

УДК 004.94:519.6

Смирнов Иван Михайлович, магистрант, 2 курс, Информатика и вычислительная техника, Кафедра «Автоматизированные системы управления», Московский автомобильно-дорожный государственный технический университет (МАДИ), г. Москва

**АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ И БИБЛИОТЕК ДЛЯ
КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ОБЪЕКТОВ С
РАСПРЕДЕЛЕННЫМИ ПАРАМЕТРАМИ**

Аннотация:

В статье рассматриваются программные средства и библиотеки, применяемые для компьютерного моделирования объектов с распределенными параметрами. Актуальность темы связана с тем, что при решении задач такого типа исследователю необходимо выбрать не только численный метод, но и подходящую программную среду для его реализации. Рассмотрены возможности MATLAB, Simulink, Partial Differential Equation Toolbox, а также библиотек Python, используемых для научных вычислений и визуализации результатов. Показано, что разные инструменты отличаются по удобству задания модели, поддержке численных методов, возможностям визуализации, требованиям к ресурсам и стоимости использования. Сделан вывод о целесообразности комбинированного применения готовых вычислительных пакетов и самостоятельной программной реализации при исследовании эффективности алгоритмов.

Ключевые слова: компьютерное моделирование, распределенные параметры, численные методы, MATLAB, Simulink, Python, программные средства.

**ANALYSIS OF SOFTWARE TOOLS AND LIBRARIES FOR COMPUTER
MODELING OF OBJECTS WITH DISTRIBUTED PARAMETERS**

Annotation:

The article considers software tools and libraries used for computer modeling of objects with distributed parameters. The relevance of the topic is related to the fact that, when solving such problems, the researcher has to choose not only a numerical method, but also a suitable software environment for its implementation. The paper discusses MATLAB, Simulink, Partial Differential Equation Toolbox, and Python libraries used for scientific computing and visualization of results. It is shown that different tools differ in model specification, support for numerical methods, visualization capabilities, resource requirements, and cost of use. The conclusion is made that it is reasonable to combine ready-made computing packages with custom software implementation when studying the efficiency of algorithms.

Key words: computer modeling, distributed parameters, numerical methods, MATLAB, Simulink, Python, software tools.

Введение

Компьютерное моделирование является одним из основных способов исследования сложных технических и физических систем. Особенно важную роль оно играет при анализе объектов с распределенными параметрами, состояние которых зависит не только от времени, но и от пространственных координат. К таким объектам относятся тепловые поля, процессы диффузии, колебательные системы, распределение давления в среде, электромагнитные поля и другие процессы, описываемые функциями нескольких переменных.

Математическое описание таких объектов часто строится на основе дифференциальных уравнений в частных производных. Для их решения в практических задачах используются численные методы: метод конечных разностей, метод конечных элементов, метод конечных объемов, а также итерационные методы решения больших систем уравнений. После дискретизации непрерывная задача преобразуется в расчетную схему, которая должна быть реализована в программной среде.

При этом выбор программного инструмента оказывает значительное влияние на результат исследования. Одна среда может быть удобна для быстрого построения графиков и проверки модели, другая - для решения сложных PDE-задач, третья - для самостоятельной реализации алгоритма и анализа его производительности. Поэтому программное средство следует выбирать с учетом цели работы, требуемой точности, доступных библиотек, возможности визуализации и ограничений вычислительных ресурсов.

Цель статьи - рассмотреть основные программные средства и библиотеки, которые могут применяться при компьютерном моделировании объектов с распределенными параметрами, а также определить их преимущества, ограничения и возможные области рационального использования.

1. Особенности выбора программных средств для моделирования объектов с распределенными параметрами

Объекты с распределенными параметрами отличаются тем, что их состояние описывается функциями координат и времени. В отличие от простых динамических систем, где достаточно конечного числа переменных, в распределенных системах требуется учитывать пространственное распределение искомой величины. Например, температура в пластине зависит от двух координат и времени, а давление в трубопроводе - от координаты, времени и режима движения среды.

Для компьютерного расчета непрерывная область заменяется расчетной сеткой. В результате формируется массив значений в узлах сетки, система алгебраических уравнений или набор дифференциальных уравнений по времени. Поэтому программная среда должна поддерживать работу с матрицами, массивами, численными решателями, графиками и большими объемами данных.

При выборе программного средства для таких задач важно учитывать несколько критериев:

- наличие встроенных численных методов и средств решения систем уравнений;
- удобство задания начальных и граничных условий;
- поддержка работы с разреженными матрицами;
- возможность построения расчетной сетки;
- средства визуализации результатов в 1D, 2D и 3D;
- скорость выполнения расчетов и возможность оптимизации;
- стоимость, доступность лицензии и требования к компьютеру.

Таким образом, программное средство должно соответствовать не только математической постановке задачи, но и практическим условиям исследования. Для дипломной работы, связанной с эффективностью вычислительных алгоритмов, особенно важно, чтобы выбранная среда

позволяла не только получить численное решение, но и оценить время выполнения, точность, устойчивость и требования к памяти.

Таблица 1 – Критерии выбора программного средства для моделирования

Критерий	Значение для моделирования	Пример оценки
Численные методы	Определяют возможность решения уравнений и систем	Наличие решателей ОДУ, PDE, СЛАУ
Визуализация	Позволяет анализировать распределение величин	Графики, тепловые карты, поверхности
Работа с данными	Важна при больших сетках и матрицах	Массивы, sparse-матрицы, экспорт результатов
Производительность	Влияет на длительность вычислительного эксперимента	Векторизация, параллельные расчеты
Доступность	Определяет возможность использования в учебной работе	Бесплатная среда или коммерческая лицензия

2. MATLAB и Simulink как средства моделирования динамических и распределенных систем

MATLAB является универсальной средой для инженерных и научных расчетов. Ее сильной стороной является работа с матрицами и массивами, что особенно важно при реализации численных методов. После дискретизации объектов с распределенными параметрами часто возникают матрицы коэффициентов, векторы неизвестных и массивы значений в узлах сетки. MATLAB позволяет выполнять такие операции в компактной форме и быстро строить графики результатов.

В задачах моделирования MATLAB может использоваться двумя способами. Первый способ заключается в самостоятельной реализации численного алгоритма, например явной или неявной разностной схемы. В этом случае исследователь полностью контролирует расчетную сетку, порядок вычислений и способ хранения данных. Второй способ связан с применением

готовых расширений, например Partial Differential Equation Toolbox, Optimization Toolbox или Statistics and Machine Learning Toolbox.

Simulink дополняет MATLAB и применяется для блочного моделирования динамических систем. В такой среде модель представляется в виде блок-схемы, где отдельные блоки задают входные воздействия, усилители, интеграторы, передаточные функции, вычислительные элементы и блоки вывода результатов. Такой подход особенно удобен при моделировании систем управления, переходных процессов и объектов, которые после пространственной дискретизации можно представить в виде конечномерной динамической системы.

Для объектов с распределенными параметрами Simulink целесообразно использовать не всегда напрямую, а после предварительной дискретизации. Например, тепловая задача может быть сведена к системе обыкновенных дифференциальных уравнений, описывающих изменение температуры в узлах сетки. После этого такую систему можно представить в виде структурной модели и исследовать ее динамику.



Рисунок 1 – Пример структурной модели в среде Simulink

Рисунок 1 показывает общий принцип блочного представления модели. Входной сигнал поступает на расчетный блок, далее результат передается в блок визуализации, а при необходимости используется обратная связь. Такая схема не является копией конкретного интерфейса MATLAB, но отражает типовую логику построения динамических моделей в Simulink.

Преимуществами MATLAB и Simulink являются наглядность расчетов, большое количество встроенных функций, удобная визуализация и возможность проведения вычислительных экспериментов. К ограничениям можно отнести коммерческий характер среды, значительный объем установки и зависимость возможностей от набора установленных дополнительных продуктов.

Таблица 2 – Возможности MATLAB и Simulink при моделировании

Средство	Что позволяет делать	Практическое значение
MATLAB	Реализовывать численные схемы, работать с матрицами, строить графики	Подходит для алгоритмов, расчетов и анализа результатов
Simulink	Создавать блочные модели динамических систем	Удобен для систем управления и переходных процессов
Optimization Toolbox	Решать задачи оптимизации и подбора параметров	Полезен при настройке модели
Statistics and Machine Learning Toolbox	Анализировать данные и результаты экспериментов	Полезен при сравнении расчетов и обработке статистики

3. Partial Differential Equation Toolbox для решения задач с распределенными параметрами

Partial Differential Equation Toolbox является специализированным расширением MATLAB, предназначенным для решения задач, описываемых дифференциальными уравнениями в частных производных. Именно поэтому данный инструмент имеет прямое отношение к моделированию объектов с распределенными параметрами. Он позволяет задавать геометрию расчетной области, формировать сетку, описывать коэффициенты уравнений, задавать начальные и граничные условия, выполнять расчет и визуализировать результаты.

Основу вычислений в данном инструменте составляет метод конечных элементов. Этот метод удобен для областей сложной формы, так как расчетная область разбивается на конечные элементы, внутри которых приближенно описывается поведение искомой функции. В простых задачах пользователь может сосредоточиться на физической постановке модели, не реализуя вручную генерацию сетки и формирование матриц.

Типовая последовательность работы с Partial Differential Equation Toolbox включает следующие этапы:

- создание или импорт геометрии расчетной области;
- задание типа физической задачи и коэффициентов модели;
- определение начальных и граничных условий;
- построение конечно-элементной сетки;
- запуск численного решателя;
- построение графиков и анализ распределения искомой величины.

Конечно-элементная сетка расчетной области

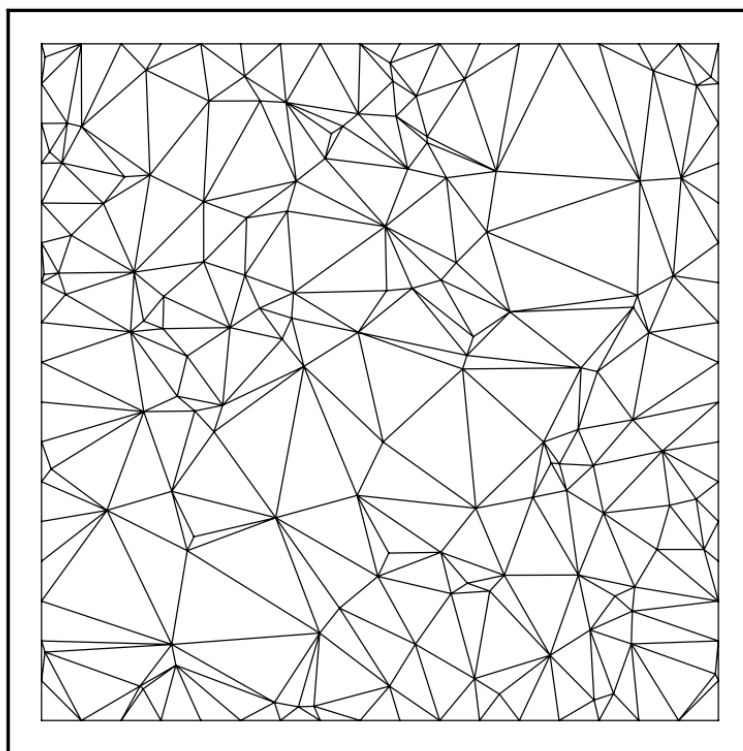


Рисунок 2 – Пример конечно-элементной сетки расчетной области

На рисунке 2 показан пример конечно-элементной сетки. Такая сетка используется для замены непрерывной области конечным набором элементов. Чем мельче сетка, тем выше потенциальная точность решения, но тем больше объем вычислений и требования к оперативной памяти.

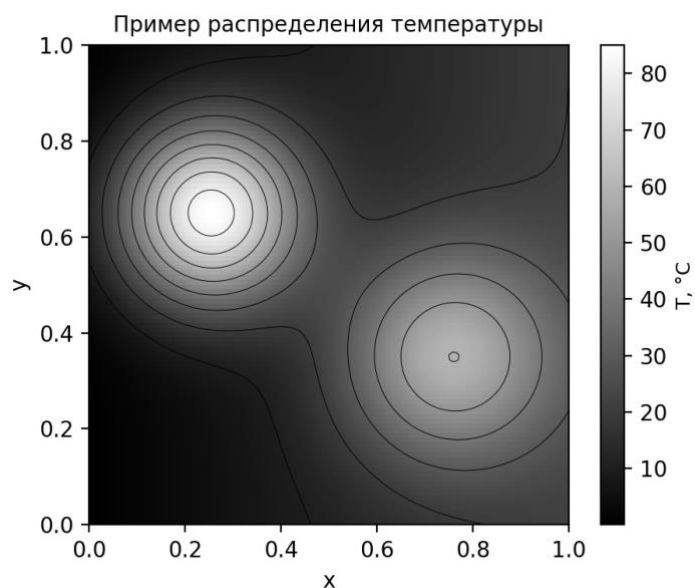


Рисунок 3 – Пример визуализации распределения температуры в расчетной области

На рисунке 3 приведен пример визуализации результата расчета в виде распределения температуры по двумерной области. Подобные изображения

удобны при анализе объектов с распределенными параметрами, так как позволяют быстро определить зоны максимальных и минимальных значений, характер изменения величины и возможные особенности решения.

Преимуществом Partial Differential Equation Toolbox является наличие готового рабочего процесса для PDE-задач. Однако для исследования эффективности самих алгоритмов этот инструмент имеет ограничение: часть внутренних вычислительных процедур скрыта от пользователя. Поэтому его целесообразно применять для проверки модели, визуализации и сравнения результатов, а не как единственный инструмент анализа алгоритмической эффективности.

Таблица 3 – Основные этапы работы с Partial Differential Equation Toolbox

Этап	Содержание	Результат
Геометрия	Создание или импорт расчетной области	Модель области
Физика задачи	Выбор уравнения и коэффициентов	Математическая постановка
Граничные условия	Задание условий на границах	Корректная расчетная задача
Сетка	Разбиение области на конечные элементы	Дискретная модель
Решение	Запуск численного решателя	Массив результатов
Визуализация	Построение графиков и карт распределения	Анализ решения

4. Python-библиотеки NumPy, SciPy и Matplotlib в задачах численного моделирования

Python является открытым и гибким языком, который широко применяется в научных вычислениях. В отличие от MATLAB, базовый язык Python сам по себе не является специализированной средой для численного моделирования. Его возможности раскрываются за счет библиотек, предназначенных для работы с массивами, линейной алгеброй, оптимизацией, визуализацией и обработкой данных.

Для моделирования объектов с распределенными параметрами наиболее важными являются библиотеки NumPy, SciPy и Matplotlib. Они позволяют реализовывать численные схемы, хранить значения в узлах расчетной сетки, решать системы уравнений и представлять результаты в виде графиков. Такой подход особенно полезен, если требуется самостоятельно реализовать алгоритм и исследовать его работу.

Таблица 4 – Назначение основных Python-библиотек при численном моделировании

Библиотека	Назначение	Применение в задачах с распределенными параметрами
NumPy	Многомерные массивы, матричные и векторные операции	Хранение сеточных значений, реализация разностных схем, векторизация вычислений
SciPy	Численные методы, оптимизация, интегрирование, линейная алгебра	Решение систем уравнений, работа с разреженными матрицами, поиск параметров модели
Matplotlib	Построение графиков и визуализация данных	Линейные графики, тепловые карты, контурные диаграммы, поверхности
Pandas	Табличные данные и обработка результатов	Сохранение и анализ результатов вычислительных экспериментов
SymPy	Символьные вычисления	Проверка формул, аналитические преобразования, работа с уравнениями
Numba	Ускорение Python-кода путем компиляции отдельных функций	Оптимизация расчетных циклов при больших сетках

Библиотека NumPy используется как основа для большинства научных вычислений на Python. Она предоставляет массивы, которые работают значительно быстрее обычных списков Python, и позволяет выполнять операции сразу над целыми наборами данных. В задачах с распределенными параметрами NumPy удобно использовать для хранения значений функции в узлах сетки и выполнения расчетов по всей сетке без избыточных циклов.

Библиотека SciPy расширяет возможности NumPy. Для рассматриваемой темы особенно важны модули линейной алгебры и работы с разреженными матрицами. После дискретизации PDE-задач часто возникают большие системы линейных уравнений, в которых большая часть элементов матрицы равна нулю. Использование разреженных форматов позволяет экономить память и ускорять расчеты.

Matplotlib применяется для визуализации результатов. С ее помощью можно строить графики зависимости величины от координаты, тепловые карты, контурные диаграммы и трехмерные поверхности. Для статьи, диплома и отчета это особенно важно, поскольку распределенные параметры сложно анализировать только в виде числовых таблиц.

Дополнительно могут использоваться Pandas для хранения результатов серий вычислительных экспериментов, SymPy для символьных преобразований и Numba для ускорения отдельных участков кода. Такое сочетание делает Python удобным инструментом для исследовательских задач, особенно когда требуется свободно изменять алгоритм и контролировать все этапы расчета.

Таблица 5 – Сравнение открытых и коммерческих решений

Тип решения	Примеры	Преимущества	Ограничения
Открытые библиотеки	Python, NumPy, SciPy, Matplotlib	Бесплатность, гибкость, доступность кода, удобство автоматизации	Требуют навыков программирования и оптимизации
Коммерческие вычислительные среды	MATLAB, Simulink, Mathematica	Единая среда, документация, готовые инструменты, удобная визуализация	Необходима лицензия, большой объем установки
Инженерные пакеты	COMSOL Multiphysics, ANSYS	Готовые физические интерфейсы, сложные геометрии, развитая визуализация	Высокая стоимость, сложность освоения, скрытость внутренних алгоритмов

5. Специализированные инженерные пакеты и их место в моделировании

Кроме MATLAB и Python, в инженерной практике применяются специализированные пакеты моделирования, такие как COMSOL Multiphysics и ANSYS. Они ориентированы на прикладное решение сложных физических задач и содержат готовые интерфейсы для теплообмена, механики, гидродинамики, электромагнитных процессов и мультифизического моделирования.

Главное преимущество таких систем заключается в том, что пользователь может работать на уровне физической постановки задачи: задавать геометрию, свойства материалов, нагрузки, граничные условия и получать результат с помощью встроенных решателей. Это удобно для инженерного анализа, когда основной целью является получение практического результата, а не разработка численного метода.

Однако для исследования эффективности вычислительных алгоритмов такие пакеты имеют ограничения. Они часто скрывают внутреннюю реализацию решателей, используют собственные настройки сетки и оптимизации, а также требуют значительных вычислительных ресурсов.

Поэтому их полезно применять для сравнения и проверки результатов, но не всегда удобно использовать как основной инструмент анализа алгоритмов.

6. Сравнение возможностей программных средств

Для выбора программного средства удобно сравнивать инструменты по нескольким параметрам: назначение, степень контроля над алгоритмом, возможности визуализации, требования к подготовке пользователя и доступность. Такое сравнение позволяет определить, какой инструмент лучше подходит для конкретного этапа исследования.

Таблица 6 – Сравнение программных средств для моделирования объектов с распределенными параметрами

Средство	Основное назначение	Преимущества	Ограничения	Когда целесообразно использовать
MATLAB	Численные расчеты и визуализация	Удобная работа с матрицами, графиками и алгоритмами	Коммерческая среда, зависимость от toolbox-пакетов	Прототипирование и вычислительные эксперименты
Simulink	Блочное моделирование динамических систем	Наглядность, готовые блоки, связь с MATLAB	Не всегда удобен для больших пространственных сеток	Динамические модели и системы управления
PDE Toolbox	Решение задач с PDE	Сетка, метод конечных элементов, граничные условия	Скрыты многие внутренние вычислительные процедуры	Теплопередача, механика, общие PDE-задачи
Python	Самостоятельная реализация алгоритмов	Бесплатность, гибкость, библиотеки	Нужна грамотная оптимизация кода	Исследование алгоритмов и автоматизация расчетов
COMSOL, ANSYS	Инженерное моделирование	Готовые физические интерфейсы и сложные геометрии	Высокая стоимость и ресурсоемкость	Прикладные инженерные задачи

Из таблицы 6 видно, что универсального средства для всех задач не существует. Если цель состоит в исследовании алгоритма, полезнее применять

Python или MATLAB с самостоятельной реализацией расчетной схемы. Если необходимо быстро получить визуальный результат для PDE-задачи, удобнее использовать Partial Differential Equation Toolbox или инженерный пакет. Если требуется представить динамическую модель в виде структуры, целесообразно использовать Simulink.

7. Рекомендации по выбору инструмента для дипломного исследования

Для дипломной работы, связанной с исследованием эффективности вычислительных алгоритмов при моделировании объектов с распределенными параметрами, рационально использовать комбинированный подход. Он позволяет не ограничиваться одной программной средой и сравнивать разные способы решения задачи.

На первом этапе целесообразно реализовать простую модель самостоятельно, например на Python или MATLAB. Это позволит контролировать расчетную сетку, численную схему, шаг по времени, критерии остановки и способ хранения данных. Такой подход хорошо подходит для анализа точности, устойчивости и скорости работы алгоритма.

На втором этапе можно использовать Partial Differential Equation Toolbox для проверки результатов и построения наглядных графиков распределения величин. Это особенно полезно, если в работе необходимо показать конечно-элементную сетку, распределение температуры или другой параметр в пространстве.

На третьем этапе можно использовать Simulink для представления динамической части модели или системы управления, если распределенная модель после дискретизации сводится к системе уравнений во времени. Это позволит показать связь между математической моделью и структурной схемой объекта.

Таким образом, наиболее удачным вариантом является не выбор одного программного средства, а разумное сочетание нескольких инструментов в зависимости от этапа исследования.

Таблица 7 – Рекомендуемое применение программных средств в дипломной работе

Этап исследования	Рекомендуемое средство	Цель применения
Построение простой модели	Python или MATLAB	Самостоятельная реализация численной схемы
Проверка PDE-модели	Partial Differential Equation Toolbox	Сравнение решения и построение сетки
Анализ динамики	Simulink	Представление модели в виде блок-схемы
Визуализация результатов	MATLAB или Matplotlib	Графики, тепловые карты, поверхности
Сравнение производительности	Python, MATLAB	Измерение времени расчета и затрат памяти

Заключение

Выбор программного средства является важным этапом компьютерного моделирования объектов с распределенными параметрами. От него зависит удобство задания модели, скорость получения результата, качество визуализации, возможность контроля численного алгоритма и трудоемкость исследования.

Проведенный анализ показывает, что MATLAB удобен для матричных вычислений и вычислительных экспериментов, Simulink - для блочного моделирования динамических систем, Partial Differential Equation Toolbox - для решения задач, описываемых дифференциальными уравнениями в частных производных, а Python с библиотеками NumPy, SciPy и Matplotlib - для гибкой самостоятельной реализации алгоритмов и визуализации результатов.

Коммерческие инженерные пакеты, такие как COMSOL Multiphysics и ANSYS, позволяют решать сложные прикладные задачи, но не всегда подходят для детального исследования внутренней эффективности алгоритмов. Поэтому при выполнении дипломной работы целесообразно сочетать готовые программные средства с самостоятельной реализацией отдельных численных методов.

Таким образом, наиболее эффективным является комплексный подход, при котором разные инструменты используются для разных задач: разработка алгоритма, проверка корректности, визуализация, сравнение производительности и оформление результатов исследования.

Литература

1. Самарский А.А., Гулин А.В. Численные методы. — М.: Наука, 1989. — 432 с.
2. Самарский А.А. Теория разностных схем. — М.: Наука, 1989. — 616 с.
3. Калиткин Н.Н. Численные методы. — М.: Наука, 1978. — 512 с.
4. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. — М.: БИНОМ. Лаборатория знаний, 2012. — 636 с.
5. Зенкевич О. Метод конечных элементов в технике. — М.: Мир, 1975. — 541 с.
6. MATLAB Documentation: официальный сайт MathWorks. [Электронный ресурс]. URL: <https://www.mathworks.com/help/matlab/index.html> (дата обращения: 24.05.2026).
7. Simulink Documentation: официальный сайт MathWorks. [Электронный ресурс]. URL: <https://www.mathworks.com/help/simulink/index.html> (дата обращения: 24.05.2026).
8. Partial Differential Equation Toolbox Documentation: официальный сайт MathWorks. [Электронный ресурс]. URL: <https://www.mathworks.com/help/pde/index.html> (дата обращения: 24.05.2026).
9. Sparse arrays: SciPy Documentation. [Электронный ресурс]. URL: <https://docs.scipy.org/doc/scipy/reference/sparse.html> (дата обращения: 24.05.2026).
10. Sparse linear algebra: SciPy Documentation. [Электронный ресурс]. URL: <https://docs.scipy.org/doc/scipy/reference/sparse.linalg.html> (дата обращения: 24.05.2026).

Literature

1. Samarskii A.A., Gulin A.V. Numerical Methods. — Moscow: Nauka, 1989. — 432 p.
2. Samarskii A.A. Theory of Difference Schemes. — Moscow: Nauka, 1989. — 616 p.
3. Kalitkin N.N. Numerical Methods. — Moscow: Nauka, 1978. — 512 p.
4. Bakhvalov N.S., Zhidkov N.P., Kobelkov G.M. Numerical Methods. — Moscow: BINOM. Laboratory of Knowledge, 2012. — 636 p.
5. Zienkiewicz O. The Finite Element Method in Engineering. — Moscow: Mir, 1975. — 541 p.
6. MATLAB Documentation: MathWorks official website. [Electronic resource]. URL: <https://www.mathworks.com/help/matlab/index.html> (accessed: 24.05.2026).
7. Simulink Documentation: MathWorks official website. [Electronic resource]. URL: <https://www.mathworks.com/help/simulink/index.html> (accessed: 24.05.2026).
8. Partial Differential Equation Toolbox Documentation: MathWorks official website. [Electronic resource]. URL: <https://www.mathworks.com/help/pde/index.html> (accessed: 24.05.2026).
9. Sparse arrays: SciPy Documentation. [Electronic resource]. URL: <https://docs.scipy.org/doc/scipy/reference/sparse.html> (accessed: 24.05.2026).
10. Sparse linear algebra: SciPy Documentation. [Electronic resource]. URL: <https://docs.scipy.org/doc/scipy/reference/sparse.linalg.html> (accessed: 24.05.2026).