

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Проректор по учебной работе и  
довузовской подготовке**

**А.А. Воронов**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Компьютерные технологии решения научных задач
<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Общая и прикладная физика Физтех-школа физики и исследований им. Ландау кафедра информатики и вычислительной математики
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 0 час.

семинары: 0 час.

лабораторные занятия: 60 час.

Самостоятельная работа: 120 час.

Всего часов: 180, всего зач. ед.: 4

Количество контрольных работ, заданий: 4

Программу составил: К.А. Беклемышева, канд. физ.-мат. наук, доцент

Программа обсуждена на заседании кафедры информатики и вычислительной математики 06.02.2020

## Аннотация

Данный курс предназначен для того, чтобы познакомить студентов с методиками решения несложных актуальных задач из различных лабораторий и основными инструментами, которыми пользуются научные программисты.

В рамках курса будет дан обзор некоторых технологий, применяемых при решении научных задач, по каждой технологии будет сжатый рассказ в формате "теория + демонстрация". Основная суть курса – проектная деятельность, решение небольших практически значимых задач из институтских лабораторий. Предполагается работа в группах по 2-5 человек, погружение конкретно в выбранную задачу, работа над ней весь семестр, представление итогового результата.

## 1. Цели и задачи

### Цель дисциплины

Научить студентов использовать типовые компьютерные технологии решения научных задач, научить работать с комплексной задачей от этапа постановки до реализации на одном из языков программирования и представления результата, научить работать в группе с использованием системы контроля версий git.

### Задачи дисциплины

- Формирование навыков работы с комплексной задачей от этапа уточнения требований и определения строгой постановки до реализации на одном из языков программирования и представления результата;
- формирование навыка проектирования и разработки программного обеспечения с использованием системы контроля версий, в том числе в рабочей группе;
- формирование базовых знаний о распространённых компьютерных технологиях решения научных задач — вопросы машинной точности вычислений, библиотеки работы с геометрией, средства научной визуализации, API некоторых библиотек для расчётов с использованием сеток и частиц, средства разработки графического интерфейса, различные технологии распараллеливания расчётного кода.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-2 Способен использовать современные информационные технологии и	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности

программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
ОПК-5 Способен участвовать в проведении фундаментальных и прикладных исследований и разработок, самостоятельно осваивать новые теоретические, в том числе, математические методы исследований, и работать на современной экспериментальной научно-исследовательской, измерительно-аналитической и технологической аппаратуре	ОПК-5.1 Способен решать поставленные задачи в области теоретических и экспериментальных исследований и разработок
	ОПК-5.2 Обладает способностью к освоению новых знаний на основе изучения литературы, научных статей и других источников
	ОПК-5.3 Способен к профессиональной эксплуатации современной экспериментальной научно-исследовательской (измерительно-аналитической и технологической) аппаратуры
ПК-1 Способен планировать и проводить научные эксперименты (в избранной предметной области) и (или) теоретические (аналитические и имитационные) исследования	ПК-1.3 Владеет культурой постановки научной задачи и моделирования естественнонаучных объектов и систем
	ПК-1.6 Знает основные правила поведения и работы в современной научной лаборатории
	ПК-1.4 Умеет строить математические модели для описания и исследования процессов и явлений в соответствующих научных областях
	ПК-1.8 Владеет навыками работы с современными языками программирования и программными пакетами для научных расчетов
	ПК-1.7 Способен оценивать требуемые ресурсы (материальные и временные) для планирования и проведения научного эксперимента
ПК-2 Способен анализировать полученные в ходе научно-исследовательской работы данные и делать научные выводы (заключения)	ПК-2.1 Владеет методами статистической обработки и анализа научных данных
	ПК-2.2 Умеет находить ключевые параметры, определяющие изучаемое явление, и производить численные оценки по порядку величины
ПК-3 Способен выбирать и применять подходящее оборудование, инструменты и методы исследований для решения задач в избранной предметной области	ПК-3.2 Знает области и критерии применимости используемых теоретических подходов и умение оценивать точность приближенных аналитических методов вычислений
	ПК-3.3 Умеет производить оценку точности численных методов, используемых на ЭВМ, вычислительной сложности используемых алгоритмов и объема требуемых вычислительных ресурсов
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- Особенности написания расчётного кода с учётом машинной точности вычислений;
- одну из библиотек работы с геометрией и построения расчётных сеток;
- технологии научной визуализации с использованием VTK;
- одну из библиотек для расчётов с использованием сеток или частиц;
- базовые принципы разработки графического интерфейса;
- базовые принципы распараллеливания расчётного кода с использованием OpenMP, MPI, CUDA.

уметь:

- Отображать физическую задачу на компьютерные технологии её решения;
- определять допустимое приближение задачи с учётом имеющихся входных данных;
- выполнять проектирование и разработку программного обеспечения для поставленной задачи;
- использовать систему контроля версий git для совместной разработки;
- представлять выполненную работу в формате доклада.

владеть:

- Навыками использования git;
- навыками использования git средств визуализации;
- навыками написания кроссплатформенных оконных интерфейсов с использованием Qt.

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Средства совместной разработки с использованием git.			4	10
2	Средства визуализации. Научная визуализация с использованием vtk/paraview.			4	10
3	Средства визуализации. 2D и 3D графика средствами OpenGL.			4	10
4	Средства работы с геометрией. Геометрические модели - построение, импорт в свою программу.			4	10
5	Средства работы с геометрией. Расчётные сетки средствами gmsh, Ansys, CGAL - построение, импорт в свою программу.			8	10
6	Техники моделирования физических задач. Некоторые библиотеки и API для сеточных методов.			8	10
7	Техники моделирования физических задач. Некоторые библиотеки и API для методов частиц.			8	10
8	Особенности написания расчётного кода. Вопросы производительности. Машинная точность вычислений.			4	10
9	Кроссплатформенные оконные интерфейсы с использованием Qt. Базовый обзор технологии.			4	10
10	Технологии распараллеливания на CPU. OpenMP. Базовый обзор технологии.			4	10

11	Технологии распараллеливания на CPU. MPI. Базовый обзор технологии.			4	10
12	Технологии распараллеливания на GPU. CUDA. Базовый обзор технологии.			4	10
Итого часов				60	120
Подготовка к экзамену		0 час.			
Общая трудоёмкость		180 час., 4 зач.ед.			

#### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 4 (Весенний)

##### 1. Средства совместной разработки с использованием git.

Централизованные и распределённые системы контроля версий. Общая организация системы контроля версий git. Понятие коммита (commit), ветки (branch), репозитория (repository). Варианты организации работы при совместной разработке проекта с использованием общего репозитория. Понятие форка (fork), клонирования (clone), слияния (merge), запроса на изменение (pull request). Консольный интерфейс git. Некоторые графические клиенты git.

##### 2. Средства визуализации. Научная визуализация с использованием vtk/paraview.

Различные подходы к визуализации. Понятие об асинхронности расчёта и визуализации вычислительных экспериментов. Общая архитектура библиотеки VTK. Работа со структурированными и неструктурированными сетками из расчётных модулей на C++ и Python. Запись динамического процесса как последовательности кадров. Просмотр, анализ и постобработка расчётных данных в Paraview.

##### 3. Средства визуализации. 2D и 3D графика средствами OpenGL.

Высокоуровневый обзор средств интерактивной машинной графики. Взаимодействие прикладного расчётного кода, процессора (CPU) и видеокарты (GPU). Общая логика организации OpenGL как типового инструмента визуализации. Примеры использования OpenGL в своей программе. Библиотеки GLUT, GLFW. Библиотека SFML.

##### 4. Средства работы с геометрией. Геометрические модели - построение, импорт в свою программу.

Общие подходы к хранению данных о геометрии сложных моделей или гетерогенных сред. Геометрические примитивы для описания поверхностей и объёмов. Некоторые редакторы геометрических моделей. Формат STL как пример типового формата хранения данных геометрии. Сопоставление подходов к заданию и хранению геометрии с подходами при расчёте и визуализации.

##### 5. Средства работы с геометрией. Расчётные сетки средствами gmsh, Ani3D, CGAL - построение, импорт в свою программу.

Необходимость построения расчётных сеток из входных геометрических данных. Типовые требования к расчётным сеткам со стороны вычислительных модулей. Библиотеки gmsh, Ani3D, CGAL - общая архитектура, использование из своей программы на C++ или Python.

##### 6. Техники моделирования физических задач. Некоторые библиотеки и API для сеточных методов.

Общий подход к моделированию физических задач, в которых объектом выступает сплошная среда. Философия сеточных методов. Некоторые библиотеки для реализации сеточных методов - Deal.II, DOLFIN, FEniCS. Примеры использования библиотек в своей программе.

7. Техники моделирования физических задач. Некоторые библиотеки и API для методов частиц.

Общий подход к моделированию физических задач с использованием методов частиц. Сходства и отличия методов молекулярной динамики и сглаженных частиц. Некоторые библиотеки для реализации методов частиц - PySPH, SPLISHSPLASH. Примеры использования библиотек в своей программе.

8. Особенности написания расчётного кода. Вопросы производительности. Машинная точность вычислений.

Вопросы производительности вычислительных программ, обусловленные архитектурой современного оборудования - эффекты использования кэшей процессора, обмена данными с оперативной памятью, конвейеризации и векторизации операций. Вопросы машинной точности при работе с малыми и большими величинами. Необходимость обезразмеривания данных и приведения их к одному порядку.

9. Кроссплатформенные оконные интерфейсы с использованием Qt. Базовый обзор технологии.

Базовый обзор технологии Qt. Базовое понятие о среде разработки Qt Creator. Логика построения оконных интерфейсов, основанная на обработке событий. Необходимость использования библиотек для достижения кроссплатформенности приложения. Необходимость использования многопоточности и асинхронных операций при разработке интерфейса пользователя.

10. Технологии распараллеливания на CPU. OpenMP. Базовый обзор технологии.

Понятие о распараллеливании на CPU в общей памяти - возможности и ограничения подхода. Базовый обзор технологии распараллеливания OpenMP. Некоторые директивы OpenMP. Сборка программы на C++ с использованием OpenMP. Использование OpenMP в программе на Python. Примеры использования OpenMP на разных логических уровнях программы, приводящие к заметному отличию в эффективности.

11. Технологии распараллеливания на CPU. MPI. Базовый обзор технологии.

Понятие о распараллеливании на CPU для суперкомпьютера с распределённой памятью - возможности и ограничения подхода, необходимость информационных обменов и синхронизации. Базовый обзор технологии распараллеливания MPI. Некоторые функции MPI, достаточные для запуска базовых примеров. Сборка и запуск программы на C++ с использованием MPI. Использование MPI в программе на Python.

12. Технологии распараллеливания на GPU. CUDA. Базовый обзор технологии.

Понятие об использовании сопроцессоров для вычислений. GPU как сопроцессор. Необходимость информационных обменов и синхронизации с сопроцессором. Базовый обзор технологии CUDA. Некоторые функции CUDA, достаточные для запуска базовых примеров. Сборка и запуск программы на C++ с использованием CUDA. Использование CUDA в программе на Python.

## **5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Компьютерный класс с доской, проектором или телевизором, подключенный к сети Интернет.

## 6. Перечень рекомендуемой литературы

### Основная литература

1. Программирование на Python 3 : Подробное руководство [Текст] = Programming in Python 3 : [учеб. пособие для вузов] / М. Саммерфилд; пер. с англ. А. Киселева .— СПб : Символ-Плюс, 2015 .— 608 с.
2. Язык программирования C++ для профессионалов / Б. Страуструп .— 2-е изд., испр. — Москва : ИНТУИТ, 2016 .— Электрон. версия печ. публикации .

### Дополнительная литература

1. Численные методы, алгоритмы и программы. Введение в распараллеливание [Текст] : учеб. пособие для вузов / В. Е. Карпов, А. И. Лобанов .— М. : Физматкнига, 2014 .— 192 с.

## 7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. <https://github.com>
2. <https://vtk.org/>
3. <https://www.opengl.org/>
4. <https://www.cgal.org/>
5. <http://gmsh.info/>
6. <https://fenicsproject.org/>
7. <https://pysph.readthedocs.io/>
8. <http://moodle.phystech.edu/enrol/index.php?id=281>
9. <https://www.qt.io/>
10. <https://www.openmp.org/>
11. <https://www.open-mpi.org/>
12. <https://developer.nvidia.com/cuda-toolkit>

## 8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

На лабораторных занятиях для демонстрации технологий используются мультимедийные технологии, включая демонстрацию презентаций и показ работы приложений. Для командной работы над проектами обучающимся во время занятий необходим доступ в Интернет, в частности к сайту <https://github.com>. Студенты могут использовать среды разработки CLion и PyCharm. При их отсутствии используются компилятор g++ и интерпретатор python в командной строке.

## 9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий курс «Компьютерные технологии решения научных задач» должен, с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать особенности написания расчётного кода с учётом машинной точности вычислений, одну из библиотек работы с геометрией и построения расчётных сеток, технологии научной визуализации с использованием VTK, одну из библиотек для расчётов с использованием сеток или частиц, базовые принципы разработки графического интерфейса, базовые принципы распараллеливания расчётного кода с использованием OpenMP, MPI, CUDA.

Успешное освоение курса требует напряжённой работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;

- проработку учебного материала (по учебной и научной литературе), доказательство отдельных утверждений, свойств;
- решение задач, предлагаемых студентам на лабораторных работах;
- подготовку к лабораторным работам.
- решение заданий

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Показателем владения материалом служит умение решать теоретические и практические задачи. Для формирования умения применять теоретические знания на практике студенту необходимо решать как можно больше практических задач. При решении задач каждое действие необходимо аргументировать, ссылаясь на известные теоретические сведения. Программы должны легко читаться и иметь подробные комментарии.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю, проводящему лабораторные работы.



**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Общая и прикладная физика Физтех-школа физики и исследований им. Ландау кафедра информатики и вычислительной математики
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

**Разработчик:** К.А. Беклемышева, канд. физ.-мат. наук, доцент

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
ОПК-5 Способен участвовать в проведении фундаментальных и прикладных исследований и разработок, самостоятельно осваивать новые теоретические, в том числе, математические методы исследований, и работать на современной экспериментальной научно-исследовательской, измерительно-аналитической и технологической аппаратуре	ОПК-5.1 Способен решать поставленные задачи в области теоретических и экспериментальных исследований и разработок
	ОПК-5.2 Обладает способностью к освоению новых знаний на основе изучения литературы, научных статей и других источников
	ОПК-5.3 Способен к профессиональной эксплуатации современной экспериментальной научно-исследовательской (измерительно-аналитической и технологической) аппаратуры
ПК-1 Способен планировать и проводить научные эксперименты (в избранной предметной области) и (или) теоретические (аналитические и имитационные) исследования	ПК-1.3 Владеет культурой постановки научной задачи и моделирования естественнонаучных объектов и систем
	ПК-1.6 Знает основные правила поведения и работы в современной научной лаборатории
	ПК-1.4 Умеет строить математические модели для описания и исследования процессов и явлений в соответствующих научных областях
	ПК-1.8 Владеет навыками работы с современными языками программирования и программными пакетами для научных расчетов
	ПК-1.7 Способен оценивать требуемые ресурсы (материальные и временные) для планирования и проведения научного эксперимента

ПК-2 Способен анализировать полученные в ходе научно-исследовательской работы данные и делать научные выводы (заключения)	ПК-2.1 Владеет методами статистической обработки и анализа научных данных
	ПК-2.2 Умеет находить ключевые параметры, определяющие изучаемое явление, и производить численные оценки по порядку величины
ПК-3 Способен выбирать и применять подходящее оборудование, инструменты и методы исследований для решения задач в избранной предметной области	ПК-3.2 Знает области и критерии применимости используемых теоретических подходов и умение оценивать точность приближенных аналитических методов вычислений
	ПК-3.3 Умеет производить оценку точности численных методов, используемых на ЭВМ, вычислительной сложности используемых алгоритмов и объема требуемых вычислительных ресурсов
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Компьютерные технологии решения научных задач (ЛФИ)» обучающийся должен:

### знать:

- Особенности написания расчётного кода с учётом машинной точности вычислений;
- одну из библиотек работы с геометрией и построения расчётных сеток;
- технологии научной визуализации с использованием VTK;
- одну из библиотек для расчётов с использованием сеток или частиц;
- базовые принципы разработки графического интерфейса;
- базовые принципы распараллеливания расчётного кода с использованием OpenMP, MPI, CUDA.

### уметь:

- Отображать физическую задачу на компьютерные технологии её решения;
- определять допустимое приближение задачи с учётом имеющихся входных данных;
- выполнять проектирование и разработку программного обеспечения для поставленной задачи;
- использовать систему контроля версий git для совместной разработки;
- представлять выполненную работу в формате доклада.

### владеть:

- Навыками использования git;
- навыками использования git средств визуализации;
- навыками написания кроссплатформенных оконных интерфейсов с использованием Qt.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Текущий контроль проводится при помощи презентаций и отчетов по итогам решения практических задач. В течение семестра от студента требуется подробное и развернутое решение трех задач.

Перечень задач:

1. Вычислить коэффициенты эллипса минимальной площади, включающего в себя множество всех точек, заданных координатами в двумерном пространстве.
2. Известно, что в некой двумерной среде показатель преломления зависит от координат. Зная точку входа в эту среду и нужную точку выхода определить угол альфа, под которым луч должен попадать в эту среду. Указание: считать, что показатель преломления среды изменяется плавно.
3. На орбите под действием внешних сил движется космический аппарат. В качестве сил рассматривается притяжение Земли и давление Солнечного излучения. Определить, где аппарат будет через сутки. В гравитационном потенциале учитывать первых 4 гармоники. Считать, что сила давления солнечного излучения пропорциональна освещаемой площади.

4. Нужно какое минимальное количество топлива потребуется, чтобы рассчитать перелет с геостационарной орбиты (36 000 км над уровнем моря) на орбиту Луны 100 км над уровнем поверхности Луны, зная, что масса полезной нагрузки 100 кг, тяга двигателя 1 мН, удельный импульс 20 км в секунду. Считать, что сила тяги всегда направлена по скорости аппарата. Считать, что Луна летает вокруг Земли. Гравитационные поля Земли и Луны считать центральными (т.е. создаваемыми точечными массами).
5. Моделирование транспортного потока с точностью до отдельной машины.
  - a. Очередь на светофоре с учетом времени на разгон и остановку, дорога однополосная.
  - b. Фрагмент многополосной дороги. Учитывать время на разгон, торможение, минимальную дистанцию, возможность перестроения из полосы в полосу, ограничение скорости. Оценка пропускной способности в зависимости от загрузки дороги. Анализ того, как разные параметры влияют на пропускную способность.
  - c. Перекресток, регулируемый, с разными режимами светофора, многополосными дорогами и разной разметкой.
  - d. Перекресток нерегулируемый, машины ориентируются на приоритеты и область видимости.
  - e. «Квартал» - небольшая сеть из дорог и перекрестков.
  - f. «Город» - большая сеть из дорог и перекрестков.
  - g. «Настоящий город» - перерисовать или выгрузить реальную карту.
6. Молекулярное моделирование.
  - a. Одноатомный газ, трехмерный, в замкнутом объеме. Моделировать с точностью до атома. Анализировать зависимость давления от температуры и объема.
  - b. Нагрев с одной стороны, динамика распределения температуры.
  - c. Смесь нескольких газов.
  - d. Многоатомные газы, учет вращения молекулы.
  - e. Химические реакции.
7. Анализ геоданных.
  - a. Оценка площади суши. Модельная задача – картинка в bmp/png, размеченная синим и зеленым.
  - b. Оценка площади суши по фрагментам фото океана с гугл/яндекс карт.
8. Моделирование медицинского ультразвука при помощи рейтрейсинга.
9. Моделирование нагрева, охлаждения, таяния и намораживания льда в арктических условиях.
10. Расчет гравитационного маневра при помощи численного решения задачи n тел. Решение обратной задачи – подбор начальных параметров с целью достижения определенной точки пространства или набора определенной скорости.

#### **4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся**

Промежуточная аттестация по дисциплине осуществляется в форме дифференцированного зачёта. Он проводится в устной форме.

Перечень контрольных вопросов:

1. Средства совместной разработки с использованием git.
2. Средства визуализации. Научная визуализация с использованием vtk/paraview.
3. Средства визуализации. 2D и 3D графика средствами OpenGL.
4. Средства работы с геометрией. Геометрические модели - построение, импорт в свою программу.
5. Средства работы с геометрией. Расчётные сетки средствами gmsh, Ani3D, CGAL - построение, импорт в свою программу.
6. Техники моделирования физических задач. Некоторые библиотеки и API для сеточных методов.
7. Техники моделирования физических задач. Некоторые библиотеки и API для методов частиц.
8. Особенности написания расчётного кода. Вопросы производительности. Машинная точность вычислений.
9. Кроссплатформенные оконные интерфейсы с использованием Qt. Базовый обзор технологии.
10. Технологии распараллеливания на CPU. OpenMP. Базовый обзор технологии.

11. Технологии распараллеливания на CPU. MPI. Базовый обзор технологии.
12. Технологии распараллеливания на GPU. CUDA. Базовый обзор технологии.

Примеры задач дифференцированного зачета:

1. Выполнить расчёт распространения волны в среде (по заданным аналитическим соотношениям), результат записать в формате VTK и визуализировать средствами Paraview.
2. Выполнить расчёт задачи N тел, визуализировать результат с использованием любой выбранной библиотеки.
3. Загрузить геометрическую модель из файла, задать на ней скалярное поле по заданным аналитическим соотношениям), результат записать в формате VTK и визуализировать средствами Paraview.
4. Построить в заданном геометрическом примитиве (шар, цилиндр, призма) сетку из тетраэдров, задать на ней скалярное поле по заданным аналитическим соотношениям), результат записать в формате VTK и визуализировать средствами Paraview.
5. Напишите код, демонстрирующий эффект переполнения мантиссы. Напишите код, демонстрирующий эффект антипереполнения. Определите минимальное по модулю число, которым может оперировать компьютер для типов float и double. Сравните производительность работы сопроцессора на нормализованных и денормализованных числах.
6. Найти среднее арифметическое массива из вещественных чисел, сравнить время работы при оптимизациях O0, O1, O2 и O3.
7. Оценить, как влияет исполнение различных операций с числами с плавающей точкой (сложение, вычитание, умножение, деление и т.д.) на время работы. Ускорить программу за счет конвейера операций. Измерить производительность в FLOPS, сравнить с теоретическими показателями процессора.
8. Выполнить корректное слияние веток разработки в рамках одного git-репозитория.
9. Построить график заданной функции в оконном интерфейсе с использованием Qt.
10. Реализовать заданные операции над матрицами на C++ и на Python. Сравнить время выполнения. Обсудить скорость работы при самостоятельной реализации и при использовании различных библиотек, объяснить различия.

#### Критерии оценивания

- 10 - Выставляется студенту, успешно защитившему все необходимые проекты и дополнительный проект повышенной сложности, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, а также способность самостоятельно развивать исходно поставленную задачу проекта.
- 9 - Выставляется студенту, успешно защитившему все необходимые проекты, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, а также способность самостоятельно развивать исходно поставленную задачу проекта.
- 8 - Выставляется студенту, успешно защитившему все необходимые проекты, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины.
- 7 - Выставляется студенту, успешно защитившему все необходимые проекты, если он твердо знает материал и, по существу, излагает его.
- 6 - Выставляется студенту, успешно защитившему все необходимые проекты, если он знает материал и, по существу, излагает его, но допускает в ответе на зачёте или на защите проектов существенные неточности.
- 5 - Выставляется студенту, успешно защитившему хотя бы один проект, если он знает материал и, по существу, излагает его.
- 4 - Выставляется студенту, не защитившему проекты, если он даёт правильные формулировки базовых понятий и может применять полученные знания по образцу в стандартной ситуации.
- 3 - Выставляется студенту, не защитившему проекты, если он даёт недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом может применять полученные знания по образцу в стандартной ситуации.
- 2 - Выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

1 - Выставляется студенту, который не способен сформулировать основные понятия дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Дифференцированный зачет проводится по билетам, включающим контрольные вопросы и практические задачи. При выставлении итоговой оценки учитываются результаты решения задач текущего контроля.