

Государственное бюджетное учреждение дополнительного
профессионального образования Республики Мордовия «Центр
непрерывного повышения профессионального мастерства педагогических
работников – «Педагог 13.ру»
Республика Мордовия г. Саранск ул. Транспортная, 19

Связь математики и информатики с физикой

Коняшкина Любовь Ивановна,
преподаватель
ГБПОУ РМ «Зубово-Полянский
педагогический колледж»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 СВЯЗЬ МАТЕМАТИКИ И ИНФОРМАТИКИ С ФИЗИКОЙ.....	4
2 ПРОФЕССИОНАЛЬНАЯ НАПРАВЛЕННОСТЬ СОДЕРЖАНИЯ ОБУЧЕНИЯ.....	6
3 ИСПОЛЬЗОВАНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ И МЕТОДИК ПРЕПОДАВАНИЯ.....	9
4 КОНКРЕТНЫЙ ПРАКТИЧЕСКИЙ ПРИМЕР.....	12
5 ФОРМИРОВАНИЕ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ БУДУЩИХ СПЕЦИАЛИСТОВ.....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17

Введение

Современная IT-индустрия, особенно в сферах разработки игр, компьютерной графики, виртуальной реальности и научных симуляций, предъявляет высокие требования к программистам. Для создания реалистичных и сложных программных продуктов необходимы не только навыки кодирования, но и глубокое понимание физических законов, которые описывают окружающий мир, а также математический аппарат для их моделирования. В этом контексте интеграция знаний из математики, информатики и физики становится не просто полезной, а необходимой для подготовки высококвалифицированных IT-специалистов.

Проблема заключается в том, что у студентов IT-направлений зачастую отсутствует понимание практической ценности курсов физики и высшей математики. Эти дисциплины воспринимаются как абстрактные и не связанные с их будущей профессией, что приводит к формальному усвоению материала и невозможности применять эти знания в реальных проектах.

Целью данного проекта является разработка методики реализации междисциплинарных связей математики, информатики и физики через практико-ориентированные задания и проекты для повышения качества подготовки будущих программистов.

1 Связь математики и информатики с физикой

Связь между этими тремя дисциплинами является фундаментальной и взаимодополняющей.

Физика предоставляет содержательную основу, ставит реальные задачи, которые требуют решения: расчет траекторий, моделирование физических полей, описание динамики твердых и мягких тел, распространение волн и многое другое. Без понимания физики программист не может создать правдоподобную симуляцию или реалистичную графику.

Математика выступает в роли универсального языка для формализации физических задач. Ключевую роль играют:

Математический анализ (дифференциальные и интегральные исчисления) для описания непрерывных изменений, таких как скорость, ускорение и потоки.

Линейная алгебра (векторы, матрицы, тензоры) для работы с пространственными преобразованиями, компьютерной графикой и квантовой механикой.

Дифференциальные уравнения для моделирования динамических систем, от колебаний маятника до уравнений Максвелла для электромагнитных полей.

Численные методы для нахождения приближенных решений сложных уравнений, не имеющих аналитического решения.

Информатика является инструментом практической реализации и вычислительным "двигателем". Она предоставляет не просто инструменты, а целые экосистемы для решения задач:

Языки программирования: Выбор языка зависит от задачи. Python с библиотеками (NumPy, SciPy) идеален для прототипирования и научных вычислений, C++ — для высокопроизводительных симуляций и игровых движков, а Julia — для современных научных расчетов.

Алгоритмы и структуры данных: Это основа эффективности. Быстрые алгоритмы сортировки и поиска, древовидные структуры данных, хеш-таблицы и графы позволяют управлять сложными данными и минимизировать время вычислений.

Специализированные библиотеки и фреймворки: Готовые решения для конкретных задач:

Для моделирования: Nvidia PhysX, Bullet Physics, ANSYS.

Для научных вычислений: NumPy, SciPy, MATLAB.

Для визуализации: Matplotlib, VTK, ParaView, а также игровые движки (Unity, Unreal Engine) для интерактивного представления результатов.

Высокопроизводительные вычисления (HPC): Информатика позволяет решать грандиозные задачи, используя параллельные вычисления на кластерах и суперкомпьютерах (например, с помощью MPI, OpenMP), что необходимо для прогнозирования климата, моделирования ядерных реакций или расшифровки структуры белков.

Среды разработки (IDE) и системы контроля версий (Git), которые делают процесс создания сложного программного обеспечения для физики управляемым и коллективным.

Таким образом, физика задает вопрос «что смоделировать?», математика — «как это описать?», а информатика — «как это эффективно реализовать в коде и визуализировать?». Их симбиоз рождает современные технологии — от точных прогнозов погоды и создания новых материалов до разработки беспилотных автомобилей и освоения космоса. Среды разработки и специализированные библиотеки для визуализации и ускорения расчетов.

Таким образом, физика задает вопрос «что смоделировать?», математика — «как это описать?», а информатика — «как это реализовать в коде?».

2 Профессиональная направленность содержания обучения

Содержание образовательных программ в среднем профессиональном образовании должно быть целенаправленно выстроено вокруг формирования конкретных, востребованных на рынке труда компетенций. Для специальности 09.02.07 «Информационные системы и программирование» это означает, что каждый учебный модуль должен вносить вклад в подготовку высококвалифицированного разработчика, способного решать комплексные прикладные задачи. Для студентов данной специальности умение создать, запрограммировать и проанализировать математическую модель физического процесса является не абстрактным навыком, а ключевой профессиональной компетенцией.

Ключевая роль интеграции фундаментальных дисциплин.

В этом контексте стратегическая интеграция математики, информатики и физики приобретает первостепенное значение и перестает быть просто формальным требованием образовательного стандарта. Эта триада дисциплин формирует неразрывный симбиоз, создающий прочный фундамент для профессиональной деятельности, будущего программиста и специалиста по информационным системам.

Математика (включая математический анализ, линейную алгебру, дискретную математику и теорию вероятностей) предоставляет универсальный язык и аппарат для формализации задач, описания закономерностей и создания алгоритмов.

Информатика и программирование дают практический инструментарий для реализации математических моделей в виде программного кода, алгоритмов и вычислительных методов.

Физика выступает в роли основного поставщика реальных, содержательных проблемных областей, процессы в которых требуют точного количественного описания и прогнозирования.

Математическое моделирование как ключевая профессиональная компетенция

Для студентов данной специальности умение создать, запрограммировать и провести всесторонний анализ математической модели физического, технического или экономического процесса является не абстрактным академическим навыком, а ключевой профессиональной компетенцией, определяющей их востребованность и конкурентоспособность.

Этот навык подразумевает целый комплекс умений:

1. Формализация: Выделение существенных параметров реального процесса и абстрагирование от второстепенных.

2. Построение модели: Перевод физических законов (например, законов Ньютона, термодинамики, электродинамики) в систему уравнений или алгоритмических правил.

3. Программная реализация: Выбор оптимальных структур данных, численных методов (например, методов Рунге-Кутты для решения дифференциальных уравнений) и написание эффективного, надежного кода на одном из языков программирования (C++, Python, Java и др.).

4. Верификация и валидация: Проверка корректности работы программы и соответствия результатов модели реальному поведению системы.

5. Анализ и интерпретация результатов: Извлечение содержательных выводов из данных, полученных в ходе компьютерного эксперимента, и их представление в понятной форме (например, с помощью средств визуализации).

Прикладное значение для будущей профессии

Овладение этой компетенцией открывает выпускникам дорогу в такие высокотехнологичные отрасли, как:

Разработка игр и компьютерная графика: Моделирование физики движения, столкновений, поведения жидкостей и света.

Системы автоматического управления и робототехника: Создание алгоритмов управления на основе моделей динамики систем.

Научные и инженерные вычисления (CAE): Моделирование прочности конструкций, аэродинамических свойств, тепловых режимов.

Обработка сигналов и данных: Анализ и фильтрация сигналов, прогнозное моделирование на основе статистических данных.

Таким образом, интегративное изучение математики, информатики и физики трансформирует обучение из процесса получения разрозненных знаний в целенаправленное формирование специалиста, готового к проектной работе, способного не просто писать код, но и понимать глубинную суть решаемых задач и предлагать эффективные, научно обоснованные решения.

3 Использование современных технологий и метода преподавания

Для успешной и эффективной реализации интеграции дисциплин необходима соответствующая трансформация педагогического инструментария. Традиционные лекционные форматы оказываются недостаточными для формирования комплексных профессиональных умений. В связи с этим предлагается использование проектно-ориентированного метода (ПОМ) в качестве стержневого подхода организации учебного процесса.

Данный метод не просто дополняет учебный план, а кардинально меняет роль студента: из пассивного получателя информации он превращается в активного участника исследовательско-конструкторской деятельности. Студенты сталкиваются с необходимостью не просто заучивать теорию, а самостоятельно инициировать ее поиск, критически оценивать и применять на практике для достижения конкретного, осязаемого результата в рамках учебного проекта.

В рамках проектно-ориентированного метода предлагается последовательное и целенаправленное применение следующего набора современных технологий, которые стали отраслевым стандартом:

1. Язык программирования Python и его научный экосистема

Выбор языка Python в качестве основного инструмента обусловлен его уникальным сочетанием простоты синтаксиса, мощи и универсальности. Это делает его идеальным для образовательных целей и быстрого прототипирования. Ключевые библиотеки формируют готовую среду для научных и инженерных расчетов:

NumPy и SciPy: выступают в роли виртуальной лаборатории для работы с многомерными массивами, реализации сложных численных методов, решения дифференциальных уравнений и проведения статистического анализа, фактически заменяя традиционные инженерные калькуляторы и специализированное ПО.

Matplotlib и Plotly: обеспечивают мощные средства визуализации данных — от простых двумерных графиков до интерактивных трехмерных моделей и анимаций, что позволяет превращать абстрактные численные результаты в наглядные материалы для анализа и презентации.

2. Интерактивные среды разработки (Jupyter Notebook)

Jupyter Notebook представляет собой не просто среду для написания кода, а революционную образовательную платформу. Она позволяет создавать «живые» документы, где код, его исполнение, математические формулы (с помощью LaTeX), графики, пояснения и выводы объединены в единое целое. Это идеально отражает итеративный процесс научного исследования: гипотеза -> код -> эксперимент -> визуализация -> анализ -> вывод.

3. Игровые движки (на примере Unity)

Применение игровых движков, таких как Unity, выводит мотивацию студентов на качественно новый уровень. Они позволяют наглядно продемонстрировать, как абстрактные физические расчеты (механика, кинематика, столкновения, траектории) напрямую воплощаются в создании динамичного и интерактивного виртуального контента. Студент видит немедленный и зрелищный результат своей работы — будь то траектория полета снаряда или поведение физического объекта под действием силы тяжести, что превращает учебу из рутины в увлекательный творческий процесс.

4. Системы контроля версий (Git)

Обучение работе с Git (и платформами типа GitHub или GitLab) является обязательным элементом подготовки к реальной профессиональной деятельности. Эта технология прививает культуру командной разработки, позволяет обучать студентов управлению изменениями в коде, совместной работе над одним проектом, отслеживанию ошибок и ведению документации. Таким образом, студенты осваивают не только технические, но и крайне важные организационно-коммуникативные компетенции.

Совместное применение проектно-ориентированного метода и современного технологического стека создает синергетический эффект. Оно моделирует в учебной аудитории среду, максимально приближенную к условиям реальной IT-индустрии. Студенты проходят полный жизненный цикл разработки — от постановки задачи и теоретического обоснования до реализации, тестирования, документирования и презентации готового продукта, что в конечном итоге формирует из них высококвалифицированных и конкурентоспособных специалистов.

4 Конкретный практический пример

Для закрепления теоретических принципов интеграции и демонстрации их практической ценности рассмотрим детально проработанный учебный проект, который может быть ключевым в рамках курса или модуля

Название проекта: «Численное моделирование движения планет в гравитационном поле звезды»

Специальность: 09.02.07 «Информационные системы и программирование»

Ключевая задача: Разработать на языке Python программу, которая не только рассчитывает, но и визуализирует в реальном времени движение планеты под действием гравитации звезды, наглядно демонстрируя фундаментальные законы небесной механики.

Цели и решаемые педагогические задачи

Дидактическая цель: Сформировать у студентов целостное представление о полном цикле решения прикладной задачи: от физической постановки и математической формализации до программной реализации и анализа результатов.

Профессиональные задачи:

Научиться переводить физические законы в алгоритмы.

Освоить на практике базовые методы численного моделирования.

Приобрести навыки визуализации результатов вычислений для их интерпретации.

Мотивационная задача: Показать, как абстрактные математические и физические концепции оживают в виде динамической модели, созданной собственными руками, что резко повышает вовлеченность и интерес к дисциплинам.

Детальная реализация междисциплинарных связей.

Процесс работы над проектом разбивается на логические этапы, каждый из которых опирается на конкретную дисциплину.

1. Физический этап: Формализация проблемы

Содержание: Студенты опираются на фундаментальные законы:

Закон всемирного тяготения Ньютона: $F = G * (m_{\text{star}} * m_{\text{planet}}) / r^2$.

Определяет величину силы притяжения.

Второй закон Ньютона: $F = m * a$. Позволяет перейти от силы к ускорению: $a = F / m$.

Результат этапа: Физическая модель системы, где гравитационная сила рассматривается как вектор, направленный от планеты к звезде. Студенты осознают, что задача по своей природе векторная и динамическая.

2. Математический этап: Создание вычислительной модели

Содержание: Осознание того, что аналитически решить уравнения движения для системы двух тел сложно, приводит к необходимости применения численных методов. Задача сводится к решению системы обыкновенных дифференциальных уравнений второго порядка.

Выбор метода:

Метод Эйлера: Используется как простейший для понимания концепции численного интегрирования. Позволяет пошагово обновлять координаты и скорость: $v_{\text{new}} = v_{\text{old}} + a * dt$, $pos_{\text{new}} = pos_{\text{old}} + v_{\text{new}} * dt$.

Метод Верле (Velocity Verlet): Предлагается как более точная и устойчивая альтернатива для серьезных проектов. Этот метод лучше сохраняет полную энергию системы, что критически важно для моделирования стабильных орбит без искусственного "затухания" или "разгона".

Результат этапа: Дискретная математическая модель, готовая к программной реализации.

3. Этап информатики и программирования: Реализация и визуализация

Содержание: Создание программного кода на Python, который воплощает математическую модель.

Структура программы:

1. Инициализация параметров (массы, начальное положение, скорость, гравитационная постоянная, размер временного шага dt).

2. Основной цикл моделирования (на каждую итерацию, представляющую шаг по времени dt):

Расчет силы: Вычисление расстояния r между телами и вектора гравитационной силы.

Расчет ускорения: $a = F / m_{\text{planet}}$.

Интегрирование: Обновление скорости и положения планеты с помощью выбранного численного метода (Эйлера или Верле).

3. Визуализация в реальном времени: С помощью библиотеки Matplotlib (или Plotly) создается анимация, где положение планеты отображается на каждом шаге, формируя ее траекторию.

Результат этапа: Работоспособное приложение, представляющее собой законченный программный продукт.

Итоговый образовательный результат и компетенции

В результате выполнения проекта студенты не просто "пишут код". Они создают работающую симуляцию, которая наглядно подтверждает законы Кеплера:

Орбита планеты принимает форму эллипса, в одном из фокусов которого находится звезда.

Наблюдается прямой переход от физического закона к его математическому описанию и далее — к программной реализации.

Студенты на практике осознают:

Важность выбора адекватного численного метода и размера временного шага (dt) для точности и устойчивости симуляции.

Мощь современных библиотек научных вычислений (NumPy для векторных операций) и визуализации (Matplotlib).

Ценность итеративного подхода: модель можно усложнять — добавлять больше планет, учитывать сопротивление среды, создавать графический интерфейс.

Данный проект служит краеугольным камнем, на основе которого студенты в дальнейшем смогут браться за более сложные задачи моделирования, будучи вооруженными пониманием полного цикла создания вычислительной модели.

5 Формирование профессиональных компетенций будущих специалистов

Реализация описанного подхода способствует формированию у будущих программистов следующих профессиональных компетенций:

Способность к формализации прикладных задач и переводу их в математическую модель.

Умение выбирать и применять адекватные численные методы для решения поставленных задач.

Навык разработки программного обеспечения для математического моделирования и анализа данных.

Готовность к участию в проектах, требующих междисциплинарных знаний (физика, математика, информатика).

Способность верифицировать и анализировать результаты computational experiments.

Это формирует портрет современного IT-специалиста, способного решать комплексные задачи на стыке различных научных дисциплин.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Губанова, О. А. Междисциплинарный подход к обучению: интеграция информатики, математики и физики через основы алгоритмизации и программирования в основной школе / О. А. Губанова. — Текст : непосредственный // Молодой ученый. — 2024. — № 45 (544). — С. 77-80. — URL: <https://moluch.ru/archive/544/119122> (дата обращения: 18.09. 2025)
2. Лактина, Д. С. Роль математики в физике / Д. С. Лактина, М. М. Клименко. — Текст : непосредственный // Юный ученый. — 2022. — № 11 (63). — С. 17-20. — URL: <https://moluch.ru/young/archive/63/3262>. (дата обращения: 18.09. 2025)
3. Математика, информатика, физика: проблемы и перспективы. Международная научно-практическая конференция: сборник научных статей международной научно – практической конференции «Математика, информатика, физика: проблемы и перспективы», Оренбург, 24–25 апреля 2025 г./ ответственный редактор доктор педагогических наук И.В. Игнатушина; Министерство просвещения РФ; ФГБОУ ВО «Оренбургский государственный педагогический университет». – Оренбург: [б.и.], 2025. – 646 с.: ил.
4. Математика, информатика, физика в науке и образовании: сб. научных трудов / Моск. пед. гос. ун-т. — М. : МПГУ, 2012. — 380 с.
5. Ухабина, Е. А. Межпредметные связи математики и физики / Е. А. Ухабина. — Текст: непосредственный // Молодой ученый. — 2023. — № 18 (465). — С. 253-255. — URL: <https://moluch.ru/archive/465/102207> (дата обращения: 18.09. 2025)