросы динамического управления развертыванием интеллектуальных приложений, интеграции конвейеров обработки данных, веб-сервисов и наборов данных для выполнения распределенных вычислительных экспериментов.

Ключевые слова: распределенные вычислительные эксперименты, машинное обучение, облачные технологии, MLOps.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Хританков А.С., Полежаев В.А., Жуликов Г.А. и др. Выполнение распределенных вычислительных экспериментов на MLOps платформе НИУ ВШЭ // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2025. Т. 14, № 2. С. 42–66. DOI: 10.14529/cmse250203.

Введение

Распределенные вычислительные системы используются во многих областях при проведении вычислительных экспериментов, обучении моделей машинного обучения, обработки больших данных, в научных вычислениях. Вычислительные кластеры, в том числе со спе-

*Статья рекомендована к публикации программным комитетом Всероссийской научной конференции с международным участием «Параллельные вычислительные технологии (ПаВТ) 2025».

специализированными ускорителями вычислений, Грид-системы [1], системы общественных вычислений (public/volunteer computing), облачные центры обработки данных совместного использования, туманные вычисления (fog computing) [2] с использованием ресурсов оконечных и промежуточных устройств и другие составляют современный спектр подходов к решению задач в области.

Разработка таких распределенных систем [3] и создание методов и средств, облегчающих разработчикам и исследователям доведение результатов своих прикладных научных исследований до промышленного применения на их основе, составляют одну из важнейших задач прикладной науки и экономики.

Интегрированный процесс создания приложений в области машинного обучения (MLOps) объединяет этапы моделирования, разработки программного обеспечения, тестирование, внедрение и использование таких приложений и интеллектуальных сервисов в их составе. Сходные с MLOps подходы применяются организациями для создания и последующего развития нескольких интеллектуальных приложений разными командами.

Центр искусственного интеллекта НИУ ВШЭ [4] разрабатывает **MLOps фреймворк** — библиотеку программных средств автоматизации процедур процесса MLOps, на ее основе создана распределенная вычислительная **платформа MLOps** развертывания интеллектуальных веб-сервисов. Фреймворк автоматизирует процедуры процесса MLOps и скрывает излишнюю для исследователей данных сложность создания и эксплуатации облачных веб-сервисов и выполнения расчетных конвейеров в распределенной среде. Платформа объединяет облачные вычислительные ресурсы, облачные средства хранения данных и суперкомпьютерный кластер sHARISMa. Платформа может быть использована разработчиками-исследователями как облачный сервис (SaaS) при создании интеллектуальных сервисов и приложений с подключением конечных потребителей сервисов через API по сети Интернет.

Для этого необходимо решить проблему интеграции доступных платформе разнородных вычислительных ресурсов [5], как локально на ПК разработчика приложения, так и в центре обработки данных и ресурсов вычислительного кластера НИУ ВШЭ. Задача возникает при проведении распределенных вычислительных экспериментов в области машинного обучения и в настоящий момент часто решается вручную, от случая к случаю с использованием подручных программных средств и с привлечением специалистов в области высокопроизводительных вычислений, разработчиков облачного программного обеспечения и других.

В настоящей работе проблема интеграции вычислительных ресурсов рассматривается, как совокупность задач модельно-ориентированной разработки ПО (MDSD), то есть разработки модели предметной области интеллектуальных приложений, объединяющей различные их компоненты на общих правилах, и задач создания программных средств трансляции моделей приложений и исполнения приложений с использованием доступных вычислительных ресурсов. Выделение концептуальной модели предметной области, описывающей структуру приложения и его сервисов позволяет решать дальнейшие задачи воспроизводимости и прослеживаемости распределенных вычислительных экспериментов сразу для широкого класса приложений в области машинного обучения. Не связанные с фреймворком разработчики приложений в области биоинформатики, анализа данных физических экспериментов, медицинской диагностики, интеллектуального поиска, заботы и аналитики по сотрудникам и других могут использовать реализованную в MLOps фреймворке модель для описания структуры своих приложений, фреймворк обеспечивает интеграцию компо-

нентов этих приложений, в том числе конвейеров обучения расчетных моделей в рамках вычислительных экспериментов.

Предлагаемый подход к интеграции разнородных вычислительных ресурсов на единой платформе на основе модели и его реализация составляют основной результат данной работы. Анализ проблемы интеграции и существующих решений в области приведены в разделе 1. В разделе 2 более полно изложены основные решения по автоматизации процесса MLOps и интеграции вычислительных ресурсов для его автоматизации. В разделе 3 изложены алгоритмы и структура предлагаемого решения. Экспериментальная проверка его реализуемости и расчеты эффективности использования вычислительных ресурсов приведены в разделе 4.

1. Анализ литературы и альтернативные решения

Задача интеграции вычислительных ресурсов состоит в их объединении в общую среду с единым доступом и автоматическим планированием и распределением задач. При этом вычислительные ресурсы могут быть разнородными. Выделяют несколько типовых решений. Вычислительные кластеры (compute cluster) состоят из обычных ЭВМ с централизованным управлением. Распространенная гибридная архитектура кластеров сочетает использование процессоров (CPU), графических карт (GPU) и иногда специализированных ускорителей [6]. Грид-системы (grid computing) объединяют кластеры, суперкомпьютеры и другие ресурсы, разделенные географически и часто связанные с экспериментальными установками. Грид-системы отличает отсутствие централизованного управления, разнородность и низкая связанность вычислительных узлов. Виртуальные организации объединяют пользователей, использующих Грид-системы для решения задач [1]. К разновидности Грид-систем относят системы общественных вычислений (public/volunteer computing), в которых расчеты проводятся пользователями, добровольно предоставляющими свои компьютеры. Технология облачных вычислений (cloud computing) позволяет проводить вычисления на сторонних серверах центра обработки данных, предоставляемых в аренду — «облаке». Преимуществами облачных вычислений являются отсутствие необходимости приобретения и самостоятельного обслуживания оборудования и гибкая тарификация.

Традиционными подходами к интеграции удаленных ресурсов являются использование стандартных сетевых протоколов, таких как SSH, и написание программ-агентов, работающих на стороне ресурса [7]. Известно решение задачи интеграции суперкомпьютера НИЦ «Курчатовский институт» с Грид-системой и облачной платформой с помощью унифицированных интерфейсов и программы PanDA [8]. Также произведена интеграция суперкомпьютера «Говорун», двух Грид-кластеров, кластера NICA и облака ОИЯИ с помощью библиотеки DIRAC Interware [9]. Для решения задачи интеграции гетерогенных вычислительных ресурсов применяются технологии виртуализации [10]. Для интеграции суперкомпьютеров и облачных вычислений предложена концепция MC2E [5].

В научной среде популярна организация процесса вычислений в виде конвейера (workflow) из множества соединенных инструментов [11], в том числе контейнеров [12], изолирующих исполнение отдельных этапов. Конвейеры расчетов (pipeline) используются для обучения моделей и проверки гипотез. В работе [6] рассматривается задача интеграции высокопроизводительных вычислений, технологий анализа данных и машинного обучения в рамках одного конвейера и предлагается парадигма HPC Workflow-as-a-Service.

Вычислительный эксперимент проводится над математической моделью объекта с помощью вычислительных и логических процедур, осуществляемых соответствующими программными средствами на вычислительных системах. Проведение вычислительных экспериментов в области машинного обучения предполагает определение цели эксперимента, проверяемых гипотез о свойствах данных, критериев успешности, используемых данных, структуры расчетной модели и связанной задачи оптимизации для подбора параметров модели, определения конвейеров с необходимыми расчетами для проверки выбранных гипотез.

При проведении вычислительных экспериментов для улучшения воспроизводимости и повышения доверия к их результатам, разработки и улучшения конвейеров расчетов [13, 14], необходимо устанавливать происхождение данных и определять произведенные с ними операции, что составляет задачу прослеживаемости (data provenance) [15]. Для ее решения используются технологии распределенных реестров; моделирование конвейера расчетов с помощью графов знаний, семантических сетей и онтологий [16].

MLOps платформа автоматизирует процессы, связанные с проведением вычислительных экспериментов, обучением, внедрением и наблюдением за использованием расчетных моделей и интеллектуальных сервисов на их основе. Платформа решает задачу интеграции разнородных вычислительных ресурсов, суперкомпьютера сHARISMa НИУ ВШЭ, локальных вычислительных ресурсов исследователя и облачных ресурсов для проведения распределенных вычислительных экспериментов.

На решение сходных задач направлены несколько программных систем. Система Asperitas [17] предназначена для управления виртуальными сетями и вычислительными кластерами и хранения данных и включает в себя облачную среду, оркестраторы и приложения для контроля на основе программного средства автоматизации сборки программ Ansible. Система Fanlight (Desktop-as-a-Service) [18] предназначена для создания виртуальных рабочих мест с доступом к интегрированным прикладным математическим пакетам и аппаратным ресурсам высокопроизводительных вычислительных комплексов, подключенных систем хранения данных, вычислительных серверов. При построении системы Everest была произведена интеграция облачных ресурсов и Грид-систем [3]. При этом вышеперечисленные платформы не имеют специализации на технологиях машинного обучения и собственных вычислительных ресурсах, доступных пользователям системы.

Платформа DataMall [19] использует методы машинного обучения, инфраструктуру БД и единую облачную экосистему для работы с объектами на основе больших данных. Также платформа играет роль биржи специалистов, данных и ресурсов. Об интеграции вычислительных кластеров, Грид-систем и других ресурсов не упоминается.

Коммерческие платформы, такие как VK ML Cloud, Google Cloud AI, Amazon SageMaker и Azure AI Studio, предоставляют доступ только к облачным ресурсам их разработчиков с возможностью подключения дополнительных вычислителей в режиме гибридного облака (hybrid computing). Платформа HuggingFace использует облачные ресурсы партнеров, таких как Amazon и Google. Платформа MLSpace интегрирует суперкомпьютеры Christofari и Christofari Neo. Система ClearML рассчитана на использование ресурсов пользователя. Таким образом, крупнейшие коммерческие ML-платформы [20] решают задачи интеграции меньшего числа ресурсов по сравнению с разрабатываемой MLOps платформой. Программные средства с открытым исходным кодом, такие как MLflow, Kuberflow,

Airflow и TensorFlow Extended [20], предназначены для общей организации выполнения конвейеров расчетов и не решают задачу интеграции вычислительных ресурсов.

2. Процесс MLOps и архитектура платформы

В области машинного обучения вычислительные эксперименты проводятся для проверки гипотез о соответствии или оптимизации параметров предлагаемой расчетной модели имеющимся данным. Результаты такого эксперимента в на практике используют для создания интеллектуальных сервисов, предоставляющих обученные расчетные модели для использования через интерфейсы прикладного программирования (API). Большой объем доступных данных и вычислительная сложность моделей требует применения методов высокопроизводительных научных вычислений, использования аппаратных ускорителей и параллельных алгоритмов на вычислительных кластерах [21].

Одним из перспективных подходов к гибкому созданию интеллектуальных приложений на практике считается объединение в одном процессе моделирования данных, программной реализации и развертывание их на облачной инфраструктуре. Такой процесс поставки и предоставления в использование прогнозных моделей кратко называют MLOps. При разработке фреймворка в этом процессе были выделены следующие процедуры, выполняемые командами исследователей и разработчиков, схематично представленные на рис. 1.

На первом этапе моделирования ставится задача прикладного исследования, выполняется сбор наборов данных, исследовательский анализ данных, разработка моделей и экспериментальное исследование их качества.

На этапе реализации выполняется разработка интеллектуального приложения, в том числе алгоритма воспроизводимого обучения моделей, так называемого конвейера расчетов (pipeline), и веб-сервисов, представляющих результаты обучения по API. После этого выполняется оценка качества обучения с помощью конвейера расчетов. При необходимости, конвейер используется для дообучения и адаптации ранее разработанных моделей под новые задачи или новые наборы данных.

На третьем этапе применения проводится тестирование и верификация разработанных приложений и контроль качества. После этого, сборка и поставка модулей для развертывания на целевой платформе, предоставление доступа и подключение разнородных вычислительных ресурсов. После развертывания выполняется наблюдение за входными, выходными данными приложения и качеством его работы.

Основные задачи, на решение которых направлен MLOps фреймворк при использовании командами исследователей и разработчиков процессу, следующие: стандартизация описания конвейеров обучения, автоматизация дообучения и адаптации интеллектуальных приложений, автоматизация построения отчетов по результатам экспериментов, упрощение процедур создания и развертывания интеллектуальных приложений, повышение уровня технологической готовности исследовательских расчетных моделей, интерпретация результатов расчетов и построенных модельных прогнозов, облегчение поиска и обнаружения развернутых приложений и наборов данных.

Решение представленных сложных задач в комплексе требует проработки архитектуры программной системы и определение принципов ее разработки и развития.

Во-первых, рассматриваемый фреймворк использует декларативное описание структуры и конфигурации приложений, входящих в их состав конвейеров, веб-сервисов и других компонентов. В отличие от программного способа определения состояния, при де-



Рис. 1. Решаемые системой задачи автоматизации процесса MLOps

кларативном описании именно MLOps фреймворк определяет алгоритм разбора различий (reconciliation) между указанным разработчиком целевой и текущей конфигурацией приложения на MLOps платформе.

Во-вторых, возникает задача предоставления разработчикам удобного языка для описания такой конфигурации. В отличие от имеющегося низкоуровневого декларативного описания в инфраструктуре Kubernetes, ориентированной на широкий класс облачных приложений, MLOps фреймворк предлагает высокоуровневый предметно-ориентированный язык, специализированный для интеллектуальных приложений и программных средств машинного обучения. В основе языка лежит объектная модель, определяющая конфигурации и структуры компонентов приложений и вычислительных экспериментов. Для совместимости, язык описания конфигурации разработан как внутренний DSL с использованием средств расширения самой инфраструктуры Kubernetes.

Выполненное на предлагаемом языке описание структуры конвейера расчетов уже не зависит от реализации алгоритмов фреймворка и внутренней структуры и технологий MLOps платформы, что позволяет решать задачу выполнения распределенных вычислительных экспериментов и интегрировать разнородные вычислительные ресурсы. При этом MLOps платформа обеспечивает необходимую настройку, обмен данными и запуск компонентов конвейера расчетов на каждом используемом вычислителе.

Решение двух указанных задач позволит использовать возможности высокопроизводительных вычислительных систем и методы научных вычислений при создании интеллектуальных приложений в отраслях на основе объединенного процесса MLOps.

3. Реализация платформы и распределенные вычислительные эксперименты

MLOps платформа объединяет облачные вычислительные ресурсы под управление инфраструктуры Kubernetes, подключаемое через API облачное объектное хранилище, системные сервисы, такие как система управления версиями, реестр образов и пакетов, API

шлюзы, системы мониторинга, библиотеки программ, предоставляемые разработчикам-исследователям и другие подсистемы.

В данном разделе изложены основные решения по реализации платформы MLOps и организации распределенных вычислительных экспериментов на ее основе.

3.1. Интеграция на основе общей объектной модели

Методология модельно-управляемой разработки программного обеспечения (model-driven software development) предусматривает разработку предметно-ориентированных языков (DSL), использование концептуальных, имитационных, расчетных моделей для формализации решаемых задач в выбранной предметной области, и автоматической трансляции таких моделей в программное обеспечение или интерпретации этих моделей. Система с открытым кодом MLDev реализует расширяемую модель вычислительного эксперимента и интерпретатор такой модели, следуя указанному подходу. Модель эксперимента задается декларативно на внутреннем DSL, который может быть расширен пользователем. В ходе интерпретации система MLDev обеспечивает интеграцию программных средств, используемых в вычислительном эксперименте, и воспроизводимое выполнение заданных на этом DSL конвейеров расчетов для проверки гипотез [14].

Декларативное описание структуры вычислительного эксперимента также применяется в инфраструктуре Kubernetes для указания конфигурации приложений, развертываемых на вычислительных ресурсах в центрах обработки данных. Предлагаемая Kubernetes объектная модель позволяет описывать широкий класс веб и серверных приложений, веб-сервисы и исполняемые фоновые задачи. Конфигурация приложения задается набором документов, называемых манифестами.

Таким образом, использование подходов, аналогичных примененным в MLDev, и разработка совместимой, понятной специалистам в машинном обучении объектной модели распределенных вычислительных экспериментов для инфраструктуры Kubernetes позволит интегрировать разнородные вычислительные ресурсы.

Суть предлагаемого решения состоит в следующем. Пользователи фреймворка записывают структуру размещаемого интеллектуального приложения, в том числе интеллектуальных сервисов, вычислительных экспериментов и наборов данных в виде структурированных документов — манифестов. Фреймворк выполняет трансляцию из этих манифестов в объектной модели фреймворка в объектную модель приложений Kubernetes и отслеживает их соответствие. Kubernetes приводит в соответствие указанную в модели целевую конфигурацию приложения и фактически исполняемые в облачной вычислительной среде процессы, тома дисков, сетевые ресурсы.

Структура интеллектуального программного приложения и его компонентов в предлагаемой объектной модели приведена на рис. 2 ниже.

Компонент — это любой размещаемый разработчиками-исследователями на MLOps платформе исполняемый программный модуль. Экземпляры компонента, как такового, создавать не допускается, в модели определено несколько видов компонентов, которые пользователь может использовать.

Для размещения расчетной модели пользователь описывает в конфигурации приложения интеллектуальный веб-сервис (ML-компонент). После создания, экземпляр ML-компонента предоставляет API для вызова расчетной модели в интерактивном режиме. ML-компонент не имеет значимого состояния, параметры модели, сторонние данные и дру-

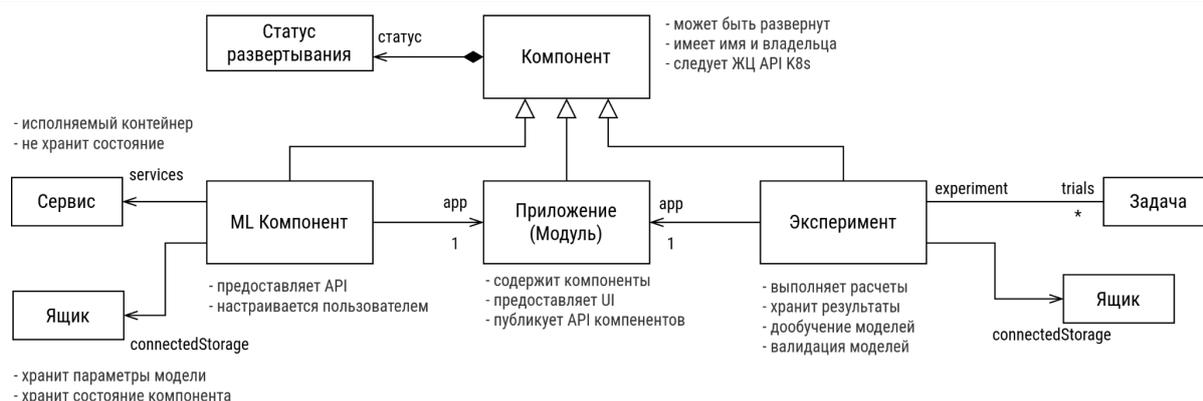


Рис. 2. Фрагмент объектной модели интеллектуального приложения с пояснениями

гие настройки хранятся в ящике данных (Ящик). Поэтому ML-компонент может быть перезапущен фреймворком в любое время.

Конвейер вычислительного эксперимента (Эксперимент) подразумевает исполнение длительной расчетной задачи, результатом которой будет набор файлов, размещаемых в ящиках данных. Этот набор файлов может содержать как отчеты об эксперименте, так и данные статического веб-сайта для отображения в пользовательском интерфейсе MLOps платформы. Расчетная задача может выполняться как на ресурсах платформы, на стороннем подключенном вычислительном кластере, так и выполнена заранее и ее результаты загружены до запуска эксперимента.

Приложение группирует другие компоненты, позволяя им взаимодействовать между собой и с пользовательским интерфейсом приложения. Например, дообученная в эксперименте модель может быть использована в ML-компоненте того же приложения. Изменяемые системой атрибуты приложения содержат сведения о статусе развертывания и готовности приложения к работе, ссылки и адреса URL, по которым можно получить доступ к компонентам приложения, использованные для развертывания версии источников из Git репозитория приложения.

Пользователи используют ML-компонент для предоставления результатов обучения, например, расчетных моделей, для использования в интерактивном режиме через формируемый фреймворком API, и принимающие на вход наборы данных эксперименты для выполнения конвейеров расчетов (pipelines) в фоновом режиме.

Структура вычислительного эксперимента в объектной модели фреймворка предусматривает как использование системы MLDev с несколькими конвейерами, подключаемыми модулями на этапах конвейера, модулями наборов данных, так и упрощенный режим с использованием одного конвейера в эксперименте. Трансляция структуры эксперимента с объектной модели MLDev в объектную модель фреймворка, записанную на внутреннем DSL, выполняется с использованием дополнительных программных модулей библиотеки MLDev.

Хранимые в приложении наборы данных могут быть использованы повторно другими приложениями на MLOps платформе и вне ее. Объектная модель фреймворка предусматривает возможность версионирования и добавления метаданных к данным в формате FAIR [22]. При этом версии набора данных становятся доступны по уникальным URL, пригодным для цитирования.

Для реализации указанного выше подхода фреймворк использует средства расширения объектной модели Kubernetes (Kubernetes operator framework) и определения вида ресурсов (CRD) при реализации объектной модели. Поэтому для создания экземпляров ресурсов из объектной модели фреймворка пользователям достаточно стандартного API Kubernetes.

Для создания ресурсов Kubernetes по добавленным пользователям манифестам фреймворк реализует программные компоненты управления ресурсами инфраструктуры Kubernetes (контроллеры CRD), для каждого класса объектной модели отдельный контроллер.

Определение ресурса CRD содержит контрольные соотношения (правила валидации) манифестов, проверяемые перед добавлением таких CRD самим Kubernetes. При обработке созданного CRD системой, обработчик системы дополнительно проверяет выполнение соотношений и ограничений, устанавливаемых моделью предметной области. В случае ошибки при добавлении, система сохраняет подробное сообщение об ошибке для просмотра пользователем стандартными средствами Kubernetes и устанавливает подходящий статус ресурса.

3.2. Выполнение распределенных вычислительных экспериментов

Для проведения вычислительного эксперимента в составе интеллектуального приложения необходимо подготовить манифесты с описанием конвейера расчетов и других параметров эксперимента в терминах ресурсов фреймворка.

Пример описания структуры эксперимента в объектной модели фреймворка приведен на рис. 3.

Объектная модель фреймворка содержит конвейер расчетов (ExperimentPipeline), который определяет этапы вычислительного эксперимента, ящик данных (DataBox) задает область хранения данных, используемых приложением, набор данных (DatasetComponent) определяет повторно используемый набор файлов для обучения расчетных моделей, интеллектуальный сервис (MLComponent) определяет программный сервис вычислений с использованием расчетных моделей машинного обучения, интерфейсный компонент (APIComponent) определяет API для доступа к сервисам эксперимента.

Каждому этапу конвейера сопоставлен образ контейнера, реализующий расчет. Этап принимает данные через входные переменные, выполняет расчет, возвращает данные через выходные переменные. Каждая переменная ассоциирована с ящиком данных, значения входных и выходных переменных передаются в виде файлов. При запуске эксперимента для каждой переменной область хранения данных подключенного ящика монтируется в контейнер этапа. Программа, выполняющая расчет, получает путь к файлам каждой переменной через переменные окружения. Таким образом, широкий класс программ, которые используют файлы в качестве входов и выходов, могут быть включены в конвейер как этап вычислительного эксперимента.

В текущей реализации фреймворк реализует хранение данных ящиков в объектном хранилище, совместимом с API S3.

После создания ресурсов эксперимента фреймворком, конвейер может быть исполнен. Для запуска необходимо отправить запрос к сервису конвейеров фреймворка, передав значения входных переменных или пути к ящикам данных в теле запроса и указав перечень запрашиваемых результатов.

Далее алгоритм выполнения конвейера эксперимента следующий. Получив запрос на запуск, сервис конвейеров проверяет входные данные, следуя структуре конвейера, форми-

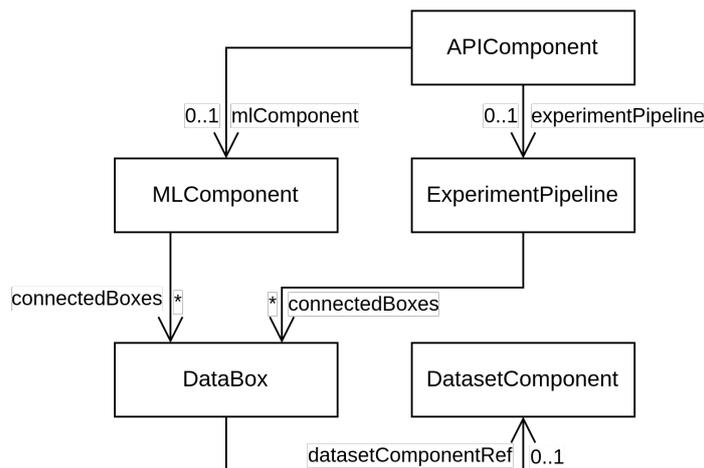


Рис. 3. Пример структуры эксперимента в объектной модели фреймворка

рует спецификацию вычислительной задачи Kubernetes, добавляя последним этап с проверкой результатов расчетов и создает ресурс задачи (Job), запуская таким образом конвейер. При этом этап конвейера сопоставляется отдельному контейнеру системы контейнеризации containerd при запуске этапа в инфраструктуре Kubernetes, или контейнера Singularity, при запуске на вычислительном кластере.

Каждому запуску присваивается уникальный идентификатор отслеживания. Это позволяет узнать статус выполнения конвейера и получить результаты расчетов. Фреймворк реагирует на события вычислительной задачи и обновляет статус запуска в системной базе данных.

3.3. Библиотека расширений конвейера эксперимента

Фреймворк предлагает готовые программные модули, которые включаются в вычислительный эксперимент при помощи механизма расширения объектной модели MLDev. Такие программные модули составляют библиотеку тегов фреймворка.

Теги библиотеки позволяют включить в вычислительный эксперимент готовую функциональность, объединить программные модули библиотеки фреймворка и разработанные программы этапов эксперимента в единый конвейер.

Для использования этих тегов в вычислительном эксперименте, необходимо описать эксперимент MLDev с тегом конвейера фреймворка Unipipeline. Этапы конвейера указываются тегами контейнеров ContainerStage. В режиме подготовки запуска MLDev prepare реализация тега контейнера выполняет сборку образа контейнера, в режиме запуска run исполняет контейнер на основе собранного образа. Тег контейнера указывает исходный код, используемые библиотеки, команду или скрипт запуска и позволяет запускать в контейнере широкий класс программ и вычислительных экспериментов MLDev, используемых в расчетах в машинном обучении.

Библиотека фреймворка содержит следующие классы на языке Python, реализующие следующие теги разметки YAML для использования при написании схемы эксперимента:

- Тег SingularValues выполняет регуляризацию нейросетевых методов обучения путем ограничения собственных значений параметров модели.

- Тег StageSrc позволяет использовать в этапах конвейера набор данных DatasetComponent.
- Тег CharismaStage позволяет запускать этап конвейера в вычислительном кластере сHARISMa НИУ ВШЭ или любом другом кластере с планировщиком задач Slurm.
- Тег SynthesizeDataStage обучает статистическим свойствам набора данных генеративные модели и с их помощью генерирует синтетические данные.
- Тег Report создает по шаблону отчеты в виде веб-сайтов по проведенному вычислительному эксперименту.
- Тег UnipPipeline определяет конвейер расчетов для запуска на MLOps платформе и преобразует описание эксперимента MLDev в манифесты фреймворка.
- Тег AttentionInterpStage реализует алгоритм интерпретации результатов расчетов с использованием карты внимания в глубоких нейронных сетях.
- Тег StructuredInterpStage реализует алгоритмы интерпретации результатов классификации с помощью методов LIME и SHAP.

```

report: !Report
name: report
inputs:
  - &template !StageVar
    name: template
    path: report/data/template
    # ...
    # другие входные данные
outputs:
  - !StageVar
    name: report
    path: report/data/report
params:
  template: *template
  # ...
  # другие параметры отчета
model:
  pic1: *pic1
  table: !Table
  title: "Финальные метрики"
  data: *table
  # ...
  # заполнение остальных полей
  # шаблона
    
```

Рис. 4. Пример фрагмента YAML с описанием отчета

3.4. Генератор отчетов Report

Тег Report библиотеки тегов фреймворка предоставляет функциональность для ускорения подготовки повторяющихся отчетов в формате веб-страниц о проведенных вычислительных экспериментах посредством параметризуемых шаблонов. Тег предоставляет возможность отобразить полученные в ходе работы конвейера работы файлы в формате текста, изображений, графиков или таблиц. Параметры тега в YAML манифесте определяют содержание отчета, в том числе задают однозначное соответствие файлов, создаваемых в ходе работы предыдущих этапов, и маркеров, с помощью которых данные будут размещены в отчете, а также их формат. Пример описания отчета приведен на рис. 4. Шаблон отчета и текст, который не будет изменяться при повторном запуске, можно указать в шаблоне отчета в формате Markdown. В этом же файле может быть указано расположение маркеров, определенных в манифесте.

В ходе подготовки отчета указанные в манифесте файлы будут преобразованы в соответствии с указанным ранее форматом в текст, изображения, графики или таблицы. Изоб-

ражения будут сжаты и закодированы в base64. При преобразовании графиков и таблиц они будут вначале сформированы в формате RST, а впоследствии будет сформирован соответствующий им HTML. После этого полученные данные будут вставлены в шаблон отчета. Конечным результатом будет HTML-файл со встроенными в него стилями, таблицами, графиками и изображениями для предоставления пользователю.

3.5. Представление результатов расчетов в виде сервисов

Результаты расчетов вычислительных экспериментов могут быть представлены MLOps платформой в виде RESTful веб-сервисов. Эти интеллектуальные сервисы позволяют использовать обученные ранее модели для обработки данных с помощью определяемого MLOps фреймворком API.

Для работы сервисам необходимо различное количество вычислительных ресурсов, которое может изменяться со временем в зависимости от количества запросов или объема обрабатываемых данных. Также могут возникнуть ситуации при которых сервисы резервируют вычислительные ресурсы, но не производят обработку данных ввиду отсутствия запросов через API сервиса. Это приводит к уменьшению утилизации выделенных вычислительных ресурсов MLOps платформы.

Задачу повышения эффективности использования вычислительных ресурсов MLOps платформы решает модуль автомасштабирования интеллектуальных сервисов. Модуль автомасштабирования интеллектуальных сервисов позволяет автоматически изменять количество реплик интеллектуального сервиса. В отличие от стандартных средств масштабирования для Kubernetes, модуль позволяет сократить количество реплик сервиса до нуля и увеличить при поступлении запроса через стандартный API сервиса.

Конфигурация алгоритма масштабирования определяет набор состояний целевого сервиса. Состояния сервиса в свою очередь определяют необходимое количество реплик и набор правил перехода в другие состояния. Перевод сервиса из одного состояния в другое модуль масштабирования осуществляет согласно правилам перехода на основе метрик, которые модуль масштабирования получает от Сервера метрик через определенные промежутки времени. Структура обменов данными между частями фреймворка и модуля приведена на рис. 5.

Особым образом обрабатывается выключение сервиса при его неиспользовании. Выключение интеллектуального сервиса может быть осуществлено как на основе правил перехода, задаваемых в конфигурации масштабирования, так и при получении сигнала от Менеджера оповещений при отсутствии запросов клиентов.

При выключении интеллектуального сервиса запросы к API сервиса перенаправляются на модуль автомасштабирования. Запуск сервиса после выключения производится при получении перенаправленного запроса модулем. Запуск сервиса может занять значительное время и может привести к ошибке истечения таймаута HTTP соединения. Чтобы избежать получения этой ошибки клиентом интеллектуального сервиса модуль автомасштабирования MLOps платформы периодически перенаправляет клиента на тот же адрес. Таким образом модуль будет поддерживать активное соединение, пока интеллектуальный сервис не будет полностью готов обрабатывать запросы клиента. После полного запуска целевого сервиса модуль масштабирования восстанавливает маршруты обработки запросов и только после этого перенаправляет клиента на целевой сервис.

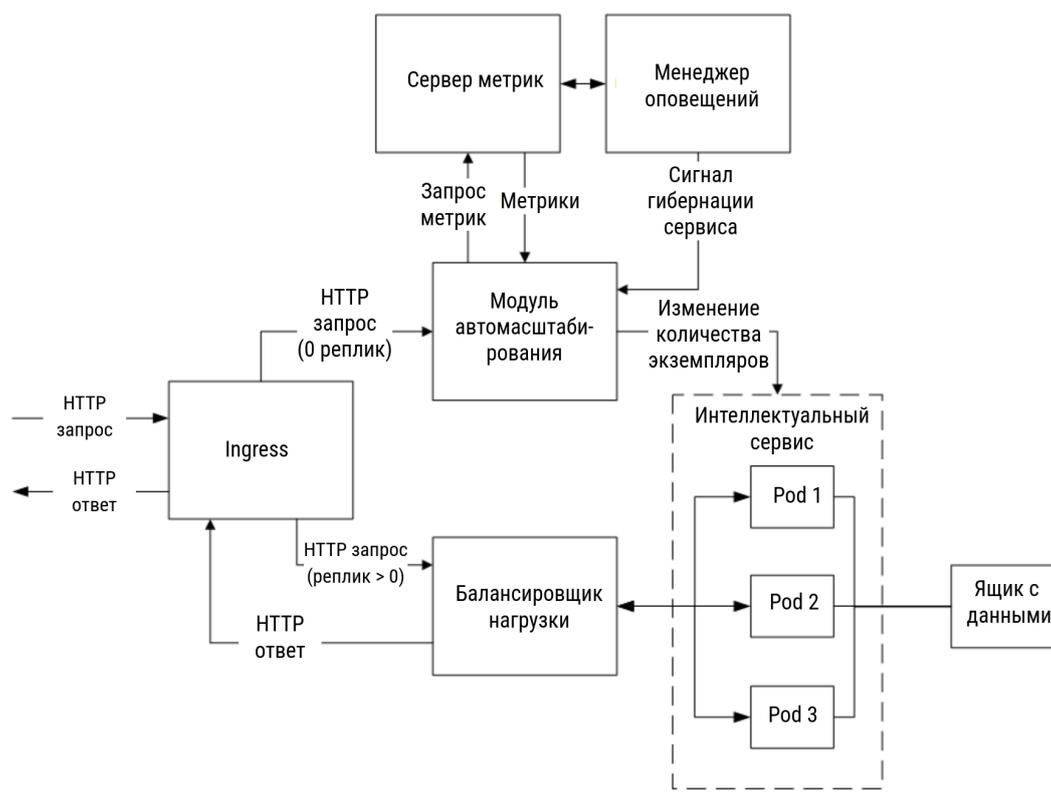


Рис. 5. Структура обменов данными при масштабировании интеллектуального сервиса

3.6. Интеграция суперкомпьютера сHARISMa

Основная способ запуска выполнения задач на вычислительном кластере сHARISMa использует систему управления заданиями SLURM. Данная система позволяет автоматически запустить задачу на подходящих узлах или поставить задачу в очередь, если в данный момент ресурсы недоступны.

Для взаимодействия конвейеров MLOps платформы с вычислительным кластером в рамках работы решены следующие задачи: передача самой вычислительной задачи на головной узел кластера, загрузка входных данных для вычислений, отслеживание текущего состояния задачи и выгрузку выходных данных после завершения.

Для передачи данных фреймворк использует соединение SSH. Во-первых, такой способ доступен для подключения к кластеру обычным пользователям, и он позволяет обеспечить защиту передаваемым данным. Во-вторых, тот же способ подключения используется для формирования расчетного задания для SLURM на головном узле в окружении пользователя, от имени которого фреймворк производит подключение.

Запуск задач через SLURM на кластере сHARISMa возможен в двух режимах: `sbatch` — запуск задач в «пакетном» режиме и `sgen` — запуск в интерактивном режиме. Так как основное преимущество режима `sbatch` — это возможность подготовки окружения через скрипт, то этот способ и был выбран для дальнейшей реализации.

Для решения задачи передачи окружения пользовательской расчетной задачи фреймворк использует образы контейнеров Singularity в формате SIF, принимаемые вычислительным кластером. Данное решение позволяет сделать выполнение кода на кластере безопасным и дополнительно позволяет пользователю фреймворка подготовить вычислительную задачу без учета особенностей доступного на вычислительном кластере окружения, про-

тестировать запуск задачи на своей стороне. При подготовке конвейера с использованием системы MLDev и библиотеки тегов фреймворка `tag CharismaStage` создает SIF образ контейнера непосредственно из образа контейнера Docker, что упрощает работу пользователей уже имеющих опыт работы с контейнерами.

Для решения задачи мониторинга процесса расчетов необходимо в дополнение к системе HPC TaskMaster иметь постоянный процесс для слежения за вычислительной задачей. Теоретически данный процесс мог быть запущен и на головном узле вычислительного кластера и отправлять статусы на внешний адрес MLOps платформы. Однако такое решение требовало, чтобы этот процесс работал постоянно, что не соответствовало концепции работы с вычислительным кластером.

Вместо этого процесс опроса статуса вычислительной задачи запускается как часть конвейера расчетов и через соединение SSH просматривает содержимое файла со статусом, доступное на головном узле. В зависимости от текущего состояния задачи `sbatch` в данном файле происходит постепенное изменение статусов: `running`, `error` и `finished`. Далее этот статус транслируется в статусы этапа конвейера расчетов в объектной модели фреймворка. В случае отключения соединения SSH фреймворк выполняет повторное переподключение через некоторое время ожидания.

После завершения процесса на вычислительном кластере выходные данные копируются через SSH обратно на узел MLOps платформы и передаются далее по конвейеру через ящики данных.

Таким образом весь процесс взаимодействия управляемой фреймворком MLOps платформы с вычислительным кластером можно описать следующими шагами:

- подготовка описания конвейера с входными данными и исполняемыми процессами;
- сборка SIF контейнера с реализацией вычислительной задачи;
- подготовка реквизитов для подключения по SSH и к репозиторию образов контейнеров;
- загрузка контейнера в ящик данных в хранилище S3;
- загрузка входные данные в ящик данных в хранилище S3;
- запуск конвейера на выполнение через подключение к головному узлу кластера.

Для анализа эффективности использования вычислительных ресурсов на вычислительном кластере сHARISMa в НИУ ВШЭ использована специализированная система мониторинга эффективности HPC TaskMaster [23]. Система позволяет автоматически обнаруживать неэффективные и некорректно запущенные пользовательские задачи за счет настройки подходящих индикаторов, параметров и тегов задач [24]:

- непараллельные задачи, ошибочно запущенные в параллельном режиме;
- неравномерное распределение ресурсов между узлами; простои выделенных вычислительных ресурсов;
- низкая загрузка CPU или GPU;
- слишком объемная запись данных на локальный SSD;
- слишком частые обращения к СХД и другие.

Использование системы HPC TaskMaster на вычислительном кластере сHARISMa более чем в 5 раз снижает время ожидания задач в очереди и освобождает более 7% вычислительных ресурсов в пользу эффективных вычислительных задач.

Для обеспечения взаимодействия с MLOps фреймворком система HPC TaskMaster расширена путем реализации REST API на базе Django REST Framework. Данное расширение

позволяет MLOps фреймворку получать детальную информацию о выполняемых задачах, включая основные сведения о задаче и запрошенных ресурсах, а также агрегированные метрики использования ресурсов. Данные в API представлены в формате JSON, который далее передается и обрабатывается в MLOps фреймворке. Для обеспечения безопасности и изоляции данных в API HPC TaskMaster реализован механизм авторизации на основе WskeyToken. Доступ к информации о задачах ограничивается значением wskey — специального идентификатора, который указывается фреймворком при постановке задачи в очередь планировщика SLURM. API предоставляет данные только о тех задачах, которые имеют соответствующий wskey, что позволяет организовать контролируемый доступ к задачам, запущенным через MLOps фреймворк.

4. Пример применения для реализации MLOps платформы

Проект «MLOps-система исполнения и мониторинга ИИ-моделей» выполняется в составе стратегического проекта № 4 «Цифровая трансформация: технологии, эффекты, эффективность» программы академического лидерства «Приоритет 2030». Проект направлен на создание и развитие центров компетенций по развитию и применению технологий искусственного интеллекта в НИУ ВШЭ и их интеграцию в образовательный процесс, в сферах управления и медицины совместно с партнерами. Разрабатываемая в проекте MLOps платформа позволяет автоматизировать процессы развертывания, поддержки и эксплуатации цифровых сервисов на основе искусственного интеллекта.

Ключевая ценность для пользователей и разработчиков моделей машинного обучения состоит в снижении трудоемкости развертывания интеллектуальных приложений и сервисов на основе данных информационных систем НИУ ВШЭ; снижении модельного риска при применении расчетных моделей за счет своевременного определения снижения качества их работы и изменений в распределениях входных данных, повышении устойчивости работы сервисов за счет балансировки и предоставления релевантных вычислительных ресурсов.

На платформе развернуты интеллектуальные сервисы для обнаружения текста, написанного с помощью нейросетей, подготовки юридических документов на основе больших языковых моделей, сервис прогнозирования и классификации тикетов в системе управления задачами Jira, интеллектуальный ассистент сотрудника учебного офиса, отвечающий на вопросы по образовательному процессу, интеллектуальный сервис анализа изображений глазного дна и другие. В 2024 г. на платформе развернуто 8 интеллектуальных приложений и сервисов, с ожидаемым приростом по 3–5 сервисов в год до 2030 г.

4.1. Приложение обучения моделей классификации изображений

Для демонстрации работы и расчета показателей эффективности проведен вычислительный эксперимент в ходе которого решается задача идентификации объектов на изображениях. В рамках вычислительного эксперимента обучена модель RTMDet [25], предназначенная для решения задачи идентификации на наборе данных COCO. Модель реализована в модуле синтеза моделей машинного зрения для детектирования объектов и действий (AutoOD) [26, 27].

Набор данных состоит из порядка 5 тыс. примеров, разделенных на две выборки. В обучающей выборке 4 тыс. изображений и 1 тыс. в тестовой выборке. Также в набор данных входит разметка изображений в формате COCO с указанием координат объектов.

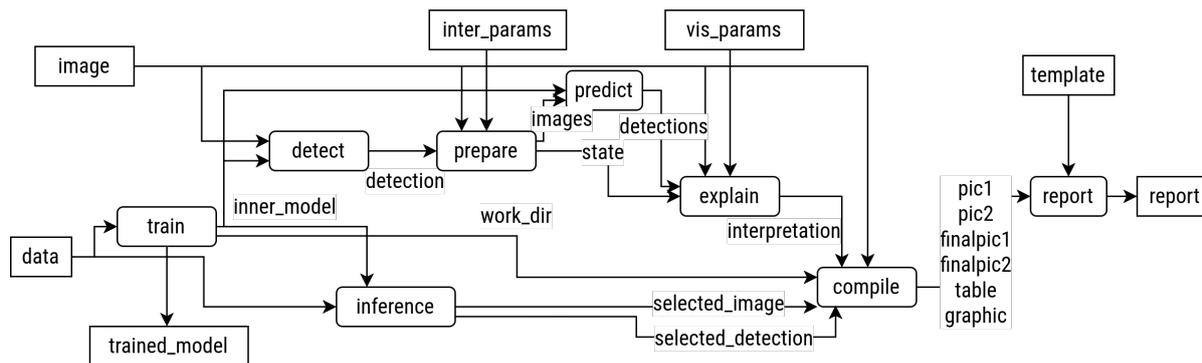


Рис. 6. Схема этапов конвейера расчета в эксперименте

Использованы следующие критерии качества модели — средняя точность (mean average precision) при пороге отношения пересечения к объединению 0.5 и 0.75 между рассчитанной моделью областью нахождения объекта на изображении и областью в разметке.

4.2. План и структура вычислительного эксперимента

Конвейер расчетов вычислительного эксперимента состоит из восьми этапов. Схема конвейера показана на рис. 6. Первый этап — обучение (train) на вычислительном кластере sHARISMa. На этом этапе используется базовый модуль AutoOD для обучения модели идентификации объектов на изображениях [27]. На вход передаются данные для обучения (data), в качестве результатов модуль возвращает обученную модель (trained_model) и передает ее в следующие этапы (inner_model). Также модуль возвращает содержимое рабочей директории обучения модели (work_dir), которая используется для анализа процесса обучения.

На втором этапе выполняется применение расчетной модели (inference). Обученная модель (inner_model) используется для идентификации объекта заданного класса на изображении из набора данных (data). Выбранное изображение (selected_image) и результат обнаружения (selected_detection) передаются далее в этап сбора данных (compile) для включения в отчет о запуске.

Этапы с третьего по шестой — это шаги процесса построения интерпретации результатов применения расчетной модели, выполняемые повторно используемым MLOps модулем библиотеки модулем фреймворка. Третий этап — обнаружение объекта (detect). В этом этапе обученная модель (inner_model) используется для идентификации объекта на исходном изображении (image). Результат идентификации (detection) передается в следующий, четвертый, этап — подготовку к интерпретации (prepare). На этапе подготовки на основе одного исходного изображения создается множество зашумленных версий (images) с вырезанной областью вокруг объекта. Количество созданных изображений, а также вид шума определяются параметрами интерпретации (inter_params). Также на этапе подготовки к интерпретации сохраняется информация о соответствии зашумленных участков изображениям (state). Пятый этап — идентификация объектов на наборе созданных изображений (predict). На пятом этапе обученная модель (inner_model) используется для обнаружения объектов на зашумленных изображениях (images) этапа подготовки к интерпретации. Результаты идентификации (detections) передаются в этап построения интерпретации. Шестой этап — этап расчета интерпретации. На основе результатов идентификации (detections), и с использованием информации о соответствии зашумленных участков изображению (state), рассчитыва-

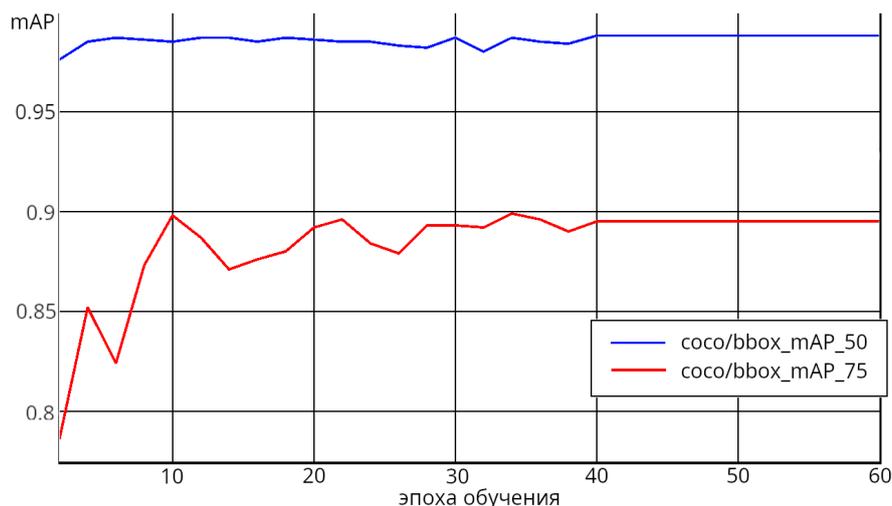


Рис. 7. Кривые обучения модели идентификации объектов в эксперименте в зависимости от пройденных эпох (по горизонтали); coco/bbox_mAP_50 — средняя точность при пороге 50%, coco/bbox_mAP_75 — средняя точность при пороге 75%

ются области, наиболее сильно влияющие на точность работы модели (interpretation). Для построения визуализации используется само исходное изображение (image) и параметры визуализации (vis_params).

Седьмой этап — сбор данных для включения в отчет (compile). В качестве входных данных этот этап принимает исходное изображение (image), результат идентификации объекта на исходном изображении (detection), изображение, выбранное на этапе применения модели (selected_image), результат идентификации на нем (selected_detection), рабочую директорию обучения модели (work_dir) и интерпретацию работы модели на исходном изображении (interpretation). На данном этапе создается визуализация идентификации объектов на двух входных изображениях и формируются таблицы с показателями точности обучения. Все эти данные передаются в этап построения отчета.

Восьмой этап — построение отчета с помощью поставляемой в составе фреймворка расширенной версии библиотеки mldev-reporting [14]. Из этапа сбора данных передаются два исходных изображения, два изображения с идентификацией объектов, изображение с интерпретацией, таблица финальных показателей качества и таблица изменения показателей в процессе обучения. Также в этот этап передается шаблон отчета, на основе которого в ходе этапа формируется финальный отчет о запуске эксперимента в виде веб-сайта (report).

4.3. Результаты и анализ эффективности

Для демонстрации работы и проведения измерений конвейер обучения исполнен со следующими параметрами: 60 эпох обучения, 8 изображений в одном минибатче, начальный шаг обучения 0.004. Полученные кривые обучения приведены на рис. 7. Значения показателей качества составили $\text{coco/mAP}_{0.5} = 0.976$, $\text{coco/mAP}_{0.75} = 0.786$.

Для анализа эффективности использования вычислительных ресурсов платформы воспользуемся методов эталонных систем [28]. В качестве эталона рассмотрим гипотетический способ организации расчетов на тех же вычислительных ресурсах, при котором пользовательская задача задействует все доступные ресурсы без накладных расходов и увеличивает

объем решаемой задачи за то же время. Для данного расчета считаем задачу произвольно делимой между вычислителями.

Обозначим через p_{ijt} доступную мощность вычислительного ресурса j для решения задачи i в интервал времени t продолжительности Δ_t . Пусть в тот же момент времени только доля c_{ijt} этого выделенного ресурса задействована для расчетов, остальная часть не используется или составляет накладные расходы. Тогда эффективность E решения группы вычислительных задач по отношению к рассмотренной эталонной модели составит

$$E = \frac{\sum_{ijt} p_{ijt} c_{ijt} \Delta_t}{\sum_{ijt} p_{ijt} \Delta_t}. \quad (1)$$

Данную эффективность можно интерпретировать, как отношение выполненной полезной работы в выделенные интервалы времени к теоретически возможному объему работы.

В эксперименте для выполнения этапов detect, explain, inference, predict, prepare, compile, report конвейера расчета использован узел с одним vCPU ядром Intel® Xeon® Gold 6338 (примем $p_{ijt} = 1$) и ускорителем NVIDIA® Tesla® T4 с 16 ГБ RAM (оценочно при экспериментальном сравнении, $p_{ijt} \approx 10$), и на вычислительном кластере sHARISMa для этапа train использована выделенная Slurm часть узла с четырьмя ядрами CPU (без hyper-threading) на Intel Xeon Gold 6152 2.1–3.7 GHz (оценочно, $p_{ijt} \approx 1$) и одной NVIDIA Tesla V100 32 GB (оценочно, $p_{ijt} \approx 40$). Для этапа сборки build использован ПК разработчика с восемью vCPU ядрами Intel i5-8300H (примерно $p_{ijt} \approx 1$). Доступная пропускная способность коммуникационной сети между узлами и с головным узлом кластера не ограничивалась и не измерялась. Результаты расчетов показателей эффективности приведены в табл. 1, названия этапов соответствуют схеме на рис. 3. Указанное в таблице время расчета измеряется от момента выделения вычислительных ресурсов задаче и до их освобождения.

Таблица 1. Эффективность E_i использования вычислительных ресурсов на этапе расчета i

Этап i	E_i	E_i^{cpu}	p_{it}^{cpu}	E_i^{gpu}	p_{it}^{gpu}	Время, с.
(build)	0.00	0.00	≈ 8	0.00	0	833
train	0.57	0.70	4	0.56	≈ 40	12240
(charisma-stage)	0.00	0.00	0.5	0.00	0	13053
detect	0.08	0.93	1	0.00	≈ 10	13
explain	0.04	0.41	1	0.00	≈ 10	5
inference	0.05	0.58	1	0.00	≈ 10	22
predict	0.37	0.96	1	0.31	≈ 10	69
prepare	0.04	0.48	1	0.00	≈ 10	241
compile	0.02	0.27	1	0.00	≈ 10	3
report	0.02	0.27	1	0.00	≈ 10	5

Учитывая весьма приблизительные оценки сравнительной производительности использованных вычислителей и сделанные предположения о делимости задачи, получаемое оценочное значение эффективности всего расчета лежит около $E \approx 0.55..0.56$ при вариации оценочных значений производительности вычислительных ресурсов от 67% до 150% и составляет примерно 97% от эффективности расчетов на этапе обучения.

Анализируя результаты расчетов, следует отметить следующее. Первое, использование постоянно работающего процесса для отслеживания поставленной в очередь задачи, обозначенного в таблице `charisma-stage`, не вносит существенного вклада в эффективность всего конвейера. Сборка и подготовка эксперимента `build` средствами библиотеки тегов фреймворка на ПК разработчика также не вносит существенного вклада в эффективность. Второе, подключение вычислительного кластера для запуска коротких задач менее 10 минут может быть неэффективным. В этом случае предпочтительно использовать облачные вычислительные ресурсы центра обработки данных.

Тем не менее, среди способов дальнейшего повышения эффективности можно выделить следующие:

- Расчет оценок эффективности в зависимости от используемых ресурсов для запускаемых задач в помощь разработчикам при планировании их использования на вычислительном кластере.
- Разделение исполнения этапов между разными узлами для исключения простоя высокопроизводительных ресурсов, когда они не запрашиваются задачей этапа.

Заключение

В работе рассмотрена актуальная проблема создания технологии создания программных средств интеллектуального анализа и обработки данных для решения отдельных прикладных задач. Исследованы задачи применения и автоматизации процесса MLOps на базе единой гибридной облачной вычислительной платформы, в том числе интеграции разнородных облачных, локальных и высокопроизводительных вычислительных ресурсов для проведения распределенных экспериментов, реализации конвейеров настройки моделей машинного обучения, решения задачи обработки и анализа данных

Основные результаты работы состоят в создании на основе модельно-ориентированного подхода (MDSD) расширяемой объектной модели интеллектуальных приложений, ее реализация с применением библиотеки `MLDev` предоставления пользователям возможности создания собственных конвейеров машинного обучения, выполняемых в распределенной среде. Для описания структуры и конфигурации интеллектуальных приложений разработан формальный язык на основе `YAML` и разработан программный фреймворк, управляющий исполнением приложений в инфраструктуре `Kubernetes`.

На примере задачи классификации и детектирования объектов на изображениях продемонстрирована возможность расширения пользователями объектной модели и создание сложных многоэтапных конвейеров расчетов прозрачным для пользователя образом. По результатам предварительных экспериментов и расчетов эффективности, вносимые MLOps фреймворком дополнительные накладные расходы невелики по сравнению с затратами на обучение расчетных моделей. В настоящее время разработанный фреймворк используется в MLOps платформе НИУ ВШЭ.

Благодарности

Исследование выполнено с использованием суперкомпьютерного комплекса НИУ ВШЭ [29].

Литература

1. Кореньков В. Грид-технологии: статус и перспективы // Вестник Международной академии наук. Русская секция. 2010. № 1. С. 41–44. DOI: 10.3997/2214-4609.20142827.
2. Pimenov A., Fedorov I., Bezzateev S. Fog computing architecture using blockchain technology // Information and Control Systems. 2022. Oct. No. 5. P. 40–48. DOI: 10.31799/1684-8853-2022-5-40-48.
3. Sukhoroslov O.V., Afanasiev A. Everest: A Cloud Platform for Computational Web Services // CLOSER. 2014. P. 411–416. DOI: 10.5220/0004941404110416.
4. Центр искусственного интеллекта – Национальный исследовательский университет «Высшая школа экономики». 2024. URL: <https://cs.hse.ru/aicenter/>.
5. Antonenko V., Chupakhin A., Kolosov A., *et al.* On HPC and Cloud Environments Integration // Performance Evaluation Models for Distributed Service Networks. Springer, 2021. P. 159–185. DOI: 10.1007/978-3-030-67063-4_8.
6. Ejarque J., Badia R.M., Albertin L., *et al.* Enabling dynamic and intelligent workflows for HPC, data analytics, and AI convergence // Future generation computer systems. 2022. Vol. 134. P. 414–429. DOI: 10.1016/j.future.2022.04.014.
7. Сухорослов О. Комбинированное использование высокопроизводительных ресурсов и Грид-инфраструктур в рамках облачной платформы Everest // Суперкомпьютерные дни в России. 2015. С. 706–711.
8. Велихов В.Е., Климентов А.А., Машинистов Р.Ю. и др. Интеграция гетерогенных вычислительных мощностей НИЦ «Курчатовский институт» для проведения масштабных научных вычислений // Известия Южного федерального университета. Технические науки. 2016. № 11 (184). С. 88–100.
9. Кутовский Н., Мицын В., Мошкин А. и др. Интеграция распределенных гетерогенных вычислительных ресурсов для эксперимента mpd с помощью DIRAC Interware // Физика элементарных частиц и атомного ядра. 2021. Т. 52, № 4. С. 999.
10. Feoktistov A.G., Sidorov I.A., Sergeev V.V., *et al.* Virtualization of heterogeneous HPC-clusters based on OpenStack platform // Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2017. Vol. 6, no. 2. P. 37–48. DOI: 10.14529/cmse170203.
11. Silva R.F.D., Badia R.M., Bard D., *et al.* Frontiers in scientific workflows: Pervasive integration with high-performance computing // Computer. 2024. Vol. 57, no. 8. P. 36–44. DOI: 10.1109/mc.2024.3401542.
12. Stubbs J., Cardone R., Packard M., *et al.* Tapis: An API platform for reproducible, distributed computational research // Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Vol. 1. Springer. 2021. P. 878–900. DOI: 10.1007/978-3-030-73100-7_61.
13. Воронцов К., Игловиков В., Стрижов В. и др. Проблемы проведения экспериментов и воспроизводимости исследований в науках о данных // Труды Московского физико-технического института. 2021. Т. 13, № 2 (50). С. 100–108. DOI: 10.53815/20726759_2021_13_2_100.

14. Khritankov A., Pershin N., Ukhov N., Ukhov A. MLDev: Data Science Experiment Automation and Reproducibility Software // International Conference on Data Analytics and Management in Data Intensive Domains. Springer. 2021. P. 3–18. DOI: 10.1007/978-3-031-12285-9_1.
15. Alam K., Roy B. Challenges of provenance in scientific workflow management systems // 2022 IEEE/ACM Workshop on Workflows in Support of Large-Scale Science (WORKS). IEEE. 2022. P. 10–18. DOI: 10.1109/works56498.2022.00007.
16. Dhruv A., Dubey A. Managing software provenance to enhance reproducibility in computational research // Computing in Science & Engineering. 2023. Vol. 25, no. 3. P. 60–65. DOI: 10.1109/mcse.2023.3314288.
17. Зыбин Р., Швецова В., Бадалян Д. и др. Облачная среда «Асперитас». 2022. Свидетельство о государственной регистрации программы для ЭВМ RU 2022682679.
18. Группин Д., Самоваров О., Хашба Э. SaaS платформа организации единой web-среды исследований, разработок и образования «Fanlight». 2018. Свидетельство о государственной регистрации программы для ЭВМ RU 2018615444.
19. Насонов Д., Бутаков Н., Бухановский А. и др. Технология организации управления и обработки больших данных – DataMall. 2020. Свидетельство о государственной регистрации программы для ЭВМ RU 2020664222.
20. Kreuzberger D., Kühl N., Hirschl S. Machine learning operations (MLOPS): Overview, definition, and architecture // IEEE Access. 2023. Vol. 11. P. 31866–31879. DOI: 10.1109/access.2023.3262138.
21. Тютляева Е.О., Одинцов И.О., Мармузов Г.В. и др. Тенденции развития вычислительных узлов современных суперкомпьютеров // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 3. С. 92–114. DOI: 10.14529/cmse190305.
22. Wilkinson M.D., Dumontier M., Aalbersberg I.J., *et al.* The FAIR Guiding Principles for scientific data management and stewardship // Scientific Data. 2016. Vol. 3, no. 1. P. 1–9. DOI: 10.1038/sdata.2016.18.
23. Kostenetskiy P., Shamsutdinov A., Chulkevich R., *et al.* HPC TaskMaster - Task Efficiency Monitoring System for the Supercomputer Center // Parallel Computational Technologies / ed. by L. Sokolinsky, M. Zymbler. Cham: Springer International Publishing, Jan. 2022. P. 17–29. DOI: 10.1007/978-3-031-11623-0_2.
24. Kostenetskiy P., Kozyrev V., Chulkevich R., Raimova A. Enhancement of the Data Analysis Subsystem in the Task-Efficiency Monitoring System HPC TaskMaster for the cHARISMa Supercomputer Complex at HSE University // Parallel Computational Technologies / ed. by L. Sokolinsky, M. Zymbler, V. Voevodin, J. Dongarra. Cham: Springer Nature Switzerland, 2024. P. 49–64. DOI: 10.1007/978-3-031-73372-7_4.
25. Lyu C., Zhang W., Huang H., *et al.* RTMDet: An Empirical Study of Designing Real-Time Object Detectors // CoRR. 2022. Vol. abs/2212.07784. DOI: 10.48550/ARXIV.2212.07784. arXiv: 2212.07784.
26. Слостников С., Чертова Э. Модуль синтеза моделей машинного зрения для детектирования объектов и действий. 2024. URL: https://cs.hse.ru/aicenter/rid_detection.

27. Слестников С., Чертова Э. Модуль синтеза моделей машинного зрения для детектирования объектов и действий. 2023. Свидетельство о государственной регистрации программы для ЭВМ RU 2023660157.
28. Хританков А.С. Метод анализа производительности распределенных приложений на основе эталонных моделей // Параллельные вычислительные технологии (ПаВТ'2011). 2011. С. 343–354.
29. Kostenetskiy P., Chulkevich R., Kozyrev V. HPC Resources of the Higher School of Economics // Journal of Physics: Conference Series. 2021. Jan. Vol. 1740, no. 1. P. 012050. DOI: 10.1088/1742-6596/1740/1/012050.

Хританков Антон Сергеевич, к.ф.-м.н., факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

Полежаев Валентин Александрович, факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

Жуликов Григорий Александрович, факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

Галынчик Максим Сергеевич, факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

Климин Никита Андреевич, отдел разработки программных систем, МИЭМ ВШЭ (Москва, Российская Федерация)

Сахаров Кирилл Евгеньевич, отдел разработки программных систем, МИЭМ ВШЭ (Москва, Российская Федерация)

Минченков Виктор Олегович, отдел разработки программных систем, МИЭМ ВШЭ (Москва, Российская Федерация)

Спирин Иван Вячеславович, факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

Крупнов Иван Игоревич, факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

Якушева Софья Федоровна, к.т.н., факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

Маратканова Александра Сергеевна, факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

Козырев Вячеслав Иванович, отдел суперкомпьютерного моделирования, НИУ ВШЭ (Москва, Российская Федерация)

Костенецкий Павел Сергеевич, к.ф.-м.н., доцент, отдел суперкомпьютерного моделирования, НИУ ВШЭ (Москва, Российская Федерация)

Салех Хади Мухаммед, к.т.н., факультет компьютерных наук, НИУ ВШЭ (Москва, Российская Федерация)

DISTRIBUTED COMPUTATIONAL EXPERIMENTS IN THE MLOPS PLATFORM OF HSE UNIVERSITY

© 2025 A.S. Khritankov, V.A. Polezhaev, G.A. Zhulikov, M.S. Halynchik,
N.A. Klimin, K.E. Sakharov, V.O. Minchenkov, I.V. Spirin, I.I. Krupnov,
S.F. Yakusheva, A.S. Maratkanova, V.I. Kozyrev, P.S. Kostenetskiy, H.M. Salekh

National Research University Higher School of Economics

(11 Pokrovskiy blvd., Moscow, 109028 Russia)

E-mail: akhritankov@hse.ru

Received: 19.05.2025

Despite the wide spread and successful application of data mining and processing tools for solving individual applied problems, the problem of developing a technology for creating such software tools has not yet been solved. In the context of a unified MLOps process for creating machine learning technologies, this paper considers the emerging problems of automating and executing distributed computing experiments on a hybrid cloud computing platform. The MLOps platform being developed at HSE University is designed to deploy intelligent services and data analysis software. The platform shall manage heterogeneous resources available locally and in the cloud environment and combine them with the resources of the HSE cHARISMa computing cluster managed with Slurm. Thus, relevant is the problem of integrating the specified resources for conducting computational experiments, implementing pipelines for setting up machine learning models, solving problems of data processing and analysis. The features of the problem being solved are the consideration of the computation process as an integral part of the technology for creating intelligent services, the need for using heterogeneous resources for this technology, and the use of the hybrid platform for the execution of computations. The paper proposes a solution to the problem of integrating computations and presents the results of testing the solution for intelligent services. We show the feasibility of such integration of heterogeneous resources in the same computational experiment based on an object model of the experiment extended by the user and a domain-specific language for its specification, and resolve the issues of dynamic management of the deployment of intelligent applications, integration of data processing pipelines, services and data sets for performing distributed computational experiments.

Keywords: distributed computing experiments, machine learning, cloud technologies, MLOps.

FOR CITATION

Khritankov A.S., Polezhaev V.A., Zhulikov G.A., *et al.* Distributed Computational Experiments in the MLOps Platform of HSE University. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2025. Vol. 14, no. 2. P. 42–66. (in Russian) DOI: 10.14529/cmse250203.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Korenkov V. GRID technologies: status and perspectives. Herald of the International Academy of Science. Russian Section. 2010. No. 1. P. 41–44. (in Russian) DOI: 10.3997/2214-4609.20142827.
2. Pimenov A., Fedorov I., Bezzateev S. Fog computing architecture using blockchain technology. Information and Control Systems. 2022. Oct. No. 5. P. 40–48. (in Russian) DOI: 10.31799/1684-8853-2022-5-40-48.

3. Sukhoroslov O.V., Afanasiev A. Everest: A Cloud Platform for Computational Web Services.. CLOSER. 2014. P. 411–416. DOI: 10.5220/0004941404110416.
4. Centre of Artificial Intelligence – HSE University. 2024. URL: <https://cs.hse.ru/aicenter/> (in Russian).
5. Antonenko V., Chupakhin A., Kolosov A., *et al.* On HPC and Cloud Environments Integration. Performance Evaluation Models for Distributed Service Networks. Springer, 2021. P. 159–185. DOI: 10.1007/978-3-030-67063-4_8.
6. Ejarque J., Badia R.M., Albertin L., *et al.* Enabling dynamic and intelligent workflows for HPC, data analytics, and AI convergence. Future generation computer systems. 2022. Vol. 134. P. 414–429. DOI: 10.1016/j.future.2022.04.014.
7. Sukhoroslov O. Combined use of high-performance resources and Grid infrastructures within the Everest cloud platform. Supercomputer Days in Russia. 2015. P. 706–711. (in Russian).
8. Velikhov V., Klimentov A., Mashinistov R., *et al.* Integration of heterogeneous computing resources at NRI “Kurchatov Institute” for large-scale scientific computations. Izvestiya SFedU. Engineering Sciences. 2016. No. 11 (184). P. 88–100. (in Russian).
9. Kutovskiy N., Mitsyn V., Moshkin A., *et al.* Integration of distributed heterogeneous computing resources for the mpd experiment with DIRAC Interware. Physics of Particles and Nuclei. 2021. Vol. 52, no. 4. P. 999. (in Russian).
10. Feoktistov A.G., Sidorov I.A., Sergeev V.V., *et al.* Virtualization of heterogeneous HPC-clusters based on OpenStack platform. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2017. Vol. 6, no. 2. P. 37–48. (in Russian) DOI: 10.14529/cmse170203.
11. Silva R.F.D., Badia R.M., Bard D., *et al.* Frontiers in scientific workflows: Pervasive integration with high-performance computing. Computer. 2024. Vol. 57, no. 8. P. 36–44. DOI: 10.1109/mc.2024.3401542.
12. Stubbs J., Cardone R., Packard M., *et al.* Tapis: An API platform for reproducible, distributed computational research. Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Vol. 1. Springer. 2021. P. 878–900. DOI: 10.1007/978-3-030-73100-7_61.
13. Vorontsov K., Iglonikov V., Strijov V., *et al.* Roundtable: Challenges in repeatable experiments and reproducible research in data science. Proceedings of Moscow Institute of Physics and Technology. 2021. Vol. 13, no. 2 (50). P. 100–108. (in Russian) DOI: 10.53815/20726759_2021_13_2_100.
14. Khritankov A., Pershin N., Ukhov N., Ukhov A. MLDev: Data Science Experiment Automation and Reproducibility Software. International Conference on Data Analytics and Management in Data Intensive Domains. Springer. 2021. P. 3–18. DOI: 10.1007/978-3-031-12285-9_1.
15. Alam K., Roy B. Challenges of provenance in scientific workflow management systems. 2022 IEEE/ACM Workshop on Workflows in Support of Large-Scale Science (WORKS). IEEE. 2022. P. 10–18. DOI: 10.1109/works56498.2022.00007.
16. Dhruv A., Dubey A. Managing software provenance to enhance reproducibility in computational research. Computing in Science & Engineering. 2023. Vol. 25, no. 3. P. 60–65. DOI: 10.1109/mcse.2023.3314288.

17. Zybin R., Shvetsova V., Badalyan D., *et al.* Cloud environment “Asperitas”. 2022. (in Russian). Certificate of state registration of a computer program RU 2022682679.
18. Grushin D., Samovarov O., Hashba E. SaaS platform for organizing a unified web environment for research, development and education “Fanlight”. 2018. (in Russian). Certificate of state registration of a computer program RU 2018615444.
19. Nasonov D., Butakov N., Bukhanovsky A., *et al.* Technology for organizing management and processing big data – DataMall. 2020. (in Russian). Certificate of state registration of a computer program RU 2020664222.
20. Kreuzberger D., Kühn N., Hirschl S. Machine learning operations (MLOPS): Overview, definition, and architecture. *IEEE Access*. 2023. Vol. 11. P. 31866–31879. DOI: 10.1109/access.2023.3262138.
21. Tyutlyaeva E.O., Odintsov I.O., Marmuzov G.V., *et al.* Development trends of modern supercomputers. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2019. Vol. 8, no. 3. P. 92–114. (in Russian) DOI: 10.14529/cmse190305.
22. Wilkinson M.D., Dumontier M., Aalbersberg I.J., *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*. 2016. Vol. 3, no. 1. P. 1–9. DOI: 10.1038/sdata.2016.18.
23. Kostenetskiy P., Shamsutdinov A., Chulkevich R., *et al.* HPC TaskMaster - Task Efficiency Monitoring System for the Supercomputer Center. *Parallel Computational Technologies / ed. by L. Sokolinsky, M. Zymbler*. Cham: Springer International Publishing, Jan. 2022. P. 17–29. DOI: 10.1007/978-3-031-11623-0_2.
24. Kostenetskiy P., Kozyrev V., Chulkevich R., Raimova A. Enhancement of the Data Analysis Subsystem in the Task-Efficiency Monitoring System HPC TaskMaster for the cCHARISMa Supercomputer Complex at HSE University. *Parallel Computational Technologies / ed. by L. Sokolinsky, M. Zymbler, V. Voevodin, J. Dongarra*. Cham: Springer Nature Switzerland, 2024. P. 49–64. DOI: 10.1007/978-3-031-73372-7_4.
25. Lyu C., Zhang W., Huang H., *et al.* RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *CoRR*. 2022. Vol. abs/2212.07784. DOI: 10.48550/ARXIV.2212.07784. arXiv: 2212.07784.
26. Slastnikov S., Chertova E. Machine vision model synthesis module for object and action detection. 2024. URL: https://cs.hse.ru/aicenter/rid_detection (in Russian).
27. Slastnikov S., Chertova E. A program for synthesis of machine vision models to detect objects and activities. 2023. (in Russian). Certificate of state registration of a computer program RU 2023660157.
28. Khritankov A.S. A method for performance analysis of distributed applications based on reference models. *Parallel Computational Technologies (PCT’2011)*. 2011. P. 343–354. (in Russian).
29. Kostenetskiy P., Chulkevich R., Kozyrev V. HPC Resources of the Higher School of Economics. *Journal of Physics: Conference Series*. 2021. Jan. Vol. 1740, no. 1. P. 012050. DOI: 10.1088/1742-6596/1740/1/012050.

СУПЕРКОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СВЕРХЗВУКОВОГО ТЕЧЕНИЯ ВОКРУГ ТЕЛ ВРАЩЕНИЯ РАЗЛИЧНОЙ ФОРМЫ*

© 2025 С.В. Поляков, В.О. Подрыга, Н.И. Тарасов

Институт прикладной математики им. М.В. Келдыша РАН

(125047 Москва, Миусская пл., д. 4)

E-mail: polyakov@imamod.ru, pvictoria@list.ru, nikita_tarasov@imamod.ru

Поступила в редакцию: 18.04.2025

В работе представлены численный подход и его параллельная программная реализация для исследования процессов обтекания твердых тел сложной геометрии сверхзвуковыми потоками газа. Цель исследования состояла в анализе эффективности численных схем на неструктурированных гибридных сетках, аппроксимирующих квазигазодинамические (КГД) уравнения. В качестве примера была выбрана задача обтекания тел вращения различной формы. Газовая среда представлена сухим воздухом. Система КГД уравнений рассматривалась в однокомпонентной постановке. Она дополнялась уравнениями состояния идеального газа и зависимостями кинетических коэффициентов от температуры и давления. В работе исследовались зависимость результатов от параметров численного метода и особенности параллельной реализации. В первом случае было проанализировано влияние параметров регуляризации, используемых в КГД подходе. Во втором случае анализировались различные варианты распараллеливания. В итоге этих исследований были предложены оптимальные значения вышеуказанных параметров и выявлены зависимости эффективности от алгоритма распараллеливания. В численных экспериментах были рассмотрены осесимметричные течения газа вокруг цилиндра, эллипсоида и составного тела. Расчеты проводились для трех значений скорости входного потока. Полученные результаты подтвердили корректность разработанной вычислительной технологии.

Ключевые слова: течения газа вблизи твердых тел, тела вращения разной формы, квазигазодинамические уравнения, численные методы, параллельные алгоритмы, суперкомпьютерные вычисления.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Поляков С.В., Подрыга В.О., Тарасов Н.И. Суперкомпьютерное моделирование сверхзвукового течения вокруг тел вращения различной формы // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2025. Т. 14, № 2. С. 67–85. DOI: 10.14529/cmse250204.

Введение

Работа посвящена исследованиям в области прикладной газовой динамики [1]. К настоящему моменту имеется множество теоретических и практических подходов к решению задач прикладной газовой динамики компьютерными методами [2–8]. Один из таких подходов развивается с 1980-х годов и базируется на квазигазодинамической (КГД) системе уравнений [9–18]. Данная система является обобщением (регуляризацией) уравнений Навье—Стокса и в последнее время приобрела определенную популярность при расчетах сверхзвуковых течений разреженного газа в областях сложной геометрии [13, 16]. Основное преимущество КГД системы состоит в возможности проведения расчетов при числах Кнудсена вплоть до 1, а также при больших и очень больших числах Маха в условиях сильно разреженной газовой среды. Также система КГД достаточно просто обобщается на

*Статья рекомендована к публикации программным комитетом Всероссийской научной конференции с международным участием «Параллельные вычислительные технологии (ПаВТ) 2025».

случай газовых смесей. При ее численной реализации не приходится использовать сложный аппарат лимиттеров. Явная по времени численная схема достаточно просто распараллеливается. При ее использовании реализуется конечная скорость распространения возмущений газового потока, соответствующая физике процессов обтекания.

В настоящей работе исследуется численная реализация КГД подхода в случае обтекания тел вращения. Данная тематика востребована со времени запуска первых спутников в космос. В настоящее время возобновился интерес к разработке многоцелевых возвращаемых космических аппаратов различных конструкций и назначения. Среди проблем проектирования таких аппаратов имеется множество вопросов, решаемых на уровне предпроектного математического моделирования. Одной из актуальных проблем остается режим обтекания аппарата в атмосфере Земли. Ранее авторами рассматривались подобные задачи в контексте использования гибридных неструктурированных сеток для расчета газодинамических течений вокруг возвращаемых летательных аппаратов [19, 20]. Однако форма этих тел была затупленной и не имела острых углов. В данном исследовании рассмотрены различные геометрии, в том числе с острыми углами. Основной задачей было установить диапазоны расчетных параметров дискретной КГД модели и детали параллельной реализации используемой численной методики.

Статья организована следующим образом. В разделе 1 представлена постановка задачи. Раздел 2 посвящен рассмотрению основных уравнений и уравнений связи. В разделе 3 описаны используемые в работе граничные и начальные условия. В разделе 4 приведены численный алгоритм и программная реализация разработанной математической модели. В разделе 5 представлены результаты моделирования и их анализ. В заключении приводится краткий обзор результатов, полученных в работе, а также указаны направления дальнейших исследований.

1. Постановка модельной задачи

Рассмотрим прямоугольную расчетную область Ω в (R, Z) -геометрии (рис. 1). Внутри области расположен контур тела вращения. Во внешности контура пространство изначально заполнено покоящимся газом с плотностью ρ_0 , давлением p_0 , температурой T_0 . Предполагается, что тело вращения абсолютно твердое. Теплообмен тела с окружающим его газом не учитывается ввиду наличия теплоизоляции на поверхности тела. Слева в расчетную область входит газовый поток, отличающийся от покоящегося газа лишь скоростью u_{in} .

Размеры расчетной области Ω — $R_1 \times (L_1 + L_2 + L_3)$. Твердое тело представлено следующими тремя формами:

1. половиной центрального сечения прямоугольного цилиндра длины L_2 и ширины R_2 (см. рис. 1а);
2. половиной эллипсоида с диаметрами L_2 и $2R_2$ (см. рис. 1б);
3. половиной центрального сечения прямоугольного цилиндра длины $(L_2 - L_4)$, дополненного слева половиной сечения конуса с высотой L_4 и радиусом R_2 (см. рис. 1в).

Целью анализа выбранной модельной задачи является анализ качества численного подхода и его параллельной программной реализации.

2. Основные уравнения

Анализ поставленной задачи проведем для случая однокомпонентного газа, соответствующего по параметрам воздушной среде. Для описания его состояния будем использо-

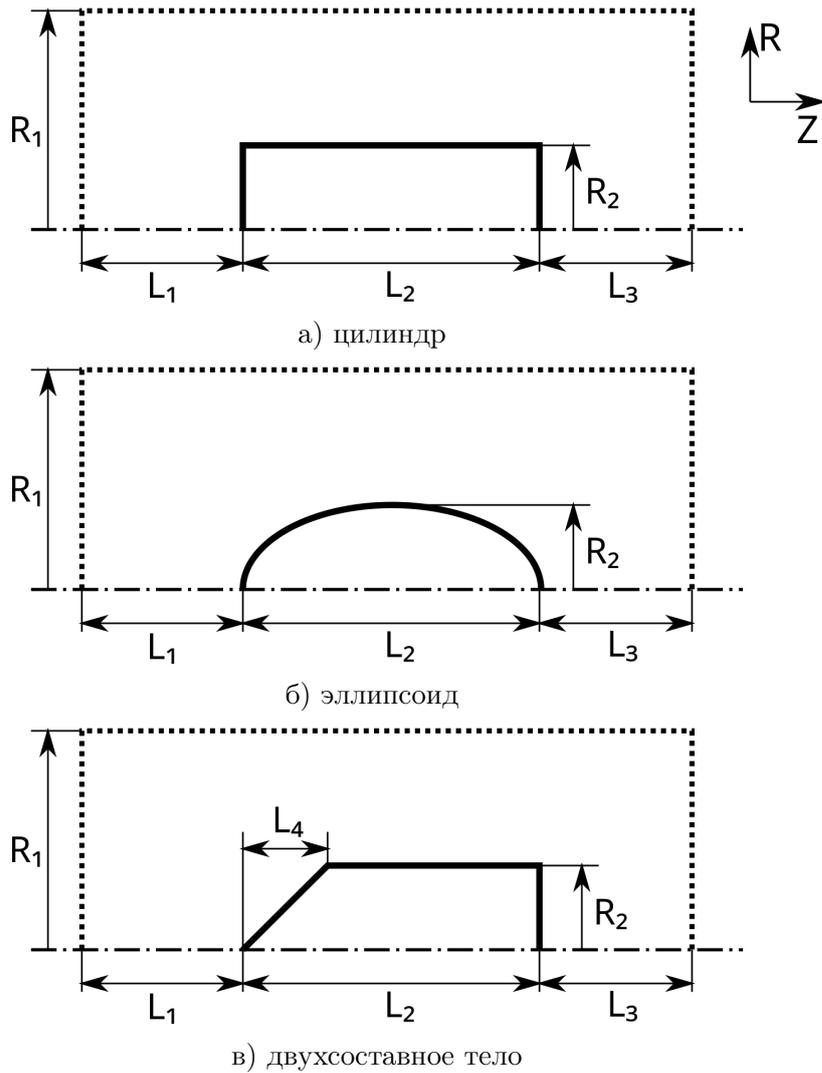


Рис. 1. Расчетная геометрия для трех форм тел вращения

вать квазигазодинамическую систему уравнений [11]. В безразмерных переменных в декартовой системе координат эта система вместе с уравнениями состояния и связей запишется следующим образом:

$$\frac{\partial \rho}{\partial t} + \operatorname{div} \mathbf{W}^{(\rho)} = 0, \quad (1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \operatorname{div} \mathbf{W}^{(I)} = 0, \quad (2)$$

$$\frac{\partial E}{\partial t} + \operatorname{div} \mathbf{W}^{(E)} = 0, \quad (3)$$

$$E = \rho |\mathbf{u}|^2 / 2 + \varepsilon, \quad \varepsilon = p / (\gamma - 1), \quad H = (E + p) / \rho, \quad p = \rho T / \gamma, \quad (4)$$

$$\mu = \tau p \operatorname{Sc}, \quad \chi = \tau p \operatorname{Sc} / (\operatorname{Pr} (\gamma - 1)), \quad \tau = \operatorname{Ma} T^\omega / (\operatorname{Re} p \operatorname{Sc}) + \alpha h / c, \quad c = \sqrt{T}.$$

Здесь предполагается, что газовая среда характеризуется массовой плотностью ρ , температурой T , давлением p и скоростью \mathbf{u} . Эти параметры нормированы соответственно на ρ_0 , T_0 , p_0 . Вектор скорости \mathbf{u} нормирован на скорость звука в газе при температуре T_0 и давлении p_0 . Также в уравнениях (1)–(3) используются стандартные обозначения для частной производной по времени $\partial/\partial t$ и оператора дивергенции div в декартовых координатах.

Другие параметры газа используются в уравнениях связи (4). Это соответственно E — плотность полной энергии, H — энтальпия и ε — внутренняя энергия. Также в (4) используются $\mu = \mu(T, p)$ и $\chi = \chi(T, p)$ — коэффициенты динамической вязкости и теплопроводности, $\tau = \tau(T, p)$ — характерное время релаксации газовой среды к равновесному состоянию, введенное в рамках КГД подхода.

Параметрами модели являются: γ , α , h , c , ω — показатель адиабаты, положительный коэффициент регуляризации, характерный пространственный масштаб задачи (часто совпадает с шагом сетки при дискретизации), безразмерная местная скорость звука, показатель степени температурной зависимости в формулах для μ , χ , τ ; Pr, Sc, Re и Ma — числа Прандтля, Шмидта, Рейнольдса и Маха.

Векторы $\mathbf{W}^{(\rho)}$, $\mathbf{W}^{(E)}$ и тензор $\mathbf{W}^{(I)}$ с точностью до знака совпадают с векторами потоков массовой плотности и плотности полной энергии и тензором потока плотности импульса. Они имеют следующий вид:

$$\begin{aligned} \mathbf{W}^{(\rho)} &= \rho(\mathbf{u} - \mathbf{w}), \quad \mathbf{W}^{(I)} = \rho(\mathbf{u} - \mathbf{w}) \otimes \mathbf{u} + p\mathbf{I} - \mathbf{\Pi}, \\ \mathbf{W}^{(E)} &= H\rho(\mathbf{u} - \mathbf{w}) + \mathbf{q} - \mathbf{\Pi}\mathbf{u}, \end{aligned} \quad (5)$$

$$\begin{aligned} \mathbf{\Pi} &= \mathbf{\Pi}^{NS} + \mathbf{\Pi}^{QGD}, \quad \mathbf{\Pi}^{NS} = \{\Pi_{ij}^{NS}; i, j = X, Y, Z\}, \\ \Pi_{ij}^{NS} &= \mu(\partial u_j / \partial i + \partial u_i / \partial j) - 2\mu \operatorname{div} \mathbf{u} / 3, \end{aligned} \quad (6)$$

$$\begin{aligned} \mathbf{\Pi}^{QGD} &= \rho \mathbf{u} \otimes \hat{\mathbf{w}} + \tau [\mathbf{u} \nabla p + \rho c^2 \operatorname{div} \mathbf{u}] \mathbf{I}, \\ \mathbf{q} &= -\chi \nabla T - \tau [\rho (\mathbf{u} \nabla \varepsilon + p (\mathbf{u} \nabla) (1/\rho))] \mathbf{u}, \end{aligned} \quad (7)$$

$$\mathbf{w} = \frac{\tau}{\rho} (\operatorname{div} (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p), \quad \hat{\mathbf{w}} = \frac{\tau}{\rho} [\rho (\mathbf{u} \nabla) \mathbf{u} + \nabla p]. \quad (8)$$

Здесь \mathbf{w} и $\hat{\mathbf{w}}$ — КГД поправки к вектору скорости, \otimes — знак прямого векторного произведения, \mathbf{I} — единичный тензор, $\mathbf{\Pi}$ — тензор вязких напряжений, состоящий из двух слагаемых $\mathbf{\Pi}^{NS}$ и $\mathbf{\Pi}^{QGD}$, где первое совпадает с тензором Навье—Стокса, а второе появляется в результате применения КГД подхода.

Далее уравнения (1)–(3) и дополнительные связи (4)–(8) будем использовать для решения задачи обтекания твердого тела вращения в случае цилиндрической (R, Z) -геометрии расчетной области с учетом аксиальной симметрии тела (см. рис. 1).

3. Граничные и начальные условия

Система уравнений (1)–(3) со связями (4)–(8) дополняется граничными условиями на поверхности обтекаемого тела, а также традиционными граничными условиями на входе в среду и на свободных границах [11]. Они формулируются следующим образом.

- На левой границе области задаются параметры входного потока:
 $\rho = 1$, $p = 1/\gamma$, $T = 1$, $u_R = 0$, $u_Z = Ma$;
- на верхней и правой границах области задаются условия свободного выхода:
 $\partial \rho / \partial n = 0$, $\partial u_j / \partial n = 0$, $\partial E / \partial n = 0$, $j = R, Z$;
- на оси симметрии ($R = 0$) задаются условия:
 $\partial \rho / \partial n = 0$, $u_R = 0$, $\partial u_Z / \partial n = 0$, $\partial E / \partial n = 0$;
- на поверхности объекта задаются условия прилипания при нулевом тепловом потоке:
 $\partial \rho / \partial n = 0$, $u_j = 0$, $\partial E / \partial n = 0$, $j = R, Z$.

В качестве начальных условий будем использовать условия покоя среды:
 $\rho = 1$, $u_j = 0$, $p = 1/\gamma$, $T = 1$, $j = R, Z$.

4. Численный алгоритм и программная реализация

Численная реализация уравнений (1)–(3) с учетом (4)–(8) базируется на применении сеточного метода конечных объемов [21, 22] на неструктурированных сетках с различной формой ячеек. В данном случае мы выбрали ячейки треугольной формы. По временной переменной используются явные разностные схемы. Расчеты задачи проводились на треугольной пространственной сетке Ω_h с параметрами N_P , N_E , N_C , обозначающими число точек, число ребер и число треугольных ячеек соответственно. Сетка по времени ω_t характеризуется шагом Δt и количеством шагов N_t .

Общая численная схема имела следующий вид:

$$\hat{\rho}_h = \rho_h - \Delta t \sum_k \left(\mathbf{w}_{h,k}^{(\rho)}, \mathbf{n}_k \right), \quad (9)$$

$$\hat{\rho}_h \hat{\mathbf{u}}_h = \rho_h \mathbf{u}_h - \Delta t \sum_k \left(\mathbf{w}_{h,k}^{(I)}, \mathbf{n}_k \right), \quad (10)$$

$$\hat{E}_h = E_h - \Delta t \sum_k \left(\mathbf{w}_{h,k}^{(E)}, \mathbf{n}_k \right). \quad (11)$$

Здесь введены сеточные аналоги массовой плотности, плотности энергии и вектора скорости, векторов и тензора потоков, обозначаемые нижним индексом h , а также внешние нормали \mathbf{n}_k к ребрам контрольных объемов, сформированных вокруг точек сетки Ω_h . Суммирование ведется по всем ребрам контрольных объемов, включая границу расчетной области $\partial\Omega$. Более подробное описание этой части численной методики приведено в работе [19].

Шаг по времени в общем случае Δt является переменным и определяется на каждом слое по критерию устойчивости:

$$\Delta t \leq \beta h_{min}/Ma, \quad (12)$$

где h_{min} — минимальный размер ребер треугольников, β — коэффициент из полуинтервала $(0, 1]$. Вместе с коэффициентом регуляризации α эти два параметра являются управляющими сходимостью предложенной численной схемы.

В табл. 1 и на рис. 2 представлены базовые расчетные конфигурации и фрагменты использованных треугольных сеток. Во всех конфигурациях $R_1 = 5$, $R_2 = 0.5$, $L_1 = 1.5$, $L_2 = 3$, $L_3 = 1.5$. В конфигурациях 3–5 величина L_4 определяется углом наклона φ образующей конуса относительно оси симметрии: $L_4 = R_2 \text{ctg}(\varphi)$ и составляет соответственно 0.289, 0.5, 0.866. Треугольные сетки построены с помощью свободно распространяемого пакета Gmsh [23]. Длины ребер построенных треугольников во всех конфигурациях лежат в диапазоне $[0.01, 0.015]$.

Параллельная программная реализация итогового численного алгоритма основана на методе декомпозиции расчетной области [24] на компактные домены примерно одинаковой вычислительной емкости. Это достигается путем распределения ячеек сетки по вычислителям с помощью геометрического подхода и последующей коррекции в рамках использования алгоритма динамической балансировки загрузки [25].

Для расчета конфигураций на сетках умеренного объема (порядка 1 млн. ячеек) распараллеливание производилось с помощью стандарта OpenMP [26]. При использовании двух-

Таблица 1. Базовые расчетные конфигурации

№	Геометрия	Число узлов	Число ребер	Число ячеек
1	Круговой цилиндр	168 970	2 021 484	336 402
2	Эллипсоид	170 578	2 040 960	339 663
3	Составное тело, $\varphi = 60^\circ$	169 332	2 025 884	337 140
4	Составное тело, $\varphi = 45^\circ$	170 023	2 034 196	338 527
5	Составное тело, $\varphi = 30^\circ$	170 537	2 040 384	339 560

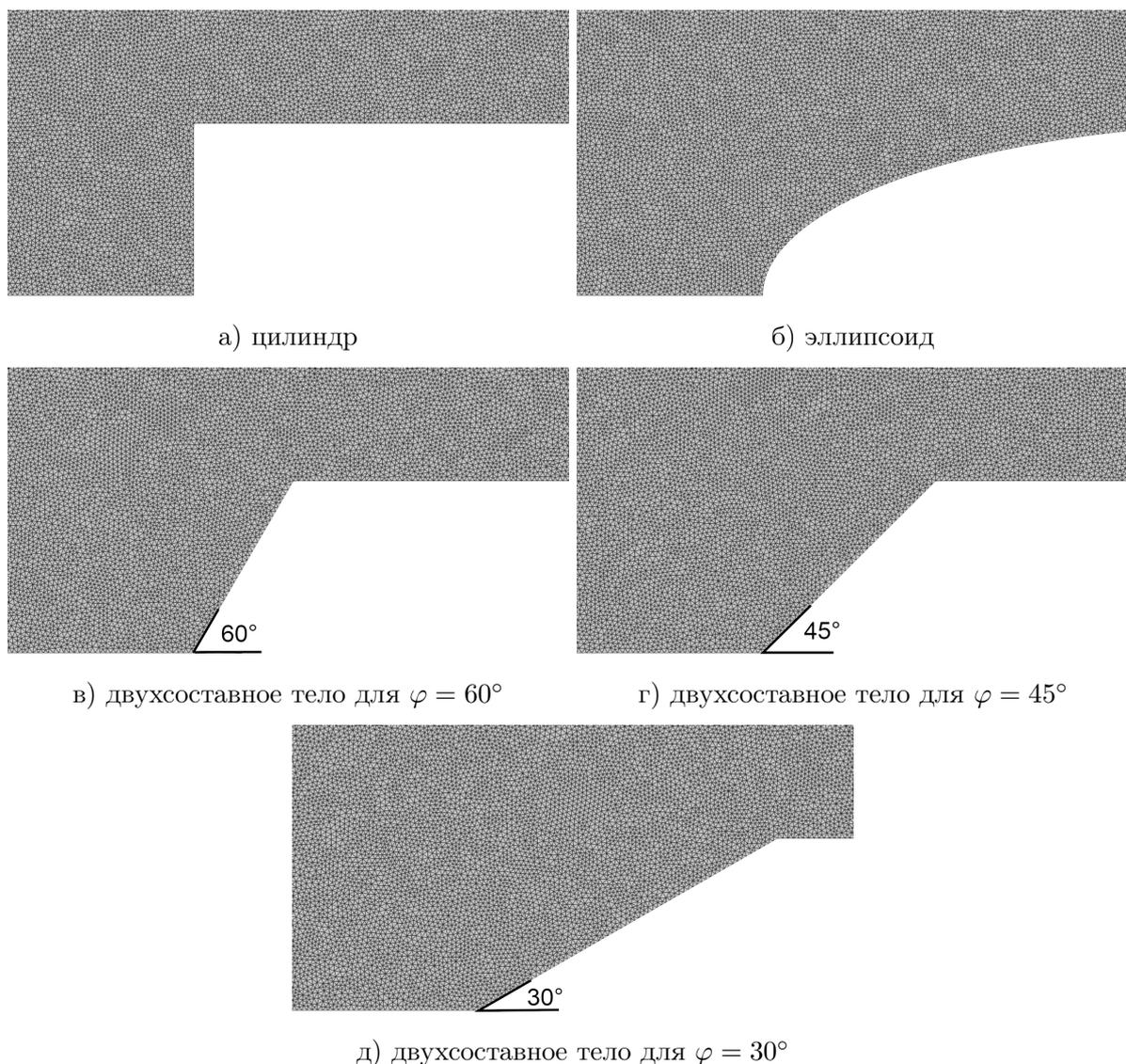


Рис. 2. Расчетные сетки для выбранных тел вращения

или четырехпроцессорной рабочей станции использовалось дополнительное разделение вычислений по оперативной памяти процессоров с использованием механизма NUMA [27]. В этой ситуации необходимые обмены данными между отдельными блоками вычислений производились через общий сегмент памяти. Фактически в этом случае используется двухуровневое разбиение сетки: сначала на процессорные блоки, а затем внутри каждого такого блока на домены, которые распределяются между всеми ядрами (трэдами) соответствующего процессора.

Для расчета больших сеточных конфигураций (более 10 млн. ячеек) предполагается использование гибридной схемы распараллеливания, использующей одновременно стандарты MPI [28] и OpenMP. В этом случае на каждом узле кластера необязательно пользоваться механизмом NUMA, поскольку его можно заменить помещением нескольких MPI-процессов на узел соответственно числу имеющихся там процессоров. Привязка MPI-процесса к конкретному процессору узла производится с помощью управления процессорной маской.

Расчеты умеренных сеточных конфигураций проводились на кластере IMM23, установленном в Суперкомпьютерном центре коллективного пользования ИПМ им. М.В. Келдыша РАН (ЦКП). На каждом узле кластера установлены 2 процессора AMD EPYC 9124, каждый из них имеет 16 двухпоточковых ядер и работает на частоте 3.0 ГГц. Расчеты больших сеточных конфигураций проводились на суперкомпьютере K60 ЦКП [29].

5. Результаты моделирования

Для проведения конкретных расчетов была выбрана лабораторная геометрия с характерными параметрами длины R_n , $Z_n = 0.01$ м. С их помощью несложно пересчитать размеры расчетных областей в конфигурациях 1–5 (см. табл. 1). В качестве покоящегося и набегающего потоков газа рассматривалась однокомпонентная сухая воздушная смесь с параметрами

$$\begin{aligned} \rho_0 &= 1.184 \text{ кг/м}^3, & T_0 &= 297 \text{ К}, & p_0 &= 101\,325 \text{ Па}, \\ \mu_0 &= 1.7894 \cdot 10^{-5} \text{ кг/(м} \cdot \text{с)}, & \chi_0 &= 0.0242 \text{ Вт/(м} \cdot \text{К)}. \end{aligned}$$

Скорость звука в этой смеси $c_0 \approx 343$ м/с служила нормировкой скорости. Остальные параметры были следующие. Показатель адиабаты $\gamma = 1.42$, нормировочное число Рейнольдса $Re = \rho_0 c_0 R_n / \mu_0 \approx 226\,954$, число Прандтля $Pr = 0.67$, число Шмидта $Sc = 0.77$, показатель температурной зависимости кинетических коэффициентов $\omega = 0.5$. Расчеты проводились для трех значений числа Маха: $Ma = 1.0, 2.0, 3.0$.

Целью расчетов на основе выбранных конфигураций и параметров был анализ качества реализованной численной схемы и эффективности параллельного кода. Для верификации численного алгоритма расчеты проводились для нескольких значений управляющих параметров α и β . Для валидации разработанной программы использовался известный программный пакет ANSYS CFD, солвер Fluent [26] (Лицензия ИПМ им. М.В. Келдыша РАН — ANSYS CFD Enterprise, Permanent, № 511564).

Обсуждение результатов начнем с выбора параметров α и β . Как было показано в работе [11], параметр α выступает в качестве коэффициента искусственной вязкости и подбирается из диапазона $[0, 1.0]$, исходя из условий конкретной задачи. Основанием для его выбора является обеспечение устойчивого сквозного счета разработанного вычислительного алгоритма. При этом излишнее завышение параметра α приводит к существенному размыванию фронта ударных волн в сеточной модели. В рамках данного исследования параметр α принимался равным 0.5 независимо от скорости входного потока и минимального линейного размера сеточных элементов. При предварительном исследовании были опробованы несколько значений данного параметра. Оно показало, что уже при $\alpha = 0.25$ у входной границы на первых временных шагах появляются нефизичные осцилляции всех расчетных параметров потока (плотности, скорости и энергии). А при $\alpha = 0.1$ вычислительный алгоритм расходится независимо от шага по времени. Отметим также, что в целом значение параметра α , обеспечивающее устойчивость численного решения, сильно зависит от числа

Рейнольдса. При невысоких значениях ($Re \sim 1000$) устойчивого счета удается добиться даже при $\alpha = 0$.

Выбор параметра β фактически позволяет задать шаг по времени из условия устойчивости (12). Величина β в начале расчета выбирается из полуинтервала $(0, 1.0]$. Однако далее анализируется дополнительный эмпирический критерий ограничения шага по времени

$$\max_i \left| \frac{E_i^{k+1} - E_i^k}{E_i^k} \right| \leq \varepsilon_E, \quad i \in I(\Omega_h), \quad (13)$$

где максимум модуля относительного изменения полной энергии за один шаг по времени берется по множеству индексов $I(\Omega_h)$ всех узлов расчетной сетки. Из этого критерия при заданной допустимой величине ε_E можно найти ограничение на шаг Δt и, следовательно, для β . При этом ε_E является величиной порядка 0.01.

В данной работе выбрано значение $\varepsilon_E = 0.02$. В результате в проведенных расчетах конфигурации 2 при $Ma = 1$ расчет начинался при $\beta = \beta_{max} = 0.1$. Далее этот параметр в автоматическом режиме доходил до $\beta_{min} = 0.01$ и при формировании стационарных параметров течения возвращался к β_{max} . Для расчета течений при других числах Маха величина β_{max} оставалась постоянной, а величина β_{min} составила 0.0022 и 0.0007 для $Ma = 2$ и $Ma = 3$ соответственно.

Итогом проведенного анализа является вывод о том, что параметр α позволяет эффективно сглаживать нефизичные осцилляции, а параметр β отвечает за устойчивость численного решения в сеточной норме C , в том числе позволяет обеспечить неотрицательность массовой плотности, плотности полной энергии и температуры. Отметим также, что при измельчении сетки параметры α и β можно не изменять. Для сильно вязких течений или тел малого масштаба он может быть равен 0. При больших числах Рейнольдса его приходится увеличивать.

При значениях $\alpha = 0.5$ и указанных выше параметрах ε_E и β для $Ma = 2$ были получены стационарные распределения плотности, давления, температуры и модуля скорости, представленные соответственно рис. 3–7 для конфигураций 1–5. В целях сравнения расчет конфигурации 1 был повторен в пакете ANSYS CFD с помощью солвера Fluent. В качестве модели использовалась стационарная осесимметричная модель на основе совместного решения уравнений неразрывности и импульса. В качестве уравнения состояния применялся идеальный газ с параметрами, соответствующими использованному при расчетах КГД-методом. Вычислительный алгоритм основывался на явном методе установления с противопоточной аппроксимацией конвективных членов второго порядка. Треугольная расчетная сетка соответствовала представленной в табл. 1 для расчетной конфигурации 1. Анализ расчетов показал качественное совпадение результатов по всем характеристикам: плотности, давлению, температуре и модулю скорости. Отклонение от расчетных данных ANSYS составило порядка 5%.

Также на основании динамики распределения давления были построены зависимости коэффициента сопротивления формы $C_d = \frac{2F_z}{\rho_0 A |u|^2}$, где $F_z = - \int_{body} p dS$ — сила сопротивления формы, $A = \pi R_2^2$ — площадь фронтального сечения. Полученные зависимости для каждого из объектов изображены на рис. 8. Анализ полученных результатов позволил сделать вывод об их корректности [31]. Так, максимальный коэффициент сопротивления формы характерен для торцевого обтекания цилиндра, далее в порядке убывания величины коэф-

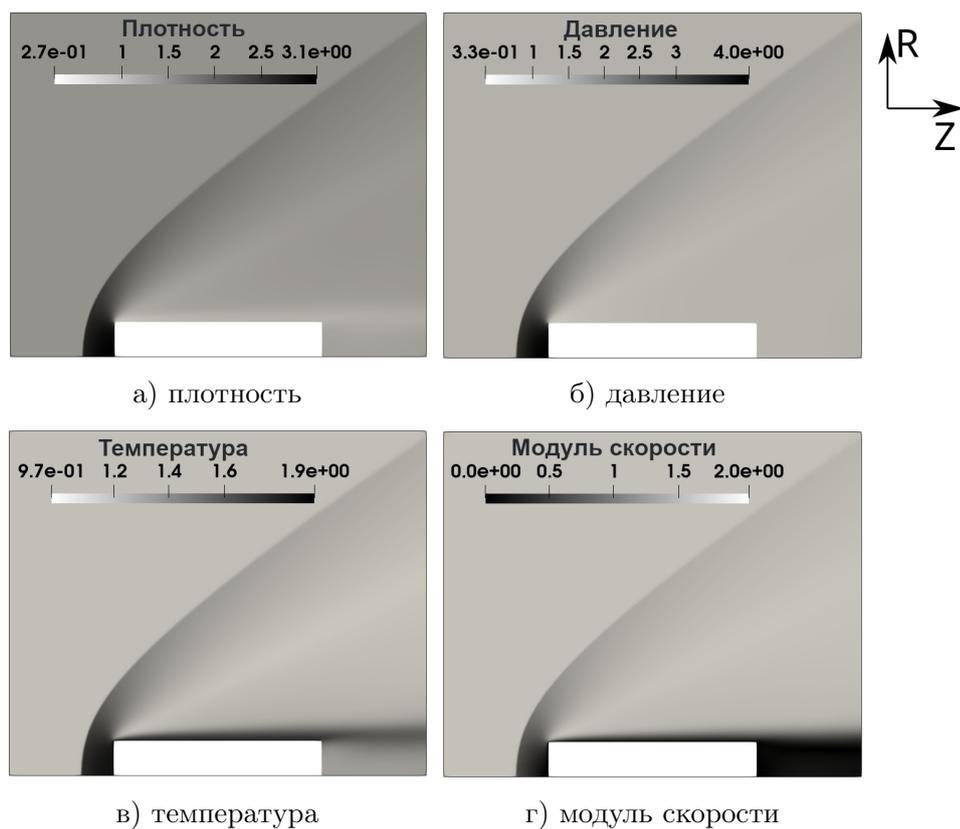


Рис. 3. Стационарные распределения параметров течения для конфигурации 1

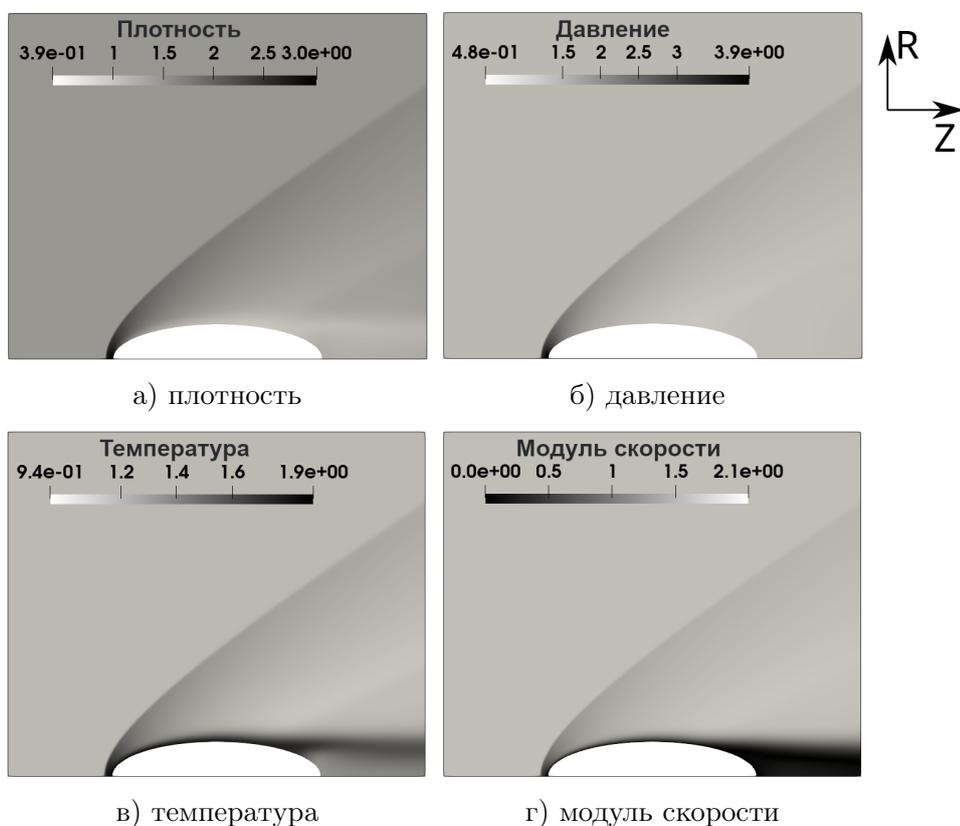


Рис. 4. Стационарные распределения параметров течения для конфигурации 2

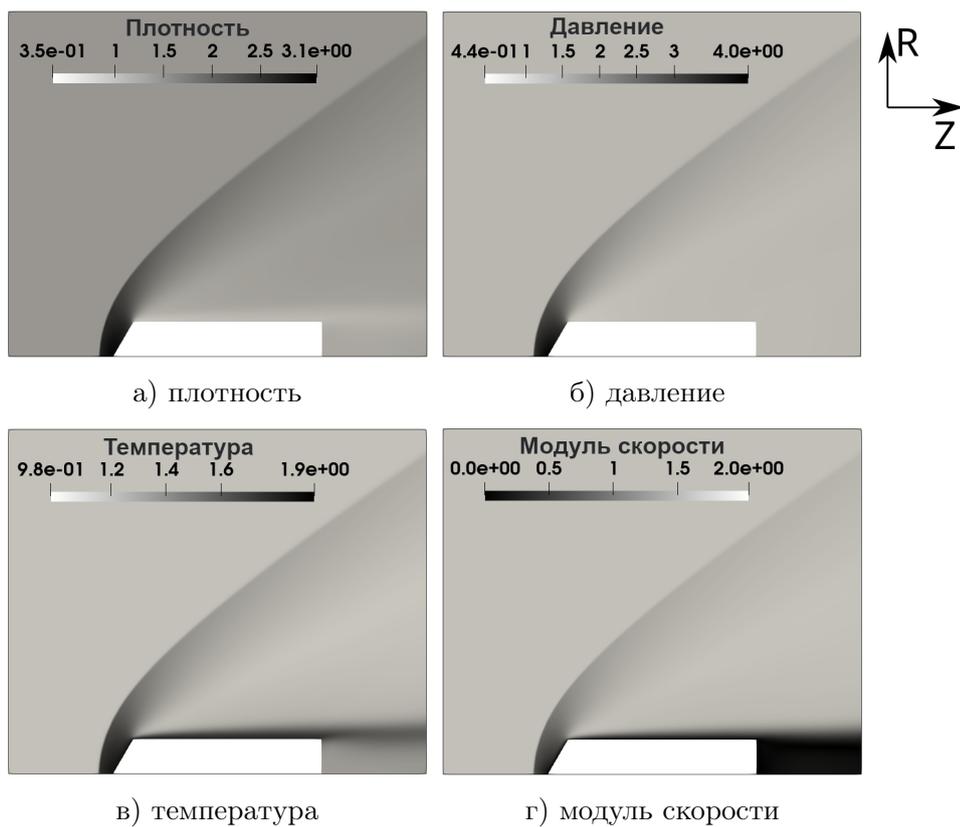


Рис. 5. Стационарные распределения параметров течения для конфигурации 3

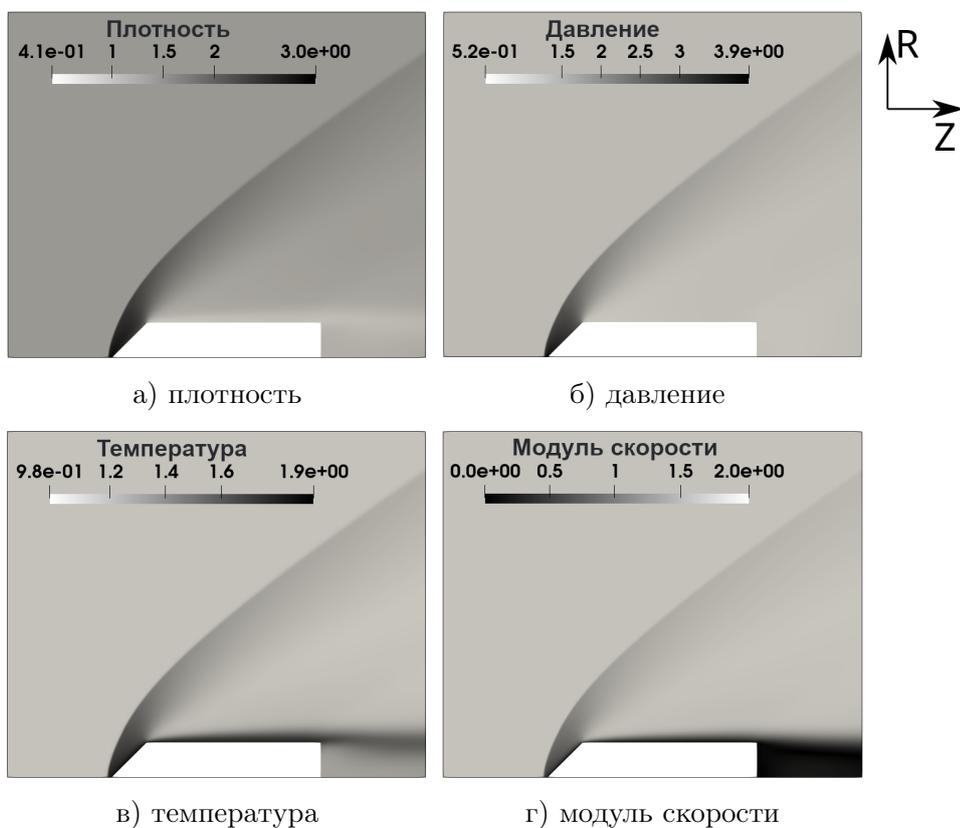


Рис. 6. Стационарные распределения параметров течения для конфигурации 4

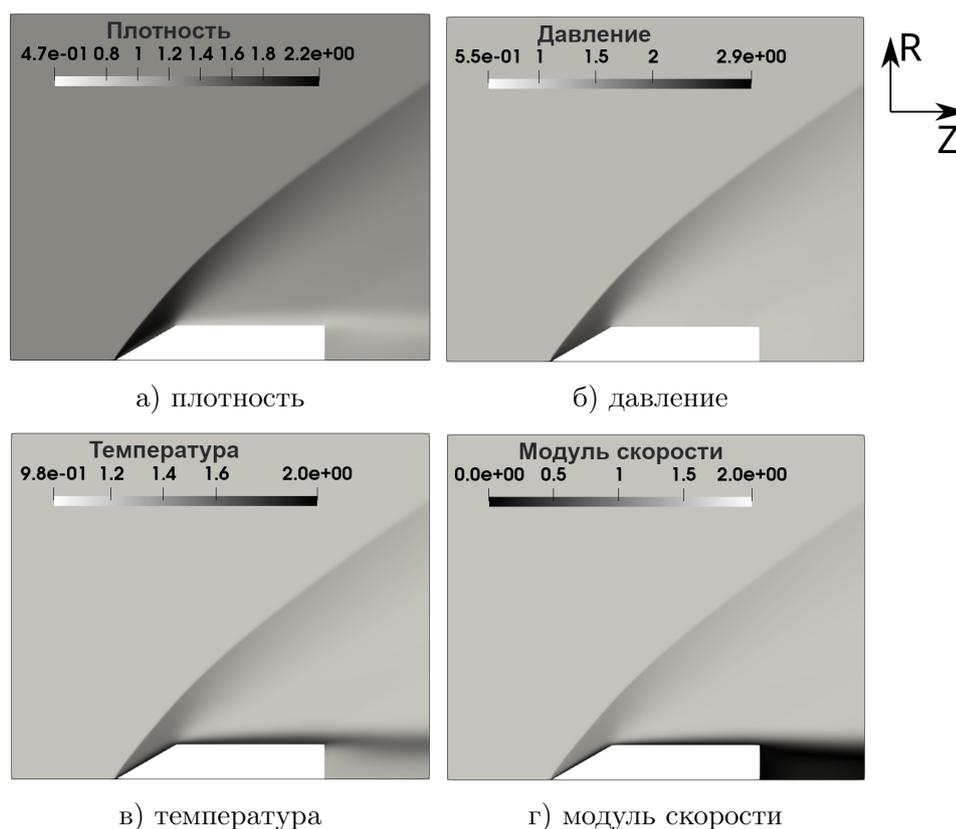


Рис. 7. Стационарные распределения параметров течения для конфигурации 5

фициента следуют составные тела с конусами на торце от 60 до 30 градусов. Наименьший же коэффициент сопротивления характерен для эллипсоида.

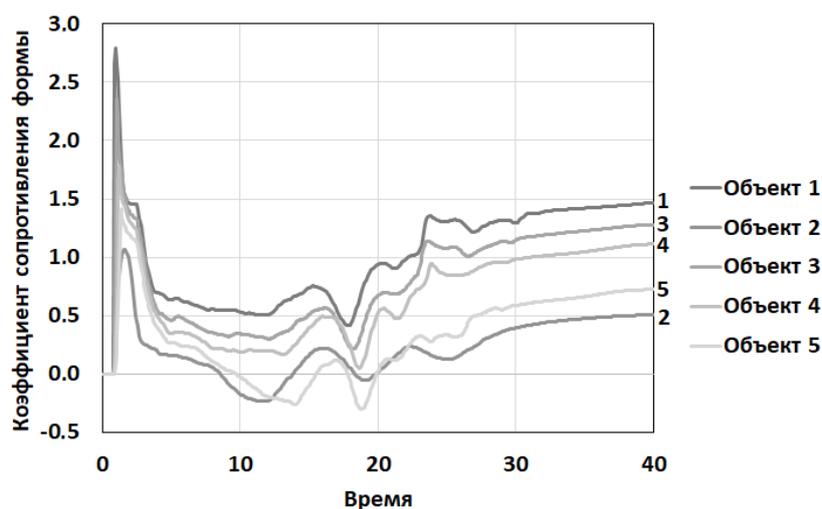


Рис. 8. Зависимость коэффициента сопротивления формы C_d от времени

Следующая серия расчетов проводилась для конфигурации 2 для $Ma = 1$ на последовательности измельченных сеток (см. табл. 2). Данные расчеты использовались как для верификации численного метода, так и для определения эффективности распараллеливания.

Таблица 2. Параметры измельченных сеток

ID	Геометрия	Число узлов	Число ребер	Число ячеек
M1	Эллипсоид	170 578	2 040 960	339 663
M2	Эллипсоид	680 818	8 157 876	1 358 652
M3	Эллипсоид	2 720 287	32 619 576	5 434 608
M4	Эллипсоид	10 875 181	130 454 448	21 738 432

При анализе точности использованного численного метода проведены расчеты на сетках M1–M3. Они подтвердили сходимость решения по сетке с порядком не ниже первого. В качестве иллюстрации сказанного на рис. 9 изображены зависимости коэффициента сопротивления формы от времени для исследуемого набора сеток при $Ma = 1$. Как видно из приведенных графиков, измельчение сетки приводит к небольшому уменьшению коэффициента сопротивления, что можно объяснить лучшей аппроксимацией границы эллипса, большей ее гладкости. При этом, при умеренных числах Рейнольдса и $\alpha = 0$ получается второй пространственный порядок точности. Полученные решения сравнивались с аналогичными, рассчитанными в пакете Ansys Fluent. Сравнение показало близость значений коэффициентов сопротивления формы.

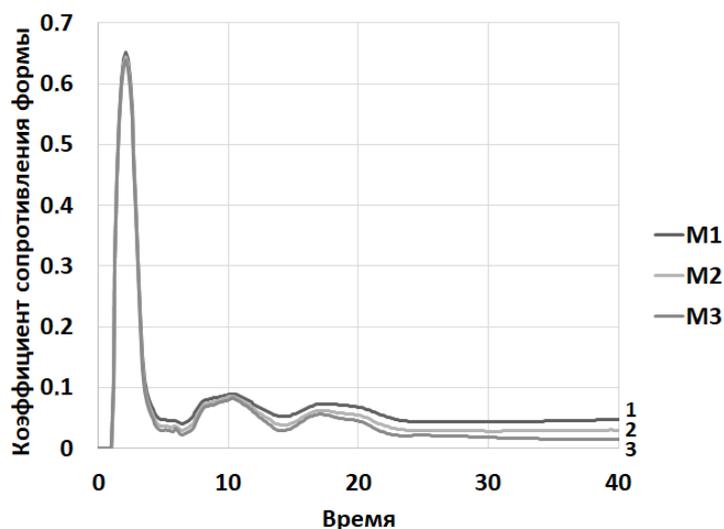


Рис. 9. Зависимость коэффициента сопротивления формы C_d от времени для сеток M1–M3

Результаты вычисления ускорения и эффективности распараллеливания на сетках умеренного размера (M1, M2, M3) представлены на рис. 10 и рис. 11. Они показывают, что при малом количестве процессов преобладают линейный рост ускорения и слабое падение эффективности, что соответствует теоретическим оценкам этих величин для явных схем по времени. Дальнейший загиб ускорения и более интенсивное падение эффективности связаны с архитектурными особенностями использованного оборудования, в том числе с включенным режимом гипертрейдинга. Менее эффективное поведение программы на сетке с большим числом узлов M3 связано с превышением обрабатываемых данных размера кэша процессора. В целом можно отметить, что полученное максимальное ускорение для сеток

M1, M2 позволило существенно сократить времена расчета. Таким образом, реализованная технология на основе стандарта OpenMP полностью себя оправдала.

Данные о расчетах 100 временных шагов на большой сетке M4 представлены в табл. 3. Они приведены для сравнения для кластеров IMM23 и K60 (секция с CPU). В таблице приведены времена расчета для суммарного количества параллельных процессов от 16 до 512. Для результатов, полученных на K60, дополнительно рассчитаны ускорение и эффективность. Анализ таблицы показывает достаточно высокую эффективность распараллеливания при использовании гибридной технологии MPI+OpenMP.

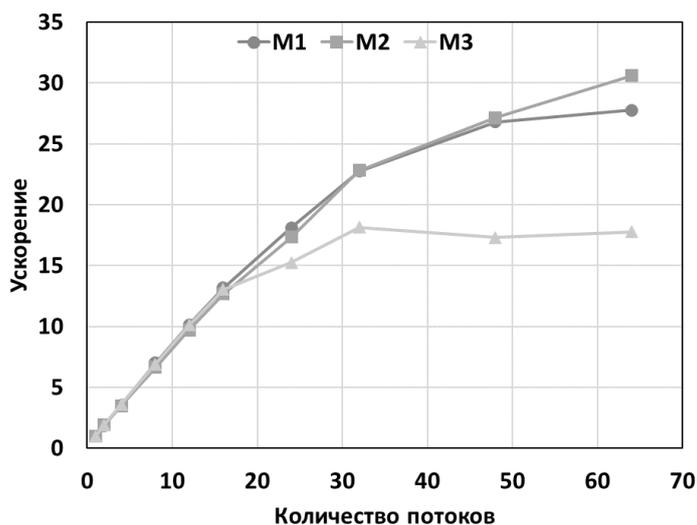


Рис. 10. Зависимости ускорения распараллеливания от количества параллельных процессов для сеток M1, M2, M3

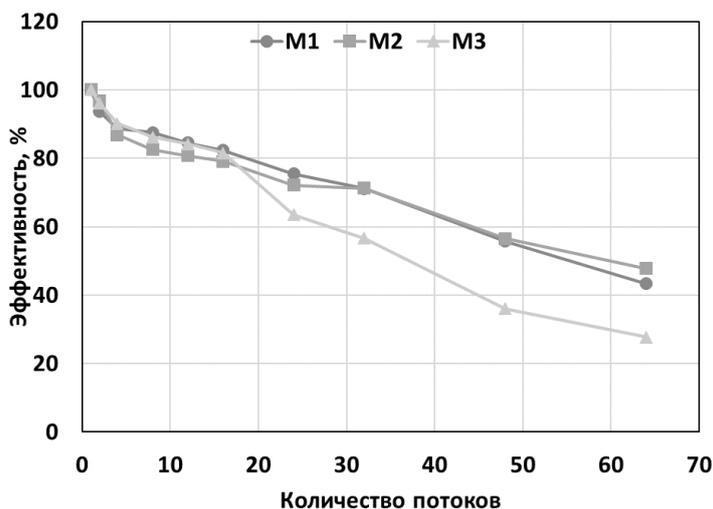


Рис. 11. Зависимости эффективности распараллеливания от количества параллельных процессов для сеток M1, M2, M3

Таблица 3. Параметры распараллеливания на сетке М4

Параметр	Кол-во процессов					
	16	32	64	128	256	512
Время расчета (ИММ23), с	1 149	789	—	—	—	—
Время расчета (К60_ CPU), с	1 033	562	304	172	98	65
Ускорение	1.00	1.84	3.40	6.00	10.54	15.89
Эффективность, %	100.00	91.90	84.95	75.00	65.88	49.66

Заключение

Рассмотрена проблема обтекания тел вращения различной формы сверхзвуковым потоком вязкого теплопроводного газа. Актуальность таких расчетов связана с повышенным вниманием к проектированию спускаемых космических аппаратов нового поколения. Основная цель работы состояла в анализе качества численных алгоритмов на неструктурированных сетках и их параллельной реализации. Особенностью использованного подхода было применение в качестве модели течения газа квазигазодинамической системы уравнений. Использование данной системы определяется ее повышенной численной устойчивостью к схемным возмущениям по сравнению с численными подходами на основе уравнений Навье—Стокса. Главными направлениями работы стали анализ качества результатов моделирования при варьировании параметров численного метода и изучение особенностей параллельной программной реализации. Итогом проведенного исследования стали оптимальные значения параметров регуляризации КГД-подхода, а также зависимости ускорения и эффективности от методики распараллеливания. В качестве примера были проведены расчеты сверхзвуковых осесимметричных течений сухого воздуха вокруг цилиндра, эллипсоида и составного тела. В проведенных численных исследованиях были установлены критерии выбора параметров явной по времени конечно-объемной схемы с целью повышения ее устойчивости и точности. Полученные результаты подтвердили корректность разработанной вычислительной технологии и эффективность разработанного параллельного кода. В перспективе на основе разработанного численного подхода предполагается перейти к решению задач технической аэродинамики, предполагающей учет реального уравнения состояния газа и свойств поверхности обтекаемого им тела.

Работа выполнена в рамках государственного задания ИПМ им. М.В. Келдыша РАН. Расчеты проведены с помощью кластера ИММ23 и суперкомпьютера К60, установленных в Суперкомпьютерном ЦКП ИПМ им. М.В. Келдыша РАН.

Литература

1. Абрамович Г.Н. Прикладная газовая динамика. Москва: Наука, 1969. 824 с.
2. Годунов С.К., Забродин А.В., Иванов М.Я. и др. Численное решение многомерных задач газовой динамики. Москва: Наука, 1976. 400 с.
3. Самарский А.А., Попов Ю.П. Разностные методы решения задач газовой динамики. Москва: Наука, 1992. 424 с.
4. Приходько А.А. Компьютерные технологии в аэродинамике и теплообмене. Киев: Наукова думка, 2003. 379 с.

5. Крайко А.Н. Теоретическая газовая динамика: классика и современность. Москва: Торус пресс, 2010. 430 с.
6. Pert G.J. Introductory Fluid Mechanics for Physicists and Mathematicians. New York: Wiley, 2013. 490 p.
7. Georgantopoulou C.G., Georgantopoulos G.A. Fluid Mechanics in Channel, Pipe and Aerodynamic Design Geometries. Vol. 1 and Vol. 2. New York: Wiley–ISTE, 2018. 416 p. and 304 p. DOI: 10.1002/9781119457015.
8. Rathakrishnan E. Applied Gas Dynamics. New York: Wiley, 2019. 656 p.
9. Chetverushkin B.N. Kinetic Schemes and Quasi Gas Dynamic System of Equations. Barcelona: CIMNE, 2008. 298 p.
10. Zlotnik A.A., Chetverushkin B.N. Parabolicity of the Quasi-Gasdynamics System of Equations, Its Hyperbolic Second-Order Modification, and the Stability of Small Perturbations for Them // Comput. Math. Math. Phys. 2008. Vol. 48, no. 3. P. 420–446. DOI: 10.1134/S0965542508030081.
11. Elizarova T.G. Quasi-Gas Dynamic Equations. Berlin, Heidelberg, New York: Springer–Verlag, 2009. 286 p. DOI: 10.1007/978-3-642-00292-2.
12. Zlotnik A., Gavrilin V. On Quasi-Gasdynamics System of Equations with General Equations of State and Its Application // Mathematical Modelling and Analysis. 2011. Vol. 16, no. 4. P. 509–526. DOI: 10.3846/13926292.2011.627382.
13. Elizarova T.G., Shirokov I.A. Numerical Simulation of the Non-Stationary Flow in a Vicinity of the Hypersonic Vehicle // Math. Models Comput. Simul. 2012. Vol. 4, no. 4. P. 410–418. DOI: 10.1134/S2070048212040035.
14. Chetverushkin B.N. Kinetic Models for Solving Continuum Mechanics Problems on Supercomputers // Math. Models Comput. Simul. 2015. Vol. 7, no. 6. P. 531–539. DOI: 10.1134/S2070048215060034.
15. Elizarova T.G., Zlotnik A.A., Shilnikov E.V. Regularized Equations for Numerical Simulation of Flows of Homogeneous Binary Mixtures of Viscous Compressible Gases // Comput. Math. Math. Phys. 2019. Vol. 59, no. 11. P. 1832–1847. DOI: 10.1134/S0965542519110058.
16. Shirokov I.A., Elizarova T.G. Computational Experiment in the Problem of Supersonic Flow around a Blunt Body with Tail Expansion // Math. Models Comput. Simul. 2020. Vol. 12, no. 3. P. 433–444. DOI: 10.1134/S2070048220030163.
17. Elizarova T.G., Shilnikov E.V. Numerical Simulation of Gas Mixtures Based on the Quasi-Gasdynamics Approach as Applied to the Interaction of a Shock Wave with a Gas Bubble // Comput. Math. Math. Phys. 2021. Vol. 61, no. 1. P. 118–128. DOI: 10.1134/S0965542521010048.
18. Zlotnik A.A. Conditions for Dissipativity of an Explicit Finite-Difference Scheme for a Linearized Multidimensional Quasi-Gasdynamics System of Equations // Dokl. Math. 2022. Vol. 106, no. 1. P. 236–242. DOI: 10.1134/S1064562422040196.
19. Kudryashova T.A., Polyakov S.V., Sverdlin A.A. Calculation of Gas Flow Parameters around Reentry Vehicle // Math. Models and Comput. Simul. 2009. Vol. 1, no. 4. P. 445–452. DOI: 10.1134/S2070048209040036.

20. Polyakov S.V., Kudryashova T.A., Sverdlin A.A., *et al.* Parallel Software Package for Simulation of Continuum Mechanics Problems on Modern Multiprocessor Systems // Math. Models and Comput. Simul. 2011. Vol. 3, no. 1. P. 46–57. DOI: 10.1134/S2070048211010091.
21. Eymard R., Gallouet T.R., Herbin R. The Finite Volume Method // Handbook of Numerical Analysis (Vol. 7). Amsterdam: North Holland Publishing Company. 2000. P. 713–1020. DOI: 10.1016/S1570-8659(00)07005-8.
22. LeVeque R.J. Finite Volume Methods for Hyperbolic Problems. Cambridge: Cambridge Univ. Press, 2002. 558 p. DOI: 10.1017/CBO9780511791253.
23. Geuzaine C., Remacle J.-F. Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-processing Facilities // International Journal for Numerical Methods in Engineering. 2009. Vol. 79, no. 11. P. 1309–1331. DOI: 10.1002/nme.2579.
24. Smith B.F. Domain Decomposition Methods for Partial Differential Equations // Parallel Numerical Algorithms. ICASE/LaRC Interdisciplinary Series in Science and Eng. / ed. by D.E. Keyes, A. Sameh, V. Venkatakrishnan. Springer, 1997. Vol. 4. P. 225–243. DOI: 10.1007/978-94-011-5412-3_8.
25. Alakeel A.A. Guide to Dynamic Load Balancing in Distributed Computer Systems // International Journal of Computer Science and Network Security. 2009. Vol. 10, no. 6. P. 153–160.
26. Specifications – OpenMP. URL: <https://www.openmp.org/specifications> (дата обращения: 24.01.2025).
27. Patterson D.A., Hennessy J.L. Computer Organization and Design, Second Edition: The Hardware/Software Interface. Morgan Kaufmann, 1997. 965 p.
28. OpenMPI. URL: <https://www.open-mpi.org/> (дата обращения: 24.01.2025).
29. Суперкомпьютерный центр коллективного пользования ИПМ им. М.В. Келдыша РАН. URL: <https://ckp.kiam.ru/> (дата обращения: 24.01.2025).
30. ANSYS CFD. URL: <https://www.ansys.com/products/fluids> (дата обращения: 24.01.2025).
31. Lindsey W.F. Drag of Cylinders of Simple Shapes: NASA Technical Reports // NASA. 1938. No. 619. P. 169–176. URL: <https://ntrs.nasa.gov/citations/19930091694>.

Поляков Сергей Владимирович, д.ф.-м.н., в.н.с., Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация)

Подрыга Виктория Олеговна, д.ф.-м.н., в.н.с., Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация)

Тарасов Никита Игоревич, к.ф.-м.н., н.с., Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация)

SUPERCOMPUTER MODELING OF SUPERSONIC FLOW AROUND ROTATION BODIES OF DIFFERENT SHAPES

© 2025 S.V. Polyakov, V.O. Podryga, N.I. Tarasov

Keldysh Institute of Applied Mathematics of RAS (Miusskaya sq. 4, Moscow, 125047 Russia)

E-mail: polyakov@imamod.ru, pvictoria@list.ru, nikita_tarasov@imamod.ru

Received: 18.04.2025

The numerical approach and its parallel software implementation for studying the processes of flow around solids of complex geometry by supersonic gas flows are discussed. The aim of the study is to analyze the effectiveness of numerical schemes on unstructured hybrid grids approximating quasi-gasdynamics (QGD) equations. As an example, the problem of flowing around bodies of rotation of various shapes is chosen. The gaseous medium is represented by dry air. The system of QGD equations is considered in a one-component formulation. It is supplemented by equations of state of an ideal gas and the dependencies of kinetic coefficients on temperature and pressure. Axisymmetric gas flows around a cylinder, an ellipsoid, and a composite body are considered in numerical experiments. Calculations are performed for three values of the input flow velocity. The obtained results confirm the correctness of the developed computing technology.

Keywords: gas flows near solids, rotation bodies of different shapes, quasi-gasdynamics equations, numerical methods, parallel algorithms, supercomputer modeling.

FOR CITATION

Polyakov S.V., Podryga V.O., Tarasov N.I. Supercomputer Modeling of Supersonic Flow around Rotation Bodies of Different Shapes. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2025. Vol. 14, no. 2. P. 67–85. (in Russian) DOI: 10.14529/cmse250204.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Abramovich G.N. Applied Gas Dynamics. Moscow: Nauka, 1969. 824 p. (in Russian)
2. Godunov S.K., Zabrodin A.V., Ivanov M.Ya., *et al.* Numerical Solution of Multidimensional Problems of Gas Dynamics. Moscow: Nauka, 1976. 400 p. (in Russian)
3. Samarskii A.A., Popov Yu.P. Difference Methods for Solving Gas Dynamics Problems. Moscow: Nauka, 1992. 424 p. (in Russian)
4. Prikhodko A.A. Computer Technologies in Aerodynamics and Heat and Mass Transfer. Kiev: Naukova Dumka, 2003. 379 p. (in Russian)
5. Krayko A.N. Theoretical Gas Dynamics: Classics and Modernity. Moscow: Torus Press, 2010. 430 p. (in Russian)
6. Pert G.J. Introductory Fluid Mechanics for Physicists and Mathematicians. New York: Wiley, 2013. 490 p.
7. Georgantopoulou C.G., Georgantopoulos G.A. Fluid Mechanics in Channel, Pipe and Aerodynamic Design Geometries. Vol. 1 and Vol. 2. New York: Wiley–ISTE, 2018. 416 p. and 304 p. DOI: 10.1002/9781119457015.

8. Rathakrishnan E. Applied Gas Dynamics. New York: Wiley, 2019. 656 p.
9. Chetverushkin B.N. Kinetic Schemes and Quasi Gas Dynamic System of Equations. Barcelona: CIMNE, 2008. 298 p.
10. Zlotnik A.A., Chetverushkin B.N. Parabolicity of the Quasi-Gasdynamics System of Equations, Its Hyperbolic Second-Order Modification, and the Stability of Small Perturbations for Them. *Comput. Math. Math. Phys.* 2008. Vol. 48, no. 3. P. 420–446. DOI: 10.1134/S0965542508030081.
11. Elizarova T.G. Quasi-Gas Dynamic Equations. Berlin, Heidelberg, New York: Springer–Verlag, 2009. 286 p. DOI: 10.1007/978-3-642-00292-2.
12. Zlotnik A., Gavrilin V. On Quasi-Gasdynamics System of Equations with General Equations of State and Its Application. *Mathematical Modelling and Analysis.* 2011. Vol. 16, no. 4. P. 509–526. DOI: 10.3846/13926292.2011.627382.
13. Elizarova T.G., Shirokov I.A. Numerical Simulation of the Non-Stationary Flow in a Vicinity of the Hypersonic Vehicle. *Math. Models Comput. Simul.* 2012. Vol. 4, no. 4. P. 410–418. DOI: 10.1134/S2070048212040035.
14. Chetverushkin B.N. Kinetic Models for Solving Continuum Mechanics Problems on Supercomputers. *Math. Models Comput. Simul.* 2015. Vol. 7, no. 6. P. 531–539. DOI: 10.1134/S2070048215060034.
15. Elizarova T.G., Zlotnik A.A., Shilnikov E.V. Regularized Equations for Numerical Simulation of Flows of Homogeneous Binary Mixtures of Viscous Compressible Gases. *Comput. Math. Math. Phys.* 2019. Vol. 59, no. 11. P. 1832–1847. DOI: 10.1134/S0965542519110058.
16. Shirokov I.A., Elizarova T.G. Computational Experiment in the Problem of Supersonic Flow around a Blunt Body with Tail Expansion. *Math. Models Comput. Simul.* 2020. Vol. 12, no. 3. P. 433–444. DOI: 10.1134/S2070048220030163.
17. Elizarova T.G., Shilnikov E.V. Numerical Simulation of Gas Mixtures Based on the Quasi-Gasdynamics Approach as Applied to the Interaction of a Shock Wave with a Gas Bubble. *Comput. Math. Math. Phys.* 2021. Vol. 61, no. 1. P. 118–128. DOI: 10.1134/S0965542521010048.
18. Zlotnik A.A. Conditions for Dissipativity of an Explicit Finite-Difference Scheme for a Linearized Multidimensional Quasi-Gasdynamics System of Equations. *Dokl. Math.* 2022. Vol. 106, no. 1. P. 236–242. DOI: 10.1134/S1064562422040196.
19. Kudryashova T.A., Polyakov S.V., Sverdlin A.A. Calculation of Gas Flow Parameters around Reentry Vehicle. *Math. Models and Comput. Simul.* 2009. Vol. 1, no. 4. P. 445–452. DOI: 10.1134/S2070048209040036.
20. Polyakov S.V., Kudryashova T.A., Sverdlin A.A., *et al.* Parallel Software Package for Simulation of Continuum Mechanics Problems on Modern Multiprocessor Systems. *Math. Models and Comput. Simul.* 2011. Vol. 3, no. 1. P. 46–57. DOI: 10.1134/S2070048211010091.
21. Eymard R., Gallouet T.R., Herbin R. The Finite Volume Method. *Handbook of Numerical Analysis (Vol. 7)*. Amsterdam: North Holland Publishing Company. 2000. P. 713–1020. DOI: 10.1016/S1570-8659(00)07005-8.
22. LeVeque R.J. Finite Volume Methods for Hyperbolic Problems. Cambridge: Cambridge Univ. Press, 2002. 558 p. DOI: 10.1017/CBO9780511791253.

23. Geuzaine C., Remacle J.-F. Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-processing Facilities. *International Journal for Numerical Methods in Engineering*. 2009. Vol. 79, no. 11. P. 1309–1331. DOI: 10.1002/nme.2579.
24. Smith B.F. *Domain Decomposition Methods for Partial Differential Equations. Parallel Numerical Algorithms*. ICASE/LaRC Interdisciplinary Series in Science and Eng. / ed. by D.E. Keyes, A. Sameh, V. Venkatakrisnan. Springer, 1997. Vol. 4. P. 225–243. DOI: 10.1007/978-94-011-5412-3_8.
25. Alakeel A.A. Guide to Dynamic Load Balancing in Distributed Computer Systems. *International Journal of Computer Science and Network Security*. 2009. Vol. 10, no. 6. P. 153–160.
26. Specifications – OpenMP. URL: <https://www.openmp.org/specifications> (accessed: 24.01.2025).
27. Patterson D.A., Hennessy J.L. *Computer Organization and Design, Second Edition: The Hardware/Software Interface*. Morgan Kaufmann, 1997. 965 p.
28. OpenMPI. URL: <https://www.open-mpi.org/> (accessed: 24.01.2025).
29. Supercomputer Centre of Collective Usage of KIAM RAS. URL: <https://ckp.kiam.ru/> (accessed: 24.01.2025).
30. ANSYS CFD. URL: <https://www.ansys.com/products/fluids> (accessed: 24.01.2025).
31. Lindsey W.F. Drag of Cylinders of Simple Shapes: NASA Technical Reports. NASA. 1938. No. 619. P. 169–176. URL: <https://ntrs.nasa.gov/citations/19930091694>.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Научный журнал «Вестник ЮУрГУ. Серия «Вычислительная математика и информатика» основан в 2012 году.

Учредитель — Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет» (национальный исследовательский университет).

Главный редактор — Л.Б. Соколинский.

Свидетельство о регистрации ПИИ ФС77-57377 выдано 24 марта 2014 г. Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

Журнал включен в Реферативный журнал и Базы данных ВИНТИ; индексируется в библиографической базе данных РИНЦ. Журнал размещен в открытом доступе на Всероссийском математическом портале MathNet. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

Решением Президиума Высшей аттестационной комиссии Министерства образования и науки Российской Федерации журнал включен в «Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук» по научным специальностям и соответствующим им отраслям науки: 1.2.3 – Теоретическая информатика, кибернетика (физико-математические науки), 2.3.5 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (физико-математические науки).

Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год.

Адрес редакции, издателя: 454080, г. Челябинск, проспект Ленина, 76, Издательский центр ЮУрГУ, каб. 32.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии <https://vestnikvmi.susu.ru>. Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются.
2. Адрес редакционной коллегии научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:
Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, ЮУрГУ, НОЦ ИИКТ,
зам. главного редактора Цымблеру М.Л.
3. Адрес электронной почты редакции: vestnikvmi@susu.ru.
4. Плата с авторов за публикацию рукописей не взимается, гонорары авторам не выплачиваются.

ВЕСТНИК
ЮЖНО-УРАЛЬСКОГО
ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
Серия
«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»
2025 Том 14, № 2

16+

Техн. редактор А.В. Миних

Издательский центр Южно-Уральского государственного университета

Подписано в печать 30.05.2025. Дата выхода в свет 10.07.2025. Формат 60×84 1/8. Печать цифровая.
Усл. печ. л. 10,23. Тираж 500 экз. Заказ 107/133. Цена свободная.

Отпечатано в типографии Издательского центра ЮУрГУ.
454080, г. Челябинск, проспект Ленина, 76.