

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

УТВЕРЖДЕНО
Проректор по учебной работе

А.А. Воронов

| | |
|----------------------------|--|
| | Рабочая программа дисциплины (модуля) |
| по дисциплине: | Параллельные вычисления и алгоритмы решения дифференциальных уравнений |
| по направлению: | Ядерная физика и технологии |
| профиль подготовки: | Ядерная физика, УТС и компьютерные методы в физике Физтех-школа физики и исследований им. Ландау кафедра информатики и вычислительной математики |
| курс: | 1 |
| квалификация: | магистр |

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Дифференцированный зачет
- 2 (весенний) - Экзамен

Аудиторных часов: 120 всего, в том числе:

- лекции: 60 час.
- семинары: 0 час.
- лабораторные занятия: 60 час.

Самостоятельная работа: 165 час.

Подготовка к экзамену: 30 час.

Всего часов: 315, всего зач. ед.: 7

Программу составили:

- В.Е. Карпов, канд. физ.-мат. наук, доцент
- А.И. Лобанов, д-р физ.-мат. наук, профессор

Программа обсуждена на заседании кафедры информатики и вычислительной математики 04.04.2022

Аннотация

В курсе рассматриваются общие подходы к постановке параллельного численного эксперимента для математического моделирования различных процессов. Разбираются вопросы изменения парадигмы программирования в современных условиях, сравнения корректных алгоритмов для решения задач математического моделирования, однозначного представления алгоритмов при помощи их графов, декомпозиции существующих алгоритмов и программ. Описываются способы распараллеливания программ с помощью их эквивалентного преобразования без изменения алгоритмов. Теоретический материал сопровождается практикой в средах программирования MPI и OpenMP на виртуальном кластере. Основные цели курса — дать представление о подходах к распараллеливанию алгоритмов и программ и научить не бояться писать новые параллельные программы или распараллеливать существующие.

Рассматриваются классы алгоритмов численного решения задач для обыкновенных дифференциальных уравнений и нестационарных уравнений в частных производных. Анализируются возможности параллельной реализации алгоритмов.

1. Цели и задачи

Цель дисциплины

освоение студентами знаний в области применения современных высокопроизводительных комплексов различной архитектуры в научных исследованиях и прикладных областях, в частности — в математическом моделировании и обработке больших массивов данных.

Задачи дисциплины

- формирование основных знаний в области применения высокопроизводительных вычислительных комплексов различной архитектуры на основе курсов информатики, знания операционных систем, языков программирования и курсов вычислительной математики для обеспечения технологических основ математического моделирования в современных инновационных сферах деятельности;
- обучение студентов принципам создания эффективных параллельных алгоритмов и программ, анализа существующих программ и алгоритмов на параллельность; знакомство с основными методами и принципами параллельного программирования, основными технологиями параллельного программирования;
- формирование подходов к выполнению исследований студентами в области параллельных вычислений и математического моделирования с использованием современных технологий и программных средств параллельного программирования в рамках магистерских диссертационных работ.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

| Код и наименование компетенции | Индикаторы достижения компетенции |
|--|---|
| ОПК-1 Способен формулировать цели и задачи исследования, выбирать критерии оценки, выявлять приоритеты решения задач | ОПК-1.1 Имеет представление о современном состоянии исследований в рамках тематической области своей профессиональной деятельности |
| | ОПК-1.2 Способен составлять общий план работы по заданной теме, предлагать методы исследования и способы обработки результатов, проводить исследования по согласованному с руководителем плану, представлять полученные результаты |
| | ОПК-1.3 Владеет систематическими знаниями по направлению деятельности; углубленными знаниями по выбранной направленности подготовки, базовыми навыками проведения научно-исследовательских работ по предложенной теме |
| ОПК-2 Способен применять современные методы исследования, оценивать и представлять результаты выполненной работы | ОПК-2.3 Способен аргументировано выбирать способ проведения научного исследования, применять знания в области профессиональной деятельности для решения поставленной задачи, формулирования выводов и оценки полученных результатов |

| | |
|--|---|
| ОПК-3 Способен оформлять результаты научно-исследовательской деятельности в виде статей, докладов, научных отчетов и презентаций с использованием систем компьютерной верстки и пакетов офисных программ | ОПК-3.1 Владеет профессиональной терминологией, используемой в современной научно-технической литературе, обладает навыками устного и письменного изложения результатов научной деятельности в рамках профессиональной коммуникации |
| | ОПК-3.2 Владеет навыками оформления результатов научно-исследовательской деятельности в виде статей, докладов, научных отчетов и презентаций с использованием систем компьютерной верстки и пакетов офисных программ |
| ПК-1 Способен к созданию теоретических и математических моделей в области ядерной физики и технологий | ПК-1.1 Знает физическое описание явлений и процессов в области ядерной физики и технологий |
| | ПК-1.2 Умеет создавать теоретические и математические модели в области ядерной физики и технологий |
| | ПК-1.3 Владеет навыками работы с современными расчетными программными средствами |
| ПК-2 Готов применять методы исследования и расчета процессов, происходящих в современных физических установках и устройствах в области ядерной физики и технологий | ПК-2.3 Владеет навыками использования информационных технологий и пакетов прикладных программ при проектировании и расчете устройств или объектов (установок, материалов, приборов) в своей предметной области |

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- историю эволюции вычислительных систем и историческую необходимость использования параллельных вычислений;
- основы архитектуры параллельных вычислительных комплексов;
- основные технологические этапы разработки параллельных программ;
- принципы асимптотического анализа алгоритмов;
- методы декомпозиции последовательных алгоритмов;
- способы эквивалентных и неэквивалентных преобразований последовательных программ, позволяющие использовать их на параллельных вычислительных комплексах;
- основные идеи при реализации численных алгоритмов, позволяющие избежать случая низкой эффективности распараллеливания;
- основные идеи разработки эффективных параллельных алгоритмов для решения задач математической физики.

уметь:

- оценивать асимптотическую сложность используемых алгоритмов и выбирать оптимальные алгоритмы для современных программ;
- анализировать последовательные программы для выявления возможности их распараллеливания;
- оценивать эффективность работы распараллеленных программ;
- выбирать эффективные численные методы для поставленных задач математического моделирования.

владеть:

- приемами распараллеливания алгоритмов и программ;
- средствами и технологиями разработки приложений, обеспечивающих проведение параллельного вычислительного эксперимента.

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

| | | |
|--|--|---|
| | | Трудоемкость по видам учебных занятий, включая самостоятельную работу, час. |
|--|--|---|

| № | Тема (раздел) дисциплины | Лекции | Семинары | Лаборат. работы | Самост. работа |
|-----------------------|---|---------|----------|-----------------|----------------|
| 1 | Проблемы эволюции вычислительных систем. Парадигмы последовательного и параллельного программирования | 4 | | | 5 |
| 2 | Процессы в операционных системах. Знакомство с операционной системой UNIX | 2 | | 4 | 6 |
| 3 | Элементы асимптотического анализа алгоритмов | 4 | | | 5 |
| 4 | Введение в технологию программирования MPI | | | 3 | 5 |
| 5 | Декомпозиция алгоритмов на уровне операций | 2 | | | 5 |
| 6 | Укрупнение параллельных ярусов | 2 | | | 5 |
| 7 | Взаимодействие MPI-процессов | | | 4 | 6 |
| 8 | Параллельность циклов | 2 | | | 5 |
| 9 | Групповые операции в MPI | | | 4 | 1 |
| 10 | Эквивалентные преобразования программ и алгоритмов | 2 | | | 5 |
| 11 | Основные подходы к организации размещения задач на процессорах | 2 | | | 2 |
| 12 | Оркестрирование исполнения параллельных программ | 2 | | | 3 |
| 13 | Назначение и оркестровка в MPI | | | 4 | 2 |
| 14 | Нити исполнения (threads) в операционных системах. Введение в технологию программирования OpenMP | 2 | | 2 | 4 |
| 15 | Общие и приватизированные переменные в OpenMP | 2 | | 2 | 5 |
| 16 | Директивы OpenMP для распределения работы нитей исполнения | 2 | | 3 | 5 |
| 17 | Гибридное распараллеливание | 2 | | 4 | 6 |
| 18 | Исследование систем ОДУ. Ляпуновские показатели | 2 | | | 8 |
| 19 | Основные идеи параллельной реализации методов решения ЖС ОДУ | 3 | | 5 | 10 |
| 20 | Векторные и матричные операции | 2 | | 5 | 9 |
| 21 | Линейные краевые задачи. «Параллельная прогонка» и редуцированные алгоритмы | 4 | | 5 | 10 |
| 22 | Нелинейные краевые задачи | 3 | | | 10 |
| 23 | Явные разностные схемы решения уравнения теплопроводности | 3 | | 5 | 10 |
| 24 | Реализация неявных разностных схем для решения уравнения теплопроводности | 3 | | 5 | 10 |
| 25 | Реализация разностных схем для решения уравнений и систем гиперболического типа | 5 | | 5 | 13 |
| 26 | Контрольные работы | 5 | | | 10 |
| Итого часов | | 60 | | 60 | 165 |
| Подготовка к экзамену | | 30 час. | | | |

| | |
|--------------------|---------------------|
| Общая трудоёмкость | 315 час., 7 зач.ед. |
|--------------------|---------------------|

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

Семестр: 1 (Осенний)

1. Проблемы эволюции вычислительных систем. Парадигмы последовательного и параллельного программирования

Три кризиса в развитии математического обеспечения. Архитектурный и программный параллелизм. Проблемы использования параллельных систем. Парадигма последовательного программирования. Модели последовательного программирования. Парадигма параллельного программирования. Этапы декомпозиции, назначения, оркестрирования, отображения. Задачи, решаемые на каждом этапе. Модели параллельного программирования.

2. Процессы в операционных системах. Знакомство с операционной системой UNIX

Понятие процесса. Процесс и программа. Состояния процесса. Управляющий блок процесса и его

контекст. Операции над процессами. Переключение контекста.

Системные вызовы и библиотека libc. Понятия login и password. Упрощенное устройство файловой системы в UNIX. Полные имена файлов Текущая директория. Относительные имена файлов.

Домашняя директория пользователя. Команда man – универсальный справочник. Команды cd и ls.

Перенаправление стандартного ввода и стандартного вывода. Простейшие команды работы с файлами – cat, cp, mkdir, mv, rm Шаблоны имен файлов. Пользователь и группа. Команды chown и

chgrp. Права доступа к регулярному файлу и к директории. Команда chmod. Доступ к виртуальному

кластеру МФТИ. Редактирование файлов, компиляция и запуск программ. Написание и прогон последовательной программы численного вычисления значения определенного интеграла.

3. Элементы асимптотического анализа алгоритмов

Основные предположения. Вычислительная модель RAM. Терминология и обозначения.

Асимптотические отношения. Оптимальный по поведению последовательный алгоритм. Примеры

асимптотического анализа сложности последовательного алгоритмов. Рекуррентные соотношения.

Основная теорема асимптотического анализа. Расширенная квалификация Флинна. Примеры SISD,

SIMD, MISD, MIMD машин. Вычислительные модели PRAM. Ускорение при распараллеливании.

Закон Амдала. Стоимость параллельного алгоритма. Оптимальность алгоритма по стоимости. Ограниченность асимптотического анализа.

4. Введение в технологию программирования MPI

Модель программирования MPI. MPI-процессы, группы процессов, коммуникаторы. Функции MPI_Init(), MPI_Finalize(), MPI_Abort(). Функции определения ранка процесса и количества процессов в группе MPI_Comm_rank() и MPI_Comm_size(). Использование системы очередей TORQUE/PBS. Компиляция и запуск MPI-программ. Написание и прогон параллельной программы

«Hello!».

5. Декомпозиция алгоритмов на уровне операций

Понятие о графе алгоритма. Строго параллельные формы графа, каноническая параллельная форма.

Соотнесение строго параллельных форм с выполнением алгоритма на конкретных архитектурных

решениях. Ярусы параллельной формы, их ширина и высота. Концепция неограниченного параллелизма. Определение максимально возможного ускорения по ярусно-параллельной форме алгоритма.

6. Укрупнение параллельных ярусов

Декомпозиция алгоритмов и программ на уровне действий и операторов. Условия Бернштейна и их

нарушение. Истинная или потоковая зависимость, антизависимость, зависимость по выходным

данным. Графы зависимостей. Связь зависимостей операторов с возможностью их одновременного выполнения.

7. Взаимодействие MPI-процессов

Понятие point-to-point взаимодействия. Что такое MPI-сообщение. Типы данных в MPI-сообщениях. Функции MPI_Send() и MPI_Recv(). Задача на передачу информации между MPI-процессами. Параллельное численное вычисление значения определенного интеграла.

8. Параллельность циклов

Простые циклы: расстояние зависимости; зависимости, связанные и несвязанные с циклом.

Вложенные циклы. Вектора зависимости и направлений. Их использование для определения возможности распараллеливания циклов.

9. Групповые операции в MPI

Понятие групповых операций. Барьерная синхронизация – функция MPI_Barrier().

Широковещательная рассылка и редукционные операции. Функции MPI_Bcast() и MPI_Reduce().

Модификация параллельного численного вычисления значения определенного интеграла с использованием групповых операций. Сбор данных. Функции MPI_Gather() и MPI_Gatherv(). Задача на сбор рассчитанных данных.

10. Эквивалентные преобразования программ и алгоритмов

Способы устранения зависимостей, связанных с циклом: loop distribution, code replication, loop alignment, приватизация переменных, индукция и редукция.

11. Основные подходы к организации размещения задач на процессорах

Динамическое, потоковое, статическое планирование, work pool, pipeline, competition, divide & conquer. Их недостатки и достоинства. Проблемы балансировки загрузки процессоров.

Гомогенные и гетерогенные вычислительные системы.

12. Оркестрирование исполнения параллельных программ

Где и как синхронизировать вычисления и обмениваться данными. Перекрытия. Ухудшение последовательного алгоритма для улучшения параллельного. Масштабирование параллельных программ.

13. Назначение и оркестровка в MPI

Параллельное решение одномерного дифференциального уравнения в частных производных с использованием явной четырехточечной схемы.

14. Нити исполнения (threads) в операционных системах. Введение в технологию программирования OpenMP

Понятие нитей исполнения. Отличия нитей исполнения от процессов. Состояния нитей исполнения и их связь с состояниями процессов. Рождение и завершение нитей исполнения. Модель программирования OpenMP. Директивы и вспомогательные процедуры, последовательные и параллельные области программы. Директива `parallel`. Компиляция и запуск

OMP программы на отдельном компьютере. Написание и прогон OMP-программы с одной параллельной и двумя последовательными частями. Использование системы очередей TORQUE/PBS для запуска OMP-программ. Определение количества нитей и номера нити с помощью вспомогательных процедур `omp_get_num_threads()` и `omp_get_thread_num()`. Написание и прогон параллельной OMP-программы «Hello!».

15. Общие и приватизированные переменные в OpenMP

Распределение переменных на общие и приватизированные по умолчанию. Опции директив `private`, `shared`, `firstprivate`. Редукционные переменные и операции. Опция `reduction`. Низкоуровневое распараллеливание. Параллельное численное вычисление значения определенного интеграла на низком уровне в OpenMP.

16. Директивы OpenMP для распределения работы нитей исполнения

Выполнения работы в параллельной части только одной нитью. Директивы `single` и `master`. Функциональная декомпозиция. Директивы `sections` и `section`. Декомпозиция по данным. Распараллеливание циклов. Директива `for`. Параллельное решение одномерного дифференциального уравнения в частных производных с использованием явной четырехточечной схемы в среде OpenMP.

17. Гибридное распараллеливание

Совместное использование технологий MPI и OpenMP. Использование системы очередей TORQUE/PBS для гибридного распараллеливания. Гибридное распараллеливание решения одномерного дифференциального уравнения в частных производных с использованием явной четырехточечной схемы.

Семестр: 2 (Весенний)

18. Исследование систем ОДУ. Ляпуновские показатели

Математические модели динамических систем. Явные методы решения ОДУ. Явные методы Рунге-Кутты. Невозможность распараллеливания явных методов Рунге-Кутты. Проблемы качественного анализа систем ОДУ. Показатели Ляпунова, функции чувствительности к параметрам. Фундаментальная система решений (ФСР) уравнения в вариациях. Вычисление матрицы ФСР. Способы оценки показателей Ляпунова.

Алгоритмы с нормировкой векторов при подсчете показателей Ляпунова.

19. Основные идеи параллельной реализации методов решения ЖС ОДУ

Численные методы решения ЖС ОДУ. Неявные методы Рунге-Кутты. Анализ графа алгоритма. «Ослабление» неявных численных методов – методы Розенброка и W-методы. Способы распараллеливания матричных операций.

Лабораторная работа. Ускорение решения систем ОДУ с использованием систем с общей памятью. Пример – численное решение систем «умеренной» размерности (биология, химическая кинетика) с использованием технологии OpenMP.

20. Векторные и матричные операции

Векторные и матричные операции. Различные алгоритмы матричного умножения.

21. Линейные краевые задачи. «Параллельная прогонка» и редукционные алгоритмы

Алгоритм прогонки. Невозможность распараллеливания прогонки. Алгоритм Яненко–Коновалова («параллельная прогонка»). Метод редукции (четно-нечетного исключения) для решения линейной краевой задачи. Матричный вариант метода редукции. Лабораторная работа. Реализация алгоритмов Яненко–Коновалова и редукционного алгоритма на MPI.

22. Нелинейные краевые задачи

Метод стрельбы и «метод параллельной стрельбы». Квазилинеаризация и редукционные алгоритмы.

23. Явные разностные схемы решения уравнения теплопроводности

Явные методы решения задачи теплопроводности. Графы зависимостей. Характеристики уравнений. Способы распараллеливания явных методов решения. Геометрическое распараллеливание без перекрытия областей и с перекрытием областей. Динамическая схема распараллеливания. Сравнение эффективности различных вариантов распараллеливания.

Понятия о методах для решения задач с несколькими пространственными переменными.

24. Реализация неявных разностных схем для решения уравнения теплопроводности

Неявные методы решения линейного уравнения теплопроводности. Сведение к ЖС ОДУ – методы прямых. Неявные разностные схемы, метод редукции, использование алгоритма Яненко–Коновалова.

Лабораторная работа. Решение нелинейной задачи для уравнения параболического типа на примерах задачи о бегущей волне для квазилинейного уравнения (волна Самарского–Соболя–Зельдовича) или волновых взаимодействий в биологических активных средах (полулинейная система уравнений) с использованием различных подходов к распараллеливанию.

25. Реализация разностных схем для решения уравнений и систем гиперболического типа

Численные методы решения уравнений параболического типа (модельная задача). Явные схемы. Схема на шаблоне «кабаре». Сеточно-характеристические численные методы. Задача оптимизации схем в пространстве неопределенных коэффициентов. Понятия о методах для решения задач с несколькими пространственными переменными. Компактные разностные схемы. Бикompактные разностные схемы. Способы их параллельной реализации.

Лабораторная работа. Численное решение квазилинейных уравнений гиперболического типа (примеры – уравнение Хопфа, уравнение КдФ, одномерная система уравнений газовой динамики, уравнения мелкой воды)

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

Учебная аудитория, оснащенная компьютерами с доступом к виртуальному кластеру и мультимедийным оборудованием (проектор, звуковая система).

6.Перечень рекомендуемой литературы

Основная 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.
2. Технологии параллельного программирования MPI и OpenMP [Текст] : учеб. пособие для вузов / А. С. Антонов ; Моск. гос. ун-т им. М. В. Ломоносова .— М. : Изд-во Моск. ун-та, 2012 .— 344 с.
3. Карпов В. Е., Коньков К. А. Основы операционных систем: курс лекций. Издание 3-е. — М., Физматкнига, 2019. — 328 с

Дополнительная литература

1. Параллельные вычисления [Текст], учеб. пособие для вузов /В. В. Воеводин, Вл. В. Воеводин. -СПб., БХВ-Петербург, 2002
2. Вычислительная математика и структура алгоритмов [Текст] : 10 лекций о том, почему трудно решать задачи на вычислительных системах параллельной архитектуры и что надо знать дополнительно, чтобы успешно преодолевать эти трудности : учебник для вузов / В. В. Воеводин ; Моск. гос. ун-т им. М. В. Ломоносова .— 2-е изд., стереотип. — М. : Изд-во Моск. ун-та, 2010 .— 168 с.
3. Высокопроизводительные вычисления для многопроцессорных многоядерных систем [Текст] : учебник для вузов / В. П. Гергель ; Нижегород. гос. ун-т им. Н. И. Лобачевского .— М. : Изд-во Моск. ун-та : Физматлит, 2010 .— 544 с.
4. Последовательные и параллельные алгоритмы: Общий подход [Текст] : [учеб. пособие для вузов] / Р. Миллер, Л. Боксер ; пер. с англ. А. В. Козвониной ; под ред. С. М. Окулова .— М. : БИНОМ. Лаб. знаний, 2006 .— 406 с.
5. Головизнин В.М., Зайцев М.А., Карабасов С.А., Короткин И.А. Новые алгоритмы вычислительной гидродинамики для многопроцессорных вычислительных систем — М.: Издательство Московского университета, 2013 — 467 с.

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

Информационные ресурсы: <http://www.intuit.ru/>, <http://www.parallels.ru/>, www.hpc-education.ru, <https://algowiki-project.org> электронные конспекты лекций, учебные пособия и сборники задач, разработанные для данного курса.

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

На лекциях используются мультимедийные технологии, включая демонстрацию экрана.

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

Студент, изучающий дисциплину "Параллельные вычисления и алгоритмы решения дифференциальных уравнений", 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, понятия, теоремы и алгоритмы.

Успешное освоение курса требует напряжённой самостоятельной 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 конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- выполнение лабораторных работ, для осознания связей между теорией и практическими навыками;
- подготовку к дифференцированному зачету и экзамену.

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

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

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

| | |
|----------------------------|--|
| по направлению: | Ядерная физика и технологии |
| профиль подготовки: | Ядерная физика, УТС и компьютерные методы в физике Физтех-школа физики и исследований им. Ландау кафедра информатики и вычислительной математики |
| курс: | 1 |
| квалификация: | магистр |

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Дифференцированный зачет
- 2 (весенний) - Экзамен

Разработчики:

В.Е. Карпов, канд. физ.-мат. наук, доцент
А.И. Лобанов, д-р физ.-мат. наук, профессор

1. Компетенции, формируемые в процессе изучения дисциплины

| Код и наименование компетенции | Индикаторы достижения компетенции |
|--|---|
| ОПК-1 Способен формулировать цели и задачи исследования, выбирать критерии оценки, выявлять приоритеты решения задач | ОПК-1.1 Имеет представление о современном состоянии исследований в рамках тематической области своей профессиональной деятельности |
| | ОПК-1.2 Способен составлять общий план работы по заданной теме, предлагать методы исследования и способы обработки результатов, проводить исследования по согласованному с руководителем плану, представлять полученные результаты |
| | ОПК-1.3 Владеет систематическими знаниями по направлению деятельности; углубленными знаниями по выбранной направленности подготовки, базовыми навыками проведения научно-исследовательских работ по предложенной теме |
| ОПК-2 Способен применять современные методы исследования, оценивать и представлять результаты выполненной работы | ОПК-2.3 Способен аргументировано выбирать способ проведения научного исследования, применять знания в области профессиональной деятельности для решения поставленной задачи, формулирования выводов и оценки полученных результатов |
| ОПК-3 Способен оформлять результаты научно-исследовательской деятельности в виде статей, докладов, научных отчетов и презентаций с использованием систем компьютерной верстки и пакетов офисных программ | ОПК-3.1 Владеет профессиональной терминологией, используемой в современной научно-технической литературе, обладает навыками устного и письменного изложения результатов научной деятельности в рамках профессиональной коммуникации |
| | ОПК-3.2 Владеет навыками оформления результатов научно-исследовательской деятельности в виде статей, докладов, научных отчетов и презентаций с использованием систем компьютерной верстки и пакетов офисных программ |
| ПК-1 Способен к созданию теоретических и математических моделей в области ядерной физики и технологий | ПК-1.1 Знает физическое описание явлений и процессов в области ядерной физики и технологий |
| | ПК-1.2 Умеет создавать теоретические и математические модели в области ядерной физики и технологий |
| | ПК-1.3 Владеет навыками работы с современными расчетными программными средствами |
| ПК-2 Готов применять методы исследования и расчета процессов, происходящих в современных физических установках и устройствах в области ядерной физики и технологий | ПК-2.3 Владеет навыками использования информационных технологий и пакетов прикладных программ при проектировании и расчете устройств или объектов (установок, материалов, приборов) в своей предметной области |

2. Показатели оценивания компетенций

В результате изучения дисциплины «Параллельные вычисления и алгоритмы решения дифференциальных уравнений» обучающийся должен:

знать:

- историю эволюции вычислительных систем и историческую необходимость использования параллельных вычислений;
- основы архитектуры параллельных вычислительных комплексов;
- основные технологические этапы разработки параллельных программ;
- принципы асимптотического анализа алгоритмов;
- методы декомпозиции последовательных алгоритмов;
- способы эквивалентных и неэквивалентных преобразований последовательных программ, позволяющие использовать их на параллельных вычислительных комплексах;
- основные идеи при реализации численных алгоритмов, позволяющие избежать случая низкой эффективности распараллеливания;
- основные идеи разработки эффективных параллельных алгоритмов для решения задач математической физики.

уметь:

- оценивать асимптотическую сложность используемых алгоритмов и выбирать оптимальные алгоритмы для современных программ;
- анализировать последовательные программы для выявления возможности их распараллеливания;
- оценивать эффективность работы распараллеленных программ;
- выбирать эффективные численные методы для поставленных задач математического моделирования.

владеть:

- приемами распараллеливания алгоритмов и программ;
- средствами и технологиями разработки приложений, обеспечивающих проведение параллельного вычислительного эксперимента.

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

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлого занятия.

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

Смотри прикрепленный файл.

Примеры вопросов:

1. Максимальная пиковая производительность наиболее мощных современных параллельных вычислительных систем измеряется: в единицах EFLOPs, в десятках PFLOPs, в единицах PFLOPs, в сотнях TFLOPs?
2. Производительность компьютера, достигнутая при выполнении некоторой программы, выражена в TFLOPs. Это значение говорит о: среднем количестве операций над вещественными данными, представленными в форме с фиксированной запятой, выполненных за секунду в процессе обработки данной программы; общем числе команд, выполненных за время работы программы; средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой; средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой, достигнутой при выполнении данной программы; высокой реальной производительности данного компьютера.

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 зачет выставляется по результатам работы в семестре без дополнительного опроса на основе оценок за лабораторные задания и контрольную работу. Во время проведения контрольной работы обучающиеся могут пользоваться программой дисциплины и любыми рукописными материалами.

Экзаменационная (итоговая) оценка за дисциплину выставляется на основе опроса студента по теории в устной форме и результатов сдачи лабораторных работ. Студенты, сдавшие менее 20% лабораторных работ, к экзамену не допускаются.

Перечень типовых вопросов для дисциплины «Параллельные вычисления и алгоритмы решения дифференциальных уравнений»

Перечень типовых вопросов для контрольных работ в 10 семестре:

1. Максимальная пиковая производительность наиболее мощных современных параллельных вычислительных систем измеряется: в единицах EFLOPs, в десятках PFLOPs, в единицах PFLOPs, в сотнях TFLOPs?
2. Производительность компьютера, достигнутая при выполнении некоторой программы, выражена в TFLOPs. Это значение говорит о: среднем количестве операций над вещественными данными, представленными в форме с фиксированной запятой, выполненных за секунду в процессе обработки данной программы; общем числе команд, выполненных за время работы программы; средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой; средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой, достигнутой при выполнении данной программы; высокой реальной производительности данного компьютера.
3. Умножение двух квадратных плотных вещественных матриц компьютер выполнил за 5 сек с производительностью 50 GFLOPs. Какого размера были матрицы: 500*500, 1000*1000, 2000*2000, 5000*5000, 7000*7000, верного ответа нет?
4. Сколько кризисов software насчитывается за всю историю развития электронных вычислительных систем: 4, 3, 2, 1, ни одного?
5. Отметьте правильные утверждения на тему машинного представления чисел в современных ЭВМ: все существовавшие до сих пор ЭВМ используют в качестве базовой двоичную систему счисления; машинное эpsilon в основном определяется длиной мантиссы в представлении вещественных чисел; мантисса числа в двоичном представлении - та же, что и мантисса его десятичного логарифма; машинные ноль и эpsilon не могут быть получены с помощью фортран-программы, их следует найти в документации к компьютеру; машинное сложение коммутативно; машинное сложение ассоциативно; машинное умножение коммутативно; машинное умножение ассоциативно?
6. Действительные числа в машинном представлении: всегда хранятся точно; всегда хранятся с ненулевой ошибкой округления; хранились на всех существовавших вычислительных системах в двоичном представлении; имеют относительную ошибку округления не более машинного нуля; имеют абсолютную ошибку округления не более машинного эpsilon; иногда хранятся точно?
7. Отметьте правильные утверждения про компьютеры: классификация Флинна содержит 3 типа компьютеров; классификация Флинна содержит 4 типа компьютеров; классификация Флинна содержит 6 типов компьютеров; одним из признаков векторно-конвейерного компьютера является многопроцессорность; одним из признаков векторно-конвейерного компьютера является наличие хотя бы одного конвейера; одним из признаков векторно-конвейерного компьютера является наличие векторных регистров; концепция неограниченного параллелизма при развитии компьютерной техники в отдалённом будущем может стать реальностью.
8. Какие из технологических этапов присущи только парадигме параллельного программирования: построение математической модели, декомпозиция, аранжировка, написание программы?
9. Чем декомпозиция по данным отличается от декомпозиции по вычислениям?

- 10.** В чем заключаются достоинства и недостатки статического и динамического способов назначения задач виртуальным исполнителям?
- 11.** Каковы основные цели этапа назначения: сокращение загрузки исполнителей, балансировка загрузки исполнителей, сокращение обменов данными между исполнителями, равномерный обмен данными между исполнителями, сокращение накладных расходов на назначение?
- 12.** Верно ли следующее утверждение: модель передачи сообщений в параллельном программировании применима только на системах с распределенной памятью? Обоснуйте свой ответ.
- 13.** Приведите пример программного кода с внутренним параллелизмом, который, с Вашей точки зрения, нельзя распараллелить автоматически (если такие существуют).
- 14.** В чем заключаются достоинства и недостатки различных подмоделей программирования в модели общей памяти?
- 15.** На каком этапе в параллельной программе задачи назначаются реальным физическим исполнителям: на этапе декомпозиции, на этапе назначения, на этапе аранжировки, на этапе отображения?
- 16.** Современная парадигма параллельного программирования включает: 5, 6, 7, 8, 9, 10 этапов.
- 17.** Перечислите три основных принципа асимптотического анализа алгоритмов.
- 18.** Пусть $T_1(n)$ и $T_2(n)$ – времена работы последовательных алгоритмов 1 и 2 соответственно. Если $T_1(n) = O(T_2(n))$, то алгоритм 1: лучше алгоритма 2, не хуже алгоритма 2, хуже алгоритма 2, ничего нельзя сказать.
- 19.** Сформулируйте понятие оптимальности последовательного алгоритма.
- 20.** Какие из нижеперечисленных формулировок относятся к модели вычислительной системы RAM: время доступа к памяти одинаково для всех ячеек, независимо от того рассматривается операция чтения или операция записи; время выполнения всех операций на процессоре считается одинаковым; время выполнения основных операций на процессоре есть - (1)?
- 21.** Пусть два последовательных алгоритма решения одной задачи являются оптимальными по поведению. Что можно сказать о реальных временах решения при заданном параметре масштаба N : времена будут одинаковыми, ничего сказать нельзя?
- 22.** Пусть имеется два последовательных алгоритма решения одной и той же задачи — 1 и 2. Алгоритм 1 является оптимальным по поведению, алгоритм 2 не является оптимальным. Что можно сказать о реальных временах решения при заданном параметре масштаба N : время работы алгоритма 1 всегда будет меньше времени работы алгоритма 2, времена будут одинаковыми, ничего сказать нельзя?
- 23.** Используя основную теорему асимптотического анализа, оцените асимптотическое поведение алгоритма сортировки слиянием. Является ли этот алгоритм оптимальным?
- 24.** Какие схемы разрешения конфликта при разрешенной одновременной записи различными исполнителями в одну ячейку памяти в модели вычислительной системы PRAM вы знаете?
- 25.** При вычислении теоретического значения ускорения для параллельных алгоритмов решения некоторой задачи на 4-х исполнителях были получены следующие значения: 2, 3.9, 5. Какие из значений являются корректными? Почему?
- 26.** При вычислении реального значения ускорения для параллельных алгоритмов решения некоторой задачи на 4-х исполнителях были получены следующие значения: 2, 3.9, 5. Какие из значений являются корректными? Почему?
- 27.** Что означает высказывание, «некоторый параллельный алгоритм является оптимальным по стоимости»?

- 28.** Перечислите основные свойства графа алгоритма, реализованного программой.
- 29.** Сколько строгих параллельных форм может существовать у графа алгоритма: 1, ограниченное количество, но больше 1; бесконечно много?
- 30.** Докажите единственность канонической строгой параллельной формы графа алгоритма.
- 31.** Для графа алгоритма, содержащего 500 вершин, высота некоторой строгой параллельной формы равна 25. Что можно сказать о максимальном ускорении, которое может быть получено при реализации алгоритма на параллельной вычислительной системе в модели PRAM — он будет всегда меньше 20, строго равно 20, может быть больше 20?
- 32.** Для программного текста, реализованного на Фортране $b(1) = a(1) * 2$ do $i = 2, 5$ $b(i) = b(i-1) + a(i) * 2$ enddo постройте граф алгоритма, каноническую строгую параллельную форму и оцените максимально возможное ускорение при реализации алгоритма на параллельной вычислительной системе.
- 33.** Объясните понятия interleaving, недетерминированный и детерминированный набор активностей.
- 34.** Приведите пример набора активностей, для которого нарушены все условия Бернштейна, но который, тем не менее, является детерминированным.
- 35.** Если в программе встречаются два оператора, приведенных в динамическом порядке следования, $a = b+c$; $d = e+f$; то можно ли их исполнять параллельно? Обоснуйте свое утверждение.
- 36.** Перечислите виды зависимостей между операторами (блоками операторов), которые Вам известны.
- 37.** Какие из видов зависимостей по данным могут являться принципиальным препятствиям к распараллеливанию: истинная зависимость, зависимость по выходным данным, антизависимость?
- 38.** Постройте граф зависимостей по данным между операторами для фрагмента текста программы, расположенного в динамическом порядке следования:
 S1: $x = e + 2z$
 S2: $y = 2f + x$
 S3: $z = z + y$
 S4: $y = z + x$
- 39.** Постройте граф зависимостей по данным между операторами для фрагмента текста программы, расположенного в динамическом порядке следования:
 S1: $x = x + 2y$
 S2: $y = 2f + x$
 S3: $z = z + z$
 S4: $y = z + x$
- 40.** Для цикла
 do $i = 1, u$
 S1: $a(i) = d(i) + 5 * i$ S2: $c(i) = a(i-1) * 2$ enddo определите расстояние зависимости, тип зависимости и возможность распараллеливания.
- 41.** Для цикла
 do $i = 1, u$
 S1: $a(i) = d(i) + 5 * i$ S2: $c(i) = a(i+1) * 2$ enddo определите расстояние зависимости, тип зависимости и возможность распараллеливания.
- 42.** Для цикла
 do $i = 1, u$
 S1: $a(i) = d(i) + 5 * i$ S2: $c(i) = a(i) * 2$ enddo определите расстояние зависимости, тип зависимости и возможность распараллеливания.

43. С чем связаны основные проблемы определения вектора расстояний при анализе вложенных циклов на зависимость по данным?

44. Для цикла $\text{do } j1 = 1, u1 \text{ do } j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i, j - 1) + 1$

enddo enddo определите вектор расстояний, вектор направлений и тип зависимости.

45. Для цикла $\text{do } j1 = 1, u1 \text{ do } j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i, j + 2) + 1$

enddo enddo определите вектор расстояний, вектор направлений и тип зависимости.

46. Для цикла $\text{do } j1 = 1, u1 \text{ do } j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i - 1, j + 2) + 1$

enddo enddo определите вектор расстояний, вектор направлений и тип зависимости.

47. Для цикла $\text{do } j1 = 1, u1 \text{ do } j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i + 1, j + 3) + 1$

enddo enddo определите вектор расстояний, вектор направлений и тип зависимости.

48. Для цикла $\text{do } j1 = 1, u1 \text{ do } j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i, j + 2) + 1$

enddo enddo определите вектор расстояний, вектор направлений и тип зависимости.

49. Для цикла $\text{do } j1 = 1, u1 \text{ do } j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i, j) + 1$

enddo enddo определите вектор расстояний, вектор направлений и тип зависимости.

50. Для цикла

do $j1 = 1, u1$

do $j2 = 1, u2$

S1: $a(i, j) = b(i, j) * 2$

S2: $c(i, j) = a(i + 2, j - 3) + 1$

enddo enddo определите вектор расстояний, вектор направлений и тип зависимости.

51. Пусть вектор направлений для двумерного цикла имеет вид («=», «=»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

52. Пусть вектор направлений для двумерного цикла имеет вид («<», «=»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

53. Пусть вектор направлений для двумерного цикла имеет вид («<», «<»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

54. Пусть вектор направлений для двумерного цикла имеет вид («=», «<»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

55. Пусть вектор направлений для двумерного цикла имеет вид («>», «=»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

56. Пусть вектор направлений для двумерного цикла имеет вид («=», «>»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

57. Пусть вектор направлений для двумерного цикла имеет вид («>», «>»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

58. Пусть вектор направлений для двумерного цикла имеет вид («<», «>»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

59. Пусть вектор направлений для двумерного цикла имеет вид («>», «<»). Какие из следующих утверждений являются верными: цикл может быть распараллелен по внешнему индексу безо всяких ограничений, по внешнему индексу при условии дублирования входных данных, по внутреннему индексу при условии барьерной синхронизации между итерациями внешнего цикла, по двум индексам без ограничений, не может быть распараллелен?

60. Пусть для некоторого многомерного цикла вектор направлений имеет вид («=», «<», ..., «>»). Что можно сказать про тип зависимости, обнаруживаемой между операторами после раскрытия цикла: это — истинная зависимость, это — антизависимость, ничего сказать нельзя?

61. Пусть для некоторого многомерного цикла вектор направлений имеет вид («>», «<», ..., «>»). Что можно сказать про тип зависимости, обнаруживаемой между операторами после раскрытия цикла: это — истинная зависимость, это — антизависимость, ничего сказать нельзя?

62. Пусть для некоторого многомерного цикла вектор направлений имеет вид ($\langle \leftarrow, \rightarrow, \dots, \rightarrow \rangle$). Что можно сказать про тип зависимости, обнаруживаемой между

операторами после раскрытия цикла: это — истинная зависимость, это — антизависимость, ничего сказать нельзя?

63. Эквивалентное преобразование программы — это: преобразование динамического порядка следования операторов, сохраняющее результат при любых входных данных; преобразование, сохраняющее граф алгоритма; преобразование, сохраняющее зависимости по данным между операторами, ничего из вышеперечисленного.

64. Почему знание о возможности эквивалентного изменения порядка вложенности во вложенных циклах для распараллеливания программ является важным?

65. Перечислите дополнительные условия в теореме о достаточности приемов разделения цикла, выравнивания цикла и допустимой перестановки операторов в теле цикла для устранения зависимостей, связанных с циклом.

66. Что означает выражение: в цикле нет рекурсивных зависимостей по данным?

67. Справедливо ли утверждение, если между операторами цикла существует истинная зависимость — цикл по итерациям распараллелить невозможно? Обоснуйте ответ.

68. Справедливо ли утверждение, если между операторами невложенного цикла существует истинная зависимость с расстоянием зависимости равным 1 — цикл по итерациям распараллелить принципиально невозможно? Обоснуйте ответ.

69. Какие приемы распараллеливания циклов при наличии зависимости операторов цикла по скалярным переменным являются эквивалентным преобразованием программ: приватизация, «ликвидация» индукционных переменных, редукция?

70. Справедливо ли следующее утверждение: распараллеливание с помощью приема редукции всегда является неэквивалентным преобразованием программы? Обоснуйте свой ответ.

71. Справедливо ли следующее утверждение: распараллеливание с помощью приема «ликвидации» индукционных переменных всегда является неэквивалентным преобразованием программы? Обоснуйте свой ответ.

72. Можно ли устранить в последовательной программе истинные зависимости по данным для получения корректной параллельной версии?

Перечень типовых (примерных) вопросов для проведения промежуточной аттестации обучающихся во 10 семестре:

1. Какие схемы распределения работ на этапе назначения между виртуальными исполнителями Вам известны?

2. Система ОДУ не является жесткой и решается явным методом Рунге-Кутты. Будет ли реализованная для такого уравнения последовательная программа эффективно распараллеливаться?

3. Жесткая система обыкновенных дифференциальных уравнений решается W-методом заменой точного обращения матрицы приближенным по методу Шульца. Какая версия метода допускает более эффективную параллельную реализацию — с несколькими итерациями для вычисления матрицы или с учетом нескольких последовательных степеней невязки?

4. Будет ли однократно диагонально неявный метод, основанный на методе Розенброка, более эффективным с точки зрения распараллеливания по сравнению со всеми остальными методами того же класса?

5. Метод параллельной стрельбы будет более эффективным по сравнению с методом стрельбы:

А. Если число исполнителей превышает количество добавленных систем линейных дифференциальных уравнений в вариациях

Б. В любом случае, так как быстрее сходятся итерации по пристрелочным параметрам

В. Если исходная краевая задача хорошо обусловлена, и в этом случае быстрее сходятся итерации по пристрелочным параметрам

6. Есть ли смысл при реализации метода параллельной стрельбы использовать распараллеливание решения системы линейных уравнений в процессе итераций по пристрелочным параметрам?

7. При реализации параллельной пристрелки реализовано две версии метода. При первой реализации сначала решаются нелинейные уравнения, данные сохраняются, а затем решаются уравнения в вариациях, во второй реализации все эти уравнения решаются одновременно. Какая реализация более эффективна с точки зрения распараллеливания?

8. Решаются две близкие задачи – уравнение диффузии и уравнение диффузии с конвекцией. Используется неявная схема. Можно ли в каждом из этих случаев использовать алгоритм редукции для решения системы линейных уравнений с трехдиагональной матрицей?

9. Решаются две близкие задачи – уравнение диффузии и уравнение диффузии с конвекцией. Используются схемы, в которых диффузионный оператор аппроксимируется на верхнем слое по времени, а конвективные члены – на нижнем слое по времени. Можно ли в каждом из этих случаев использовать алгоритм редукции для решения системы линейных уравнений с трехдиагональной матрицей?