МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра математического и компьютерного моделирования

Математическое моделирование движения тел методом конечных объемов

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 413 группы

направления 01.03.02 — Прикладная математика и информатика

Механико-математический факультет

Дьяченко Егора Олеговича

Научный руководитель
доцент, к.т.н. И.А.Панкратов

Зав. кафедрой
зав.каф., д.ф.м.н. Ю. А. Блинков

Саратов 2019

Введение. В современных реалиях решение реальных задач связанных с обтеканием тела вязкой жидкостью встречаются достаточно часто. Для решения реальных задач часто прибегают к их представлению в несколько упрощенном виде при помощи моделирования с использованием простых тел, таких как, например, конус. Для ещё большего упрощения решения расчеты производятся при помощи специализированного программного обеспечения, а конкретнее пакетов прикладных программ, таких как ANSYS CFX, ANSYS FLUENT (оба пакета с некоторого времени поставляются вместе — в составе ANSYS CFD), FlowVision, FORTRAN, STAR_CD,STAR-CCM+, OpenFOAM. Такие пакеты чаще всего используют численные методы. Методы вычислительной гидрогазодинамики возникли в семидесятых годах двадцатого века. В частности, возник и достиг высокого уровня эффективности метод конечных объемов (МКО).

В качестве инструмента моделирования и получения решения в данной работе пакет OpenFOAM.

Целью работы является изучение движения тела простейших форм под ламинарным потоком вязкой жидкости с применением DynamicMeshDict.

В первом разделе рассмотрен пакет прикладных программ OpenFOAM. Во втором разделе рассмотрена система уравнений Навье-Стокса и их векторная форма записи. В третьем разделе рассмотрен метод конечных объемов на примере стандартного уравнения баланса некой величины в контрольном объеме. В четвертом рассмотрен принцип работы используемого в решении задачи решателя pimpleFOAM. В пятом разделе проводится наглядная проверка описанных в четвертом разделе принципов работы решателя pimpleFOAM. В шестом разделе рассмотрено влияние коэффициента кинетической вязкости жидкости на стандартны пример из туториала. В седьмом разделе проводится изучение влияния DynamicMeshDict на проведенное в шестом разделе моделирование. В конце приводится список использованных источников, состоящий из 21-го, и список приложений, содержащий программный код файла fvSolution, а так же графики зависимости скорости и давления для различных решателей и моментов времени моделирования.

Платформа для численного моделирования OpenFOAM. OpenFOAM — свободно распространяемый инструментарий вычислитель-

ной гидродинамики для операций с полями (скалярными, векторными и тензорными).

Для того, чтобы новая платформа была более гибкой и мощной, чем FORTRAN, в качестве языка программирования был выбран C++ из-за его модульности и объектно-ориентированных возможностей.

Первоначально программа предназначалась для прочностных расчетов, но в результате многолетнего академического и промышленного развития на сегодняшний момент позволяет решать множество различных задач механики сплошных сред (не ограничиваясь ею), в частности:

- Прочностные расчеты;
- Гидродинамика ньютоновских и неньютоновских вязких жидкостей как в несжимаемом, так и сжимаемом приближении с учетом конвективного теплообмена и действием сил гравитации. Для моделирования турбулентных течений возможно использование RANS-моделей, LES- и DNS методов. Возможно решение дозвуковых, околозвуковых и сверхзвуковых задач;
- Задачи теплопроводности в твердом теле;
- Многофазные задачи, в том числе с описанием химических реакций компонент потока;
- Задачи, связанные с деформацией расчетной сетки;
- Сопряженные задачи;
- Некоторые другие задачи, при математической постановке которых требуется решение дифференциальных уравнений в частных производных в условиях сложной геометрии среды;
- Распараллеливание расчета для запуска на многопроцессорных системах (в том числе кластерных).

В основе кода лежит набор библиотек, предоставляющих инструменты для решения систем дифференциальных уравнений в частных производных как в пространстве, так и во времени. Рабочим языком кода является C++. В терминах данного языка большинство математических дифференциальных и тензорных операторов в программном коде (до трансляции в исполняемый файл) уравнений может быть представлено в удобочитаемой форме, а метод дискретизации и решения для каждого оператора может быть выбран уже пользователем в процессе расчетов. Таким образом, в коде полностью инкап-

сулируются и разделяются понятия расчетной сетки (метод дискретизации), дискретизации основных уравнений и методов решения алгебраических уравнений.

Вместе с кодом поставляется набор программ-решателей, в которых реализованы различные математические модели механики сплошных сред.

Моделируемый случай включает данные для сетки, полей, свойств, параметров контроля решения, и т.д. В OpenFOAM эти данные хранятся в ряде файлов в пределах директории с примером, а не в единственном файле примера, как во многих других пакетах вычислительной гидродинамики (CFD).

Для решения поставленной задачи также придется использовать DynamicMeshDict для контроля деформации и морфинга (визуального эффекта, создающего впечатление плавной трансформации объекта) сетки во время моделирования. Этот словарь необходим только для решателей, которые вызывают движение сетки. Например, рассматриваемый и используемый в данной работе решатель pimpleFoam. [6]

Вместе с OpenFOAM также поставляется ParaView — открытый графический кросс-платформенный пакет для интерактивной визуализации в исследовательских целях, разрабатываемый Национальной Лабораторией Сандиа, компанией Kitware и Национальной Лабораторией Лос-Аламоса.

Для решения некоторых задач в Open FOAM необходимо использовать файлы формата stl. Для их создания используется SALOME — открытая интегрируемая платформу для численного моделирования. Представляет собой набор пре- и постпроцессинга [8].

Уравнение Навье-Стокса. Уравнения Навье-Стокса получаются путем применения второго закона механики к сплошной среде [13]. Получаем систему из трех уравнений в трехмерной декартовой прямоугольной системе координат [14].

Запишем данную систему в векторном виде.

$$\rho \cdot \frac{dV}{dt} = \rho \cdot \vec{F} - \nabla p + (\mu' + \mu)\nabla div\vec{V} + \mu \cdot \Delta\vec{V} \tag{1}$$

В общем случае уравнение (1) учитывает теплопроводность. То есть коэффициенты вязкости μ, μ' — это функции, зависящие от температуры. Однако,

если взять $\mu, \mu' = const$, то уравнение существенно упрощается, так коэффициенты вязкости можно свободно вытаскивать из под знака производной. Однако, если температурное поле сильно менялось с течением времени, такое упрощение приведет к сильному расхождению с реальным результатом.

Если обезразмерить уравнение (1), то перед вязкими членами появляется множитель:

$$\frac{1}{Re}, \tag{2}$$

где Re — число Рейнольдса:

$$Re = \frac{d \cdot |U|}{\nu},$$

где d — характерная размерность, $|U|$ — модуль характерной скорости, ν — кинетическая вязкость [15].

Соответственно при больших числах Рейнольдса множитель (2) будет сильно уменьшаться, что вызовет быстрое изменение функции. Из-за этого движение становится турбулентным, а решение неустойчивым.

Метод конечных объемов. Рассмотрим основные положения метода конечных объемов, рассматривая стандартное уравнение баланса некой величины ϕ (например, кинетической силы турбулентности) в контрольном объеме Ω, ограниченном поверхностью $S = \sum S_k$ с внешней нормалью $\vec{n}$:

$$\int\limits_{\Omega} \frac{\partial(p \cdot \phi)}{\partial t} d\Omega + \sum_k \int\limits_{S_k} \vec{n} \cdot \vec{q} ds = \int\limits_{\Omega} Q d\Omega, \vec{q} = p \cdot \vec{V} \cdot \phi - \alpha \cdot \nabla \phi, \tag{3}$$

где $\vec{q}$ — вектор плотности потока ϕ с конвективной и диффузионной составляющими, Q — плотность распределения объемных источников, $\vec{V}$ — вектор скорости, p — плотность среды, α — коэффициент диффузии.

Отметим, что при стягивании объема до точки, можно на основании формулы Остроградского-Гаусса записать уравнение (3) в дифференциальной форме:

$$\frac{\partial(p \cdot \phi)}{\partial t} + \nabla \cdot \vec{q} = Q.$$

Для дискретизации уравнения (3) необходимо вычислить входящие в него интегралы при помощи квадратурных форм.

Для вычисления $\vec{q_e}$ для начала рассмотрим конвективную составляющую. Расход жидкости через грань ячейки $g_e = \vec{S_e} \cdot (p \cdot \vec{V})_e$ вычисляется при помощи аппроксимации уравнения неразрывности. При известном расходе расчет конвективной составляющей потока ϕ через грань сводится к определению ϕ_e. Для этого можно использовать линейную интерполяцию вдоль сеточной линии по двум узловым значениям, ϕ_P и ϕ_E.

Аппроксимация диффузной составляющей потока не требует мер для обеспечения устойчивости. Таким образом необходимо только вычислить:

$$\vec{n} \cdot \nabla\phi \equiv \frac{\partial\phi}{\partial n}$$

в центре грани.

Для этого рассчитаем значения $\nabla\phi$ в центрах ячеек через интегральное представление градиента:

$$\int_\Omega \nabla\phi d\Omega = \int_S \vec{n} \cdot \phi dS \Rightarrow (\nabla\phi)_P \approx \frac{1}{\Omega} \cdot \sum_k \vec{S_k} \cdot \phi_k. \tag{4}$$

После для расчета $\nabla\phi$ в уравнении (4) применим линейную интерполяцию.

Принцип работы решателя pimpleFOAM. Для решения поставленной задачи будет использоваться решатель pimpleFOAM, предназначен для решения задач в условиях потока несжимаемой жидкости с использованием большого шага по времени. Имеет опцию для моделирования ламинарного потока. Использует PIMPLE алгоритм для решения уравнения неразрывности и уравнения импульса.

Стандартный пример данной задачи расположен в каталоге:

```
FOAM_TUTORIALS/incopressible/pimpleFoam/laminar/movingCone
```

Для изучения принципа работы данного решателя, изучим файл fvSolutions (приложение А), расположенный в каталоге:

```
FOAM_TUTORIALS/incopressible/pimpleFoam/laminar/movingCone/System
```

В файле в блоке с названием PIMPLE задаются параметры вычисления, в отсутствие которых решение будет осуществляться с параметрами по умолчанию. В этом случае будет использоваться алгоритм PISO. Однако, данный алгоритм имеет существенный недостаток в виде ограничения:

$$Co < 1, \tag{5}$$

где Co — число Куранта:

$$Co = \frac{\delta t \cdot |U|}{\delta x},$$

где δt — шаг по времени, $|U|$ — скорость потока через ячейку, δx — размер ячейки в направлении скорости, так как при изменении вязкости жидкости также будет меняться и ее скорость [19].

Соблюдение условия (5) избавляет от ошибок, но и требует больших вычислительных мощностей, что делает решения в реальном времени очень дорогостоящими. Чтобы ускорить моделирование необходимо преодолеть данное условие. Этого можно достичь используя алгоритм SIMPLE, разработанный для конечного, устойчивого состояния.

В данной задаче используется объединенный объединенный PISO-SIMPLE алгоритм.

Значение $nOuterCorrectors = 2$ запускает цикл PIMPLE состоящий из двух шагов для пересчета связи давление-импульс за один временной шаг.

Сначала идет построение импульсной матрицы:

$$\frac{\partial U}{\partial t} + \nabla \cdot (UU) + \nabla \cdot R = -\nabla p$$

Далее, используя данную матрицу, строится матрица давления:

$$\nabla^2 p = f(U, \nabla p)$$

Находим давление p_{new}. Корректируем скорость $U_{correctored}$ с новым полем давления.

Повторяем цикл с $p = p_{new}$, $U = U_{correctored}$ [20].

Для корректной работы алгоритма необходимо достичь плавной скорости сходимости. Для этого после блока PIMPLE в файле fvSolution есть блок relaxationFactors для расчета между отдельными временными шагами. Без блока relaxationFactors алгоритм PISO-SIMPLE по сравнению с алгоритмом PISO отличается лишь повышенной скоростью накопления ошибок.

Проверка принципа работы решателя. Сначала проверим работу решателя при настройках по умолчанию.

Для этого удалим блок relaxationFactors и содержимое блока PIMPLE в файле fvSolution. Исходя из принципа работы решателя, описанного в предыдущей части работы, соотношение (5) должно не выполниться. Тогда после вызова консольной команды, инициирующей запуск решателя, «pimpleFOAM» в консоли будет уведомление об ошибке, а так же будет указано время остановки решения.

Однако, после вызова paraview ошибки не будет. То есть, решатель произведет вычисления, однако только до указанного в консоли времени.

Действительно, при попытке инициировать решение, в консоли выводится ошибка, а так же время остановки решателя $t = 0.000005$, то есть не смог сделать даже второй шаг по времени.

Так же при попытке запустить симуляцию в paraview, график значений U и p в соответствии с рисунком Б.1 остается неизменным относительно начального момента времени $t = 0$.

Теперь проверим влияние на решатель блока relaxationFactors. Восстановим блок PIMPLE. На этот раз решатель должен проработать дольше.

Действительно, после вызова консольной команды «paraFoam» совершение вычислений занимает больше времени. Теперь решатель смог проработать до момента $t = 0.00119$.

Значение скорости U на графиках в соответствии с рисунком Б.2 и рисунком Б.3 практически не изменяется. А вот значение давление p в правой части трубы на графике решателя с блоками PIMPLE и relaxationFactors в соответствии с рисунком Б.3 ниже, в то время как на графике решателя без блока relaxationFactorsв соответствии с рисунком Б.2 оно практически не меняется, за исключением середины.

Таким образом можно сделать вывод, что без блока relaxationFactors алгоритм PISO-SIMPLE хоть и смог прорешать дольше алгоритма PISO, однако из-за ограничения на число Куранта (5) все еще не способен дорешать задачу полностью, что не дает возможность сравнить изображения с решателем с блоками PIMPLE и relaxationFactors, так как ошибка препятствует выводу изображений в paraview.

Так же стоит отметить, что количество итераций также влияет на итоговую точность решения. Для этого достаточно сравнить значения первой и второй итераций для стандартного примера.

Стоит так же отметить, что повысить число итераций можно увеличив значение nCorrectors в блоке PIMPLE файла fvSolution. В этом случае поле давления исправляется дважды, а связь между давлением и импульсом достаточно стабильна, чтобы создать неразрывное решение и избежать несоблюдения условия, наложенного на число Куранта (5) [21].

Влияние коэффициента вязкости. Рассмотрим два случая: для стандартного значения коэффициента кинетической вязкости жидкости $\nu = 0.0001$ и для $\nu = 0.00001$.

При большей вязкости давление и скорость имеют более равномерное распределение по области, а при меньшей вязкости в жидкости куда активнее возникают и протекают турбулентные течения, хотя при этом сама скорость течения жидкости ниже из-за более низкого давления.

Поведение сетки во время моделирования. DynamicMeshDict, а точнее используемый в работе dynamicMotionSolverFvMesh решатель, изменяет сетку вокруг заданного набора границ. Движение сетки рассчитывается на основе давления на этих границах. Решатель обеспечивает обратную связь для моделирований связанных с жидкостью. Он изменяет граничные условия скорости U на включенных границах для указания локальной скорости, включающей поступательные и вращательные движения (если разрешено). Такой контроль сетки практически всегда используется для решения задач, связанных с движением твердого тела.

Рассмотрим работу DynamicMeshDict на проведенных в шестом разделе вычислениях.

Из результатов следует, что динамическое изменение сетки во время моделирования позволило не только увеличивать точность вычислений в областях, где нужна была высокая точность, и уменьшать ее там, где высокая точность не требовалась, освобождая таким образом ЭВМ от лишних вычислений, но и глобально увеличить точность за счет адаптивного изменения размеры и формы ячеек сетки.

Заключение. В бакалаврской работе были изучены численные методы, их применение в рамках задач гидродинамики.

Были смоделированы процессы обтекания тел простейших форм ламинарным потоком вязкой несжимаемой жидкостью на примере конуса с использованием DynamicMeshDict.

Было установлено, что при большей вязкости давление и скорость имеют более равномерное распределение по области, а при меньшей вязкости в жидкости куда активнее возникают и протекают турбулентные течения, хотя при этом сама скорость течения жидкости ниже из-за более низкого давления. Однако вблизи стенки высокая инерция жидкости с большей кинетической вязкостью имеет куда более активные турбулентные движения.

На этом примере также были рассмотрены алгоритмы решателя pimpleFoam и различные способы его применения. Было установлено, что PISO имеет существенный недостаток из-за ограничения на число Куранта, из-за нарушения которого решатель начинает взрываться. Однако гибридный алгоритм PISO-SIMPLE имеет возможность игнорировать данное ограничение, что влечет за собой ускорение реализации работы решателся pimpleFoam и делает более пригодным для решения задач в реальном времени.

Так же было установлено, что в динамическое изменение сетки позволяет более эффективно распределять ресурсы ЭВМ между областями, а так же увеличивает точность вычислений за счет адаптивного изменения не только размеров, но и формы ячеек сетки.