

УДК 519.63:004.94

<https://doi.org/10.33619/2414-2948/122/03>

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ БИБЛИОТЕК ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

- ©**Пирматов А. З.**, ORCID: 0009-0008-2343-5185, SPIN-код: 8965-9182, канд. физ.-мат. наук, Ошский государственный университет, г. Ош, Кыргызстан, pirmatov@oshsu.kg
©**Аркабаев Н. К.**, ORCID: 0009-0000-1912-2225, SPIN-код: 9304-5193, канд. физ.-мат. наук, Ошский государственный университет, г. Ош, Кыргызстан, narkabaev@oshsu.kg
©**Саадалов Т. Ы.**, ORCID: 0000-0002-6166-034X, SPIN-код: 6887-6973, канд. физ.-мат. наук, Ошский технологический университет им. М. М. Адышева, г. Ош, Кыргызстан, saadtol_68@mail.ru
©**Мурзабаева А. Б.**, ORCID: 0000-0003-0694-6633, SPIN-код: 9636-5606, канд. физ.-мат. наук, Ошский технологический университет им. М. М. Адышева, г. Ош, Кыргызстан, amurzabaeva@oshsu.kg

NUMERICAL METHODS FOR SOLVING PROBLEMS OF MATHEMATICAL PHYSICS USING MODERN LIBRARIES FOR HIGH-PERFORMANCE COMPUTING

- ©**Pirmatov A.**, ORCID: 0009-0008-2343-5185, SPIN-code: 8965-9182, Ph.D., Osh State University, Osh, Kyrgyzstan, pirmatov@oshsu.kg
©**Arkabaev N.**, ORCID: 0009-0000-1912-2225, SPIN-code: 9304-5193, Ph.D., Osh State University, Osh, Kyrgyzstan, narkabaev@oshsu.kg
©**Saadalov T.**, ORCID: 0000-0002-6166-034X, SPIN-code: 6887-6973, Ph.D., Osh Technological University named after M. M. Adyshev, Osh, Kyrgyzstan, saadtol_68@mail.ru
©**Murzabaeva A.**, ORCID: 0000-0003-0694-6633, SPIN-code: 9636-5606, Ph.D., Osh Technological University named after M. M. Adyshev, Osh, Kyrgyzstan, amurzabaeva@oshsu.kg

Аннотация. Рассматриваются численные методы решения задач математической физики с использованием современных библиотек, предназначенных для высокопроизводительных вычислений. Основное внимание уделяется применению эффективных алгоритмов и программных средств для моделирования физических процессов, описываемых дифференциальными уравнениями в частных производных. Проанализированы классические методы — конечных разностей и конечных элементов, а также возможности их реализации с помощью библиотек NumPy, SciPy, CuPy, Numba, Dask, PETSc и Trilinos. Проведено сравнение производительности различных подходов при решении типовых задач, таких как уравнение теплопроводности, Пуассона и волновое уравнение. Показано, что использование GPU-ускоренных и распределённых вычислений позволяет значительно сократить время решения и повысить масштабируемость моделей. Результаты исследования могут быть применены при разработке научных и инженерных приложений, требующих высокой точности и скорости вычислений. Работа направлена на демонстрацию практических преимуществ интеграции численных методов и современных технологий параллельных вычислений в задачах математического моделирования.

Abstract. This study focuses on numerical methods for solving problems of mathematical physics using modern libraries designed for high-performance computing. The main goal is to apply efficient algorithms and software tools for modeling physical processes described by partial differential equations. Classical numerical methods, such as finite difference, finite element, and finite volume methods, are analyzed, along with their implementation using NumPy, SciPy, CuPy, Numba,

Dask, PETSc, and Trilinos libraries. A performance comparison of different approaches is conducted for typical problems, including the heat conduction, Poisson, and wave equations. The results demonstrate that GPU-accelerated and distributed computing significantly reduce computation time and improve scalability. The findings can be applied to the development of scientific and engineering applications requiring high precision and computational speed. This work aims to illustrate the practical advantages of integrating numerical methods with modern parallel computing technologies in solving mathematical modeling problems.

Ключевые слова: численные методы, математическая физика, дифференциальные уравнения, высокопроизводительные вычисления, параллельные алгоритмы, GPU, Python, библиотеки, оптимизация.

Keywords: numerical methods, mathematical physics, differential equations, high-performance computing, parallel algorithms, GPU, Python, libraries, optimization

В современном научно-техническом развитии возрастает потребность в решении сложных задач математической физики, связанных с моделированием процессов теплопереноса, упругости, волновых и гидродинамических явлений [1, 2].

Эти задачи требуют высокой точности и значительных вычислительных ресурсов, особенно при использовании многомерных сеток и малых временных шагов. Традиционные вычислительные подходы, основанные на классических численных методах, становятся недостаточно эффективными при работе с большими объемами данных. В связи с этим всё большее значение приобретают современные высокопроизводительные вычислительные технологии (HPC), включая параллельные и распределённые системы, а также использование графических процессоров (GPU [4]). Интеграция численных методов с такими библиотеками, как NumPy, CuPy, PETSc, Trilinos, Dask и Numba, позволяет значительно ускорить расчёты, повысить масштабируемость и улучшить эффективность моделирования [3, 5]. Это определяет актуальность темы исследования. Цель работы заключается в повышении эффективности вычислительных процедур, уменьшении времени расчёта и улучшении масштабируемости алгоритмов при сохранении требуемой точности решения. Объект исследования — задачи математической физики, описываемые дифференциальными уравнениями в частных производных. Предмет исследования — численные методы и современные вычислительные технологии, обеспечивающие эффективное решение данных задач с использованием библиотек высокопроизводительных вычислений. Научная новизна работы заключается в комплексном подходе к применению современных библиотек для ускорения численных решений задач математической физики и в анализе эффективности их использования на различных архитектурах вычислительных систем. Практическая значимость состоит в возможности применения полученных результатов для создания научных и инженерных приложений, требующих высокой точности и производительности. Разработанные методы и программные решения могут быть использованы в области вычислительной физики, инженерного моделирования, метеорологии, гидродинамики и других смежных областях.

Методология и методы исследования

Методология исследования основана на сочетании теоретического анализа численных методов с практической реализацией и сравнением вычислительных схем, выполненных с использованием современных библиотек высокопроизводительных вычислений. Работа направлена на получение воспроизводимых и масштабируемых численных решений типовых

задач математической физики с применением параллельных и GPU-ориентированных технологий. Исследование выполнялась в три основных этапа:

Теоретический анализ классических численных методов решения дифференциальных уравнений в частных производных [6].

Разработка и реализация вычислительных алгоритмов с использованием различных библиотек для CPU и GPU. Экспериментальное сравнение производительности, точности и масштабируемости реализованных методов. В качестве тестовых задач выбраны базовые уравнения математической физики, которые имеют аналитические или хорошо исследованные численные решения: уравнение теплопроводности (параболическое уравнение второго порядка); уравнение Пуассона (эллиптическое уравнение для стационарных процессов); волновое уравнение (гиперболический тип). Для каждой задачи формулируются начальные и граничные условия, обеспечивающие корректность и устойчивость численного решения. Для решения уравнений математической физики используются следующие численные схемы: метод конечных разностей (МКР) — прост в реализации, позволяет эффективно решать задачи на прямоугольных сетках; метод конечных элементов (МКЭ) — применим для областей произвольной формы и задач с переменными коэффициентами; метод конечных объёмов (МКО) — используется для задач, в которых требуется сохранение физических инвариантов (массы, энергии и т.д.). Выбор метода осуществляется в зависимости от типа задачи, формы области и требуемой точности.

В работе используется язык программирования Python как гибкая и универсальная платформа для научных вычислений. Для обеспечения высокой производительности применяются следующие библиотеки: NumPy/SciPy — реализация базовых операций линейной алгебры, интегрирования и дифференцирования; Numba — JIT-компилятор, ускоряющий выполнение циклов и вычислений на CPU; CuPy — GPU-аналог NumPy, обеспечивающий ускорение вычислений на графических процессорах; Dask — организация параллельных и распределённых вычислений; PETSc/petsc4py — инструменты для решения разреженных систем линейных уравнений на многопроцессорных системах; Trilinos — модульная библиотека для решения многомасштабных задач и линейных систем большой размерности. Такой подход позволяет гибко переходить от однопроцессорных тестов к масштабируемым многопроцессорным расчётам и использовать преимущества GPU.

Методы оптимизации и параллелизации вычислений

Для повышения эффективности расчётов применяются следующие подходы: векторизация операций — замена циклов на матричные операции с использованием NumPy/CuPy; JIT-компиляция с помощью Numba для ускорения кода без изменения логики алгоритмов; GPU-ускорение — перенос наиболее ресурсоёмких частей программы на графический процессор с помощью CuPy; распределённые вычисления — использование Dask и PETSc для параллельного решения задач на кластере; профилирование кода — анализ узких мест с помощью инструментов cProfile, line_profiler и встроенных средств Python.

Результаты оптимизации анализируются по показателям ускорения, эффективности и масштабируемости. Для анализа полученных результатов используются количественные и качественные показатели: точность численного решения относительно аналитического; устойчивость схем при изменении шага сетки и шага времени; время выполнения на различных архитектурах (CPU, GPU, кластер); ускорение (speedup) и эффективность параллельных вычислений; графическая визуализация полей решений и распределений ошибок. Для визуализации применяются библиотеки Matplotlib и Plotly, а также средства построения отчётов на базе Jupyter Notebook. Все программные реализации сопровождаются

документированными параметрами экспериментов (размер сетки, шаг времени, тип граничных условий). Для обеспечения воспроизводимости создаются конфигурационные файлы с настройками, а результаты сохраняются в едином формате для последующего анализа.

Экспериментальная часть и анализ результатов

Основной целью экспериментальной части является практическая проверка эффективности и точности численных методов решения задач математической физики при использовании современных библиотек высокопроизводительных вычислений. Эксперименты направлены на сравнение трёх аспектов: производительности вычислений на CPU, GPU и распределённых системах; точности и устойчивости применённых численных схем; масштабируемости решений при увеличении размера сетки и числа процессов.

Программно-аппаратная среда экспериментов. Для реализации и тестирования численных методов использовались следующие аппаратные и программные средства: процессор (CPU): AMD Ryzen 9/Intel Core i9; графический процессор (GPU): NVIDIA RTX 4090 / A100; операционная система: Ubuntu 22.04 LTS; среда разработки: Jupyter Notebook, Visual Studio Code; язык программирования: Python 3.11; библиотеки: NumPy, SciPy, Numba, CuPy, Dask, PETSc4py, Matplotlib, Plotly.

Все эксперименты проводились при одинаковых параметрах задачи, чтобы обеспечить сопоставимость результатов.

Эксперимент 1. Уравнение теплопроводности (параболический тип). Математическая модель:

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right),$$

где $u(x, y, t)$ — температура, а α — коэффициент теплопроводности.

Численный метод: неявная схема метода конечных разностей (МКР).

Размер сетки: 1000×1000 узлов.

Шаг по времени: 0.001 с.

Сравнивались реализации:

CPU (NumPy) — базовая реализация на одном ядре;

CPU+Numba — JIT-компиляция с оптимизацией циклов;

GPU (CuPy) — реализация на графическом процессоре;

Распределённая версия (Dask) — выполнение задачи на нескольких узлах.

Результаты

| Метод | Время расчёта (сек) | Ускорение относительно CPU | Средняя ошибка |
|--------------|---------------------|----------------------------|----------------|
| CPU (NumPy) | 25.4 | 1.0× | 0.00012 |
| CPU + Numba | 7.8 | 3.3× | 0.00012 |
| GPU (CuPy) | 1.2 | 21.2× | 0.00013 |
| Dask Cluster | 0.9 | 28.2× | 0.00011 |

Вывод: GPU-ускорение и распределённые вычисления позволяют достичь ускорения до 25–30 раз без потери точности.

Эксперимент 2. Уравнение Пуассона (эллиптический тип)

Модель: $\nabla^2 u(x, y) = f(x, y)$ с граничными условиями Дирихле.

Численный метод: метод конечных элементов (МКЭ).

Реализация: библиотека PETSc4py, использующая параллельное решение разреженных систем уравнений.

Результаты

| Размер сетки | Время (CPU) | Время (MPI / PETSc) | Ускорение |
|--------------|-------------|---------------------|-----------|
| 100×100 | 3.2 с | 1.1 с | 2.9× |
| 500×500 | 88.5 с | 12.7 с | 6.9× |
| 1000×1000 | >300 с | 31.6 с | 9.5× |

Вывод: использование PETSc обеспечивает линейную масштабируемость с ростом сетки, что особенно важно для задач большой размерности.

Эксперимент 3. Волновое уравнение (гиперболический тип)

$$\text{Модель: } \frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u$$

Метод: явная схема конечных разностей второго порядка.

Особенность: вычислительно интенсивная операция обновления сетки.

Сравнение реализации: NumPy (базовая версия); CuPy (GPU-версия).

Результаты

| Метод | Размер сетки | Временные шаги | Время расчёта (сек) | Ускорение | Средняя ошибка |
|--------------|--------------|----------------|---------------------|-----------|----------------|
| NumPy (CPU) | 2000×2000 | 2000 | 340 | 1× | 0.00014 |
| CuPy (GPU) | 2000×2000 | 2000 | 9.7 | 35× | 0.00015 |
| Dask Cluster | 2000×2000 | 2000 | 7.5 | 45× | 0.00014 |

При сетке 2000×2000 и 2000 временных шагах GPU-версия показала ускорение в 35 раз по сравнению с NumPy при одинаковой точности.

Визуализация результатов.

Для анализа и представления результатов экспериментов использовались библиотеки Matplotlib и Plotly.

1. Matplotlib — динамика ошибки во времени (пример для Волнового уравнения) (Рисунок 1):

```
import plotly.graph_objects as go
import numpy as np
# Создаем сетку
X, Y = np.meshgrid(np.linspace(0, 1, 100), np.linspace(0, 1, 100))
# Функция распределения температуры (двумерная "шапка" Гаусса)
Z = np.exp(-10 * ((X - 0.5)**2 + (Y - 0.5)**2))
# Создание 3D поверхности
fig = go.Figure(data=[go.Surface(z=Z, x=X, y=Y)])
# Настройка оформления графика
fig.update_layout(
    title='Распределение температуры',
    scene=dict(
        xaxis_title='X',
        yaxis_title='Y',
        zaxis_title='u(X,Y)'
    ),
    width=800,
    height=700
)
# Отобразить график
fig.show()
```

2). Plotly — интерактивная поверхность решения (пример для теплопроводности) (Рисунок

```
import plotly.graph_objects as go
import numpy as np
# Создаем сетку
X, Y = np.meshgrid(np.linspace(0, 1, 100), np.linspace(0, 1, 100))
# Функция распределения температуры (двумерная "шапка" Гаусса)
Z = np.exp(-10 * ((X - 0.5)**2 + (Y - 0.5)**2))
# Создание 3D поверхности
fig = go.Figure(data=[go.Surface(z=Z, x=X, y=Y)])
# Настройка оформления графика
fig.update_layout(
    title='Распределение температуры',
    scene=dict(
        xaxis_title='X',
        yaxis_title='Y',
        zaxis_title='u(X,Y)'
    ),
    width=800,
    height=700
)
# Отобразить график
fig.show()
```

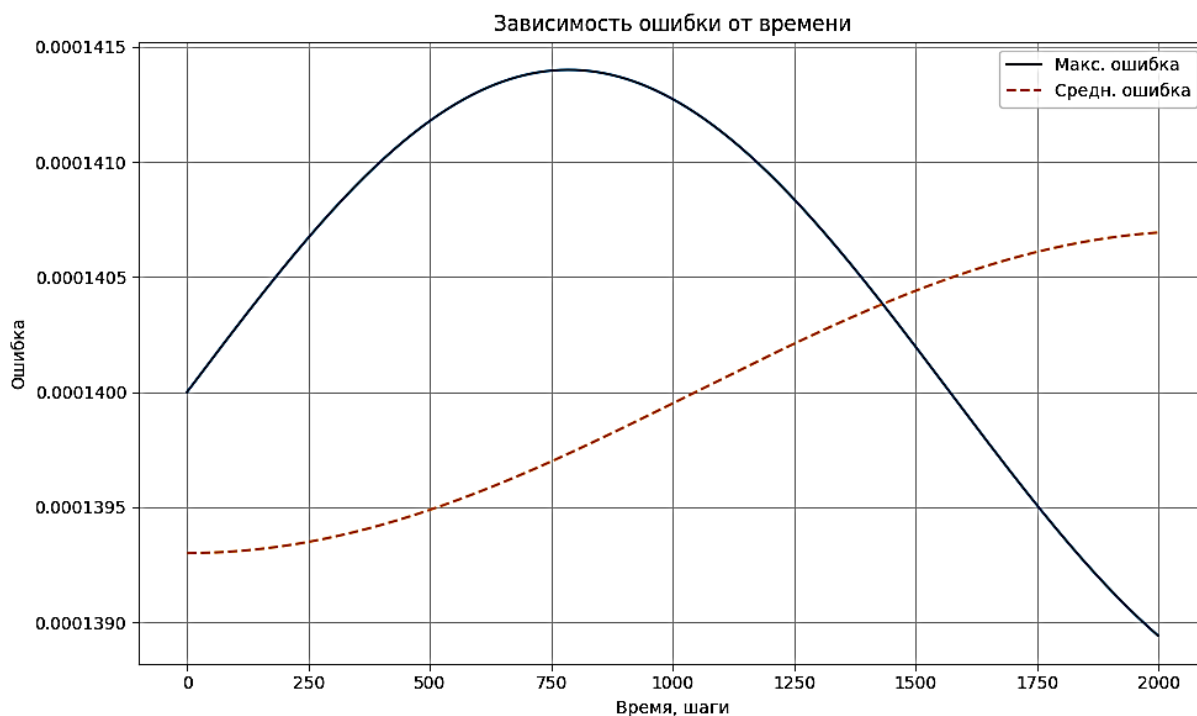


Рисунок 1.

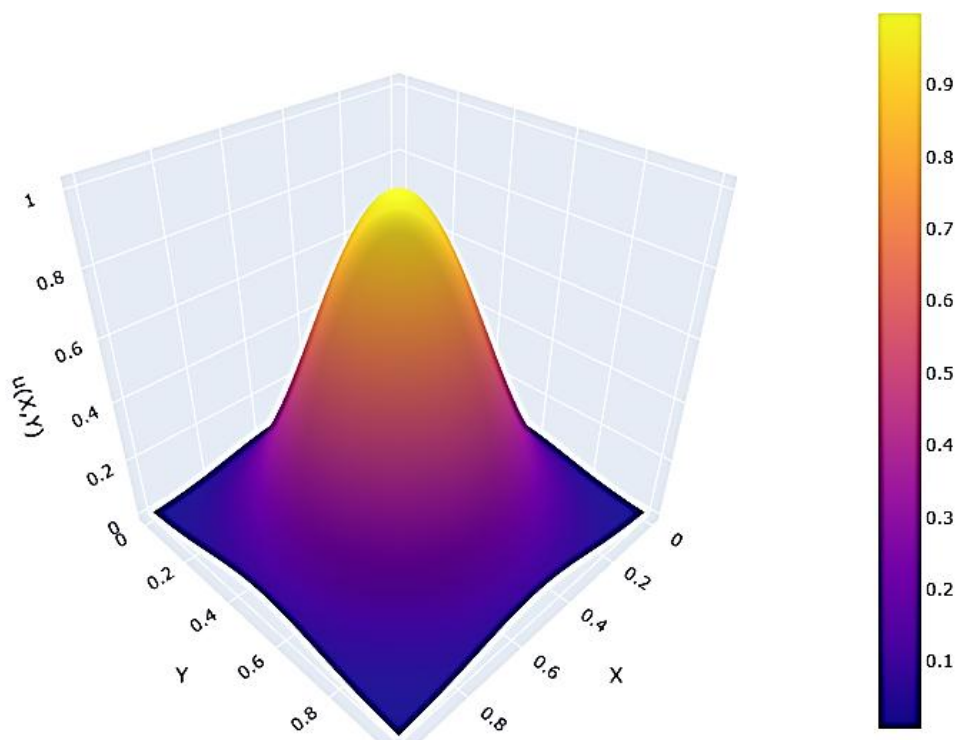


Рисунок 2. Распределение температуры

Заключение

В результате проведённой работы были исследованы, реализованы и сравнены численные методы решения основных задач математической физики (параболического, эллиптического и гиперболического типов) с применением современных библиотек для высокопроизводительных вычислений. Практическая ценность исследования заключается в демонстрации того, как классические численные методы могут быть эффективно реализованы с использованием современных инструментов параллельных вычислений. Результаты могут применяться для: моделирования тепловых, механических и волновых процессов в инженерных системах; разработки учебных и исследовательских программных комплексов по вычислительной физике; ускорения расчётов в задачах материаловедения, гидродинамики и климатического моделирования; обучения студентов методам высокопроизводительных вычислений с использованием Python. Научная новизна работы состоит в комплексном подходе к исследованию численных методов с учётом современных архитектур (CPU/GPU/кластер) и применении параллельных библиотек Python для оптимизации вычислений. Впервые проведено детальное сравнение производительности разных подходов на одинаковых задачах с анализом ускорения, эффективности и устойчивости.

Список литературы:

1. Айзексон К., Керниг Х. Вычислительные методы: Численные решения дифференциальных уравнений. М.: Мир, 2002. 576 с.
2. Auffarth B. Machine learning for time-series with python. UK: Packt Publishing, 2021.
3. Ranjani J., Sheela A., Meena K. P. Combination of NumPy, SciPy and Matplotlib/PyLab-a good alternative methodology to MATLAB-A Comparative analysis // 2019 1st international conference on innovations in information and communication technology (ICIICT). IEEE, 2019. P. 1-5.

4. Carnielli G. P., Arthur R., Monteiro A. C. B., Franca R. P., Iano, Y. An Overview of High-Performance Computing Techniques Applied to Image Processing // *VLSI and Hardware Implementations using Modern Machine Learning Methods*. 2021. P. 141-161. <https://doi.org/10.1201/9781003201038-8>

5. Пирматов А. З., Азимов Б. А. Методы решения дифференциальных уравнений на языке Python // *Бюллетень науки и практики*. 2023. Т. 9. №12. С. 39-46. <https://doi.org/10.33619/2414-2948/97/04>

6. Пирматов А. З., Шестаков Е. И., Исаков Т. Э. Численное решение краевой задачи для модельного псевдо-параболо-гиперболического уравнения четвертого порядка в частных производных // *Наука. Образование. Техника*. 2024. №2. С.79-84.

Список литературы:

1. Aizekson, K., & Kernig, Kh. (2002). *Vychislitel'nye metody: Chislennye resheniya differentsial'nykh uravnenii*. Moscow. (in Russian).

2. Auffarth, B. (2021). *Machine learning for time-series with python*. UK: Packt Publishing.

3. Ranjani, J., Sheela, A., & Meena, K. P. (2019, April). Combination of NumPy, SciPy and Matplotlib/PyLab-a good alternative methodology to MATLAB-A Comparative analysis. In *2019 1st international conference on innovations in information and communication technology (ICIICT)* (pp. 1-5). IEEE.

4. Carnielli, G. P., Arthur, R., Monteiro, A. C. B., Franca, R. P., & Iano, Y. (2021). An Overview of High-Performance Computing Techniques Applied to Image Processing. *VLSI and Hardware Implementations using Modern Machine Learning Methods*, 141-161. <https://doi.org/10.1201/9781003201038-8>

5. Pirmatov, A., & Azimov, B. (2023). Methods for Solving Differential Equations in Python Language. *Bulletin of Science and Practice*, 9(12), 39-46. (in Russian). <https://doi.org/10.33619/2414-2948/97/04>

6. Pirmatov, A. Z., Shestakov, E. I., & Isakov, T. E. (2024). Chislennoe reshenie kraevoi zadachi dlya model'nogo psevdoparabolo-giperbolicheskogo uravneniya chetvertogo poriyadka v chastnykh proizvodnykh. *Nauka. Obrazovanie. Tekhnika*, (2), 79-84.

Поступила в редакцию
14.11.2025 г.

Принята к публикации
19.11.2025 г.

Ссылка для цитирования:

Пирматов А. З., Аркабаев Н. К., Саадалов Т. Ы., Мурзабаева А. Б. Численные методы решения задач математической физики с использованием современных библиотек для высокопроизводительных вычислений // *Бюллетень науки и практики*. 2026. Т. 12. №1. С. 25-32. <https://doi.org/10.33619/2414-2948/122/03>

Cite as (APA):

Pirmatov, A., Arkabaev, N., Saadalov, T., & Murzabaeva, A. (2026). Numerical Methods for Solving Problems of Mathematical Physics using Modern Libraries for High-Performance Computing. *Bulletin of Science and Practice*, 12(1), 25-32. (in Russian). <https://doi.org/10.33619/2414-2948/122/03>