

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Лиодинович
Должность: Ректор
Дата подписания: 24.03.2026 13:02:13
Уникальный программный ключ:
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

Институт кибербезопасности и цифровых технологий

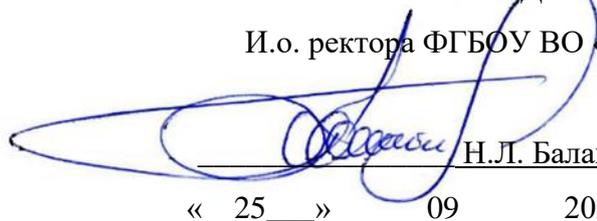
Региональный партнёр

ФГБОУ ВО

«Дагестанский государственный технический университет»

УТВЕРЖДАЮ

И.о. ректора ФГБОУ ВО «ДГТУ»


Н.Л. Баламирзоев

« 25 » 09 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДВ.02.02 ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Направленность (профиль подготовки): «Прикладной искусственный интеллект»

Квалификация выпускника: бакалавр

Форма обучения: очная

Махачкала 2023

Программа составлена в соответствии с требованиями ФГОС ВО с учетом рекомендаций ОПОП ВО по направлению подготовки 09.03.01. – Информатика и вычислительная техника, профилю «Прикладной искусственный интеллект»

Разработчик


подпись

Магомедов И.А., к.т.н, доцент

(ФИО уч. степень, уч. звание)

05.09.2023г.

Зав. кафедрой, за которой закреплена дисциплина (модуль)


подпись

Гасанова Н.М., к.э.н., доцент

(ФИО уч. степень, уч. звание)

05.09. 2023г.

Программа одобрена на заседании выпускающей кафедры УиИТСиВТ

от 12.09.2023 г., протокол № 1

Зав. выпускающей кафедрой по данному направлению (специальности, профилю)


подпись

Гасанова Н.М., к.э.н., доцент

(ФИО уч. степень, уч. звание)

от 12.09.2023 г.

Программа одобрена на заседании Методического совета факультета компьютерных технологий, вычислительной техники и энергетики от 22.09.2023 года, протокол № 1.

Председатель Методического совета факультета КТВТиЭ


подпись

Исабекова Т.И., к.ф.-м. н., доцент

(ФИО уч. степень, уч. звание)

«22» 09. 2023 г

Декан факультета


подпись

Ш.А. Юсуфов

ФИО

Начальник УО


подпись

Э.В. Магомаева

ФИО

1. Цель и задачи дисциплины

Цель дисциплины: формирование у студентов систематизированных знаний и практических навыков в области проектирования, разработки и оптимизации параллельных программ для современных многопроцессорных и многопоточных вычислительных систем, ориентированных на решение задач искусственного интеллекта.

Задачи дисциплины:

1. Изучить основные модели, парадигмы и инструменты параллельного программирования.
2. Сформировать умения разработки параллельных программ с использованием технологий OpenMP, MPI и основ CUDA.
3. Освоить методы распараллеливания типовых алгоритмов, включая алгоритмы обработки данных и машинного обучения.
4. Развить навыки анализа, отладки и оценки эффективности параллельных программ.
5. Приобрести опыт решения практических задач ИИ с использованием параллельных вычислений.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Параллельное программирование» относится к вариативной части (дисциплины по выбору). Для успешного освоения необходимы знания и компетенции, полученные при изучении дисциплин: «Программирование», «Алгоритмы и структуры данных», «Архитектура ЭВМ и систем», «Операционные системы». Является логическим продолжением и практическим углублением курсов по алгоритмам и архитектуре ЭВМ.

3. Результаты освоения дисциплины "Параллельное программирование"

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению:

Код	Наименование компетенции	Индикатор достижения компетенции	В результате освоения дисциплины обучающийся должен:	Другая дисциплина (дисциплины) / практика, участвующая в формировании компетенции
ОПК-4	Способен разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования	ОПК-4.1 Применяет технологии параллельного программирования (OpenMP, MPI) для разработки компонентов вычислительных систем.	<p> Знать: архитектурные модели параллельных систем (SMID, MIMD), основы технологий OpenMP и MPI.</p> <p>Уметь: разрабатывать параллельные программы для систем с разделяемой и распределенной памятью.</p> <p>Владеть: навыками отладки и оптимизации параллельных приложений. Архитектура ЭВМ и систем. Программирование. Операцион</p>	Архитектура ЭВМ и систем. Программирование. Операционные системы.
		ОПК-4.2 Использует технологии GPU-программирования (CUDA/OpenCL) для решения вычислительных задач.	<p>Знать: архитектуру GPU, модель выполнения CUDA, иерархию памяти.</p> <p>Уметь: разрабатывать и оптимизировать ядра (kernels) для выполнения на графических процессорах.</p> <p>Владеть: инструментами разработки и профилирования</p>	Алгоритмы и структуры данных. Математические основы искусственного интеллекта.

			CUDA-приложений. Алгоритмы и а.	
ПК-1	Способен проводить предпроектное обследование, анализ предметной области, проектировать и моделировать компоненты информационных систем	ПК-1.1 Анализирует вычислительную задачу на предмет распараллеливания и выбирает адекватную НРС-технологию.	<p>Знать: метрики производительности (ускорение, эффективность), законы Амдала и Густафсона.</p> <p>Уметь: проводить анализ алгоритма для выявления независимых ветвей вычислений и потенциальных узких мест.</p> <p>Владеть: методами проектирования параллельных алгоритмов.</p>	Методы оптимизации. Теория алгоритмов.
		ПК-1.2 Проектирует архитектуру параллельного приложения для решения задач искусственного интеллекта.	<p>знать: особенности распараллеливания алгоритмов машинного обучения (матричные операции, градиентный спуск).</p> <p>Уметь: проектировать гибридные (MPI+OpenMP) и распределенные решения для обучения моделей ИИ.</p> <p>Владеть: основами работы с системами управления заданиями (SLURM). Системы</p>	Системы искусственного интеллекта. Нейронные сети.

ПК-9	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ПК-9.1 Настраивает программную среду для разработки и выполнения параллельных приложений.	<p>Знать: состав и требования к программному стеку HPC (компиляторы, библиотеки, драйверы).</p> <p>Уметь: устанавливать и настраивать MPI-среду (OpenMPI), CUDA Toolkit, библиотеки для HPC.</p> <p>Владеть: навыками работы в среде Linux, использования контейнеризации (Docker/Singularity) для HPC.</p>	Администрирование информационных систем.
		<p>ПК-9.2</p> <p>Оценивает и подбирает аппаратные ресурсы для выполнения высокопроизводительных вычислений. </p>	<p>Знать: характеристики современных процессоров, GPU, сетевых технологий (InfiniBand).</p> <p>Уметь: оценивать требуемые вычислительные ресурсы (ядра, память, дисковое пространство) для задачи.</p> <p>Владеть: основами чтения и интерпретации спецификаций аппаратного обеспечения.</p>	Архитектура ЭВМ и систем.

4. Структура и содержание дисциплины (модуля)
«Параллельное программирование»»

4.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 3 зачетных единицы, 108 часа.

Форма обучения	Семестр	Общая трудоёмкость по дисциплине (ЗЕТ/ в часах)	Лекции, час	Практические занятия, час	Лаб. зан, час	СРС, час	Контр., час	Контроль
Очно	8	3/108	24	-	24	24	36	Экз.

4.2. Содержание дисциплины (модуля) «Параллельное программирование»

№ п/п	Раздел дисциплины, тема лекции и вопросы	Энная форма				Заочная форма			
		ЛК	ПЗ	ЛБ	СР	ЛК	ПЗ	ЛБ	СР
1	2	3	4	5	6	7	8	9	10
1.	Модуль 1. Введение в параллельное программирование и архитектурные основы. Тема 1.1. Основные понятия и мотивация. Параллелизм vs. конкуренция. Процессы, потоки, нити. Классификация параллельных вычислительных систем (классификация Флинна: SISD)	2		2	2				4

2.	<p>Модуль 2. Модели и паттерны параллельного программирования.</p> <p>Тема 2.1. Обзор моделей: разделяемая память, передача сообщений, гибридная модель, модели потоков данных и активных сообщений.</p> <p>Тема 2.2. Основные паттерны параллельных алгоритмов: «Вожак-Рабочие» (Master-Worker).</p>	2		2	2	2		2	4
3.	<p>Модуль 3. Параллельное программирование на разделяемой памяти (OpenMP). Основы.</p> <p>Тема 3.1. Архитектура и модель исполнения OpenMP. Директивы, клаузы, функции среды выполнения. Директива <code>parallel</code>. Управление об-</p>	2		2	2				4
4.	<p>Модуль 4. Параллельное программирование на разделяемой памяти (OpenMP). Продвинутое программирование.</p> <p>Тема 4.1. Синхронизация: критические секции (<code>critical</code>), атомарные операции (<code>atomic</code>), барьеры (<code>barrier</code>), явные блокировки (Lock API).</p> <p>Тема 4.2. Редукционные операции (<code>reduction</code>). Модель задач (<code>task</code>) для выражения нерегулярного и асинхронного параллелизма. Директива <code>taskloop</code>.</p>	2		2	2				4
5.	<p>Модуль 5. Программирование на распределенной памяти (MPI). Основы.</p> <p>Тема 5.1. Введение в Message Passing Interface (MPI). Концепция коммутатора (<code>MPI_COMM_WORLD</code>). Ранг и размер. Инициализация и завершение (<code>MPI_Init</code></p>	2		2	2	2		2	4

6.	<p>Модуль 6. Программирование на распределенной памяти (MPI). Коллективные операции и продвинутые темы.</p> <p>Тема 6.1. Коллективные операции связи: `MPI_Bcast`, `MPI_Scatter`, `MPI_Gather`, `MPI_Reduce`.</p>	2		2	2				4
7.	<p>Модуль 7. Основы программирования графических процессоров (CUDA).</p> <p>Тема 7.1. Архитектура GPU NVIDIA. Моделирование GPU.</p>	2		2	2				4
8.	<p>Модуль 8. Программирование GPU (CUDA): разработка и оптимизация.</p> <p>Тема 8.1. Написание и запуск CUDA-ядер.</p>	2		2	2	2		2	4
9.	<p>Модуль 9. Альтернативные технологии и гибридное программирование.</p> <p>Тема 9.1. Гибридная модель MPI + GPU.</p>	2		2	2				4
10.	<p>Модуль 10. Параллельные алгоритмы и структуры данных.</p> <p>Тема 10.1. Параллельные версии базовых алгоритмов: параллельная редукция, сканирование (prefix sum), сортировка (например, параллельная быстрая сортировка).</p> <p>Тема 10.2. Параллельные структуры данных.</p>	2		2	2				4
11.	<p>Тема 11.1. Распараллеливание базовых операций линейной алгебры: умножение матриц (GEMM), свертки. Использование специализированных библиотек (cuBLAS, cuDNN).</p> <p>Тема 11.2. Стратегии распределенного обучения нейронных сетей: Data Parallelism, Model Parallelism, Pipeline Parallelism. Обзор современных архитектур.</p>	2		2	2				4

12.	<p>Модуль 12. Производительность, отладка и инструментарий.</p> <p>Тема 12.1. Метрики и анализ производительности. Профилирование параллельных приложений: инструменты (Intel VTune, NVIDIA Nsight Systems, Scalasca). Поиск узких мест (bottlenecks).</p> <p>Тема 12.2. Отладка и верификация параллельных программ. Распространенные ошибки (гонки данных, взаимные блокировки/deadlocks).</p>	2		2	2				4
13.	<p>Форма текущего контроля успеваемости (по срокам текущих аттестаций в семестре)</p>	<p>Входная контр, работа;</p> <p>1 аттестация 1-5 тема</p> <p>2 аттестация 3-9 тема</p> <p>3 аттестация 10-12 тема</p>							
14.	<p>Форма промежуточной аттестации (по семестрам)</p>	<p>Экзамен</p> <p>1 ЗЕТ (36 часов)</p>				Экзамен			
15.	ИТОГО	24		24	24	6	-	6	48

4.3 Перечень и содержание лабораторных занятий.

№ п/п	№ разделов	Наименование лабораторных работ	Кол. ч
1.	2	1. Основы OpenMP: Параллельное вычисление числа π методом Монте-Карло. Анализ ускорения.	4
2.	2, 3	Синхронизация в OpenMP: Реализация потокобезопасного счетчика и очереди. Сравнение `critical`, `atomic`, мьютексов.	4
3.	4	Задачи (Tasks) в OpenMP: Параллельный обход графа или дерева.	4
4.	5,6	Базовые операции MPI: Обмен сообщениями между процессами. Кольцевая передача данных.	4
5.	7	Коллективные операции MPI: Параллельное умножение матрицы на вектор с использованием `MPI_Scatter` и `MPI_Gather`.	2
6.	8	Гибридная программа (MPI+OpenMP): Решение уравнения теплопроводности.	2
7.	9,10	Первое CUDA-ядро: Поэлементное сложение векторов. Сравнение производительности с CPU.	2
8.	11,12	Оптимизация в CUDA: Умножение матриц с использованием разделяемой памяти и оптимизацией по циклу тайлов.	2
Итого за семестр			24

5. Образовательные технологии

В ходе освоения дисциплины "Параллельное программирование»" при проведении аудиторных занятий используется образовательная технология, предусматривающая такие методы и формы изучения материала как лекция, лабораторное занятие, включающие активные и интерактивные формы занятий:

- Проведение лекции проблемного характера: тема 1.1. " Принципы построения численных методов поиска безусловного экстремума. Методы нулевого порядка»; тема 3.3. "Методы решения транспортных задач".
- Проведение лабораторных занятий в интерактивной форме и публичная защита отчетов по лабораторным работам, работа в малых группах.

Занятия, проводимые в интерактивной форме, составляют 25 % от общего количества аудиторных занятий.

Лабораторные занятия проводятся в с использованием специализированных пакетов.

Самостоятельная работа студентов подразумевает работу под руководством преподавателя (консультации, помощь в написании и отладке программ и др.) и индивидуальную работу студента, выполняемую как дома, так и в компьютерном классе с выходом в Интернет.

При реализации образовательных технологий используются следующие виды самостоятельной работы:

- работа с конспектом лекции и литературой;

- подготовка к лабораторной работе: изучение теоретического материала, разработка и отладка программ заданий по лабораторным работам;
- обработка результатов лабораторных работ и подготовка письменных отчетов;
- выполнение и оформление индивидуальных домашних заданий: изучение теоретического материала, разработка алгоритма решения задачи, разработка и отладка программ, вычислительный эксперимент с разработанной программой, оформление письменного отчета по индивидуальному заданию;
- поиск информации в Интернет и литературе;
- подготовка к сдаче лабораторных работ и индивидуальных заданий;
- подготовка к сдаче экзамена.

В целях реализации индивидуального подхода к обучению студентов, осуществляющих учебный процесс по собственной траектории в рамках индивидуального рабочего плана, изучение данной дисциплины базируется на следующих возможностях: обеспечение внеаудиторной работы со студентами в том числе в электронной образовательной среде с использованием соответствующего программного оборудования, дистанционных форм обучения. возможностей интернет-ресурсов, индивидуальных консультаций и т.д.

В качестве других видов контактной работы запланированы консультации при подготовке и проведении текущей и промежуточной аттестации.

При организации самостоятельной работы студентов и, при необходимости, при проведении аудиторных занятий используются /могут быть использованы дистанционные образовательные технологии.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

6.1. План самостоятельной работы студентов

ПЕРЕЧЕНЬ ТЕМ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ (СРС)

№ модуля	Наименование модуля дисциплины	Тема СРС и содержание самостоятельной работы	Объем часов	Форма отчетности / контроля
1	Введение в параллельное программирование	Изучение современных архитектур CPU. Самостоятельный анализ характеристик многоядерных процессоров (Intel, AMD, ARM) и их влияния на стратегии параллелизации. Сравнение архитектур x86 и ARM с точки зрения параллельных вычислений.	3	Краткий аналитический отчет (таблица с характеристиками, выводы)
2	Модели и паттерны	Исследование паттернов параллельного программирования. Глубокий разбор паттернов "Master-	4	Презентация с разбором одного кейса применения паттерна

		Worker", "Pipeline", "Divide and Conquer". Поиск и анализ примеров их применения в реальных программных системах (например, веб-серверы, системы обработки данных).		
3	OpenMP. Основы	Конфигурация и настройка среды разработки. Самостоятельная установка и настройка компилятора с поддержкой OpenMP (gcc, Clang, MSVC) в среде Linux/Windows. Написание и отладка простейшей параллельной программы "Hello World" с использованием <code>#pragma omp parallel</code> .	2	Скриншоты рабочей среды и исходный код программы
4	OpenMP. Продвинутое техники	Разработка параллельного алгоритма сортировки. Изучение и реализация параллельной версии алгоритма сортировки слиянием (Merge Sort) или быстрой сортировки (QuickSort) с использованием директив OpenMP (tasks, sections). Сравнение производительности с последовательной версией.	5	Отчет с кодом, графиками ускорения и анализом результатов
5	MPI. Основы	Самостоятельное изучение коммутаторов и групп. Изучение функций <code>MPI_Comm_split</code> , <code>MPI_Comm_rank</code> , <code>MPI_Comm_size</code> . Создание собственных коммутаторов для логического разделения процессов. Разработка простого приложения с несколькими коммутаторами.	3	Исходный код программы с комментариями
6	MPI. Коллективные операции	Оптимизация коллективных операций. Исследование производительности операций <code>MPI_Allreduce</code> , <code>MPI_Alltoall</code> на разных объемах данных. Сравнение с цепочкой точечных операций. Эксперименты на локальной машине и в эмуляции распределенной среды.	4	Отчет с таблицами замеров времени и рекомендациями по выбору операции
7	Архитектура CUDA	Исследование реальных GPU. Анализ спецификаций современных GPU NVIDIA (серии A100, H100, GeForce RTX 40xx). Изучение параметров: количество SM, размеры памяти, пропускная способность. Расчет теоретической пиковой производительности (FLOPS).	3	Таблица с характеристиками и расчетами

8	Программирование CUDA	Разработка оптимизированного ядра для свертки. Реализация одномерной или двумерной свертки (конволюции) с использованием разделяемой памяти и объединенного доступа. Сравнение производительности с наивной реализацией.	6	Код ядра, скрипт тестирования, анализ скорости работы
9	Гибридное программирование	Сравнение гибридных стратегий. Реализация одной вычислительной задачи (например, решение СЛАУ методом Якоби) в трех вариантах: чистая MPI, чистая OpenMP, гибрид MPI+OpenMP. Сравнение времени выполнения и эффективности использования ресурсов.	5	Сводная таблица результатов, выводы о целесообразности каждого подхода
10	Параллельные алгоритмы	Обзор библиотек параллельных алгоритмов. Изучение библиотек Intel TBB (Threading Building Blocks) или Microsoft PPL (Parallel Patterns Library). Сравнение их возможностей с OpenMP. Написание небольшой демонстрационной программы.	3	Краткий обзорный реферат с примерами кода
11	Параллельные вычисления в ИИ	Анализ параллельных фреймворков для ИИ. Сравнительное исследование возможностей Horovod, PyTorch Distributed Data Parallel (DDP) и TensorFlow Distribution Strategies. Создание простого тестового сценария обучения на синтетических данных.	5	Отчет с анализом преимуществ/недостатков каждого фреймворка
12	Инструменты и отладка	Профилирование реального приложения. Выбор небольшого open-source параллельного проекта. Его запуск, профилирование с помощью NVIDIA Nsight Systems или Intel VTune. Анализ полученного профиля, выявление узких мест и предложения по оптимизации.	4	Презентация с скриншотами профиля и списком рекомендаций
	Итоговая работа	Проектирование параллельной системы для задачи ИИ. Разработка архитектурного решения для распараллеливания конкретной задачи ИИ (например, инференс большой языковой модели, обучение рекомендательной системы). Выбор технологий, оценка требуемых	5	Техническое предложение/архитектурный меморандум (итоговый документ)

		ресурсов, проектирование схемы взаимодействия компонентов.		
Итого часов:				48

6.2 Методические указания по организации самостоятельной работы студентов

Планируются следующие виды самостоятельной работы:

- подготовка к лабораторным и лекционным занятиям,
- выполнение индивидуального задания,
- оформление отчётов по лабораторным работам,
- подготовка к экзамену.

Подготовка к лабораторным занятиям проводится посредством изучения курса лекций, дополнительной литературы, Интернет-ресурсов.

Задание к выполнению каждой лабораторной работы состоит из общей части, которая сформулирована в разделе «Задание к выполнению» и уточнения варианта, который приведен в разделе «Варианты заданий». Студент должен заранее ознакомиться со своим заданием и, если у него возникают какие-либо вопросы относительно задания, поставить эти вопросы преподавателю до начала работы.

Отчёт к лабораторной работе должен содержать:

- Тему работы;
- Цель работы;
- Задание для выполнения, включая индивидуальное задание;
- Описание алгоритма программы (при необходимости, со схемой алгоритма);
- Описание переменных и структур данных, которые применяются в программе;
- Описание ключевых программных решений, принятых при реализации алгоритма в тексте программы;
- Текст программы;
- Результат работы программы;
- Выводы.

Подготовка к экзамену проводится посредством изучения курса лекций, изучения литературы, Интернет-ресурсов.

Студентам из числа лиц с ограниченными возможностями здоровья могут быть предложены электронные образовательные ресурсы в формах, адаптированных к ограничениям их здоровья.

6.3. Материалы для проведения текущего и промежуточного контроля знаний студентов

Контроль освоения компетенций

№ п/п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
1	Текущий: проверка выполнения индивидуального задания на лабораторной работе	Разделы 1 – 2	ПК-1
2	Текущий: собеседование при защите лабораторных работ	Разделы 1 – 2	ПК-9
3	Итоговый: Экзамен	Разделы 1 – 2	ПК-1, ПК-9

Материалы для проведения текущего контроля знаний и промежуточной аттестации составляют отдельный документ – Фонд оценочных средств по дисциплине «Параллельное программирование»».

Учебно-методическое и информационное обеспечение дисциплины " Параллельное программирование "

а) основная литература:

Основная литература:

1. Энтони Уильямс. Параллельное программирование на C++ в действии. – СПб.: Питер, 2023.

2. Кулик А.С., Сысоев А.В. Параллельное программирование для многоядерных процессоров. – М.: Изд-во МГУ, 2018.

3. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах. – М.: ДМК Пресс, 2019.

Дополнительная литература:

1. Официальная документация OpenMP, MPI, CUDA.

2. Гергель В.П. Теория и практика параллельных вычислений. – М.: ИНТУИТ, 2007.

Программное обеспечение: Компиляторы C/C++ с поддержкой OpenMP (gcc, Intel, MSVC), библиотека MPI (OpenMPI, MPICH), NVIDIA CUDA Toolkit, среда разработки (Visual Studio, VS Code), система контроля версий Git.

б) Интернет-ресурсы:

№ п/п	Адрес сайта	Описание материала, содержащегося на сайте
1.	http://www.io-sotech.com/ru/	Практическое применение задач оптимизации
2.	http://matlab.exponenta.ru/optimiz/book_1/index.php	А.Г. Трифонов. "Optimization Toolbox 2.2 Руководство пользователя "
3.	http://matlab.exponenta.ru/optimiz/book_7/index.php	Статьи, материалы по практическим приложениям
4.	http://matlab.exponenta.ru/optimiz/book_2/index.php	А.Г. Трифонов. "Постановка задачи оптимизации и численные методы ее решения"
5.	http://matlab.exponenta.ru/optimiz/book_6/index.php	А.Г.Трифонов "Optimization Toolbox 3"
6.	http://matlab.exponenta.ru/optimiz/book_4/index.php	Список функций Optimization Toolbox

в) Программное обеспечение: Все лабораторные работы выполняются на персональных компьютерах с использованием специализированных математических пакетов (MATLAB)

г) Другое материально-техническое обеспечение: Реализация программы учебной дисциплины требует наличия учебной компьютерной лаборатории.

Оборудование компьютерной лаборатории: посадочные места по количеству обучающихся; рабочее место преподавателя; маркерная доска; учебно-методическое обеспечение.

Технические средства обучения: компьютеры по количеству обучающихся; локальная компьютерная сеть и глобальная сеть Интернет; лицензионное системное и прикладное программное обеспечение; лицензионное антивирусное программное обеспечение; лицензионное специализированное программное обеспечение; медиа-проектор.

Для лиц с ограниченными возможностями здоровья по ходатайству заведующего кафед-

рой на отдельные ПЭВМ может устанавливаться индивидуальный набор программного обеспечения.

**Сведения о переутверждении программы на очередной учебный год
и регистрации изменений**

Учебный год	Решение кафедры (№ протокола, дата)	Внесенные изменения	Подпись зав. кафедрой