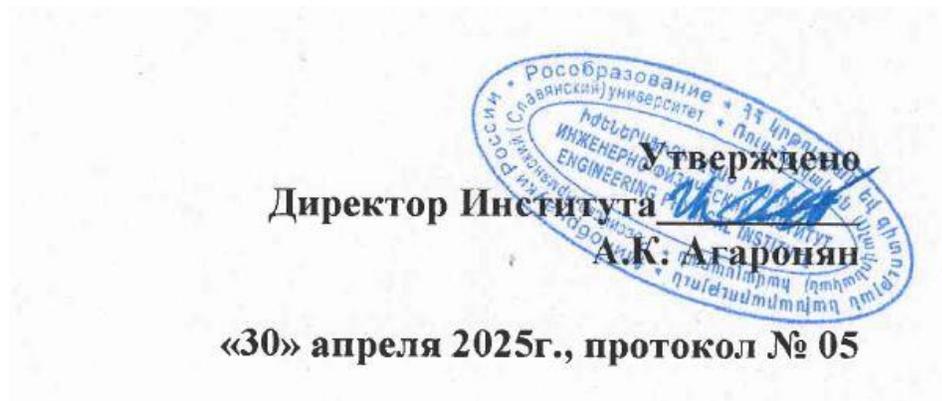


**ГОО ВПО Российско-Армянский (Славянский)  
университет**



**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ**

**Наименование дисциплины: Б1.В.08 Вычислительная физика**

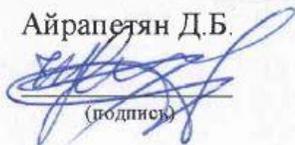
**Автор (ы) к.ф.-м.н., старший преподаватель Юрий Юрьевич Блеян**  
*Ф.И.О, ученое звание (при наличии), ученая степень (при наличии)*

**Направление подготовки: 11.03.03 Конструирование и технология  
электронных средств**

**Согласовано:**

Заведующий Кафедрой общей физики и квантовых наноструктур

Айрапетян Д.Б.

A handwritten signature in blue ink, consisting of stylized, overlapping loops and lines, positioned over a horizontal line.

(подпись)

# 1. АННОТАЦИЯ

## 1.1. Краткое описание содержания данной дисциплины;

Этот университетский курс посвящён основам вычислительной физики с акцентом на практическое применение языка программирования Python для решения физических задач. Курс разработан для предоставления студентам необходимых вычислительных инструментов для анализа, моделирования и визуализации физических систем в различных областях, включая механику, электродинамику и статистическую физику.

В рамках практических занятий студенты научатся:

- Реализовывать численные методы для решения дифференциальных уравнений и задач оптимизации.
- Моделировать физические системы и явления с использованием библиотек Python.
- Визуализировать данные и результаты моделирования с помощью мощных инструментов визуализации Python, таких как Matplotlib и Seaborn, создавая наглядные графики, диаграммы и анимации.
- Анализировать экспериментальные данные и выполнять статистические вычисления.

Курс не требует предварительного опыта программирования, что делает его доступным для всех студентов-физиков. По завершении курса студенты получают прочную основу в вычислительных методах, программировании на Python и визуализации данных, что позволит эффективно решать сложные физические задачи и ясно представлять полученные результаты.

## 1.2. Трудоемкость в академических кредитах и часах, формы итогового контроля (экзамен/зачет);

3 академических кредита / 108 часов. Форма итогового контроля — экзамен.

## 1.3. Взаимосвязь дисциплины с другими дисциплинами учебного плана специальности (направления)

Механика, Электромагнетизм, Волновые процессы, Квантовая физика, Физика макросистем,

## 1.4. Результаты освоения программы дисциплины:

<b>Код компетенции</b> (в соответствии рабочим с	<b>Наименование компетенции</b> (в соответствии рабочим с учебным планом)	<b>Код индикатора достижения компетенций</b>	<b>Наименование индикатора достижений компетенций</b> (в соответствии рабочим с учебным планом)
--	---	--	---

<i>учебным планом)</i>		<i>(в соответствии рабочим с учебным планом)</i>	
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1	Знает методики поиска, сбора и обработки информации, метод системного анализа
		УК-1.2	Умеет применять методики поиска, сбора и обработки информации, осуществлять критический анализ и синтез информации, полученной из разных источников, применять системный подход для решения поставленных задач.
		УК-1.3	Владеет методами поиска, сбора и обработки, критического анализа и синтеза информации, методикой системного подхода для решения поставленных задач.

## 2. УЧЕБНАЯ ПРОГРАММА

### 2.1. Цели и задачи дисциплины

Цель данного курса — предоставить студентам вычислительные навыки и практические знания, необходимые для решения сложных физических задач с использованием современных инструментов программирования. Интегрируя программирование на Python с базовыми и продвинутыми концепциями физики, курс направлен на:

- Развитие способности студентов анализировать, моделировать и симулировать физические системы с использованием численных методов.
- Углубление понимания вычислительных подходов в различных областях физики, включая механику, электродинамику, квантовую механику и статистическую физику.
- Ознакомление с эффективными методами визуализации данных и результатов моделирования, позволяющими интерпретировать и ясно представлять физические явления с помощью графиков, диаграмм и анимаций.

- Предоставление практического опыта работы с библиотеками Python, такими как NumPy, SciPy, Matplotlib и другими, для выполнения численных вычислений, оптимизации решений и анализа данных.
- Устранение разрыва между теоретическими знаниями и их практическим применением, стимулируя студентов к преобразованию абстрактных физических задач в вычислительные алгоритмы и их эффективной реализации.
- Улучшение навыков решения задач за счёт вовлечения студентов в сценарии, близкие к реальным, такие как моделирование динамических систем, решение дифференциальных уравнений, оптимизация параметров экспериментов и анализ больших массивов данных.
- Подготовку студентов к вычислительным требованиям современной физической науки, к работе в индустрии и к междисциплинарным приложениям, где численные симуляции и программирование являются неотъемлемой частью.

Курс направлен на формирование прочной базы вычислительного мышления, позволяя студентам подходить к решению физических задач как с теоретической, так и с вычислительной стороны. К концу курса студенты не только овладеют программированием на Python, но и приобретут уверенность в применении этих навыков для решения сложных физических задач и внесения вклада в передовые разработки в науке и технологиях.

**2.2. Трудоемкость дисциплины и виды учебной работы (в академических часах и зачетных единицах) (удалить строки, которые не будут применены в рамках дисциплины)**

Виды учебной работы	Всего, в акад. часах	Распределение по семестрам
		1 сем
1	2	3
<b>1. Общая трудоемкость изучения дисциплины по семестрам, в т. ч.:</b>	<b>108</b>	<b>108</b>
1.1. Аудиторные занятия, в т. ч.:	<b>64</b>	<b>64</b>
1.1.1. Лекции	<b>32</b>	<b>32</b>
1.1.2. Практические занятия, в т. ч.	<b>32</b>	<b>32</b>
1.1.3. Лабораторные работы		
1.2. Самостоятельная работа, в т. ч.:	<b>17</b>	<b>17</b>
1.3. Консультации		

Итоговый контроль (Экзамен, Зачет, диф. зачет - указать)	Экзамен 27	Экзамен 27
--	---------------	---------------

## 2.3. Содержание дисциплины

### 2.3.1. Тематический план и трудоемкость аудиторных занятий (модули, разделы дисциплины и виды занятий) по рабочему учебному плану

Разделы и темы дисциплины	Всего (ак. часов)	Лекции (ак. часов)	Практ. Занятия (ак. часов)
<b>1</b>	<b>2=3+4+5+6 +7</b>	<b>3</b>	<b>4</b>
Тема 1 Введение в программирование на Python	8	4	4
Тема 2 Численные методы в физике	12	6	6
Тема 3 Анализ и визуализация данных	8	4	4
Тема 4 Механика и моделирование	12	6	6
Тема 5 Электродинамика и визуализация полей	12	6	6
Тема 6 Квантовая механика и волновые функции	12	6	6
<b>ИТОГО</b>	<b>64</b>	<b>32</b>	<b>32</b>

### 2.3.2. Краткое содержание разделов дисциплины в виде тематического плана

#### Введение в программирование на Python

В этом разделе рассматриваются основы программирования на Python, включая установку, работу с IDE и понимание синтаксиса языка. Студенты изучат базовые концепции программирования, такие как переменные, типы данных, циклы и условные операторы, что создаст основу для изучения более сложных тем.

#### Численные методы в физике

Студенты познакомятся с численными методами, широко используемыми в физике, включая решение уравнений, численное интегрирование и дифференцирование. Практические примеры будут сосредоточены на моделировании физических систем, таких как движение под действием силы или вычисление энергии в потенциальных полях.

## **Анализ и визуализация данных**

Этот раздел обучает работе с экспериментальными и симуляционными данными. С использованием библиотек Python, таких как NumPy, Pandas и Matplotlib, студенты научатся визуализировать результаты с помощью графиков, диаграмм и анимаций, что упростит интерпретацию данных.

## **Механика и моделирование**

В этом разделе рассматриваются задачи классической механики, такие как движение снарядов, динамика маятника и гармонические осцилляторы. Студенты будут реализовывать численные решения для симуляции и визуализации этих систем.

## **Электродинамика и визуализация полей**

Этот раздел посвящён вычислительным методам анализа и визуализации электрических и магнитных полей. Темы включают решение уравнения Пуассона, моделирование распределений зарядов и симуляцию распространения электромагнитных волн.

## **Квантовая механика и волновые функции**

Студенты изучат основы квантовой механики с использованием вычислительных методов. Темы включают решение уравнения Шрёдингера для простых систем (например, потенциальные ямы) и визуализацию волновых функций и плотностей вероятности.

### **2.3.3. Краткое содержание практических занятий**

#### **Роль вычислительной физики в современной науке**

- Обсуждение того, как вычислительные инструменты меняют исследования в физике и приложения в промышленности.

#### **Сравнение численных методов**

- Анализ сильных и слабых сторон различных численных подходов в решении физических задач.

#### **Физика за пределами аналитических решений**

- Изучение сложных систем, где аналитические решения невозможны и требуются вычислительные подходы.

#### **Анализ данных в экспериментальной физике**

- Кейсы по анализу экспериментальных данных с использованием вычислительных методов.

#### **Визуализация в физическом исследовании**

- Анализ важности чётких и точных визуальных представлений в физике.

#### **Применение вычислительной физики в междисциплинарных областях**

- Обсуждение использования вычислительной физики в таких областях, как материаловедение, биофизика и астрофизика.

#### **Этика и надёжность в вычислительных исследованиях**

- Обсуждение этических вопросов и возможных подводных камней при использовании симуляций и вычислительных моделей в физике.

### **Презентации студентов**

- Студенты представляют свои проекты или обсуждают применение вычислительной физики к выбранной задаче.

### **Основы Python и первые программы**

- Написание простых скриптов для выполнения базовых вычислений.
- Понимание ввода/вывода и управляющих структур.

### **Численное интегрирование и дифференцирование**

- Реализация методов, таких как правило трапеций, правило Симпсона и конечные разности.

### **Решение обыкновенных дифференциальных уравнений (ОДУ)**

- Использование численных методов (например, метод Эйлера и Рунге-Кутты) для решения уравнений движения.

### **Визуализация данных**

- Создание графиков, гистограмм и 3D-визуализаций с использованием Matplotlib и Seaborn.

### **Моделирование классических систем**

- Моделирование маятника, движения снарядов и гармонических осцилляторов.

### **Вычисления электромагнитных полей**

- Визуализация электрических и магнитных полей из распределений зарядов и токов

Это разделение обеспечивает баланс между практическими навыками и теоретическим пониманием, давая студентам всесторонний опыт обучения.

## **2.3.4. Материально-техническое обеспечение дисциплины**

Для успешного освоения дисциплины "Вычислительная физика" необходимо соответствующее материально-техническое обеспечение. Это включает в себя следующие компоненты:

### **1. Компьютерные классы:**

- Оснащенные современными персональными компьютерами или ноутбуками с установленной операционной системой (Windows или Linux).
- Доступ к сети интернет для скачивания необходимых библиотек и инструментов.
- Среда разработки (IDE) для программирования на Python, такие как PyCharm, Visual Studio Code или Jupyter Notebook.

### **2. Программное обеспечение:**

- Установленный интерпретатор Python последней версии (рекомендуется Python 3.x).
- Библиотеки для численных вычислений и символьных расчётов (NumPy, SymPy, SciPy).
- Пакеты для визуализации данных (Matplotlib).
- Менеджер пакетов pip для установки и управления библиотеками.

### 3. Учебные материалы:

- Учебные пособия и методические указания по каждой теме курса.
- Доступ к онлайн-ресурсам и документации по Python и его библиотекам.
- Практические задания и проекты для самостоятельной работы.

### 4. Дополнительные ресурсы:

- Электронная библиотека

Обеспечение всеми вышеперечисленными компонентами является необходимым для полного и эффективного усвоения материала дисциплины "Структуры данных и алгоритмы (Python)".

## 2.4. Модульная структура дисциплины с распределением весов по формам контролей

Формы контролей	Вес формы (форм) текущего контроля в результирующей оценке текущего контроля (по модулям)		Вес формы промежуточного контроля в итоговой оценке промежуточного контроля		Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей		Вес итоговой оценки промежуточного контроля в результирующей оценке промежуточных контролей (семестровой оценке)	Вес результирующей оценки промежуточных контролей и оценки итогового контроля в результирующей оценке итогового контроля
	M1 <sup>1</sup>	M2	M1	M2	M1	M2		
<b>Вид учебной работы/контроля</b>	M1 <sup>1</sup>	M2	M1	M2	M1	M2		
Контрольная работа <i>(при наличии)</i>			0.5	0.5				
Устный опрос <i>(при наличии)</i>								
Лабораторные работы <i>(при наличии)</i>	0.5	0.5						
Письменные домашние задания <i>(при наличии)</i>								
<i>Решение задач</i>	0.5	0.5						
Вес результирующих оценок текущих контролей в итоговых оценках промежуточных контролей					0.5	0.5		
Вес оценок промежуточных контролей в итоговых оценках промежуточных контролей								
Вес итоговой оценки 1-го промежуточного контроля в результирующей оценке промежуточных контролей							0.5	
Вес итоговой оценки 2-го промежуточного контроля в результирующей оценке промежуточных контролей							0.5	
Вес результирующей оценки промежуточных контролей в результирующей оценке итогового контроля								0.5

<sup>1</sup> Учебный Модуль

Вес итогового контроля (Экзамен/зачет) в результирующей оценке итогового контроля								0.5
	$\Sigma = 1$							

### 3. Теоретический блок *(указываются материалы, необходимые для освоения учебной программы дисциплины)*

#### 3.1. Материалы по теоретической части курса

##### 3.1.1. Учебник(и);

1. "Python для ученых и инженеров" Мэтта Джордано
2. "Руководство студента по Python для физического моделирования" Джесси М. Киндера и Филиппа Нельсона

##### 3.1.2. Учебное(ые) пособие(я);

Python Documentation ([docs.python.org](https://docs.python.org))

### 4. Фонды оценочных средств *(указываются материалы, необходимые для проверки уровня знаний в соответствии с содержанием учебной программы дисциплины).*

#### 4.1. Планы практических занятий

##### 1. Введение в программирование на Python

- Установка Python и настройка IDE.
- Написание простых программ с базовым синтаксисом и управлением.

##### 2. Численное интегрирование и дифференцирование

- Реализация метода трапеций и численного дифференцирования.

##### 3. Решение обыкновенных дифференциальных уравнений (ОДУ)

- Решение ОДУ с использованием методов Эйлера и Рунге-Кутты.

##### 4. Визуализация данных

- Построение графиков и гистограмм с использованием Matplotlib.

##### 5. Моделирование классических систем

- Моделирование маятника, движения снарядов, осцилляторов.

##### 6. Вычисления электромагнитных полей

- Визуализация электрических и магнитных полей.

##### 7. Квантовые симуляции

- Решение уравнения Шрёдингера для потенциальной ямы.

##### 8. Роль вычислительной физики в современной науке

- Обсуждение влияния вычислений на научные исследования.

##### 9. Сравнение численных методов

- Обсуждение различных численных методов и их точности.

##### 10. Физика за пределами аналитических решений

- Обсуждение задач, решаемых только с помощью вычислений.

##### 11. Анализ данных в экспериментальной физике

- Методы подгонки данных и анализа ошибок.

##### 12. Визуализация в физическом исследовании

- Обсуждение методов визуализации данных в физике.
- 13. Применение вычислительной физики в междисциплинарных областях**
- Обсуждение применения в биофизике, материаловедении и других областях.
- 14. Этика и надёжность в вычислительных исследованиях**
- Обсуждение этических вопросов и надёжности вычислительных моделей.
- 15. Презентации студентов**
- Презентация проектов студентами.

## 4.2. Материалы по практической части курса

### 4.3.1. Учебно-методические пособия

- Марк Лутц — «Изучаем Python»
- Эрик Маттес — «Изучаем программирование на Python»

### 4.3.2. Учебные справочники

- Алекс Мартелли, Дэвид Ашер — «Python. Карманный справочник»
- Марк Саммерфилд — «Python 3: Справочник и руководство»

### 4.3.3. Задачники (практикумы)

- Дэн Бейдер — «Python. Тонкости программирования»
- Дэвид Коупер — «200 задач по Python: от основ к профессиональному уровню»

## 4.3. Вопросы и задания для самостоятельной работы студентов

**Интерактивные учебные материалы** на платформе Codecademy, Coursera и т.д.

## 4.4. Тематика рефератов, эссе и других форм самостоятельных работ

- Создание численной модели физического явления
- Анализ экспериментальных данных и их визуализация
- Исследование применения численных методов в квантовой механике
- Моделирование сложной физической системы

## 4.5. Образцы вариантов контрольных работ, тестов и/или других форм текущих и промежуточных контролей

- Тесты на знание теоретических основ вычислительной физики.
- Контрольные работы с задачами.
- Задания на моделирование сложной физической системы

## 4.6. Перечень экзаменационных вопросов

- Основные конструкции языка Python: переменные, циклы, функции. Как они используются для создания моделей физических процессов?
- Как с помощью библиотеки Matplotlib строить графики для анализа физических явлений? Приведите пример визуализации результатов моделирования.
- Опишите этапы построения численной модели физической системы (например, движения маятника или распространения тепла).

- Как вычислительные методы помогают решать задачи, которые невозможно решить аналитически? Приведите примеры из различных областей физики.

### **Основы Python**

- Объясните основные элементы синтаксиса Python, такие как переменные, операторы ввода/вывода и операторы ветвления. Приведите пример их использования в программировании для решения физических задач.

### **Численные методы**

- Опишите, как численные методы могут быть использованы для решения физических задач. Приведите примеры использования численного интегрирования и дифференцирования в механике.

### **Решение обыкновенных дифференциальных уравнений**

- Объясните, как можно решить обыкновенное дифференциальное уравнение с помощью метода Эйлера и метода Рунге-Кутты. Приведите пример из физики, в котором эти методы могут быть полезны.

### **Визуализация данных в Python**

- Как можно использовать библиотеки Python, такие как Matplotlib и Seaborn, для визуализации данных и графического представления физических процессов? Приведите примеры типов графиков, которые могут быть полезны для анализа данных.

### **Моделирование физических систем**

- Опишите, как можно смоделировать поведение механической системы, например, маятника или пружинного осциллятора, с использованием Python. Какие численные методы применяются для решения подобных задач?

### **Применение Python в квантовой механике**

- Как можно использовать Python для решения задач квантовой механики, например, для нахождения уровней энергии в потенциальной яме? Объясните, какие численные методы могут быть использованы в таких задачах.

### **Численные методы в электродинамике**

- Как численные методы могут быть использованы для расчета электромагнитных полей в различных ситуациях, например, в случае точечного заряда или тока в проводнике? Приведите примеры.

### **Работа с экспериментальными данными**

- Как в Python можно провести анализ экспериментальных данных, используя методы аппроксимации и подгонки? Опишите, как оценить точность подгонки данных и какие инструменты для этого предоставляет Python.

### **Методы оптимизации в вычислительной физике**

- Какие методы оптимизации используются в вычислительной физике для нахождения минимальных и максимальных значений в сложных физических системах? Приведите примеры использования этих методов.

### **Этика в вычислительных исследованиях**

- Какие этические проблемы могут возникнуть при использовании вычислительных методов в физике? Как обеспечить надежность и воспроизводимость результатов вычислительных экспериментов?

## **4.7. Образцы экзаменационных билетов**

**ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ**  
**Кафедра общей физики и квантовых наноструктур**  
**Направление: Конструирование и технология электронных средств**  
**Дисциплина: Вычислительная физика**  
**(бакалавриат I-ый курс, II-ой семестр)**

Экзаменационный билет № \*\*

1. Обработка исключений. Инструкция try... except... else... finally.
2. Решение задачи на языке программирования Python
3. Решение задачи на языке программирования Python

Зав. кафедрой ОФКН \_\_\_\_\_ Д.Б. Айрапетян  
20\_\_ г.

---

---

**4.8. Образцы экзаменационных практических заданий**

1. Обработка исключений. Инструкция try... except... else... finally.
2. Решение задачи на языке программирования Python
3. Решение задачи на языке программирования Python

**4.9. Банк тестовых заданий для самоконтроля**

Множественный выбор, правда/ложь и задачи на написание кода по темам курса.

**4.10. Методики решения и ответы к образцам тестовых заданий**

Подробные решения и объяснения к тестам и практическим заданиям.

**5. Методический блок (МОЖНО ИСКЛЮЧИТЬ)**

**5.1. Методика преподавания**

5.1.1. Методические рекомендации для студентов по подготовке к семинарским, практическим или лабораторным занятиям, по организации самостоятельной работы студентов при изучении конкретной дисциплины.

**Подготовка к семинарским занятиям**

1. **Изучение теоретического материала:** Перед каждым семинаром внимательно изучите соответствующий раздел учебника и дополнительные материалы, предоставленные преподавателем. Обратите внимание на основные концепции, определения и примеры.

2. **Просмотр видеолекций:** Если доступны видеолекции по теме семинара, обязательно посмотрите их. Видеолекции помогут лучше усвоить материал и подготовиться к обсуждению.
3. **Подготовка вопросов:** Сформулируйте вопросы по теме, которые остались непонятными или требуют дополнительного пояснения. Это позволит вам активнее участвовать в обсуждениях и получать разъяснения от преподавателя.

#### **Подготовка к практическим занятиям**

1. **Программирование на Python:** Убедитесь, что знакомы с основами языка Python. Выполните несколько простых упражнений, чтобы освежить знания.
2. **Решение задач:** Попробуйте решить несколько задач по теме предстоящего занятия самостоятельно. Это могут быть задачи из учебника или дополнительных источников.
3. **Анализ примеров кода:** Ознакомьтесь с примерами кода, которые будут обсуждаться на занятии. Попробуйте понять их работу и, если возможно, выполните их на своем компьютере.
4. **Подготовка к обсуждению:** Обсудите свои решения и подходы к задачам с преподавателем и одногруппниками. Запишите вопросы или проблемы, с которыми вы столкнулись при выполнении упражнений.

#### **Организация самостоятельной работы**

1. **Планирование времени:** Составьте расписание для самостоятельной работы. Определите время для чтения теоретического материала, выполнения практических заданий и повторения изученного материала.
2. **Использование дополнительных ресурсов:** Воспользуйтесь дополнительными ресурсами, такими как онлайн-курсы, видеолекции, статьи и блоги, чтобы углубить свои знания по теме.
3. **Регулярная практика:** Регулярно выполняйте упражнения и задачи, чтобы закрепить изученный материал. Постепенно увеличивайте сложность задач.
4. **Обратная связь:** Обсуждайте свои решения с одногруппниками и преподавателем. Не стесняйтесь задавать вопросы и просить разъяснений.
5. **Работа в группе:** Сотрудничайте с другими студентами. Обсуждайте решения задач, делитесь опытом и помогайте друг другу.