

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Генерация блочно-структурированных сеток для

метода конечных объемов

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 413 группы

направление 01.03.02 — Прикладная математика и информатика

механико-математического факультета

Завражнова Ильи Александровича

Научный руководитель  
доцент, к.т.н

И.А. Панкратов

Зав. кафедрой  
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2019

**Введение.** Моделирование – это замещение некоторого объекта А (оригинала) другим объектом Б (моделью). Математическая модель – это упрощенное описание реальности с помощью математических понятий. Математическое моделирование – процесс построения и изучения математических моделей реальных процессов и явлений, т.е. метод исследования объектов и процессов реального мира с помощью их приближенных описаний на языке математики – математических моделей. Крупнейшие ученые прошлого сочетали в своих трудах как построение математического описания явлений природы (математические модели), так и его исследования. Анализ усложненных моделей требовал создания новых, как правило, численных методов решения задач. Основоположником отечественного математического моделирования справедливо считают академика А.А.Самарского. Он выразил методологию математического моделирования знаменитой триадой «модель – алгоритм – программа»

1. Модель. Выбирается или строится модель исследуемого объекта, которая в математической форме отражает его важнейшие свойства. Обычно математические модели реальных процессов достаточно сложны и включают в себя системы нелинейных функционально-дифференциальных уравнений. Ядром математической модели, как правило, являются уравнения с частными производными. Для получения предварительных знаний об объекте построенная модель исследуется традиционными аналитическими средствами прикладной математики.

2. Алгоритм. Выбирается или разрабатывается вычислительный алгоритм для реализации построенной модели на компьютере, который не должен искажать основные свойства модели, должен быть адаптирующимся к особенностям решаемых задач и используемым вычислительным средствам. Проводится изучение построенной математической модели методами вычислительной математики.

3. Программа. Создается программное обеспечение для реализации модели и алгоритма на компьютере. Создаваемый программный продукт должен учитывать важнейшую специфику математического моделирования, связанную необходимостью использования набора математических моделей и многовариантностью расчетов. В результате исследователь получает в руки уни-

версальный, гибкий и недорогой инструмент, который сначала отлаживается, тестируется и калибруется на решении набора пробных задач. Затем проводится широкомасштабное исследование математической модели для получения необходимых качественных и количественных свойств и характеристик исследуемого объекта. Предложенная методология получила свое развитие в виде технологии «вычислительного эксперимента».

OpenFOAM платформа для модифицирования процессов, происходящих в телах и конструкциях. Она позволяет моделировать процессы.

Целью данной бакалаврской работы является изучение и генерация блочно-структурированных сеток для метода конечных объемов. Для этого будет использована платформа OpenFOAM.

Бакалаврская работа состоит из введения, шести разделов, заключения и списка использованных источников. В первом разделе бакалаврской работы описываются какие бывают разновидности платформ моделирования. Во втором разделе описаны основные аспекты платформы OpenFOAM для численного моделирования. В третьем разделе рассматривается генерация сетки с помощью утилиты blockMesh. В четвертом разделе рассматривается подробная инструкция для создания blockMesh и его генерация. В пятом разделе описываются способ формирования и генерации новой сетки при помощи языка программирования python. В шестом разделе описана новая генерация сетки и ее примеры использования. Список использованных источников содержит 20 наименований на которые в тексте приведены ссылки.

**Разновидности платформ моделирования.** Для решения практических задач вычислительной гидроаэродинамики (ВГАД) часто используются коммерческие пакеты прикладных программ. К числу таких пакетов относятся ANSYS CFX, ANSYS FLUENT (оба пакета с некоторого времени поставляются вместе — в составе ANSYS CFD), STAR-CD, FlowVision и др. Эти программные средства позволяют решать весьма разнообразные и сложные задачи, которые не ограничиваются лишь моделированием течений жидкости или газа, а зачастую требуют междисциплинарного подхода. Неудивительно, что такое программное обеспечение находит применение в самых разных областях науки и техники и насчитывает огромное количество пользователей.

Во многом этому способствует доступный для понимания интерфейс и сравнительная легкость обучения работе с пакетом.

Несмотря на эти достоинства, коммерческие прикладные пакеты имеют серьезные недостатки. Одним из них является высокая стоимость лицензий на использование, а также последующую поддержку. Часто немалых средств стоит даже ограниченная версия продукта, возможности которой значительно сокращены по сравнению с полноценной. Более того, реальные задачи во многих случаях не позволяют получить решение за адекватное время на персональном компьютере, требуя больших вычислительных мощностей. Выходом из этой ситуации является применение высокопроизводительных вычислительных систем — многопроцессорных кластеров. Однако стоимость лицензий для использования коммерческих пакетов на таких системах очень высока и зачастую сравнима со стоимостью аппаратного обеспечения, на котором предполагается их развертывание.

Еще одним недостатком (не столь ощутимым для инженера, в отличие от исследователя) представляется закрытость исходного кода коммерческих пакетов. Это сужает возможности для более детального изучения реализованных в пакете математических моделей и алгоритмов, а также их модификации и усовершенствования.

Среди перечисленных свободно распространяемых пакетов наиболее широко известен OpenFOAM. Он используется во многих инженерных и научных приложениях как коммерческими, так и учебными и научно-исследовательскими организациями. Этот пакет находит применение при решении задач разнообразных областей механики сплошной среды — от гидроаэродинамики, включая учет процессов горения, протекания химических реакций, турбулентности и тепломассообмена, до механики деформируемого твердого тела и электромагнетизма. Постоянно увеличивается число пользователей пакета OpenFOAM, а также растет число публикаций, связанных с его применением. Тем не менее, в нашей стране пакет OpenFOAM известен очень узкому кругу пользователей.

Далее в работе основное внимание уделено именно пакету OpenFOAM как получившему наиболее широкое распространение из свободно распространяемых пакетов для решения задач ВГАД. На примере использования

OpenFOAM рассмотрим методику решения задач ВГАД в прикладных пакетах.

**Платформа для численного моделирования OpenFOAM.** OpenFOAM — свободно распространяемый инструментальный вычислительный гидродинамики для операций с полями (скалярными, векторными и тензорными). Является одним из «законченных» и известных приложений, предназначенных для вычислений методом конечных объемов (finite volume method). Код OpenFOAM, изначально разрабатываемый в Великобритании компанией OpenCFD, Limited, в настоящее время поддерживается и развивается усилиями некоммерческой организации The OpenFOAM Foundation.

OpenFOAM состоит из двух частей: библиотеки классов, решающих дифференциальные уравнения в частных производных, которые необходимы при численном моделировании, и библиотеки программ-«решателей» (solvers), использующих эти классы для решения конкретных задач моделирования. Пакет OpenFOAM реализован в рамках объектно-ориентированного подхода на языке C++, что обеспечивает полную инкапсуляцию, иначе разделение, процессов построения сетки для дискретизации сплошной среды и методов аппроксимации уравнений в частных производных, алгебраических уравнений. Также отдельным блоком выступают различные утилиты необходимые для обработки данных.

OpenFOAM всегда работает в 3 мерной декартовой системе координат и все геометрические конфигурации производятся в 3 измерениях. OpenFOAM решает примеры в 3 измерениях по умолчанию, но может быть проинструктирован решать в 2 измерениях, определяя специальное empty (пустое) граничное условие на границах для которого никакое решение проводить не требуется.

Входные данные, относящиеся к контролю времени, чтению и записи данных решения считываются из controlDict, который расположен в директории system. Выбор ограниченного объема схем дискретизации определяется в fvSchemes. Спецификация решателей линейных уравнений (linear equation solvers) и задание точности, а также других алгоритмов контроля, определяется в fvSolution в директории system.

**Генерация сетки с помощью утилиты blockMesh.** Генерация сетки состоит из несколько блоков которые отвечают за работоспособность файла blockMeshDict.

Ключевое слово	Описание
convertToMeters/масштаб	Коэффициент масштабирования
vertices/вершины	Список координат вершин
edges/стыки	Используется для описания дуг или края сплайнов
block / блок	Упорядоченный список меток вершин и размер сетки
patches/ патч	Список патчей
mergepatchpars	Список патчей для объединения

**Вершины.** Вершины блоков сетки приведены далее в виде стандартного списка именованные vertices

vertices

```
(
    ( 0    0    0 )    // vertex number 0
    ( 1    0    0.1)  // vertex number 1
    .....
);
```

**Края.** Каждое ребро, соединяющее 2 точки вершины, по умолчанию считается прямым. Однако любое ребро может быть задано как изогнутое записями в списке с именем ребра . Список является необязательным; если геометрия не содержит криволинейных прямых, это может быть опущено.

Каждая запись для изогнутой прямой начинается с ключевого слова, указывающего тип кривая из перечисленных в соответствии с таблицей:

Подбор ключевых слов	Описание
arc/дуга	Круговая дуга
simpleSpline	Сплайновая кривая
polyLine/ полилиния	Набор линий
BSpline/ полисплайн	Набор сплайнов
line/линия	Прямая линия

**Блоки.** Определения блоков содержатся в списке с названием `blocks`. Каждое определение блока является составной записью, состоящей из списка меток вершин, порядок которых описывается , задается количество ячеек, требуемое в каждом направлении, тип и список коэффициента расширения ячеек в каждом направлении.

Определение каждого блока следующее:

**Нумерация вершин :** Первая запись является идентификатором формы блока, как определено в файле *FOAM\_ETC – 6/cellModels*. Форма всегда hex так как блоки всегда гексаэдры. Далее следует список вершин номера, упорядоченные в порядке .

**Количество клеток :** Вторая запись дает количество ячеек в каждой из  $x_1, x_2$  и  $x_3$  указания для этого блока.

**Коэффициенты расширения ячеек :** Третья запись дает коэффициенты расширения ячеек для каждого направление в блоке. Коэффициент расширения позволяет сетка быть рассортированным, или уточненный, в определенных направлениях. Отношение ширины конца ячейка  $\delta_e$  вдоль одного края блока до ширины начальной ячейки  $\delta_s$  вдоль этого края. `simpleGrading` : простое описание определяет равномерные расширения в направления  $x_1, x_2, x_3$  .

**Типы блоков** Граница сетки задается в списке с именем `boundary`. Граница разбиты на патчи (области), где каждый патч в списке имеет свое имя как ключевое слово, которое выбирает пользователь, хотя мы рекомендуем что-то, что удобно определяет заплату, например вход; имя использовано как идентификатор для задание граничных условий в файлах данных поля. Информация о патче затем содержится в под-словаре:

- `type` : тип патча, либо общий патч, на котором некоторые границы применяются условия или конкретное геометрическое условие

-faces : список граней блоков, которые составляют патч и имя которых это выбор пользователя, хотя мы рекомендуем что-то, что удобно определяет заплату, например вход ; имя использовано как идентификатор для задания граничных условий в файлах данных полей.

**Несколько блоков.** Сетка может быть создана с использованием более 1 блока. В таких обстоятельствах сетка создается как было описано в предваряющем тексте; только дополнительная проблема заключается в связи между блоками, в которых есть два различных возможности:

**Сопрягать стороны :** набор граней, которые составляют патч из одного блока сформированный из того же набора вершин, что и набор граней патч, который составьте патч из другого блока;

**Слияние граней :** группа граней из заплаты из одного блока соединены к другой группе граней из патча из другого блока, чтобы создать новый набор внутренних граней, соединяющих два блока.

Для соединения двух блоков с **сопоставлением граней** , два патча, которые формируют подключение должно просто игнорироваться из списка исправлений. blockMesh тогда определяет, что грани не образуют внешнюю границу и объединяет каждую совмещенная пара в единую внутреннюю грань, соединяющую ячейки из двух блоки.

Альтернативный вариант-слияние граней-требует слияния блоков сначала определяются в списке патчей. Каждая пара патчей, чьи лица должны быть после этого слияние должно быть включено в дополнительный список mergePatchPairs . Этот формат mergePatchPairs:

**Генерация сетки с помощью пакетов python.** В этом разделе написано то, как мы генерируем написанную программу в файл, с помощью библиотеки python - PyFoam.

PyFoam - эта Python-библиотека может использоваться для :

- Анализа журналов производства OpenFoam-solvers.
- Выполнение OpenFoam-решателей и утилит и одновременный анализ их вывода.
- Управление файлами параметров и начальными условиями выполнения неразрушающим образом.



Эта библиотека была разработана для управления OpenFOAM-моделированием с многошерным языком скриптов для анализа параметров и результатов.

Это постоянная работа. Можно добавлять функции по мере необходимости.

Эта библиотека пытается быть максимально консервативной в библиотеках, которые она использует. Это означает, что он пытается использовать только библиотеки, которые находятся в стандартном Python-дистрибутиве.

**Новая генерация сетки.** В данной бакалаврской работе была поставлена задача разработать генерирование сетки файла blockMeshDict с помощью программы написанной на языке программирования python. Была написана программа которая выводит параллелепипед. Этот код генерируем в файл с название blockMeshDict, который будем заменять. С помощью пакета PyFoam внедряем его в наш проект.

Для того что бы это реализовать нам нужно построить параллелепипед с одинаковой размерностью по всем осям , выбрать начальную точку с которой будем начинать отсчет. Пусть этой точкой будет левый нижний угол нашего параллелепипеда. Далее начинаем создавать точки которые будем задавать как начальная координата плюс шаг. Повторяем эту операцию 7 раз что бы получить все исходные точки. Далее формируем файл blockMeshDict зная все координаты точек данного параллелепипеда. После чего делаем стандартные операции и отображаем наш параллелепипед в ParaFoam.

Далее с помощью этого параллелепипеда строим похожие на него параллелепипеды. Для этого потребовалось создать функции для перемещения этого параллелепипеда в разных координатных областях. У каждого нового параллелепипеда были свои координаты, своя размерность. Это было получено следующим образом: у нас есть исходный параллелепипед к нему по оси OX прибавляем ко всем 8 точкам любое число больше чем размер нашего исходного параллелепипеда. Далее формируем обход нового параллелепипеда. После чего делаем стандартные операции и отображаем наши полученные параллелепипеды на экран с помощью ParaFoam. Так же делаем те же самые операции для каждой из осей (OY и OZ). Получили 4 параллелепипеда с одинаковой размерностью.

Для того что бы соединить все эти 4 параллелепипеда нужно поменять размер удаляемой по осям координат. Если размерность всех параллелепипедов одинаковая то они будут соединяться автоматически, так как координаты будут одинаковыми и точки будут совпадать.

**Заключение.** В данной бакалаврской работе был изучен и сгенерирован файл `blockMeshDict`. Для решения поставленных задач была использована платформа OpenFOAM. Сделана программа для генерации блочно-структурированной сетки для метода конечных объемов на языке Python. Был сгенерирован новый файл `blockMeshDict` на языке программирования Python.