КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2026 Т. 18 № 5 С. 1121–1128
DOI: 10.20537/2076-7633-2026-18-5-1121-1128

Ки&М

ЧИСЛЕННЫЕ МЕТОДЫ И ОСНОВЫ ИХ РЕАЛИЗАЦИИ

УДК: 519.8

Применение китайской теоремы об остатках для работы с длинной арифметикой

В. О. Трухин[1,2,a], В. С. Стронгин[1,2], Э. А. Лобанова[1,2], А. Г. Макаров[1,2], К. В. Нефедев[1,2]

[1]Департамент теоретической физики и интеллектуальных технологий, Институт наукоемких технологий и передовых материалов, Дальневосточный федеральный университет,
Россия, 690922, г. Владивосток, о. Русский, п. Аякс, д. 10
[2]Институт прикладной математики, Дальневосточное отделение Российской академии наук,
Россия, 690041, г. Владивосток, ул. Радио, д. 7

E-mail: [a] slavatruhin@gmail.com

Получено 20.02.2026.
Принято к публикации 28.04.2026.

В работе рассматривается применение китайской теоремы об остатках (КТО) для повышения эффективности вычислений с большими целыми числами в задачах высокопроизводительных вычислений. Актуальность исследования обусловлена тем, что в ряде задач теоретической физики, численного моделирования и переборных алгоритмов используются числа разрядности, существенно превышающей возможности стандартных машинных типов данных, что приводит к необходимости применения длинной арифметики и сопровождается значительными вычислительными затратами. Традиционные методы представления длинных чисел в виде последовательностей слов плохо масштабируются и недостаточно эффективно используют архитектуру графических процессоров (GPU).

Предложенный подход основан на представлении числа в системе вычетов по набору попарно взаимно простых модулей. Такое разложение позволяет заменить операции над многобайтными числами на независимые модульные операции над массивами остатков, что обеспечивает возможность их естественного распараллеливания. Арифметические операции сложения, вычитания и умножения выполняются покомпонентно в модульной системе, а восстановление исходного числа осуществляется с использованием алгоритмов, основанных на расширенном алгоритме Евклида. Показано, что при последовательных вычислениях предложенный метод не имеет существенного преимущества по сравнению с классической длинной арифметикой, однако при реализации на GPU достигается значительный рост производительности за счет массового параллелизма и независимости операций по модулям. Дополнительным достоинством является снижение риска переполнения и более эффективная организация доступа к памяти благодаря использованию чисел фиксированной разрядности.

Таким образом, использование КТО представляет собой эффективный инструмент для реализации многоточечных вычислений с большими числами в суперкомпьютерных системах и может применяться в задачах численного моделирования, криптографии и алгоритмах полного перебора, требующих высокой производительности и масштабируемости.

Ключевые слова: большие числа, суперкомпьютеры, модульная арифметика, параллельные вычисления

Исследование выполнено за счет гранта Российского научного фонда № 24-71-10069 (https://rscf.ru/project/24-71-10069).

© 2026 Вячеслав Олегович Трухин, Владислав Сергеевич Стронгин, Элиза Александровна Лобанова, Александр Геннадьевич Макаров, Константин Валентинович Нефедев
Статья доступна по лицензии Creative Commons Attribution-NoDerivs 3.0 Unported License.
Чтобы получить текст лицензии, посетите веб-сайт http://creativecommons.org/licenses/by-nd/3.0/
или отправьте письмо в Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

COMPUTER RESEARCH AND MODELING
2026 VOL. 18 NO. 5 P. 1121–1128
DOI: 10.20537/2076-7633-2026-18-5-1121-1128

Ки&М

NUMERICAL METHODS AND THE BASIS FOR THEIR APPLICATION

UDC: 519.8

Application of the Chinese Remainder Theorem to multi-precision computations

V. O. Trukhin[1,2,a], V. S. Strongin[1,2], E. A. Lobanova[1,2], A. G. Makarov[1,2], K. V. Nefedev[1,2]

[1]Department of theoretical physics and intellectual technologies, Institute of scientific and advanced materials, Far eastern federal university,
10 Ajax, Russky Island, Vladivostok, 690922, Russia
[2]Institute of Applied Mathematics, Far Eastern Branch, Russian Academy of Sciences,
7 Radio st., Vladivostok, 690041, Russia

E-mail: [a] slavatruhin@gmail.com

Received 20.02.2026.
Accepted for publication 28.04.2026.

This work investigates the application of the Chinese Remainder Theorem (CRT) to improve the efficiency of computations involving large integers in high-performance computing tasks. The relevance of this study arises from the fact that a number of problems in theoretical physics, numerical simulation, and exhaustive search algorithms require integers whose bit length significantly exceeds the capabilities of standard machine data types. This necessitates the use of arbitrary-precision arithmetic and leads to substantial computational overhead. Traditional representations of large integers as sequences of machine words scale poorly and do not efficiently utilize the architecture of graphics processing units (GPUs).

The proposed approach is based on representing integers in a residue number system defined by a set of pairwise coprime moduli. Such a decomposition replaces operations on multi-byte integers with independent modular operations on arrays of residues, enabling natural parallelization. Arithmetic operations including addition, subtraction, and multiplication are performed componentwise within the modular system, while reconstruction of the original integer is carried out using algorithms based on the extended Euclidean algorithm. It is shown that in sequential computations the proposed method does not provide a significant advantage over classical arbitrary-precision arithmetic; however, when implemented on GPUs, a substantial performance gain is achieved due to massive parallelism and independence of operations across moduli. An additional advantage is the reduced risk of overflow and more efficient memory access organization enabled by the use of fixed-width integers.

Thus, the use of the CRT constitutes an effective tool for implementing multipoint computations with large integers in supercomputing systems and can be applied to numerical simulation, cryptography, and exhaustive search algorithms requiring high performance and scalability.

Keywords: multi-precision arithmetic, supercomputers, modular arithmetic, parallel computing

Citation: *Computer Research and Modeling*, 2026, vol. 18, no. 5, pp. 1121–1128 (Russian).

The research was supported by the Russian Science Foundation grant No. 24-71-10069 (https://rscf.ru/en/project/24-71-10069).

© 2026 Viacheslav O. Trukhin, Vladislav S. Strongin, Eliza A. Lobanova, Aleksandr G. Makarov, Konstantin V. Nefedev
This work is licensed under the Creative Commons Attribution-NoDerivs 3.0 Unported License.
To view a copy of this license, visit http://creativecommons.org/licenses/by-nd/3.0/
or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Введение

В ряде задач теоретической физики, дискретного моделирования и переборных алгоритмов требуется выполнение вычислений с целыми числами сверхбольшой разрядности [Hao et al., 2022; Makarova et al., 2023; Trukhin et al., 2024; Trukhin et al., 2025; Васильев и др., 2020]. Эти вычисления необходимы при моделировании статистических систем, обработке больших конфигурационных пространств, анализе комбинаторных структур и реализации криптографических и факторизационных алгоритмов. Рост объемов данных и усложнение моделей приводят к тому, что разрядность используемых чисел существенно превышает возможности стандартных машинных типов, что делает применение методов длинной арифметики неизбежным.

Классические подходы к реализации длинной арифметики основаны на представлении числа в виде последовательности машинных слов и использовании библиотек произвольной точности. Несмотря на универсальность, такие методы слабо масштабируются на современные архитектуры высокопроизводительных вычислений. Их ключевым ограничением является последовательный характер обработки разрядов, что препятствует эффективному использованию массивного параллелизма графических процессоров (GPU) и других ускорителей, являющихся сегодня основной вычислительной платформой в задачах численного моделирования. Таким образом, возникает актуальная задача разработки альтернативных представлений длинных чисел, ориентированных не на последовательную, а на параллельную модель вычислений.

Одним из перспективных направлений является использование модульной арифметики, совместно с китайской теоремой об остатках (КТО) [Окулов, Лялин, 2011; Cormen et al., 2022]. Представление числа в виде набора независимых вычетов по взаимно простым модулям позволяет заменить операции над целыми числами большой разрядности на совокупность более простых модульных операций фиксированной разрядности. Такая форма записи естественным образом согласуется с архитектурой GPU, где вычисления выполняются большим числом потоков над компактными типами данных. Несмотря на известность КТО в вычислительной математике, ее потенциал как инструмента организации массового параллелизма в задачах длинной арифметики остается недостаточно исследованным применительно к современным ускорительным архитектурам.

В работе предлагается алгоритмическая схема выполнения арифметических операций с числами большой разрядности, адаптированная к параллельной реализации на GPU, а также анализируются особенности выбора модульного базиса, восстановления результата и влияния этих факторов на производительность.

Предложенный подход может служить вычислительным ядром для широкого класса задач, где критическими являются одновременно большая разрядность данных и высокая степень параллелизма: численного моделирования физических систем, криптографических приложений, переборных алгоритмов и других вычислительно трудоемких исследований.

Алгоритм работы с длинной арифметикой

В отличие от традиционной длинной арифметики, где операции выполняются над последовательностями машинных слов с переносами между разрядами, использование китайской теоремы об остатках позволяет организовать вычисления в виде набора независимых модульных операций. Такая декомпозиция устраняет межразрядные зависимости и тем самым создает предпосылки для эффективного распараллеливания.

Реализация данного подхода требует решения трех взаимосвязанных задач: выбора системы взаимно простых модулей, обеспечивающей необходимый динамический диапазон представимых чисел; организации арифметических операций над массивами вычетов, а также восста-

новления результата в позиционной системе счисления. Эти этапы образуют полный вычислительный цикл работы с числами большой разрядности в рамках предложенного метода.

Принцип работы алгоритма, основой которого является КТО, можно разбить на три этапа:

1) разбиение большого числа на составные части (набор остатков);

2) математические операции с массивами остатков от деления на простые числа;

3) расчет результата из массива остатков от деления на простые числа.

Далее рассмотрим каждый пункт более подробно.

Разбиение большого числа на набор отстатков от делений

Согласно китайской теореме об остатках [Cormen et al., 2022; Сизый, 2008] пусть $a = a_1 a_2 \dots a_k$, где a_i — попарно взаимно простые натуральные числа, $r = r_1 r_2 \dots r_k$ — натуральные числа, удовлетворяющие неравенству $0 \leqslant r_i < a_i$. Тогда найдется такое число L, для которого для всех i выполняется

$$r_i = L \bmod a_i. \tag{1}$$

Из теоремы следует, что для представления числа X необходимо вычислить его остатки по набору попарно взаимно простых модулей $a_1, a_2, \dots, a_k$. Простота самих модулей не является обязательным условием: требуется лишь выполнение соотношения

$$\gcd(a_i, a_j) = 1, \quad i \neq j. \tag{2}$$

Произведение модулей определяет динамический диапазон системы остатков:

$$M = \prod_{i=1}^{k} a_i, \quad M \geqslant X. \tag{3}$$

В настоящей реализации в качестве модулей используются различные простые числа. Такой выбор автоматически обеспечивает их попарную взаимную простоту и упрощает формирование модульного базиса. Модули выбираются максимально близкими к верхней границе используемого типа данных до тех пор, пока их произведение M не превысит требуемый диапазон представляемых чисел.

В качестве модулей также могут использоваться степени простых чисел $a_i = p_i^{s_i}$ при условии, что основания p_i для различных модулей не совпадают. Например, числа $125 = 5^3$ и $121 = 11^2$ являются взаимно простыми и могут одновременно входить в модульный базис. Напротив, различные степени одного простого числа не являются взаимно простыми и не могут одновременно использоваться в классической форме КТО.

Существует несколько подходов к выбору простых чисел для разбиения.

Первый подход состоит в выборе минимального количества простых чисел, произведение которых превосходит M, и которые помещаются в максимальный целочисленный тип данных, предоставляемый системой (2^{32} или 2^{64}).

Второй подход состоит в использовании модулей, остатки по которым могут храниться в компактном типе данных, например `int8`. Это позволяет уменьшить объем памяти, необходимый для хранения модульного представления, и снизить нагрузку на подсистему памяти GPU. При использовании знакового типа `int8` в данной реализации выбираются положительные модули $a_i < 128$. Промежуточные результаты арифметических операций вычисляются в типе большей разрядности, после чего приводятся по соответствующему модулю.

Математические операции с массивами остатков

Модульная арифметика [Omondi, Premkumar, 2007; Soderstrand et al., 1986] определяет операции сложения, вычитания и умножения следующим образом: если два числа — X_1 и X_2 — представимы в виде систем остатков $(r_{11}, r_{12}, \ldots, r_{1n})$ и $(r_{21}, r_{22}, \ldots, r_{2n})$ на модули $(a_1, a_2, \ldots, a_n)$:

$$X_1 + X_2 = ((r_{11} + r_{21}) \bmod a_1, (r_{12} + r_{22}) \bmod a_2, \ldots, (r_{1n} + r_{2n}) \bmod a_n), \quad (4)$$

$$X_1 - X_2 = ((r_{11} - r_{21}) \bmod a_1, (r_{12} - r_{22}) \bmod a_2, \ldots, (r_{1n} - r_{2n}) \bmod a_n), \quad (5)$$

$$X_1 \cdot X_2 = ((r_{11} \cdot r_{21}) \bmod a_1, (r_{12} \cdot r_{22}) \bmod a_2, \ldots, (r_{1n} \cdot r_{2n}) \bmod a_n), \quad (6)$$

то есть чтобы сложить, вычесть или умножить два числа, достаточно сложить, вычесть или умножить соответствующие элементы векторов остатков этих двух чисел.

Поскольку $r_i \in (\mathbb{N} \cup 0)$, операция деления не определена для модульной арифметики в явном виде: целочисленное деление нельзя выполнить независимо над отдельными остатками, поскольку представление в системе остаточных классов не сохраняет информацию о порядке и величине чисел. Для выполнения деления требуются специализированные алгоритмы либо восстановление исходных чисел. Это ограничение накладывает определенные сложности на использование КТО в задачах, где деление является ключевой операцией. В таких случаях необходимо либо использовать дополнительные методы для восстановления исходных чисел, либо применять алгоритмы, которые обходятся без явного деления.

Восстановление числа из массива остатков

Для восстановления исходного числа необходимо для каждого $i = 1, \ldots, k$ найти число

$$^1B_i = M_i^{-1} \bmod a_i, \quad (7)$$

то есть модульно обратный элемент для M_i по модулю a_i. Поскольку модули $a_1, \ldots, a_k$ попарно взаимно просты, выполняется

$$\gcd(M_i, a_i) = 1, \quad M_i = \frac{M}{a_i}. \quad (8)$$

Следовательно, обратный элемент ^1B_i существует.

Для его вычисления применяется расширенный алгоритм Евклида [Окулов, Лялин, 2011]. Для двух взаимно простых натуральных чисел u и v алгоритм строит последовательности R_j, S_j и T_j, для которых на каждой итерации выполняется равенство

$$R_j = S_j u + T_j v. \quad (9)$$

Начальные значения задаются следующим образом:

$$R_0 = u, \quad R_1 = v, \quad S_0 = 1, \quad S_1 = 0, \quad T_0 = 0, \quad T_1 = 1. \quad (10)$$

На каждой итерации вычисляется целая часть частного

$$q_j = \left\lfloor \frac{R_{j-1}}{R_j} \right\rfloor, \quad (11)$$

после чего последовательности обновляются согласно алгоритму 1. В рассматриваемой задаче полагается $u = M_i$ и $v = a_i$.

Для того чтобы расшифровать число обратно, используется алгоритм 2.

В алгоритме 2 для каждого модуля a_i вычисляется произведение остатка r_i, величины M_i и соответствующего первого коэффициента Безу ^1B_i. Сумма полученных произведений, взятая по модулю M, равна исходному числу.

Алгоритм 1. Вычисление обратного элемента с использованием расширенного алгоритма Евклида

Ввод: взаимно простые натуральные числа u и v.
Вывод: обратный элемент ${}^1B = u^{-1} \bmod v$.

$R_0 = u$, $R_1 = v$
$S_0 = 1$, $S_1 = 0$
$T_0 = 0$, $T_1 = 1$
$j = 1$
while $R_j \neq 0$ **do**
  $q_j = \left\lfloor \frac{R_{j-1}}{R_j} \right\rfloor$
  $R_{j+1} = R_{j-1} - q_j R_j$
  $S_{j+1} = S_{j-1} - q_j S_j$
  $T_{j+1} = T_{j-1} - q_j T_j$
  $j = j + 1$
end while
${}^1B = S_{j-1} \bmod v$

Алгоритм 2. Восстановление числа из простых чисел и остатков от делений на них

Ввод: массивы простых чисел $a[\,]$ и остатков $r[\,]$ длины k.
Вывод: расшифрованное число.

Рассчитать $M = \prod_{i=1}^{k} a_i$
Рассчитать $M_i = \frac{M}{a_i}$
Для каждого $i = 1, \ldots, k$ с использованием алгоритма 1 рассчитать ${}^1B_i = M_i^{-1} \bmod a_i$
Рассчитать искомое число $x = \left(\sum_{i=1}^{k} r_i M_i {}^1B_i \right) \bmod M$

Производительность

На рис. 1 показана производительность операции сложения массивов чисел фиксированной разрядности порядка 2^{100}. По оси абсцисс отложена длина массива (число обрабатываемых элементов), по оси ординат — производительность, выраженная в числе операций сложения в секунду, представленная в логарифмическом масштабе.

Левый график соответствует последовательному (однопоточному) режиму выполнения. Видно, что в этом случае предложенный метод на основе китайской теоремы об остатках не демонстрирует преимущества по сравнению с классической длинной арифметикой, реализованной в библиотеке GMP. Это объясняется тем, что при отсутствии параллелизма затраты на запуск ядра GPU компенсируют преимущество использования более простых модульных операций.

Правый график иллюстрирует результаты при параллельном выполнении вычислений. В этом режиме метод КТО демонстрирует кратное преимущество, которое возрастает с увеличением длины массива. Для больших объемов данных производительность предложенного подхода превышает показатели реализации на базе GMP более чем в 100 раз. Такой рост производительности обусловлен независимостью вычислений как по элементам массива, так и по модулям, что позволяет эффективно задействовать вычислительные ресурсы GPU.

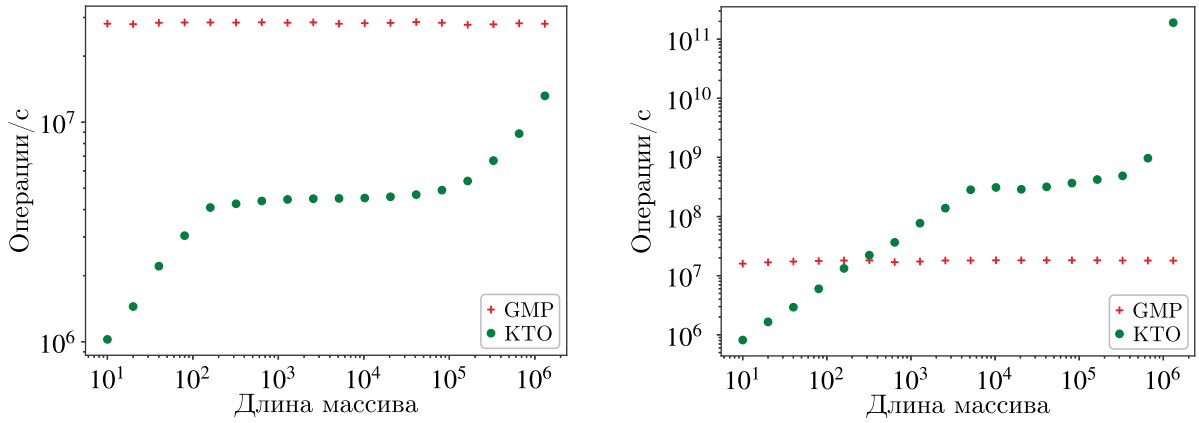


Рис. 1. Производительность операции сложения на одном потоке (слева) и при параллельном вычислении (справа)

Заключение

Предлагаемый подход обладает рядом преимуществ, которые делают его эффективным для использования в вычислениях на GPU.

Во-первых, независимые параллельные вычисления с использованием алгоритма КТО достигаются за счет работы с отдельными остатками. Это существенно ускоряет обработку массивов остатков на разных потоках или ядрах GPU в переборном алгоритме декомпозиции [Трухин и др., 2025].

Вторым важным аспектом является решение проблемы переполнения.

Кроме того, использование КТО способствует оптимизации доступа к памяти GPU. Применение модульной арифметики позволяет размещать данные в оптимальном для архитектуры GPU формате, что улучшает производительность ввода-вывода и снижает задержки доступа к памяти.

Несмотря на преимущества, подход с использованием КТО имеет ряд недостатков. Восстановление исходного числа после операций требует значительных ресурсов и может замедлить работу из-за синхронизации на GPU. Алгоритм с использованием КТО является избыточным при работе с числами, сопоставимым с простыми числами, используемыми для кодировки.

Таким образом, алгоритм с использованием КТО является мощным инструментом для работы с многоразрядными числами на GPU. Его применение позволяет эффективно реализовывать параллелизм, избежать переполнения и оптимизировать использование памяти. В задачах, требующих высокопроизводительных вычислений, таких как алгоритмы полного перебора и численные расчеты, КТО может существенно ускорить процесс обработки данных и повысить производительность системы.

Список литературы (References)

Васильев Е. В., Пержу А. В., Король А. О., Капитан Д. Ю., Рыбин А. Е., Солдатов К. С., Капитан В. Ю. Численное моделирование двумерных магнитных скирмионных структур // Компьютерные исследования и моделирование. — 2020. — Т. 12, № 5. — С. 1051–1061.
Vasil'ev E. V., Perzhu A. V., Korol' A. O., Kapitan D. Yu., Rybin A. E., Soldatov K. S., Kapitan V. Yu. Chislennoe modelirovanie dvumernykh magnitnykh skirmionnykh struktur [Numerical simulation of two-dimensional magnetic skyrmion structures] // Computer Research and Modeling. — 2020. — Vol. 12, No. 5. — P. 1051–1061 (in Russian).

Окулов С. М., Лялин А. В. Расширенный алгоритм Евклида // Информатика и образование. — 2011. — № 8. — С. 37–41.
Okulov S. M., Lyalin A. V. Rasshirennyi algoritm Evklida [Extended Euclidean algorithm] // Informatika i obrazovanie. — 2011. — No. 8. — P. 37–41 (in Russian).

Сизый С. В. Лекции по теории чисел. — 2-е изд., испр. — М.: Физматлит, 2008. — 192 с.
Sizyi S. V. Lektsii po teorii chisel [Lectures on number theory]. — 2nd ed., revised. — Moscow: Fizmatlit, 2008. — 192 p. (in Russian).

Трухин В. О., Лобанова Э. А., Анисич А. И., Макарова К. В., Макаров А. Г., Нефедев К. В. Переборный алгоритм декомпозиции для решения модели Изинга // Дальневосточный математический журнал. — 2025. — Т. 25, № 1. — С. 102–112.
Trukhin V. O., Lobanova E. A., Anisich A. I., Makarova K. V., Makarov A. G., Nefedev K. V. Perebornyi algoritm dekompozitsii dlya resheniya modeli Izinga [An enumeration decomposition algorithm for solving the Ising model] // Dal'nevostochnyi matematicheskii zhurnal. — 2025. — Vol. 25, No. 1. — P. 102–112 (in Russian).

Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to algorithms. — MIT Press, 2022.

Hao Y., Zhao Y., Liu C., Du Z., Cheng S., Li X., Hu X., Guo Q., Xu Z., Chen T. Cambricon-p: A bitflow architecture for arbitrary precision computing // 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO). — IEEE, 2022. — P. 57–72.

Makarova K., Makarov A., Strongin V., Titovets I., Shevchenko Y., Kapitan V., Rybin A., Kapitan D., Korol A., Vasiliev E., Ovchinnikov P., Soldatov K., Trukhin V., Nefedev K. Canonical Monte Carlo multispin cluster method // Journal of Computational and Applied Mathematics. — 2023. — Vol. 427. — P. 115153.

Omondi A. R., Premkumar A. B. Residue number systems: theory and implementation. — World Scientific, 2007. — Vol. 2.

Soderstrand M. A., Jenkins W. K., Jullien G. A., Taylor F. J. Residue number system arithmetic: modern applications in digital signal processing. — IEEE Press, 1986.

Trukhin V., Prokhorov E., Makarov A., Nefedev K. Low-temperature states of the Ising $\pm J$ model on a square lattice // Physica A: Statistical Mechanics and its Applications. — 2025. — Vol. 674. — P. 130729.

Trukhin V., Strongin V., Chesnokov M., Makarov A., Lobanova E., Shevchenko Y., Nefedev K. Thermodynamic equilibrium of $\pm J$ Ising model on square lattice // Physica A: Statistical Mechanics and its Applications. — 2024. — Vol. 655. — P. 130172.