

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Кафедра математики и анализа данных
Факультет информационных технологий и анализа больших данных**

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: IM7D4BNF9RPw4Uxqv2k+WTsBJScp/cj
Дата: 24.11.2025 г.

В.С. Чернышенко, И.Х. Утакаева

Основы численных методов

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике
и финансах»

Профиль:

«Прикладные информационные системы в экономике и финансах»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 03 от 16.12.2025 г.)*

Одобрено

*Кафедра математики и анализа данных
(протокол № 18 от 08.12.2025 г.)*

© Москва 2026

СОДЕРЖАНИЕ

1.	Наименование дисциплины	4
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	4
3.	Место дисциплины в структуре образовательной программы	6
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	6
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	7
5.1.	Содержание дисциплины	7
5.2.	Учебно-тематический план	9
5.3.	Содержание семинаров	10
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	12
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	12
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	14
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	18
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	22
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	22
10.	Методические указания для обучающихся по освоению дисциплины	24
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	26
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	27

1. Наименование дисциплины

«Основы численных методов».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКП-1	Способен разрабатывать и отлаживать программный код	1. Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода.	знать: Основные понятия оптимизации, типов оптимизационных задач, общих схем численных методов оптимизации, классификации задач по свойствам функций, методов спуска, теорем сходимости, оценок сложности и аффинной инвариантности. уметь: Формализовать оптимизационные задачи из экономики, финансов и анализа данных как математические модели; алгоритмизировать методы оптимизации: разбиение на шаги, выбор начального приближения, обновление итераций, реализация оракулов и критериев остановки; реализовывать численные методы оптимизации в программном коде на Python; анализировать сходимость и эффективность алгоритмов на примерах; применять регуляризацию и рестарты для улучшения алгоритмов в невыпуклых задачах.
		2. Оформляет программный код в соответствии с установленными требованиями.	знать: Стандарты оформления кода: PEP 8 для Python (правила именования, отступов, импортов, документации); лучших практик разработки: модульность кода, использование функций и классов, обработка ошибок; инструментов для тестирования и отладки: unit-тесты (pytest), профилирование

			(timeit, cProfile), визуализация результатов (matplotlib для графиков сходимости); принципов читаемости и поддерживаемости кода в контексте численных методов (например, разделение теоретической части и реализации). уметь: Документировать код: написание docstrings для функций, inline-комментарии; структурировать код: разделение на модули, использование классов для методов оптимизации; тестировать и отлаживать код: написание тестов для методов, анализ ошибок округления и устойчивости; представлять результаты.
--	--	--	---

3. Место дисциплины в структуре образовательной программы

Дисциплина «Основы численных методов» относится к «Модулю "Анализ данных"».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 7 (в часах)
Общая трудоёмкость дисциплины	3/108	108
Контактная работа-Аудиторные занятия	34	34
Лекции	16	16
Семинары, практические занятия	18	18
Самостоятельная работа	74	74
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Введение в оптимизацию.

Определение оптимизации как процесса поиска минимума/максимума функции. Исторический обзор: от аналитических методов к численным (ссылка на развитие с 1940-х годов, примеры из инженерных задач). Основные определения: целевая функция, переменные, ограничения; локальный и глобальный минимум/максимум; выпуклые множества и функции (определение, свойства); градиент, гессиан (матрица вторых производных); необходимые и достаточные условия оптимальности (первого и второго порядка). Основные типы оптимизационных задач: безусловная оптимизация; условная оптимизация (ограничения: равенства, неравенства); линейное программирование и нелинейное программирование; выпуклая и невыпуклая оптимизации (краткий обзор различий); стохастическая оптимизация (краткое упоминание). Понятие о численных методах оптимизации: отличие от аналитических методов; общая итеративная схема (начальное приближение, оракул, обновление информации, критерий остановки); оракулы (нулевого, первого, второго порядка); критерии остановки (по аргументу, по функции, по градиенту); сложность методов (аналитическая, арифметическая). Сравнение методов оптимизации: скорости сходимости (сублинейная, линейная, сверхлинейная, квадратичная с примерами графиков); классификация методов (одношаговые и многошаговые, методы спуска); теоремы сходимости. Классификация задач по свойствам функций: сильно выпуклые задачи (определение, свойства, примеры); выпуклые задачи (определение, свойства, примеры); невыпуклые задачи (сложности, примеры). Гладкие и негладкие задачи: гладкие (дифференцируемые функции, методы первого/второго порядка); негладкие (субградиенты, примеры); сравнение. Регуляризация и рестарты: регуляризация (ℓ_1 , ℓ_2 , примеры); рестарты (многостартовые методы, применение). Применение оптимизации в задачах loss-функций для определения объёма депонированного углерода ML-методами (минимизация потерь в моделях оценки углеродного секвестра, использование градиентных методов для обучения ML-моделей на данных Sentinel-2 или климатических факторах). Реализация понятий в Python с использованием библиотек `numpy` и `scipy` для моделирования задач оптимизации.

Тема 2. Методы спуска.

Два способа обоснования градиентного спуска: через направление убывания и через дискретизацию ОДУ. Анализ и мотивация подходов к выбору шага: постоянный шаг; априорно заданная последовательность; наискорейший спуск; условие существенного убывания. Требование достаточного убывания: правило Армихо (поиск шага с уменьшением); условие кривизны (правило Вольфа); требование существенного убывания (правило Гольдштейна). Теоремы сходимости градиентного спуска: базовая сходимость (градиент стремится к нулю); сходимость для выпуклых функций (сублинейная скорость); линейная сходимость для сильно выпуклых функций. Реализация градиентного спуска на Python: ручная реализация GD и Backtracking. Оценки сходимости для градиентного спуска. Плюсы и минусы градиентного спуска. Реализация методов в Python с `scipy.optimize` для сравнения.

Тема 3. Метод сопряжённых градиентов.

Связь между системой линейных уравнений и задачей безусловной минимизации. Метод сопряжённых направлений: основа для сопряжённых градиентов. Ортогонализация Грама-Шмидта. Реализация метода сопряжённых градиентов на Python. Теоремы сходимости: сходимость за число различных собственных значений; оценка сходимости. Оптимальные формулы для α_k (размер шага) и β_k ("память" о предыдущем направлении). Метод сопряжённых градиентов для неквадратичных функций: расширение на общие задачи оптимизации. Метод Флетчера-Ривса для нелинейной минимизации. Теоремы сходимости: общая сходимость метода Флетчера-Ривса. Методы Флетчера-Ривса; Полака-Рибьера; Хестенса-Штифеля. Перезапуск: ускорение и стабильность. Сравнение метода сопряжённых градиентов с градиентным спуском. Расчёт числа обусловленности матрицы A . Оценки сходимости для метода сопряжённых градиентов. Реализация в Python с использованием `numpy` для матричных операций и `scipy` для тестирования.

Тема 4. Продвинутое методы оптимизации.

Метод тяжёлого шарика: теорема сходимости. Ускоренный метод Нестерова. Адаптивный выбор константы Липшица. Метод Ньютона (квадратичная аппроксимация): теорема сходимости; аффинная инвариантность; использование метода сопряжённых градиентов в методе Ньютона. Плюсы и минусы метода Ньютона: сравнение с градиентным методом. Общая схема квазиньютоновских методов. Квазиньютоновское уравнение: сходимость. BFGS с ограниченной памятью. Реализация методов в Python с `scipy.optimize` для продвинутых задач, включая анализ устойчивости и ошибок.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Введение в оптимизацию.	14	6	4	2	8
2	Методы спуска.	30	10	4	6	20
3	Метод сопряжённых градиентов.	30	10	4	6	20
4	Продвинутые методы оптимизации.	34	8	4	4	26
	Итого	108	34	16	18	74

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Введение в оптимизацию.	Обсуждение основных определений: целевая функция, переменные, ограничения, локальный и глобальный оптимум. Анализ типов задач: безусловная и условная оптимизация, выпуклая vs. невыпуклая. Классификация задач по свойствам функций (сильно выпуклые, гладкие и негладкие). Общая итеративная схема численных методов, оракулы и критерии останова. Применение оптимизации в задачах loss-функций для определения объёма депонированного углерода ML-методами (минимизация потерь в моделях оценки углеродного секвестра, использование градиентных методов для обучения ML-моделей на данных Sentinel-2 или климатических факторах). Примеры реализации в Python (numpy, scipy).	Работа с текстом лекции, разбор вопросов по теме занятия. Изучение рекомендованных к занятию литературных источников. Участие в семинарах (обсуждение определений и типов задач), практическое занятие (реализация простых задач в Python).
Методы спуска.	Обоснование градиентного спуска (направление убывания, дискретизация ОДУ). Выбор шага: постоянный, наискорейший спуск, условия Армихо, Вольфа, Гольдштейна. Теоремы сходимости для выпуклых и сильно выпуклых функций. Плюсы и минусы метода. Ручная реализация GD и Backtracking в Python. Оценки сходимости.	Работа с текстом лекции, разбор вопросов по теме занятия. Изучение рекомендованных к занятию литературных источников. Участие в семинарах (обсуждение теорем сходимости и выбора шага). Практическое занятие (реализация и тестирование градиентного спуска в Python).
Метод сопряжённых градиентов.	Связь с системами линейных уравнений. Метод сопряжённых направлений, ортогонализация Грама-Шмидта. Формулы α_k и β_k . Расширение на неквадратичные функции (Флетчер-Ривз, Полак-Рибьер, Хестенс-Штифель). Теоремы сходимости, перезапуски. Сравнение с градиентным спуском. Расчёт числа обусловленности. Реализация в Python (numpy, scipy).	Работа с текстом лекции, разбор вопросов по теме занятия. Изучение рекомендованных к занятию литературных источников. Подготовка к семинарским и практическим занятиям. Выполнение домашних заданий.

Продвинутые методы оптимизации.	Метод тяжёлого шарика и ускоренный метод Нестерова (теоремы сходимости). Адаптивный выбор константы Липшица. Метод Ньютона: квадратичная аппроксимация, аффинная инвариантность, использование сопряжённых градиентов. Плюсы и минусы, сравнение с градиентным методом. Квазиньютоновские методы: уравнение, BFGS с ограниченной памятью. Реализация в Python (scipy.optimize).	Работа с текстом лекции, разбор вопросов по теме занятия. Изучение рекомендованных к занятию литературных источников. Подготовка к семинарским и практическим занятиям. Выполнение домашних заданий.
---------------------------------	---	--

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в оптимизацию.	Исторический обзор развития численных методов оптимизации с 1940-х годов (примеры из инженерных и экономических задач). Свойства выпуклых множеств и функций, необходимые и достаточные условия оптимальности. Классификация задач: стохастическая оптимизация, гладкие и негладкие задачи. Доказательства свойств выпуклых функций и множеств (включая теоремы о глобальной оптимальности). Анализ влияния стохастической оптимизации на задачи с шумом в данных (например, в машинном обучении). Роль оракулов высших порядков в оценке сложности алгоритмов. Применение регуляризации в задачах оценки углеродного секвестра с использованием loss-функций в ML-моделях (анализ ℓ_1 для разреженных моделей на данных Sentinel-2, сравнение с ℓ_2 для предотвращения переобучения в климатических прогнозах). Теоретические аспекты теорем сходимости для невыпуклых задач.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников; решение теоретических задач по классификации задач; самостоятельная реализация простых задач в Python (numpy, scipy) с анализом сходимости; подготовка отчета по примерам из ML (углеродный секвестр).
Методы спуска.	Мотивация выбора шага в градиентном спуске (априорно заданная последовательность, условие существенного убывания). Анализ плюсов и минусов градиентного спуска. Оценки сходимости для различных классов функций (выпуклые, сильно выпуклые). Применение условий Армихо, Вольфа и Гольдштейна в практике. Ручная реализация backtracking.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников; программирование градиентного спуска в Python с тестированием на примерах; анализ ошибок и подготовка сравнительного отчета.
Метод сопряжённых	Связь метода с системами линейных уравнений и	Работа с учебной и

градиентов.	ортогонализацией Грама-Шмидта. Формулы для α_k и β_k в нелинейных случаях. Методы Флетчера-Ривза, Полака-Рибьера, Хестенса-Штифеля. Оценки сходимости и расчет числа обусловленности матрицы. Перезапуски для стабильности. Сравнение с градиентным спуском. Предусловленность в методе сопряжённых градиентов для плохо обусловленных матриц (выбор прекондиционера). Применение метода в задачах больших данных (разреженных системах из анализа сетей). Доказательства сходимости для нелинейных вариаций (Флетчер-Ривз с перезапусками). Оценка влияния числа обусловленности на практическую эффективность (с примерами расчётов). Расширение на стохастические версии метода для ML-задач.	справочной литературой по рекомендованным источникам; решение матричных задач и теорем сходимости; самостоятельная реализация метода в Python (numpy) с визуализацией (matplotlib); подготовка презентации по приложениям в больших данных.
Продвинутые методы оптимизации.	Доказательства ускоренной сходимости метода Нестерова в сравнении с методом тяжёлого шарика (анализ констант). Вариации квазиньютоновских методов для ограниченной памяти (L-BFGS в высоких размерностях). Аффинная инвариантность метода Ньютона в контексте неевклидовых пространств. Сравнение устойчивости методов к шуму в данных (применение в регрессионных моделях). Теоретические пределы сходимости для смешанных выпукло-невыпуклых задач.	Работа с учебной и справочной литературой по рекомендованным источникам. Подготовка к решению задач по сходимости и аппроксимации; программирование методов в Python (scipy.optimize) с анализом устойчивости; подготовка отчета по практическим примерам.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

1. Реализуйте градиентный спуск с поиском шага по правилу Армихо для поиска глобального минимума функции Розенброка

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2, \text{ ТОЧНОСТЬ } \epsilon = 10^{-4},$$

максимум итераций 500. Постройте графики сходимости (норма градиента и значение функции по итерациям). Сравните поведение метода от разных стартовых точек. Почему функция Розенброка считается плохо обусловленной, и как это влияет на скорость сходимости градиентного спуска? Объясните с точки зрения числа обусловленности гессиана.

2. Реализуйте градиентный спуск с постоянным шагом для поиска глобального минимума функции Химмельблау:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2, \text{ ТОЧНОСТЬ } \epsilon = 10^{-5},$$

$$\alpha = 0.001$$

максимум итераций 1000. Вычисляйте градиент без JAX. Постройте графики сходимости. Исследуйте зависимость сходимости Вашего алгоритма от начальной точки. Какой критерий остановки (по аргументу, функции или градиенту) наиболее подходит для функции с несколькими локальными минимумами, и почему?

3. Реализуйте градиентный спуск с backtracking по Армихо для поиска глобального минимума 2D проекции функции Растригина:

$$f(x, y) = 20 + (x^2 - 10\cos(2\pi x)) + (y^2 - 10\cos(2\pi y)),$$

$$\text{ТОЧНОСТЬ } \epsilon = 10^{-3}, (\rho = 0.5, \beta = 0.1).$$

максимум итераций 300. Используйте JAX для градиента. Постройте график сходимости (норма градиента и разница значений функции по итерациям). Объясните, как правило Армихо обеспечивает достаточное уменьшение функции, и почему оно предпочтительнее фиксированного шага для функции Растригина.

4. Реализуйте градиентный спуск с уменьшающимся шагом

для поиска глобального минимума функции Экли:

$$f(x, y) = -20 \exp\left(-0.2 \sqrt{0.5(x^2 + y^2)}\right) - \exp\left(0.5\left(\cos(2\pi x) + \cos(2\pi y)\right)\right) + e + 20,$$
$$\text{ТОЧНОСТЬ } \varepsilon = 10^{-4}, \left(\alpha_k = 1/(k+1)\right).$$

Начальные точки: (0.1, 0.1), (1, -1) и (-1, 1), максимум итераций 400. Вычисляйте градиент без JAX. Сравните сходимость и укажите влияние стартовой точки на достижение глобального минимума. Почему уменьшающийся шаг

$$(\alpha_k = 1/(k+1))$$

может быть менее эффективным для плохо обусловленных функций по сравнению с поиском шага по Армихо?

5. Реализуйте градиентный спуск с поиском шага по Гольдштейну для поиска глобального минимума функции Бута:

$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2, \text{ ТОЧНОСТЬ } \epsilon = 10^{-5}, \left(c1 = 0.1, c2 = 0.9\right),$$

максимум итераций 200. Используйте JAX для градиента. Постройте графики: норма градиента и расстояние до известного минимума по итерациям. Как правило Гольдштейна отличается от правила Армихо, и в каких случаях оно может быть предпочтительнее?

6. Реализуйте градиентный спуск с backtracking по Вольфу для поиска глобального минимума 2D проекции функции Швевеля:

$$f(x, y) = 837.9658 - x \sin(\sqrt{|x|}) - y \sin(\sqrt{|y|}), \text{ ТОЧНОСТЬ } \epsilon = 10^{-4},$$
$$(c1 = 1e-4, c2 = 0.9).$$

максимум итераций 500. Вычисляйте градиент без JAX (учтите абсолютные значения). Какую роль играет оракул в градиентном спуске, и почему его локальная информация ограничивает способность метода избегать локальных минимумов?

7. Реализуйте градиентный спуск с постоянным шагом для поиска глобального минимума функции Гриванка:

$$f(x, y) = \frac{(x^2 + y^2)}{4000} - \cos(x) \cos\left(\frac{y}{\sqrt{2}}\right) + 1, \text{ ТОЧНОСТЬ } \epsilon = 10^{-3}, \left(\alpha = 0.01\right).$$

максимум итераций 600. Используйте JAX для градиента. Постройте график сходимости и проанализируйте зависимость от старта. Как постоянный шаг влияет на сходимость для функций с множеством локальных минимумов, и какие риски он несёт по сравнению с адаптивными методами?

8. Реализуйте градиентный спуск с уменьшающимся шагом

для поиска глобального минимума функции Михалевича:

$$f(x, y) = -\sin(x) \sin^{20}\left(\frac{x^2}{\pi}\right) - \sin(y) \sin^{20}\left(\frac{2y^2}{\pi}\right), \text{ ТОЧНОСТЬ } \epsilon = 10^{-4}, \\ \left(\alpha_k = 0.1 / (k + 1)\right)$$

максимум итераций 300. Вычисляйте градиент без JAX. Сравните, в какие минимумы сходится метод. Почему уменьшающийся шаг обеспечивает сходимость к стационарной точке, но может быть медленным для функций с высокой кривизной?

9. Реализуйте градиентный спуск с backtracking по Армихо для поиска глобального минимума функции Шуберта:

$$f(x, y) = \sum_{j=1}^5 j \cos((j+1)x + j) \left(\sum_{j=1}^5 j \cos((j+1)y + j) \right), \text{ ТОЧНОСТЬ } \epsilon = 10^{-5}, \\ (\rho = 0.8, \beta = 0.01)$$

максимум итераций 400. Используйте JAX для градиента. Постройте графики сходимости и укажите локальные минимумы. Как информационная модель в итеративной схеме градиентного спуска помогает адаптироваться к сложным функциям, таким как функция Шуберта?

10. Реализуйте градиентный спуск с поиском шага по Гольдштейну для поиска глобального минимума функции Леви:

$$f(x, y) = \sin^2(3\pi x) + (x - 1)^2 (1 + \sin^2(3\pi y)) + (y - 1)^2 (1 + \sin^2(2\pi y)), \text{ ТОЧНОСТЬ } \epsilon = 10^{-4}, \\ (c1 = 0.01, c2 = 0.99).$$

максимум итераций 500. Вычисляйте градиент без JAX. Проанализируйте влияние стартовой точки. Каковы преимущества и недостатки использования гессиана в оптимизации по сравнению с градиентным спуском для функций с множеством минимумов?

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ПКП-1 Способен разрабатывать и отлаживать программный код

1) 1. Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода.

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: Основные понятия оптимизации, типов оптимизационных задач, общих схем численных методов оптимизации, классификации задач по свойствам функций, методов спуска, теорем сходимости, оценок сложности и аффинной инвариантности.

Уметь: Формализовать оптимизационные задачи из экономики, финансов и анализа данных как математические модели; алгоритмизировать методы оптимизации: разбиение на шаги, выбор начального приближения, обновление итераций, реализация оракулов и критериев остановки; реализовывать численные методы оптимизации в программном коде на Python; анализировать сходимость и эффективность алгоритмов на примерах; применять регуляризацию и рестарты для улучшения алгоритмов в невыпуклых задачах.

Типовые контрольные задания

Задание 1. Реализуйте градиентный спуск с поиском шага по правилу Армихо для поиска глобального минимума функции Розенброка.

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2.$$

Точность максимум итераций 500. Постройте графики сходимости (норма градиента и значение функции по итерациям). Сравните поведение метода от разных стартовых точек.

$$\epsilon = 10^{-4},$$

Задание 2. Почему функция Розенброка считается плохо обусловленной, и как это влияет на скорость сходимости градиентного спуска? Объясните с точки зрения числа обусловленности гессиана.

Задание 3. Реализуйте градиентный спуск с backtracking по Армихо для поиска глобального минимума 2D проекции функции Растригина:

$$(\rho = 0.5, \beta = 0.1)$$

$$f(x, y) = 20 + (x^2 - 10\cos(2\pi x)) + (y^2 - 10\cos(2\pi y)).$$

Точность максимум итераций 300. Используйте JAX для градиента. Постройте семилогарифмический график сходимости (норма градиента и разница значений функции по итерациям).

$$\epsilon = 10^{-3},$$

Задание 4. Объясните, как правило Армихо обеспечивает достаточное уменьшение функции, и почему оно предпочтительнее фиксированного шага для функции Растригина.

2) 2. Оформляет программный код в соответствии с установленными требованиями.

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: Стандарты оформления кода: PEP 8 для Python (правила именования, отступов, импортов, документации); лучших практик разработки: модульность кода, использование функций и классов, обработка ошибок; инструментов для тестирования и отладки: unit-тесты (pytest), профилирование (timeit, cProfile), визуализация результатов (matplotlib для графиков сходимости); принципов читаемости и поддерживаемости кода в контексте численных методов (например, разделение теоретической части и реализации).

Уметь: Документировать код: написание docstrings для функций, inline-комментарии; структурировать код: разделение на модули, использование классов для методов оптимизации; тестировать и отлаживать код: написание тестов для методов, анализ ошибок округления и устойчивости; представлять результаты.

Типовые контрольные задания

Задание 1. Реализовать метод сопряжённых градиентов для минимизации квадратичной функции

$$f(x) = \frac{1}{2}x^T Ax - b^T x,$$

где A – симметричная положительно определённая матрица (используйте случайную матрицу размером 10×10). Точность максимум итераций 100. Постройте графики сходимости. Оформите код в соответствии с PEP 8, добавьте docstrings к функциям, структурируйте в модули, отдельные функции для расчёта напишите unit-тесты с `pytest` для проверки сходимости и визуализируйте результаты с `matplotlib`. Проанализируйте влияние числа обусловленности матрицы A на устойчивость.

$$\epsilon = 10^{-6},$$

$$\alpha_k \text{ И } \beta_k,$$

Задание 2.

Реализовать нелинейный метод сопряжённых градиентов по Флетчеру-Ривзу для поиска минимума функции Розенброка:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2.$$

Точность максимум итераций 500. Используйте JAX для градиента. Постройте график сходимости (норма градиента по итерациям). Документируйте код с `inline`-комментариями, используйте класс для метода оптимизации, протестируйте с `timeit` для профилирования и интегрировав визуализацию графиков итераций

$$\epsilon = 10^{-4},$$

Примерные вопросы для подготовки к зачету

[if !vml] , и объясните, как повторяющаяся ортогональность направлений предотвращает накопление ошибок округления в сравнении с прямым методом Гаусса. Приведите пример сценария, где потеря ортогональности из-за машинной арифметики замедляет сходимость, и предложите стратегию перезапуска. (20 баллов).

И

3. Реализуйте ускоренный метод Нестерова для минимизации выпуклой функции:

$$f(x) = \frac{1}{2}x^2 + \sin(x), \text{ где } x \in \mathbb{R} \text{ (одномерный случай для упрощения)}.$$

Начальное приближение:

$x_0 = 2$; параметр импульса $\gamma = 0.9$; постоянный шаг $\alpha = 0.1$;

точность $\epsilon = 10^{-5}$ (по норме градиента); максимум итераций – 500.

Используйте только базовые средства Python (циклы, 'math'). Вычислите градиент аналитически. Выведите финальное значение, и список норм градиентов по итерациям для сравнения с обычным градиентным спуском (реализуйте его параллельно в том же коде).

$$x^*, f(x^*)$$

Объясните механику "тяжёлого шарика" в ускоренном методе Нестерова и почему она приводит к скорости сходимости $O(1/k^2)$ для выпуклых функций, в отличие от $O(1/k)$ в стандартном градиентном спуске. Проанализируйте влияние параметра импульса γ на осцилляции траектории и предложите, как адаптировать его для недифференцируемых функций с использованием субградиентов. (20 баллов).

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Сухарев, Алексей Григорьевич Численные методы оптимизации : учебник и практикум для вузов / А. Г. Сухарев, А. В. Тимохов, В. В. Федоров. 3-е изд., испр. и доп Электрон. дан. Москва : Юрайт, 2025 367 с (Высшее образование) URL: <https://urait.ru/bcode/562179> (дата обращения: 01.09.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/562179> ISBN 978-5-534-17381-9 : 1449.00. [БИК ID: RU2fURAIT2f562179]

Дополнительная литература:

1. Седых, И. А. Методы оптимизации: теория, практические задания и лабораторные работы [Электронный ресурс] : учебное пособие для вузов / Седых И. А.,Истомин В. А.,Сёмина В. В.,Супрунов И. И. ; Истомин В. А., Сёмина В. В., Супрунов И. И. Санкт-Петербург : Лань, 2025 116 с. Книга из коллекции Лань - Математика <https://e.lanbook.com/book/489299> ISBN 978-5-507-52345-0. [БИК ID: RU-LAN-BOOK-489299]

2. Каныгин, Георгий Иванович Численные методы безусловной оптимизации : Учебное пособие / Донской государственный технический университет ; Донской государственный технический университет Вологда : Инфра-Инженерия, 2024 148 с. ВО - Бакалавриат <https://znanium.ru/catalog/document?id=452428> ISBN 978-5-9729-1761-7. [БИК ID: RU\infra-m\znanium\bibl\2170893]

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>
2. •Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>
3. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
4. <http://www.biblioclub.ru> - Университетская библиотека online
5. •Электронно-библиотечная система Znanium <http://www.znanium.ru/>
6. Образовательная платформа "ЮРАЙТ" <https://urait.ru/>

7. •Электронно-библиотечная система издательства Проспект
<http://ebs.prospekt.org/books>
8. •Электронно-библиотечная система издательства Лань <https://e.lanbook.com/>
9. •Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
10. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
11. •Национальная электронная библиотека <http://нэб.рф/>
12. •Электронные коллекции книг и журналов издательства Springer:
<http://link.springer.com/>

10. Методические указания для обучающихся по освоению дисциплины

Цель методических рекомендаций – обеспечить студенту бакалавриата (далее – студенту) оптимальную организацию процесса изучения дисциплины, а также выполнения различных форм самостоятельной работы.

Рекомендации по подготовке к лекционным занятиям:

1. При подготовке к лекции целесообразно предварительно ознакомиться с содержанием по рекомендованным источникам, выделить наиболее трудные вопросы (например, теоремы сходимости или свойства выпуклых функций).
2. Во время лекций следует конспектировать ключевые определения, формулы и алгоритмы (например, итерационную схему градиентного спуска или обновление в методе сопряжённых градиентов).
3. После занятия следует проработать конспект: отредактировать записи, оформить формулы в LaTeX (для последующего использования в отчетах). При оформлении целесообразно выделять специальным образом названия тем, формулировки теорем и примеры реализации.
4. Сделанные записи необходимо сопоставить с учебниками и учебными пособиями; в случае расхождений проконсультироваться с преподавателем. Перед очередной лекцией просмотреть материал предыдущей лекции по конспекту. При затруднениях в восприятии материала обратиться к основным литературным источникам; если разобраться не удалось, то к лектору (по графику консультаций) или преподавателю на практических занятиях. Не оставлять «белых пятен» в освоении материала, особенно в разделах по оценкам сложности и скорости сходимости.

Рекомендации по подготовке к практическим (семинарским) занятиям:

Студентам следует:

1. приносить с собой рекомендованную преподавателем литературу к конкретному занятию;
2. до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал соответствующей темы;
3. при подготовке к практическим занятиям использовать не только лекции, но и другую учебную литературу, включая онлайн-ресурсы по Python (документация NumPy и SciPy);

4. в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в понимании и освоении, при решении задач, заданных для самостоятельного решения;

5. в ходе семинара давать конкретные, четкие ответы по существу вопросов;

6. на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю; уделять отладке кода: проверять сходимость на простых примерах перед запуском на сложных функциях..

Методические рекомендации по выполнению различных форм самостоятельных домашних заданий:

Самостоятельная работа студентов включает выполнение различного рода заданий, ориентированных на более глубокое усвоение материала дисциплины (например, доказательства теорем сходимости или сравнение методов на тестовых функциях). К выполнению заданий предъявляются следующие требования: задания должны выполняться самостоятельно и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению (РЕР 8 для кода, отчеты в Jupyter Notebook).

Студентам следует:

1. руководствоваться графиком самостоятельной работы, определенным РПД;
2. выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, и разбирать на семинарах и консультациях неясные вопросы;
3. использовать при подготовке нормативные документы Финансового университета;
4. при подготовке к экзамену параллельно прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные моменты для их обсуждения на плановой консультации.

Методические рекомендации по работе с литературой:

Любая форма самостоятельной работы студента (подготовка к семинарскому занятию, выполнение расчетно-аналитической работы начинается с изучения соответствующей литературы как в библиотеке, так и дома.

К каждой теме учебной дисциплины подобрана основная и дополнительная литература.

При работе с литературой рекомендуется делать записи. Записи в той или иной форме не только способствуют пониманию и усвоению изучаемого материала, но и помогают

вырабатывать навыки явного изложения в письменной форме тех или иных теоретических вопросов.

Методические указания по проведению практических занятий:

По структуре практические занятия следует разделить на учебные и контрольные.

Учебные практические занятия структурно состоят из следующих компонент:

1. проверка наличия выполненного задания самостоятельной работы каждого студента;
2. выборочная проверка корректности выполнения домашнего задания;
3. разбор типичных ошибок, возникших в самостоятельной работе;
4. рассмотрение теоретических вопросов, связанных с текущим практическим занятием;
5. разбор методов выполнения практических заданий и решения задач;
6. корректировка заданий для самостоятельной работы студентов.

Контрольные практические занятия структурно состоят из следующих компонент:

1. проведение аудиторных самостоятельных работ;
2. подведение итогов и разбор типичных ошибок, возникших при выполнении самостоятельных работ.

Студенты должны обратить внимание на перечень основных контрольных мероприятий, которые проводятся в соответствии с рабочей программой дисциплины.

Конкретные сроки проведения этих мероприятий своевременно доводятся до сведения студентов.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Jupyter Notebook (Windows 10)
2. Anaconda
3. Kaspersky

4. Пакет офисных программ

Современные профессиональные базы данных и информационные справочные системы:

1. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>
2. Язык программирования Python 3. <https://pythonworld.ru/>

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)