

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Ректор Белорусского
государственного университета

А.Д.Король

23 мая 2025 г.

Регистрационный № УД- 13992/уч.



ПОСТРОЕНИЕ И АНАЛИЗ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

Учебная программа учреждения образования по учебной дисциплине для
специальности:

1-31 03 07 Прикладная информатика (по направлениям)

Направление специальности:

**1-31 03 07-01 Прикладная информатика (программное обеспечение
компьютерных систем)**

2025 г.

Учебная программа составлена на основе ОСВО 1-31 03 07-2021; учебных планов БГУ №G31-1-216/уч. от 22.03.2022, №G31-1-224/уч.ин. от 27.05.2022.

СОСТАВИТЕЛИ:

Н.А.Лиходед, профессор кафедры вычислительной математики факультета прикладной математики и информатики Белорусского государственного университета, доктор физико-математических наук, профессор.

А.А.Толстиков, доцент кафедры вычислительной математики факультета прикладной математики и информатики Белорусского государственного университета, кандидат физико-математических наук

РЕЦЕНЗЕНТ:

Ю.Г.Василевский, руководитель группы разработки ООО «ЯндексБел», кандидат физико-математических наук.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой вычислительной математики БГУ
(протокол № 13 от 15.05.2025);

Научно-методическим советом БГУ
(протокол № 10 от 22.05.2025)

Заведующий кафедрой



В.И.Репников



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель учебной дисциплины – овладение студентами идеологией разработки и анализа параллельных алгоритмов на основе последовательных программ, получение навыков использования сведений по алгоритмам для параллельных вычислений; формирование составной части знаний, получаемых будущими специалистами в процессе учебы и необходимых им в дальнейшем для успешной работы; формирование у студентов основ математического мышления, изучение различных подходов к построению вычислительных алгоритмов, ориентированных на параллельные компьютеры.

Задачи учебной дисциплины:

– введение студентов в проблематику организации и анализа параллельных вычислений, введение в проблематику статического распараллеливания, основанного на знании информационной структуры программ;

– изучение основных понятий, математического аппарата и анализа параллельных вычислений;

– получение теоретических и практических основ выявления параллелизма, распараллеливания алгоритмов, преобразования последовательных программ в параллельные.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к модулю «Математические методы принятия решений и интеллектуальные системы» компоненты учреждения высшего образования.

Связи с другими учебными дисциплинами: дисциплины государственного компонента модулей «Программирование», «Дискретная математика и алгоритмы», «Информатика и компьютерные системы», дисциплины компоненты учреждения высшего образования. модулей «Методы вычислений», «Прикладное программирование»

Требования к компетенциям

Освоение учебной дисциплины «Построение и анализ параллельных алгоритмов» должно обеспечить формирование следующих компетенций:

Специализированные компетенции:

Строить и анализировать математические модели для задач принятия оптимальных решений в различных прикладных областях, обосновывать методы теоретического исследования моделей, включающие аппарат математического программирования, теории игр, вариационного исчисления, оптимального управления, использовать современные эффективные алгоритмы для решения прикладных задач на многоядерных вычислительных устройствах.

В результате освоения учебной дисциплины студент должен:

знать:

– проблематику статического распараллеливания;

– основные термины и понятия, математический аппарат параллельных вычислений;

- теоретические основы организации параллельных вычислительных процессов, распараллеливания алгоритмов, преобразования последовательных программ в параллельные;

уметь:

- обнаруживать параллелизм;
- распределять операции и данные алгоритма между процессорами;
- устанавливать порядок выполнения операций и обмена данными;

владеть:

- основами разработки параллельных программ для многоядерных персональных компьютеров и компьютеров с распределенной памятью;
- навыками программной реализации методов численного решения основных задач линейной алгебры, возникающих в различных областях естествознания.

Структура учебной дисциплины

Дисциплина изучается в 7-м семестре. В соответствии с учебным планом всего на изучение учебной дисциплины «Построение и анализ параллельных алгоритмов» отведено для очной формы получения высшего образования: 100 часов, в том числе 68 аудиторных часов: лекции – 34 часа, лабораторные занятия – 34 часа. Из них:

Лекции – 34 часа, лабораторные занятия – 30 часов, управляемая самостоятельная работа (УСР) – 4 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Введение в параллельные вычисления. Информационная структура алгоритма

Тема 1.1. Основные понятия параллельных вычислений. Блочные алгоритмы

Основная цель и задача параллельных вычислений. Терминология параллельных вычислений. Блочные матрицы. Блочные алгоритмы. Локальность алгоритма. Параллельные точечные и блочные алгоритмы для реализации на многоядерном процессоре. Локальность алгоритма. Эффективное использование кэшей при реализации алгоритмов на компьютерах с общей памятью.

Тема 1.2. Анализ зависимостей

Функции, задающие индексы массивов данных. Зависимости. Представление зависимостей. Векторы зависимостей. Функции зависимостей. Графы зависимостей. Типы зависимостей. Способы устранения ложных зависимостей. Параллельные и последовательные циклы.

Тема 1.3. Инструментальные средства для получения информационной структуры алгоритма

Визуализация информационной структуры алгоритма, получение информации о зависимостях, типах зависимостей, получение функций зависимостей и графов зависимостей с помощью инструментальных средств.

Раздел 2. Организация параллельных и распределенных вычислений

Тема 2.1. Тайлинг

Разбиение множества операций алгоритма на макрооперации-тайлы. Техника тайлинга. Перестановка циклов, распределение циклов. Допустимость тайлинга.

Тема 2.2. Организация параллельных вычислительных процессов на суперкомпьютерах с распределенной памятью

Задание зерна вычислений. Функции, задающие распределение зерен между процессорами. Условия параллельности последовательностей зернистых вычислений. Разработка и реализация на суперкомпьютере параллельных алгоритмов перемножения матриц, решения систем линейных алгебраических уравнений, решения дифференциальных уравнений в частных производных (распараллеливание, задание зерна вычислений, организация обмена данными).

Тема 2.3. Параллельные алгоритмы, связанные с графами и сетями. Модель вычислений MapReduce

Параллельные алгоритмы обработки графов: поиск в ширину, нахождение кратчайшего пути, выделение компонент связности неориентированного графа, поиск множества путей, длина которых лежит в заданном диапазоне, между двумя вершинами. Алгоритм Флойда-Уоршелла поиска кратчайших путей. Модель вычислений MapReduce, система Hadoop.

Тема 2.4 Построение и анализ параллельных алгоритмов для реализации на GPU

Организация параллельных вычислений на GPU, параллельные блоки и потоки вычислений. Рациональное использование быстрых видов памяти GPU. Локальность вычислений на GPU. Построение, реализация и экспериментальное исследование на графическом ускорителе модификаций блочного параллельного алгоритма Флойда-Уоршелла.

Раздел 3. Параллельная структура алгоритма. Инструментальные средства для распараллеливания алгоритмов

Тема 3.1. Таймирующие функции

Таймирующие функции. Строгие таймирующие функции. Расщепляющие функции. Скошенный параллелизм. Векторные таймирующие функции.

Тема 3.2. Получение таймирующих функций для однородных гнезд циклов

Получение параметров таймирующих, расщепляющих, строгих таймирующих функций посредством решения вспомогательных систем неравенств и уравнений. Алгоритм получения таймирующих функций.

Тема 3.3. Аффинные преобразования гнезд вложенных циклов

Преобразования тесно вложенных циклов, задаваемые векторными таймирующими функциями. Условия параллельности внутренних циклов после преобразования. Условия параллельности внешних циклов после преобразования.

Тема 3.4. Генерация кода. Инструментальные средства для получения параллельных алгоритмов

Генерация кода: запись алгоритма после аффинного преобразования. Инструментальные средства для получения параллельных алгоритмов (в том числе и для генерации кода). Системы LooPo, CLooG, PLUTO.

Раздел 4. Некоторые теоретические аспекты параллельных вычислений

Тема 4.1. Характеристики параллельных вычислительных процессов и систем. Списки Top 500 и Graph 500

Производительность, загруженность, ускорение вычислительной системы. Законы Амдала. Сетевой закон Амдала. Закон Густавсона-Барсиса. Сравнительный анализ законов Амдала и Густавсона-Барсиса. Список Top 500. Список Graph 500.

Тема 4.2. Параллельная форма алгоритма. Концепция неограниченного параллелизма. Основные классы современных вычислительных систем

Параллельные множества операций, параллельная форма алгоритма, параллельные последовательности вычислений. Параллельные вычислительные свойства и параллельная структура алгоритма. Концепция неограниченного параллелизма. Принцип сдваивания. Процесс рекуррентного сдваивания. Основные классы современных параллельных компьютеров. Классификация Флинна параллельных компьютеров.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования с применением дистанционных образовательных технологий (ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Форма контроля знаний
		Лекции	Практические занятия	Семинарские (практические) занятия	Лабораторные занятия	Количество часов УСР	
1	2	3	4	5	6	7	8
1	Введение в параллельные вычисления. Информационная структура алгоритма	6			8	2	
1.1	Основные понятия параллельных вычислений. Блочные алгоритмы.	2			2		Отчет по лабораторной работе, экспресс-опрос
1.2	Анализ зависимостей	4			4	2	Отчет по лабораторной работе, экспресс-опрос
1.3	Инструментальные средства для получения информационной структуры алгоритма				2		Отчет по лабораторной работе (часть 2)
2	Организация параллельных и распределенных вычислений	16			14		
2.1	Тайлинг	2			2		Экспресс-опрос
2.2	Организация параллельных вычислительных процессов на суперкомпьютерах с распределенной памятью	4			4		Отчет по лабораторной работе
2.3	Параллельные алгоритмы, связанные с графами и сетями. Модель вычислений MapReduce	6			4		Отчет по лабораторной работе, экспресс-опрос
2.4	Построение и анализ параллельных алгоритмов для реализации на GPU	4			4		Отчет по лабораторной работе, экспресс-опрос

3	Параллельная структура алгоритма. Инструментальные средства для распараллеливания алгоритмов	8			8	2	
3.1	Таймирующие функции	1					Экспресс-опрос
3.2	Получение таймирующих функций для однородных гнезд циклов	1					Экспресс-опрос
3.3	Аффинные преобразования гнезд вложенных циклов	2			4		Экспресс-опрос
3.4	Генерация кода. Инструментальные средства для получения параллельных алгоритмов	4			4	2	Контрольная работа по темам 2.3, 3.3, 3.4, экспресс-опрос
4	Некоторые теоретические аспекты параллельных вычислений	4					
4.1	Характеристики параллельных вычислительных процессов и систем. Списки Top 500 и Graph 500	2					Экспресс-опрос
4.2	Параллельная форма алгоритма. Концепция неограниченного параллелизма. Основные классы современных вычислительных систем	2					Экспресс-опрос

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Роби, Р. Параллельные и высокопроизводительные вычисления / Роберт Роби и Джулиана Замора ; [пер. с англ. А. В. Логунова]. - Москва : ДМК Пресс, 2022. – 799 с.
2. Хабаров, С. П. Построение распределенных систем на базе WebSocket : Учебное пособие для вузов / С. П. Хабаров, М. Л. Шилкина. – 2-е изд., стер. – Санкт-Петербург : Лань, 2025. – 216 с. – URL: <https://e.lanbook.com/book/200510>.
3. Стин, Мартен ван. Распределенные системы / Мартен ван Стин, Эндрю С. Таненбаум ; [пер. с англ. В. А. Яроцкого]. – Москва : ДМК Пресс, 2021. – 583 с.

Дополнительная литература

1. Воеводин В.В., Воеводин Вл. В. Параллельные вычисления. – Санкт-Петербург. БХВ-Петербург. 2002. – 600 с.
2. Парфенов Д. В., Петрусевич Д. А. Параллельные и распределенные вычисления. – М.: МИРЭА - Российский технологический университет. 2022. – 92 с.
3. Лиходед Н.А. Методы распараллеливания гнезд циклов: Курс лекций. – Мн.: БГУ. 2008. – 100 с.
4. Саад Ю. Итерационные методы для разреженных линейных систем. В 2-х томах. Том 1. – М.: Издательство Московского университета, 2013. – 324 с. Том 2. – М.: Издательство Московского университета, 2014. – 306 с.
5. Vorst van der H. Iterative Krylov methods for large linear systems. – Cambridge: Cambridge University Press, 2003. 221 p.
6. Гервич Л.Р., Штейнберг Б.Я., Юрушкин М.В. Разработка параллельных программ с оптимизацией использования структуры памяти. – Ростов-на-Дону. Изд-во Южного федерального университета, 2014. – 120 с.
7. Гergель В.П. Высокопроизводительные вычисления для многопроцессорных многоядерных систем. Учебник. М.: Изд-во Московского ун-та, 2010. 544 с. (Серия «Суперкомпьютерное образование»)
8. [Электрон. ресурс – <https://edufpmi.bsu.by/> ПРВ / Лекции].
9. [Электрон. ресурс – <https://edufpmi.bsu.by/> ПРВ / Лабораторные работы].
10. LooPo. www.fmi.uni-passau.de/loopo.
11. Открытая распараллеливающая система: www.ops.rsu.ru.
12. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. – Москва. ДМК Пресс, 2019. – 232 с.
13. Лиходед Н.А., Толстиков А.А. Метод оценки локальности параллельных алгоритмов, ориентированных на компьютеры с распределенной памятью // Доклады НАН Беларуси. 2020. Т. 64. № 6. С. 647–656.
14. Толстиков А.А., Баханович С.В., Лиходед Н.А. Подход к оценке локальности зернистых вычислительных процессов, логически организованных в двумерную структуру // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 3. С. 37–55.

15. Лиходед Н.А. Условия существования бродкаста и пространственной локальности в потоках вычислений // Вес. Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. 2022. Т. 58, № 3. С. 292–299.

16. Воеводин Вл.В., Воеводин Вад.В. Спасительная локальность суперкомпьютеров // Открытые системы. 2013, № 9. С. 12–15.

17. Библиотеки BLAS, LAPACK, BLACS, ScaLAPACK: <http://www.netlib.org>

18. CLooG: The Chunky Loop Generator. <http://www.cloog.org>.

19. PLUTO: pluto-compiler.sourceforge.net.

20. Бёрнс Б. Распределенные системы. Паттерны проектирования [пер. с англ. К. Русецкого]. – Санкт-Петербург [и др.]: Питер, 2019. – 222 с. – URL: <https://ibooks.ru/bookshelf/361843>

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Объектом диагностики компетенций студентов являются знания, умения, полученные ими в результате изучения учебной дисциплины. Выявление учебных достижений студентов осуществляется с помощью мероприятий текущего контроля и промежуточной аттестации.

Для текущего контроля качества усвоения знаний студентами используется следующий диагностический инструментарий:

- отчеты по лабораторным работам;
- письменные контрольные работы;
- устные экспресс-опросы.

Лабораторные работы, как правило, представляют собой задания, включающие программную реализацию указанного численного метода, проведение вычислительного эксперимента и комментарии по его итогам. Рекомендуемая форма отчетности по лабораторной работе – письменный отчет. Лабораторная работа оценивается по 10-балльной шкале. Отметка за лабораторную работу может быть снижена в случае несвоевременного выполнения.

Письменные контрольные работы проводятся для контроля знаний по одному или нескольким разделам дисциплины. Они включают 4–5 заданий и оцениваются по 10-балльной шкале. В случае неудовлетворительной отметки контрольная работа может быть переписана.

Устный экспресс-опрос студентов проводится в свободной форме во время лабораторных и лекционных занятий. Его результаты учитываются преподавателем при выставлении итоговой отметки.

Формой промежуточной аттестации по дисциплине «Построение и анализ параллельных алгоритмов» учебным планом предусмотрен зачёт.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 1.2. Анализ зависимостей (2 ч).

Примерный перечень заданий:

1. Найти зависимости указанного алгоритма.
2. Найти функции зависимостей указанного алгоритма: вид функции, область определения.
3. Построить развернутый и редуцированный графы зависимостей указанного алгоритма.

Форма контроля – отчет по лабораторной работе.

Тема 3.4. Генерация кода. Инструментальные средства для получения параллельных алгоритмов (2 ч).

Примерный перечень заданий:

1. Применить для указанной задачи параллельный алгоритм поиска в ширину в графе с 2D разбиением графа.
2. Показать, что указанная векторная функция является таймирующей.
3. Обосновать параллельность внутреннего цикла после применения к алгоритму аффинного преобразования.
4. Записать псевдокод, получаемый после аффинного преобразования.

Форма контроля - контрольная работа по темам 2.3, 3.3, 3.4.

Примерная тематика лабораторных занятий

Занятие 1. Блочные алгоритмы.

Занятие 2. Представление зависимостей.

Занятие 3. Способы устранения ложных зависимостей.

Занятие 4. Визуализация информационной структуры алгоритма.

Занятие 5. Техника тайлинга.

Занятие 6. Разработка для реализации на суперкомпьютере параллельных алгоритмов.

Занятие 7. Реализация на суперкомпьютере параллельных алгоритмов.

Занятие 8. Параллельные алгоритмы поиска в ширину.

Занятие 9. Модель вычислений MapReduce.

Занятие 10. Организация параллельных вычислений на GPU.

Занятие 11. Построение и реализация на графическом ускорителе алгоритма Флойда-Уоршелла.

Занятие 12. Преобразования тесно вложенных циклов, задаваемые векторными таймирующими функциями.

Занятие 13. Условия параллельности внутренних циклов после преобразования.

Занятие 14. Запись алгоритма после аффинного преобразования.

Занятие 15. Система PLUTO.

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используются следующие методы:

- **практико-ориентированный подход**, который предполагает освоение содержание учебного материала через решение практических задач, приобретение навыков эффективного выполнения разных видов профессиональной деятельности;
- **метод группового обучения**, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

Методические рекомендации по организации самостоятельной работы

Для организации самостоятельной работы студентов по учебной дисциплине «Построение и анализ параллельных алгоритмов» следует использовать современные информационные технологии: разместить в сетевом доступе комплекс учебных и учебно-методических материалов (учебно-программные материалы, методические указания к лабораторным занятиям, материалы текущего контроля и промежуточной аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательных стандартов высшего образования и учебно-программной документации, в том числе вопросы для подготовки к зачету, список рекомендуемой литературы, информационных ресурсов и др.). Эффективность самостоятельной работы студентов проверяется в ходе текущего и итогового контроля знаний.

Примерный перечень вопросов к зачету

1. Определение зависимости между операциями. Типы зависимостей (истинная зависимость, антизависимость, зависимость по выходу).
2. Условие параллельности внешних циклов после применения аффинного преобразования гнезда циклов.
3. Таймирующие функции, строгие таймирующие функции, расщепляющие функции (определения и примеры).
4. Функция зависимостей. Вектор зависимости. Однородные и аффинные зависимости.
5. Понятие о локальности алгоритмов. Локальность при использовании компьютеров с распределенной памятью, CPU. Улучшение локальности данных. Пространственная локальность. Временная локальность.
6. Развернутый и редуцированный графы зависимостей.
7. Теорема о получении таймирующих функций для однородных гнезд циклов.

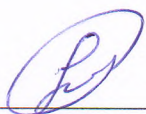
8. Разбиение множества операций алгоритма на макрооперации-тайлы. Примеры использования тайлинга.
9. Условия параллельности внутренних циклов после применения аффинного преобразования гнезда циклов.
10. Закон Амдала (третий закон Амдала).
11. Алгоритм получения таймирующих функций для однородных гнезд циклов.
12. Устранение ложных зависимостей путем переобозначения или увеличения размерности массивов.
13. Получение параллельных зернистых вычислительных процессов (на примере).
14. Векторные таймирующие функции и аффинные преобразования циклов (определения и примеры).
15. Основные задачи статического распараллеливания алгоритмов, заданных последовательными программами.
16. Этапы и пример генерация кода после аффинного преобразования гнезда циклов.
17. Классификация Флинна параллельных компьютеров.
18. Этапы организации параллельных зернистых вычислительных процессов и обмена данными.
19. Производительность, загруженность, ускорение вычислительной системы (определения). Формулы, выражающие производительность и ускорение системы из s процессоров через загруженность.
20. Цель и задача параллельных вычислений. Терминология параллельных вычислений.
21. Параллельная форма алгоритма. Наименьшая возможная высота алгоритма, зависящего от N переменных.
22. Список Top 500. Список Graph 500.
23. Закон Густавсона-Барсиса.
24. Формулы максимальных производительности, загруженности, ускорения. Первый закон Амдала.
25. Основные факторы, определяющие скорость выполнения алгоритма на параллельном компьютере. Проблемы вычислительной математики, которые возникают при использовании параллельных компьютеров.
26. Перестановка циклов: достаточные условия, применение для распараллеливания и тайлинга.
27. Сетевой закон Амдала.
28. Параллельные множества операций, параллельная форма алгоритма, параллельные последовательности вычислений. Параллельные вычислительные свойства и параллельная структура алгоритма.
29. Второй закон Амдала.
30. Распределение циклов: достаточные условия, применение для распараллеливания и тайлинга.
31. Использование таймирующих функций.

32. Концепция неограниченного параллелизма. Максимальное ускорение системы из s ядер, реализующей N операций алгоритма высоты m .
33. Сравнительный анализ второго закона Амдала и закона Густавсона-Барсиса.
34. Параллельные множества операций. Естественный параллелизм. Внутренний параллелизм. Параллельные вычислительные процессы.
35. Допустимость тайлинга (необходимые и достаточные условия, достаточные условия; формулировки и иллюстрации).
36. Сетевой аналог третьего закон Амдала.
37. Применение теоремы Фурье-Моцкина к генерации кода.
38. Расщепление операторов (устранение циклов с ложными зависимостями в редуцированных графах зависимостей).
39. 2D разбиение графа.
40. Параллельный алгоритм поиска в ширину, использующий 2D разбиение графа, для реализации на компьютере с распределенной памятью (детали 2D разбиения графа пояснять не требуется).
41. Тайлинг для улучшения локальности гнезд циклов при последовательных вычислениях. Выбор циклов для формирования тайлов.
42. Исследование локальности альтернативных вариантов параллельных зернистых алгоритмов перемножения матриц.
43. Параллельный алгоритм поиска в ширину, использующий 1D разбиение графа, для реализации на компьютере с распределенной памятью.
44. Теорема о получении таймирующих функций для аффинных гнезд циклов.
45. Исследование локальности альтернативных вариантов параллельных зернистых алгоритмов прямого хода метода Гаусса.
46. Получение наибольшего числа независимых таймирующих функций.
47. Представление о способах улучшения локальности: перерасмещение данных, тайлинг.
48. Формулировка теоремы о параллельности всех зернистых вычислительных процессов.
49. Двухуровневый тайлинг для распараллеливания алгоритмов при программировании общего назначения на графических процессорах. Пример.
50. Использование блочных алгоритмов в последовательных и в параллельных вычислениях.
51. Блочные матрицы. Операции над блочными матрицами.
52. Основные классы современных вычислительных систем.
53. Формулировка теоремы о допустимости тайлинга, применяемого для получения параллельных зернистых последовательностей вычислений.
54. Параллельный алгоритм поиска в ширину для реализации на многоядерном компьютере.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Учебная дисциплина не требует согласования			

Заведующий кафедрой
вычислительной математики
кандидат физ.-мат. наук, доцент



В.И.Репников

15.05.2025

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УО

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры
_____ (протокол № ____ от _____ 202_ г.)

Заведующий кафедрой

УТВЕРЖДАЮ
Декан факультета
