

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

**УТВЕРЖДЕНО**

**Директор физтех-школы физики  
и исследований им. Ландау**

**А.В. Рогачев**

**Рабочая программа дисциплины (модуля)**

<b>по дисциплине:</b>	Методы массивно-параллельного программирования в среде CUDA для решения задач теоретической и математической физики
<b>по направлению:</b>	Фотоника и оптоинформатика
<b>профиль подготовки:</b>	Фотоника, квантовые технологии и двумерные материалы Физтех-школа физики и исследований им. Ландау Физтех-кластер академической и научной карьеры
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: Е.Е. Перепелкин, д-р физ.-мат. наук

Программа обсуждена на заседании Физтех-кластера академической и научной карьеры 04.04.2022

## Аннотация

Курс лекций рассчитан на широкий круг студентов, аспирантов, преподавателей ВУЗов и специалистов в различных областях математического моделирования и теоретической физики, для которых программирование не является основной специальностью, а используется ими как дополнительный инструмент в численном моделировании исследуемых задач.

В курсе изложены базовые знания, необходимые, чтобы быстро и эффективно начать писать программы на графическом процессоре (GPU) без специальной подготовки в области программирования. Курс преследует цель изложить материал на простом доступном уровне, в первую очередь, пользователям, занимающимся прикладными задачами.

### 1. Цели и задачи

#### Цель дисциплины

Целью освоения дисциплины является приобретение обучающимися профессиональных компетенций в области информатики и вычислительной техники, являющихся основой профессиональных и специальных дисциплин, необходимых для разработки программного обеспечения и успешной профессиональной деятельности специалистов.

#### Задачи дисциплины

- понимание программирования на графических процессорах, его сущности и места в системе формирования математических моделей и методов моделирования физических систем;
- изучение научных физических задач, приводящих к вычислительным методам, реализуемых на параллельной архитектуре графических процессоров;
- владение полученными знаниями и применение их при решении задач распараллеливания на массивно-параллельной архитектуре графических процессоров;
- формирование у студентов способностей к анализу численных методов и программных алгоритмов на возможность их распараллеливания на графических процессорах.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

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

В результате освоения дисциплины обучающиеся должны

знать:

базовые понятия архитектуры графических процессоров и элементов программной среды CUDA.

уметь:

использовать базовые знания по программно-аппаратному стеку CUDA для распараллеливания программного кода на GPU.

владеть:

навыками анализа программного кода на возможность его распараллеливания на графическом процессоре GPU.

### 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	Архитектура и программирование массивно-параллельных вычислительных систем	1	1		1
2	Гибридная модель вычислений. Типы вычислительных архитектур. Архитектура графического процессора (GPU)	2	1		1
3	Программная модель CUDA. Гибридная модель программного кода. Понятие потока, блока, сети блоков. Функция – ядро, как параллельный код на GPU	2	1		1
4	Иерархия памяти на GPU. Регистры и локальная память. Глобальная память. Шаблон работы с глобальной памятью. Использование pinned-памяти. CUDA-22потоки	2	1		1
5	Объединение запросов. Массивы с выравниванием	2	2		2
6	Разделяемая память. Шаблон работы с разделяемой памятью. Оптимизация работы с разделяемой памятью	2	2		2
7	Статические переменные. Константная память. Текстурная память	2	2		2
8	Вопросы оптимизации приложений на CUDA	1	2		2
9	Решение дифференциальных уравнений на CUDA на примере задач аэро-гидродинамики	2	2		2
10	Решение задачи магнитостатики на GPU	1	2		2
11	Метод массивно-параллельного программирования на GPU в задачах динамики пучка	2	2		2
12	Оценка потерь пучка на GPU как задача трассировки лучей	2	2		2
13	Расчет эффекта пространственного заряда пучка на GPU	2	2		2
14	Задача трассировки пучка на GPU	2	2		2
15	Нерегулярный параллелизм в задачах обработки цифрового сигнала на GPU	2	2		2
16	Режим Multi-GPU. Нейросетевые алгоритмы на GPU	2	2		2
17	Сравнение реализаций модельных задач на OpenMP, OpenACC, CUDA	1	2		2
Итого часов		30	30		30
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

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

Семестр: 1 (Осенний)

##### 1. Архитектура и программирование массивно-параллельных вычислительных систем

Введение. Настройка программной среды в MSVisualStudio. Компиляция простейших примеров. для работы с CUDA. Оценка характеристик видеокарты. ComputeCapability видеокарты.

## 2. Гибридная модель вычислений. Типы вычислительных архитектур. Архитектура графического процессора (GPU)

Написание последовательного кода для вычисления суммы ряда. Оценка времени вычисления. Проект CUDA в MSVisualStudio.

## 3. Программная модель CUDA. Гибридная модель программного кода. Понятие потока, блока, сети блоков. Функция – ядро, как параллельный код на GPU

Написание последовательного кода для алгоритма перемножения квадратных матриц. Использование динамической памяти. Оценка времени выполнения. Работа с CUDA Toolkit.

## 4. Иерархия памяти на GPU. Регистры и локальная память. Глобальная память. Шаблон работы с глобальной памятью. Использование pinned-памяти. CUDA-2D потоки

Настройка проекта CUDA в среде MSVisualStudio. Компиляция примеров из CUDA SDK. Примеры параллельной реализации задачи многих тел и гидродинамики.

## 5. Объединение запросов. Массивы с выравниванием

Написание параллельного алгоритма сложения векторов и суммирование ряда на CUDA. Оценка времени выполнения и сравнение с последовательной реализацией на CPU.

## 6. Разделяемая память. Шаблон работы с разделяемой памятью. Оптимизация работы с разделяемой памятью

Написание параллельного алгоритма перемножения матриц с использованием глобальной памяти на CUDA. Оценка времени выполнения и сравнение с последовательной реализацией на CPU.

## 7. Статические переменные. Константная память. Текстовая память

Написание и оптимизация программного кода на CPU и GPU для работы с различными типами памяти на GPU. Глобальная память, L2 кэш, Read-Only Data кэш.

## 8. Вопросы оптимизации приложений на CUDA

Написание и оптимизация программного кода на CPU и GPU для работы с различными типами памяти на GPU. Разделяемая память, регистры. Конфликт банков.

## 9. Решение дифференциальных уравнений на CUDA на примере задач аэро-гидродинамики

Написание и оптимизация программного кода на CPU/OpenMP. Задача суммирования ряда и перемножения матриц.

## 10. Решение задачи магнитостатики на GPU

Написание и оптимизация программного кода на CPU/OpenMP. Задача многих взаимодействующих тел.

## 11. Метод массивно-параллельного программирования на GPU в задачах динамики пучка

Написание и оптимизация программного кода на CUDA для задач параллельной редукции. Работа с разделяемой памятью. Банк конфликты. Ветвление Warp.

## 12. Оценка потерь пучка на GPU как задача трассировки лучей

Написание и оптимизация программного кода на CUDA для перемножения матриц. 4-Байтовые и 8-Байтовые банки в разделяемой памяти. Банк конфликты на видеокартах с 16 и 32 банками. Работа планировщика warp. Алгоритмы ILP и TLP.

## 13. Расчет эффекта пространственного заряда пучка на GPU

Написание и оптимизация программного кода на CUDA для CFAU. Реализация с использованием глобальной памяти. Использование линейной текстурной памяти.

## 14. Задача трассировки пучка на GPU

Написание и оптимизация программного кода на CUDA для CHAU. Производная Фреше и непрерывный аналог метода Ньютона. Использование коаллесинга при доступе в глобальную память.

## 15. Нерегулярный параллелизм в задачах обработки цифрового сигнала на GPU

Написание и оптимизация программного кода на CUDA для задачи многих тел.

## 16. Режим Multi-GPU. Нейросетевые алгоритмы на GPU

Основные директивы OpenACC. Компиляция кода в среде OpenACC. Написание программного кода на OpenMP и GPU/OpenACC для задач перемножения матриц.

## 17. Сравнение реализаций модельных задач на OpenMP, OpenACC, CUDA

Написание и оптимизация программного кода на OpenMP и GPU/OpenACC для задач задачи многих тел.

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

- Экран, проектор (с HDMI выходом)
- Доступ в Интернет (WiFi подключение)
- Лазерная указка
- Компьютерный класс с видеокартами компании NVIDIA.
- На компьютерах должно быть установлено программное обеспечение для работы с CUDA.

## 6. Перечень рекомендуемой литературы

### Основная литература

1. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA, Издательство: ДМК-Пресс Год: 2010.

### Фонд литературы кафедры:

2. Перепелкин Е.Е., Садовников Б.И., Иноземцева Н.Г., Вычисления на графических процессорах (GPU) в задачах математической и теоретической физики, серия «Классический учебник МГУ», Изд. URSS Москва, ISBN 978-5-9710-6490-9, 2019, 240 стр.
3. Иноземцева Н.Г., Перепелкин Е.Е., Садовников Б.И., Оптимизация алгоритмов задач математической физики для графических процессоров, Изд. МГУ им. М.В. Ломоносова, ISBN 978-5-8279-0107-5, 2012, 256 стр.

### Дополнительная литература

Рекомендуемая литература для самостоятельного изучения:

1. David Kirck and Wen-meiHwu's. Programming Massively Parallel Processors: A Hands-on Approach. Applications of GPU Computing Series, 2007, ECE 498AL, University of Illinois.

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

<http://www.nvidia.com>

<http://developer.nvidia.com>

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

MS Visual Studio 2019, NVIDIA Driver, CUDA ToolKit, SDK

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

Студент, изучающий дисциплину, 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 основные определения и понятия, уметь применять полученные знания для решения различных задач.

Успешное освоение курса требует:

- посещения всех занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без конспекта отвечать на вопросы по темам дисциплины.

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

Возможен промежуточный контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Фотоника и оптоинформатика
<b>профиль подготовки:</b>	Фотоника, квантовые технологии и двумерные материалы Физтех-школа физики и исследований им. Ландау Физтех-кластер академической и научной карьеры
<b>курс:</b>	<u>1</u>
<b>квалификация:</b>	магистр
Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет	
<b>Разработчик:</b>	Е.Е. Перепелкин, д-р физ.-мат. наук

## **1. Компетенции, формируемые в процессе изучения дисциплины**

## **2. Показатели оценивания компетенций**

В результате изучения дисциплины «Методы массивно-параллельного программирования в среде CUDA для решения задач теоретической и математической физики» обучающийся должен:

### **знать:**

базовые понятия архитектуры графических процессоров и элементов программной среды CUDA.

### **уметь:**

использовать базовые знания по программно-аппаратному стеку CUDA для распараллеливания программного кода на GPU.

### **владеть:**

навыками анализа программного кода на возможность его распараллеливания на графическом процессоре GPU.

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

1. Использование глобальной памяти на примере перемножения матриц. Оценка производительности.
2. Разделяемая память. Запись, чтение, синхронизация, скорость доступа.
3. Запись и чтение из глобальной памяти, скорость доступа.
4. Шаблоны доступа к глобальной памяти.

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

Перечень контрольных вопросов:

1. Параллельные архитектуры на базе центрального процессора: Intel Core2 Duo, Quad, Cell, SMP, BlueGene. Особенности работы
2. Параллельная архитектура на базе графических процессоров GPU: NVIDIA Tesla 8, Tesla, Fermi, Kepler, Pascal
3. Среда программирования CUDA для массивно параллельных архитектур GPU.
4. Гибридный подход программирования на GPU/CPU.
5. Понятие функции ядра, сетки блоков, нитей.
6. Основные спецификаторы, и переменные среды CUDA.
7. Основные типы памяти на GPU.
8. Запись и чтение из глобальной памяти, скорость доступа.
9. Шаблоны доступа к глобальной памяти.
10. Понятие Compute Capability. Влияние CC на работу с памятью.
11. Объединение запросов при доступе в глобальную память.
12. Использование глобальной памяти на примере перемножения матриц. Оценка производительности.
13. Разделяемая память. Запись, чтение, синхронизация, скорость доступа.
14. Понятие банков памяти, шаблоны доступа в разделяемую память.
15. Использование разделяемой памяти на примере перемножения матриц. Оценка производительности.
16. Пример параллельной редукции с последующей оптимизацией.
17. Понятие текстуры, фильтры, преобразование типов, способы обращения.
18. Использование текстуры при доступе в глобальную память.
19. Задачи обработки сигнала. Нахождение свертки.
20. Обработка изображения, понятие шума. Подавление шума
21. Параллельное решение системы линейных и нелинейных уравнений на GPU



22. Различные типы фильтров при обработке изображения.
23. Вопросы оптимизации. Классификация задач по степени распараллеливания.
24. Использование CUDART и CUDADriverAPI.
25. Основные шаблоны работы с памятью. Примеры оптимизации кода.
26. Работа с профилировщиком.
27. Особенности работы с несколькими GPU.
28. Примеры использования GPU. Задача гидрогазодинамики.
29. Трассировка лучей.
30. Моделирование динамики пучка.
31. Параллельный алгоритм решения уравнения теплопроводности.

#### Примеры контрольных заданий:

1. Написать в среде CUDA программный код реализующий параллельное перемножение матриц с использованием разделяемой памяти
2. Написать в среде CUDA программный код реализующий параллельное перемножение матриц с использованием глобальной памяти
3. Написать в среде CUDA программный код реализующий параллельное перемножение матриц с использованием разделяемой памяти и алгоритмом ILP и TLP
4. Написать в среде CUDA программный код реализующий параллельное вычисление динамики системы многих частиц с гравитационным взаимодействием.
5. Написать программный код иллюстрирующий эффективность работы с pinned-памятью
6. Написать параллельный код сложения массивов с использованием асинхронного копирования данных между host/device/

#### Примеры билетов

##### Билет 1.

1. Понятие текстуры, фильтры, преобразование типов, способы обращения
2. Задача на работу с pinned-памятью

##### Билет 2.

1. Понятие банков памяти, шаблоны доступа в разделяемую память.
2. Задача на работу с CUDA-Stream

##### Билет 3.

1. Пример параллельной редукции с последующей оптимизацией.
2. Задача на сложение матриц с использованием текстурной памяти.

##### Билет 4.

1. Шаблоны доступа к глобальной памяти. Понятие коалессинга.
2. Задача на сложение массивов в глобальной памяти с использованием выравнивания.

##### Билет 5.

1. Организация работы в режиме Multi-GPU средствами OpenMP и CUDA-Stream
2. Задача на работу с динамической разделяемой памятью.

#### Критерии оценивания

Оценка отлично 10 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично 9 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично 8 баллов - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо 7 баллов - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо 6 баллов - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо 5 баллов - выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно 4 балла - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно 3 балла - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно 2 балла - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно 1 балл - выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

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

Дифференцированный зачёт проводится в устной форме по билетам. В каждом билете представлено два теоретических вопроса. При проведении зачёта обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося не должен превышать одного астрономического часа.