

Документ подписан простой электронной подписью
Информация о владельце: Федеральное государственное бюджетное образовательное учреждение
ФИО: Макаренко Елена Николаевна
Должность: Ректор
Дата подписания: 10.06.2026 16:23:10
Уникальный программный ключ:
с098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОСТОВСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ (РИНХ)»

УТВЕРЖДАЮ
Начальник отдела
аспирантуры и докторантуры
Е.Н. Грузднева
«26» мая 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Современные языки, методы и технологии
программирования

Группа научных специальностей
1.2. Компьютерные науки и информатика

Научная специальность
1.2.2. Математическое моделирование, численные методы
и комплексы программ

Для набора: 2026 года

Кафедра информационных систем и прикладной информатики

Распределение часов дисциплины по семестрам*

Вид занятий	Количество часов	Семестр
Лекции	12	4
Лабораторные	6	
Итого ауд.	18	
Самостоятельная работа	52	
Контроль	2	
ИТОГО	72	
Промежуточная аттестация по дисциплине (зачет)	36	

ОСНОВАНИЕ

Учебный план утвержден Ученым советом вуза 03.03.2026, протокол № 9.

Программу составили(и): д.э.н., профессор А.И. Долженко

Зав. кафедрой: д.э.н., доцент С.М. Щербаков

Научно-методический совет: проректор по научной работе и инновациям д.э.н., профессор Н.Г. Вовченко

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины: формирование способности объективно исследовать, выбирать и применять языки, методы и технологии программирования для развития теоретических основ математического моделирования, численных методов и комплексов программ.

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

<i>Код</i>	<i>Формулировка результата</i>	<i>Индикаторы достижения результата</i>
РОЗ	способен использовать углубленные теоретические знания по научной специальности для критического анализа достижений научной отрасли в целом и в рамках конкретной исследуемой проблемы	<ul style="list-style-type: none"> – знает теоретические концепции и методологические основы научной отрасли; основные тенденции и результаты новейших теоретических и прикладных исследований в рамках научной специальности; основные положения, принципы, методы сбора, обработки и анализа научной информации; – умеет использовать полученные теоретические знания для решения вопросов собственного научного исследования, учитывая современные научные тенденции; структурировать и обобщать имеющийся в литературе и электронных базах данных материал; аргументированно представлять результаты собственного исследования; – владеет навыками оценки состояния и динамики развития своей предметной области на современном этапе.

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

№	Наименование темы, краткое содержание	Вид занятия	Кол-во часов	Результат освоения
1	Раздел «Теория языков, методов и технологий программирования»			
1.1	Тема 1. Языки программирования программных приложений. Проводится обзорный анализ современных языков программирования: C++, C#, Java, Visual Basic. Возможности, характеристики, недостатки, пути совершенствования.	Лекционные занятия	2	РОЗ
1.2	Тема 2. Языки программирования веб-приложений. Проводится обзорный анализ современных языков веб-программирования: HTML5, JavaScript, CSS, PHP, Python. Возможности, характеристики, недостатки, пути совершенствования.	Лекционные занятия	2	РОЗ
1.3	Тема 3. Платформы и среды создания программных систем. Проводится обзорный анализ современных программных платформ и сред для создания программных систем: JDK, Microsoft Framework.NET, ASP.NET, Windows Azure	Лекционные занятия	2	РОЗ
1.4	Тема 4. Программные системы с открытым кодом. Проводится обзорный анализ современных программных платформ и сред с открытым кодом: веб-сервер Apache, операционная система Linux, инструментальная среда OpenIDE, GigaIDE.	Лекционные занятия	2	РОЗ
1.5	Тема 5. Методы и технологии создания программных систем. Проводится обзорный анализ современных технологий создания программных систем: управление жизненным циклом приложений, методологии командной разработки, системы контроля версий, не-	Лекционные занятия	2	РОЗ

	прерывная интеграция приложений, тестирование.			
1.6	Тема 6. Мобильные и облачные технологии. Проводится обзорный анализ современных мобильных и облачных технологий: операционные системы iOS, Android.	Лекционные занятия	2	РОЗ
2	Раздел «Практика языков, методов и технологий программирования»			
2.1	Лабораторная работа 1. Разработка программных приложений. Разработка программного приложения на языке Java, взаимодействующего с сетевыми ресурсами.	Лабораторные занятия	2	РОЗ
2.2	Лабораторная работа 2. Языки программирования веб-приложений. Разработка веб-приложения на языках веб-программирования HTML5, Java Script, CSS.	Лабораторные занятия	2	РОЗ
2.3	Лабораторная работа 3. Программные системы с открытым кодом. Разработка java-приложения в инструментальной среде OpenIDE.	Лабораторные занятия	2	РОЗ
3	Самостоятельная работа			
3.1	Выполнение индивидуального задания (перечень индивидуальных заданий представлены в приложении 1 к рабочей программы дисциплины).	Самостоятельная работа	52	РОЗ
3.2	Контроль		2	РОЗ
4	ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ			
4.1	Промежуточная аттестация по дисциплине (зачет)		36	РОЗ

4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Структура и содержание оценочных средств для проведения текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

№	Выходные данные	Количество экземпляров
5.1. Учебные, научные и методические издания		
1.	Попова Е.В. Математическое моделирование, численные методы и комплексы программ. Краткий курс лекций. Кубанский аграрный университет, 2021. – 43 с.	20
2.	Введение в программные системы и их разработку : [16+] / С. В. Назаров, С. Н. Белоусова, И. А. Бессонова [и др.]. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2020. – 650 с. : схем., табл., ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=429819 (дата обращения: 05.06.2026). – Библиогр. в кн. – Текст : электронный.	ЭБС Biblioclub
3.	Долженко, А. И. Разработка и сопровождение программных систем: технологии Microsoft.NET для разработки приложений: практикум : [16+] / А. И. Долженко, С. А. Глушенко ; Ростовский государственный экономический университет (РИНХ). – Ростов-на-Дону: Издательско-полиграфический комплекс РГЭУ (РИНХ), 2021. – 140 с.: ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=614911 (дата обращения: 05.06.2026). – Библиогр. в кн. – ISBN 978-5-7972-2626-0. – Текст : электронный.	ЭБС Biblioclub

5.2. Периодические издания	
4.	Прикладная информатика: журнал, 2020-2026
5.	Информационные технологии: журнал, 2020-2026
5.3. Профессиональные базы данных и информационные справочные системы	
ИСС КонсультантПлюс ИСС "Гарант" - https://internet.garant.ru/	
5.4. Перечень программного обеспечения	
LibreOffice ProjectLibre Python Simpy Jupyter Notebook OpenIDE	
5.5. Учебно-методические материалы для аспирантов с ограниченными возможностями здоровья	
При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.	

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения: столы, стулья; персональный компьютер/ноутбук (переносной); проектор; экран/интерактивная доска.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. **ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ АСПИРАНТОВ** учитывается по результатам выполнения индивидуальных заданий (выполнение реферата/доклада, прохождение опроса, тестов) и собеседования на лабораторных занятиях с преподавателем.

ТЕМЫ РЕФЕРАТОВ/ДОКЛАДОВ

Тема реферата/доклада должна быть выбрана в соответствии с темой диссертации и отраслью защиты конкретного аспиранта.

Темы рефератов:

1. **«Языки программирования для научных вычислений: сравнительный анализ Python, Julia и Fortran»**

Рассмотрение производительности, библиотек для численных расчётов (NumPy, SciPy, BLAS/LAPACK), экосистемы и применимости в математическом моделировании.

2. **«Параллельное программирование в задачах математического моделирования: OpenMP, MPI и CUDA»**

Анализ технологий параллелизации вычислений для ускорения численных методов. Примеры применения в решении систем дифференциальных уравнений, обработке больших данных и моделировании физических процессов.

3. **«Использование библиотек NumPy и SciPy для реализации численных методов»**

Практическое применение библиотек Python для решения задач линейной алгебры, интерполяции, численного интегрирования и решения ОДУ/УЧП. Сравнение с классическими реализациями на других языках.

4. **«Разработка комплексов программ для вычислительного эксперимента на языке C++ с применением шаблонов и метапрограммирования»**

Использование современных возможностей C++ (STL, шаблоны, constexpr) для создания эффективных и гибких программных решений в научном программировании.

5. **«Функциональное программирование и его применение в математическом моделировании (на примере Haskell или Scala)»**

Исследование преимуществ функционального подхода (неизменяемость данных, чистые функции, ленивые вычисления) для реализации алгоритмов численного анализа и символьных вычислений.

6. **«Технологии контейнеризации (Docker) и оркестрации (Kubernetes) в научных вычислениях»**

Применение контейнеров для воспроизводимости вычислительных экспериментов, масштабирования расчётов на кластерах и облачных платформах. Примеры использования в высокопроизводительных вычислениях (HPC).

7. **«Автоматизация научных расчётов с использованием Jupyter Notebook и workflow-систем (Snakemake, Nextflow)»**

Организация вычислительных пайплайнов, версионирование данных и кода, обеспечение воспроизводимости результатов. Интеграция с системами контроля версий (Git) и CI/CD.

8. **«Применение языков R и Python для статистического моделирования и анализа результатов вычислительных экспериментов»**

Сравнение инструментов статистического анализа, визуализации данных и машинного обучения. Примеры обработки результатов численного моделирования (оценка погрешностей, доверительных интервалов, проверка гипотез).

9. **«Программирование для GPU: технологии CUDA и OpenCL в задачах численного моделирования»**

Реализация численных методов (метод конечных разностей, метод Монте-Карло) с использованием графических ускорителей. Оценка прироста производительности и ограничения подходов.

10. **«Современные подходы к тестированию и верификации научных программных комплексов»**

Методы unit-тестирования, интеграционного тестирования и валидации математических моделей. Инструменты (pytest, Google Test), стандарты качества кода, документирование (Sphinx, Doxygen). Обеспечение надёжности и корректности результатов моделирования.

Критерии оценивания:

В ходе изучения дисциплины аспирант может подготовить от 1 до 4 рефератов/докладов (максимально 20 баллов). Каждый реферат/доклад оценивается максимум в 5 баллов:

4-5 баллов - системность, обстоятельность и глубина излагаемого материала; знакомство с научной и учебной литературой; способность воспроизвести основные тезисы доклада без помощи конспекта; способность быстро и развернуто отвечать на вопросы преподавателя и аудитории; наличие презентации к докладу;

3 балла - развернутость и глубина излагаемого в докладе материала; знакомство с основной научной литературой к докладу; при выступлении частое обращение к тексту доклада; некоторые затруднения при ответе на вопросы (неспособность ответить на ряд вопросов из аудитории); наличие презентации;

1-2 балла - правильность основных положений доклада; наличие недостатка информации в докладе по целому ряду проблем; использование для подготовки доклада исключительно учебной литературы; неспособность ответить на несложные вопросы из аудитории и преподавателя; неумение воспроизвести основные положения доклада без письменного конспекта; наличие презентации;

0 баллов - поверхностный, неупорядоченный, бессистемный характер информации в докладе; при чтении доклада постоянное использование текста; полное отсутствие внимания к докладу аудитории; отсутствие презентации.

ОПРОС

Вопросы для опроса:

Базовые знания и опыт

1. Какие языки программирования вы используете в своей научной работе для реализации математических моделей и численных методов? Перечислите и укажите примерный стаж работы с каждым.
2. С какими библиотеками и фреймворками для научных вычислений (например, NumPy, SciPy, BLAS, LAPACK, PETSc, Eigen, TensorFlow и т.д.) вы имеете практический опыт работы? Оцените свой уровень владения ими по шкале от 1 до 5.

Технологии и методы

3. Использовали ли вы технологии параллельного и распределённого программирования (OpenMP, MPI, CUDA/OpenCL) для ускорения расчётов в своих исследованиях? Если да, опишите кратко задачу, которую решали с их помощью.
4. Применяете ли вы инструменты контейнеризации (Docker) или оркестрации (Kubernetes) для обеспечения воспроизводимости вычислительных экспериментов? Если нет, считаете ли вы это полезным для вашей области исследований?
5. Какие методы и инструменты тестирования (unit-тесты, интеграционные тесты) вы используете для верификации корректности реализации численных алгоритмов в своих программных комплексах?

Рабочие процессы и инструменты

6. Используете ли вы системы контроля версий (например, Git) в своей работе над научными программными комплексами? Если да, опишите ваш типичный рабочий процесс (ветки, коммиты, совместная работа).
7. Какой инструмент вы предпочитаете для интерактивной разработки и документирования научных расчётов (Jupyter Notebook, R Markdown, MATLAB Live Scripts и т.д.) и почему?

Проблемы и потребности

8. С какими основными трудностями вы сталкиваетесь при разработке и отладке программного обеспечения для математического моделирования и численных экспериментов? Приведите 1–2 конкретных примера.
9. Какие современные технологии программирования или инструменты (языки, библиотеки, среды разработки), по вашему мнению, наиболее перспективны для применения в вашей области научных исследований в ближайшие 3–5 лет? Кратко обоснуйте свой выбор.

Перспективы и обучение

10. Какие темы или технологии в рамках дисциплины «Современные языки, методы и технологии программирования» вы хотели бы изучить более глубоко в ходе обучения? Сформулируйте 2–3 направления, которые, на ваш взгляд, принесут наибольшую пользу вашей научной работе.

Критерии оценивания:

Максимальное количество баллов, которые может набрать аспирант при коллоквиуме, собеседовании - 20 баллов:

15-20 баллов выставляется аспиранту, если при собеседовании он демонстрирует полное и содержательное знание материала;

9-14 баллов выставляется аспиранту, если он обнаруживает твердые, но в некоторых вопросах неточные знания материала;

3-8 баллов выставляется аспиранту, если он показывает знания основного учебно-программного материала, но допускает существенные неточности в ответе.

0-2 баллов выставляется аспиранту, если он не показывает знания основного учебно-программного материала, допускает существенные ошибки в ответе.

ТЕСТ

Тест 1. Основы языков программирования для научных вычислений

1. Какие языки программирования наиболее часто используются для научных вычислений и математического моделирования?
 - a) Python;
 - б) JavaScript;
 - в) Fortran;
 - г) Julia.
2. Какая библиотека Python предназначена для работы с многомерными массивами и матрицами?
 - a) Pandas;
 - б) NumPy;
 - в) Matplotlib;
 - г) SciPy.
3. Какой язык программирования изначально был разработан для научных и инженерных расчётов и до сих пор активно используется в высокопроизводительных вычислениях?
 - a) C++;
 - б) Java;
 - в) Fortran;
 - г) R.
4. Что из перечисленного является преимуществом языка Julia для научных вычислений?
 - a) Высокая производительность, близкая к C/Fortran;
 - б) Простота синтаксиса, похожего на Python;
 - в) Встроенная поддержка параллельных вычислений;
 - г) Всё вышеперечисленное.

Тест 2. Библиотеки и инструменты для численных методов

1. Какая библиотека Python предоставляет функции для численного интегрирования, оптимизации и решения дифференциальных уравнений?
 - a) NumPy;
 - б) SciPy;

- в) SymPy;
 - г) TensorFlow.
2. Что такое BLAS и LAPACK?
 - а) Языки программирования;
 - б) Библиотеки для линейной алгебры;
 - в) Среды разработки;
 - г) Алгоритмы машинного обучения.
 3. Для чего используется библиотека PETSc?
 - а) Для символьных вычислений;
 - б) Для решения больших систем линейных и нелинейных уравнений;
 - в) Для визуализации данных;
 - г) Для параллельных вычислений на кластерах.
 4. Какая библиотека C++ предоставляет эффективные структуры данных и алгоритмы для линейной алгебры?
 - а) STL;
 - б) Eigen;
 - в) Boost;
 - г) OpenCV.

Тест 3. Параллельное программирование

1. Что такое OpenMP?
 - а) Технология параллельного программирования для многопроцессорных систем с общей памятью;
 - б) Библиотека для работы с GPU;
 - в) Протокол передачи данных;
 - г) Язык программирования.
2. Для чего используется MPI (Message Passing Interface)?
 - а) Для параллелизации вычислений на многоядерных процессорах;
 - б) Для обмена сообщениями между процессами в распределённых системах;
 - в) Для работы с графическими ускорителями;
 - г) Для создания веб-приложений.
3. Что такое CUDA?
 - а) Язык программирования для CPU;
 - б) Платформа и модель программирования для GPU NVIDIA;
 - в) Библиотека для линейной алгебры;
 - г) Система контроля версий.
4. Какие задачи выигрывают от использования параллельных вычислений?
 - а) Решение систем дифференциальных уравнений;
 - б) Обработка больших массивов данных;
 - в) Построение графиков функций;
 - г) Моделирование физических процессов.

5. Тест 4. Инструменты разработки и воспроизводимость

1. Для чего используется Docker в научных вычислениях?
 - а) Для контейнеризации вычислительных сред;
 - б) Для ускорения вычислений на GPU;
 - в) Для обеспечения воспроизводимости экспериментов;

- г) Для написания кода.
- 2. Что такое Jupyter Notebook?
 - а) Среда для интерактивной разработки и документирования кода;
 - б) Язык программирования;
 - в) Система контроля версий;
 - г) Библиотека для визуализации.
- 3. Какая система контроля версий наиболее популярна в научных проектах?
 - а) SVN;
 - б) Git;
 - в) Mercurial;
 - г) CVS.
- 4. Для чего используются workflow-системы (Snakemake, Nextflow) в научных вычислениях?
 - а) Для автоматизации вычислительных пайплайнов;
 - б) Для написания кода;
 - в) Для управления зависимостями и версионирования данных;
 - г) Для тестирования программ.

Тест 5. Тестирование и верификация

- 1. Что такое unit-тестирование?
 - а) Тестирование всей программы целиком;
 - б) Тестирование отдельных функций или модулей;
 - в) Тестирование пользовательского интерфейса;
 - г) Тестирование производительности.
- 2. Какие инструменты используются для тестирования кода на Python?
 - а) pytest;
 - б) unittest;
 - в) Jupyter;
 - г) NumPy.
- 3. Зачем нужно тестировать численные алгоритмы?
 - а) Чтобы убедиться в корректности результатов;
 - б) Чтобы ускорить вычисления;
 - в) Чтобы проверить устойчивость к ошибкам входных данных;
 - г) Чтобы оптимизировать код.
- 4. Что такое верификация математической модели?
 - а) Проверка соответствия модели реальным данным;
 - б) Проверка корректности реализации численного метода;
 - в) Проверка производительности программы;
 - г) Проверка оформления кода.

Тест 6. Оптимизация и производительность

- 1. Какие методы могут ускорить выполнение численных алгоритмов?
 - а) Параллельные вычисления;
 - б) Оптимизация алгоритмов;
 - в) Использование более быстрых языков программирования;
 - г) Увеличение объема оперативной памяти.

2. Что такое профилирование кода?
 - а) Анализ производительности программы для выявления узких мест;
 - б) Написание комментариев к коду;
 - в) Тестирование кода;
 - г) Компиляция кода.
3. Какие инструменты профилирования доступны для Python?
 - а) cProfile;
 - б) line_profiler;
 - в) pytest;
 - г) NumPy.
4. Почему оптимизация кода важна для научных вычислений?
 - а) Чтобы сократить время расчётов;
 - б) Чтобы уменьшить потребление памяти;
 - в) Чтобы сделать код более читаемым;
 - г) Чтобы обеспечить воспроизводимость результатов.

Тест 7. Машинное обучение и анализ данных

1. Какие библиотеки Python используются для машинного обучения?
 - а) scikit-learn;
 - б) TensorFlow;
 - в) PyTorch;
 - г) Matplotlib.
2. Как можно использовать машинное обучение в математическом моделировании?
 - а) Для аппроксимации сложных функций;
 - б) Для прогнозирования результатов экспериментов;
 - в) Для оптимизации параметров моделей;
 - г) Для всех вышеперечисленных целей.
3. Что такое кросс-валидация?
 - а) Метод тестирования кода;
 - б) Метод оценки производительности моделей машинного обучения;
 - в) Метод параллельных вычислений;
 - г) Метод визуализации данных.
4. Какая библиотека Python используется для визуализации данных?
 - а) Matplotlib;
 - б) Seaborn;
 - в) Plotly;
 - г) Все вышеперечисленные.

Тест 8. Высокопроизводительные вычисления (HPC)

1. Что такое кластер в контексте HPC?
 - а) Группа компьютеров, объединённых сетью для совместной работы;
 - б) Программа для параллельных вычислений;
 - в) Язык программирования;
 - г) Тип процессора.

Критерии оценивания:

За каждое правильно отвеченное тестовое задание аспирант получает 3 балла. Максимально аспирант может получить 60 баллов.

2. ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ АСПИРАНТОВ ПО ДИСЦИПЛИНЕ ПРОВОДИТСЯ В ФОРМЕ ЗАЧЕТА. Зачетное задание включает в себя два теоретических вопроса.

ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЧЕТУ

1. Перечислите основные языки программирования, применяемые для научных вычислений и математического моделирования. Кратко охарактеризуйте преимущества и недостатки каждого.
2. В чём заключаются ключевые особенности языка Julia, делающие его перспективным для задач математического моделирования? Приведите примеры.
3. Опишите назначение и основные возможности библиотек NumPy и SciPy в контексте численных методов. Приведите примеры их использования.
4. Что такое BLAS и LAPACK? Как эти библиотеки используются в научных вычислениях?
5. Объясните принцип работы и область применения технологии OpenMP. Приведите пример параллелизации цикла.
6. В чём состоят основные отличия MPI от OpenMP? Для каких задач предпочтительнее использовать каждую из этих технологий?
7. Что представляет собой платформа CUDA? Опишите сценарии её применения в математическом моделировании и численных методах.
8. Какие методы и инструменты профилирования кода вы знаете для языков Python и C++? Как профилирование помогает оптимизировать численные алгоритмы?
9. Опишите процесс контейнеризации научных вычислений с помощью Docker. Какие преимущества это даёт для воспроизводимости вычислительных экспериментов?
10. Что такое Jupyter Notebook и как он используется в научных исследованиях? Перечислите его основные преимущества и ограничения.
11. Какие системы контроля версий применяются в научных проектах? Охарактеризуйте основные принципы работы с Git в контексте разработки научных программных комплексов.
12. Что такое unit-тестирование? Почему оно важно при реализации численных методов? Приведите пример unit-теста для функции численного интегрирования.
13. Какие инструменты и подходы используются для верификации и валидации математических моделей? Чем отличаются эти два процесса?
14. Опишите основные методы численного решения систем линейных алгебраических уравнений. Приведите примеры программных реализаций на Python или C++.
15. Как применяются методы машинного обучения в математическом моделировании? Приведите 2–3 конкретных примера использования ML для улучшения численных методов или анализа результатов моделирования.
16. Что такое workflow-системы в научных вычислениях (Snakemake, Nextflow)? Как они помогают организовать вычислительные пайплайны?
17. Опишите подходы к оптимизации производительности численных алгоритмов. Приведите конкретные примеры оптимизации кода для задач математической физики.
18. Что такое высокопроизводительные вычисления (HPC)? Как кластерные системы используются для решения задач математического моделирования? Кратко опишите архитектуру типичного HPC-кластера.
19. Какие библиотеки и инструменты Python используются для визуализации результатов численного моделирования? Сравните Matplotlib, Seaborn и Plotly по возможностям и сценариям применения.
20. Как обеспечить воспроизводимость научных вычислений? Перечислите ключевые практики (версионирование кода и данных, контейнеризация, документирование и т.д.) и кратко охарактеризуйте каждую.

Критерии оценивания:

50-100 баллов (зачтено) выставляется, если изложенный материал фактически верен, аспирант демонстрирует наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса, целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и

логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

0-49 баллов (не зачтено) - выставляется, если ответы аспиранта не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные занятия.

В ходе лекционных занятий рассматриваются темы, направленные на подготовку к сдаче кандидатского экзамена по специальности, даются рекомендации для самостоятельной работы и подготовке к лабораторным занятиям.

В ходе лабораторных занятий углубляются и закрепляются знания аспирантов по ряду рассмотренных на лекциях вопросов, развиваются навыки сбора, анализа и синтеза информации.

При подготовке к лабораторным занятиям каждый аспирант должен:

- изучить рекомендованную учебную и научную литературу;
- изучить конспекты лекций;
- подготовить ответы на все вопросы по изучаемой теме.

По согласованию с преподавателем аспирант готовит реферат по теме занятия. В процессе подготовки к лабораторным занятиям аспиранты могут воспользоваться консультациями преподавателя(ей).

Вопросы, не рассмотренные на лекциях и лабораторных занятиях, должны быть изучены аспирантами в ходе самостоятельной работы. Контроль самостоятельной работы аспирантов над учебной программой курса осуществляется в ходе лабораторных занятий.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации аспиранты могут воспользоваться электронно-библиотечными системами. Также обучающиеся могут взять на дом необходимую литературу на абонементе университетской библиотеки или воспользоваться читальными залами.