МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра Математического и компьютерного моделирования

Моделирование движения связанных плавающих объектов

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 413 группы

направление 01.03.02 — Прикладная математика и информатика

механико-математического факультета

Шамаева Евгения Владимировича

Научный руководитель
старший преподаватель В.С. Кожанов

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

Саратов 2019

Введение. Движения соединенных плавающих структур часто изучается в гидродинамическом анализе применительно к кораблестроению. Предлагаемый проект фокусируется на реализации в моделирования движений двух связанных плавающих объектов. В проекте реализованы две основные задачи. Первая реализуют движение двух несвязанных плавающих объектов с шестью степенями свободы (6DOF). Вторая - моделирует 6DOF движение двух плавающих объектов соединённых линейной пружиной между ними.

Работа выполнена на платформе OpenFOAM. Решатель interDyMFoam используется в двух случаях. Данные случаи являются модификацией из руководства по floatingObject. Несвязанный случай реализуется путем введения двух модулей - плавающих объектов и использования решателя движения сетки displacementLaplacian. Связанный случай выполняется путем разработки нового класса типа ограничения couplingLinearSpring, на основе существующего класса ограничений linearSpring.

Чтобы лучше проиллюстрировать эти случаи, размер каждого плавающего тела немного уменьшен по сравнению с размером объекта в flooatingObject. Масса каждого объекта в представленных случаях уменьшается до 4 кг, по сравнению с массой в 15 кг в учебном пособии. Исходный размер объекта, введенного в руководстве, составляет 0,3м×0,2м×0,5м. В нашем случае размер каждого объекта уменьшается до 0,2м×0,2м×0,2м, а расстояние между этими двумя объектами равно 0,4 м. Длина рассматриваемой области удваивается по сравнению с со случаев рассмотренным в руководстве, чтобы сохранить два плавающих объекта. Ширина и глубина не изменяются.

Учебники и инструкции некоторых других случаев движения тела с 6DOF с interDyMFoam и potentialFreeSurfaceDyMFoam можно ознакомиться из [1], [2] и [3].

Целью работы является проведение математического моделирования движения связанных и несвязанных плавающих объектов.

В первом разделе выполняется постановка задачи о движении несвязанных и связанных плавающих объектов.

Второй раздел посвящен моделированию задач механики твердых тел в OpenFOAM, рассмотрены решатели sixDoFRigidBodyMotion и interDyMFoam.

В третьем разделе приведены результаты, анализ моделирования динамики несвязанных плавающих тел в жидкости.

Четвертый раздел разработана модификация решателя sixDoFRigidBodyMotion для моделирования динамики связанных тел.

В пятом разделе приведены результаты, анализ моделирования динамики связанных плавающих тел в жидкости.

В заключении сделаны общие выводы по результатам полученных в работе.

Постановка задачи о движении несвязанных и связанных плавающих объектов. Метод и алгоритм решения задачи движения твердого тела в жидкости. Рассматривается моделирование 2 плавающих в жидкости тел, которые могут быть несвязанные и связанные между собой. Моделирование задачи осуществляется при помощи программы OpenFoam используя решатели sixDoFRigidBodyMotion и interDyMFoam. В случае когда 2 тела связанные между собой, изучаются различные типы соединений, включая шарнирное соединение, пружинное соединение, демпферное соединение, угловое пружинное соединение или цепное соединение или другие смешанные соединения. Некоторые типы соединений также определяются при описании взаимосвязи между плавающими структурами и неподвижными объектами. Например, упругая линия часто используется в качестве линии швартовки, которая связывает плавающие объекты и бочку или морское дно. В работе показаны разные типы соединений между двумя плавающими объектами.

Моделирование задач механики твердых тел в OpenFOAM, решатели sixDoFRigidBodyMotion и interDyMFoam. OpenFOAM — свободно распространяемый инструментарий вычислительной гидродинамики для операций с полями (скалярными, векторными и тензорными). Является одним из законченных и известных приложений, предназначенных для FVM-вычислений.

Копируем floatingObject в каталог run. Переименуем его как uncoupledFloatingObjects.

В исходном случае используется blockMesh с длиной, шириной и глубиной 1м×1м×1м. В нашем случае длина blockmesh увеличивается до 2 метров, а

размер сетки не меняется. Для дополнительного объекта добавляется еще одна граница. BlockMeshDict модифицируется следующим образом.

Модули плавающего объекта не содержат никаких отличительных частей в библиотеке blockMesh, потому что утилита topoSet будет использоваться для определения параметров модулей.

В OpenFOAM topoSet словарь topoSetDict используется для выбора ячеек или граней в простых геометриях. Плавающая коробка определяется функцией boxToCell. Чтобы задействовать один дополнительный плавающий блок, следует использовать в качестве альтернативы два словаря toposet. Приложение topoSet и приложение subsetMesh запускаются дважды в отношении разных модулей. Здесь subsetMesh используется для сетки на выбранной части, выполняемой topoSet. Во время этого процесса имена topoSet, c1 и c2, заменяются floatObject1 и floatingObject2. Они станут именами модулей.

Два словаря topoSet модифицируются на основе topoSetDict.

Здесь object1topoSet - это информация topoSet для плавающего объекта 1 и object2topoSet определяет topoSet для плавающего объекта 2.

Таким образом, плавающий объект 1 расположен в отрицательном направлении x, а плавающий объект 2 находится в положительном направлении x. Следующие команды могут визуализировать расположение двух плавающих объектов и наборов сетки.

Файл dynamicMeshDict - это словарь, который определяет динамический решатель сетки вместе с библиотеками и коэффициентами, которые будут использоваться.

В руководстве применяется решатель sixDoFRigidBodyMotion для решения одиночного объекта. DynamicMeshDict также содержит sixDoFRigidBodyMotionCoeffs, включая центр масс, массу, момент инерции и ограничения. В руководстве модуль CodeStream применяется для расчета массы и момента инерции.

Параметры innerDistance и outerDistance являются точками, удаленными от модулей объекта. Они используются для расчета коэффициента масштабирования в поле, где масштаб равен 1 до внутреннего расстояния и линейно уменьшается до 0 на внешнем расстоянии. Определения можно найти в sixDoFRigidBodyMotionSolver.C следующей командой.

```
vi $FOAM_SRC/sixDoFRigidBodyMotion/sixDoFRigidBodyMotionSolver\
/sixDoFRigidBodyMotionSolver.C
```

Однако SixDoFRigidBodyMotion не поддерживает движение нескольких тел. Модифицируем код в sixDoFRigidBodyMotionSolver.C.

Таким образом, смещение точки в поле изменяется только на основе преобразования координат одной жесткой системы. Вместо этого Displacement Laplacian motion решетель может использоваться для имитации множественных и независимых движений плавающих объектов.

DisplacementLaplacianFvMotionSolver - это динамический решеточный механизм, который решает уравнение Лапласа для смещения клеток.

В работе принимается модель диффузи inverseDistance. В соответствии с [4] и [5] модель inverseDistance вычисляет коэффициент диффузии внутреннего поля на основе обратного расстояния от границы. На этом этапе пользователь может указать одну или несколько границ.

Таким образом, в этом случае выбраны две границы плавающих объектов, чтобы рассчитать коэффициент диффузии для решателя displacementLaplacian.

Поскольку библиотека sixDoFRigidBodyMotion скомпилирована отдельно, и fvMotionSolver не включает эту библиотеку для вычисления движения тела 6DOF, необходимо добавить libVviMotionSolvers и libsixDoFRigidBodyMotion в motionSolverLibs.

В каталоге 0.org задается ряд граничных условий. Файлы каталога включают alpha.water, epsilon, k, nut, p , U и pointDisplacement. Поскольку вводятся два модуля объекта, информация о модулях должна обновляться в каждом файле. С помощью команд sed имена модулей изменяются на floatObject1 и floatingObject2. Здесь параметры граничного условия для каждого модуля остаются такими же, как в руководстве, за исключением файла pointDisplacement. Файл pointDisplacment будет представлен в следующем разделе.

Для использования решателя движения сетки displacementLaplacian, исходный pointDisplacement определяется вместе с правилами движения модели объекта.

Поле плавающего объекта определяется как тип calculated. Это примитивный тип, который означает, что граничное поле выводится из других полей. Поскольку sixDoFRigidBodyMotionSolver не используется в dynamicMeshDict, производные граничные условия движения 6DOF должны использоваться в модифицированном случае. Здесь тип sixDoFRigidBodyDisplacement используется для плавающих объектов, которые выравнивают точку модели на границе в соответствии с алгоритмом sixDoFRigidBodyMotion.

Чтобы применить этот тип sixDoFRigidBodyMotion, необходимо определить массу, момент инерции, центр масс, ограничения и ограничения свойств плавающих объектов. Ограничения лимитируют некоторые степени свободы движения, такие как ограничение вращения вокруг точек и ограничение движения по определенной оси или плоскости. Ограничения не ограничивают степень свободы движения тела, но налагают ограничительную силу для ограничения амплитуды или скорости движения. В этом случае применяется только ограничение вращения вокруг фиксированной точки.

SetFieldsDict используется для точной настройки утилиты setFields. В этом случае setFields задает начальную полевую информацию о распределении воды и воздуха.

Командами boxToCell устанавливаем глубину воды до 0,5 м и устанавливаем объем воды dambreak на одной стороне резервуара с высотой воды 0,6 м.

Моделирование динамики несвязанных плавающих тел в жидкости: результаты, анализ. Движение для несвязанного случая показаны в работе. Результаты отображаются с проекцией Y-направления.

Модификация решателя sixDoFRigidBodyMotion для моделирования динамики связанных тел. В библиотеке libsixDoFRigidBodyMotion различные соединения между плавающим объектом и неподвижным объектом предоставляются с ограничениями. Доступные типы удерживающих устройств в OpenFOAM - inearAxialAngularSpring, linearSpring, sphericalAngularSpring, tabulatedAxialAngularSpring, linearDamper и sphericalAngularDamper. Поскольку они не поддерживают связь между двумя движущимися телами, некоторые модификации должны выполняться на основе существующего типа ограничений.

Каталог linearSpring содержит исходный файл типа линейного пружинного соединения. Сила ограничений рассчитывается по следующим строкам.

```
r = restraintPosition - anchor_;
magR = mag(r);
r /= (magR + VSMALL);
restraintForce = -stiffness_*(magR-restLength_)*r-damping_*(r & v)*r;
```

Сила, вызванная жесткостью, пропорциональна удлинению пружины, а сила затухания связана со скоростью объекта. Жесткость данных двух элементов и затухание в классе linearSpring являются скалярами. Стационарное положение якоря (anchor) и точка привязки объекта (refAttachmentPt) - это данные с двумя точками. Скалярными данными restLength$member.refAttachmentPt$.

Чтобы связать два объекта вместе, точка привязки одного объекта обновляется в каждом цикле. Эта точка такая же, как и restrainPosition от другого объекта. Следовательно, информация должна передаваться между модулями. Это ключевой момент для решения этой проблемы.

Новая библиотека mysixDoFRigidBodyMotion, которая содержит новый класс connectLinearSpring, будет создана на основе sixDoFRigidBodyMotion и класса linearSpring.

Скопируем класс linearSpring и переименуем его в connectionLinearSpring. Затем заменим все строки linearSpring для couplingLinearSpring как и в именах файлов, так и внутри файлов.

В файле files удаляем исходный файл классов ненужных ограничений и добавим новый класс в список файлов. Затем установим библиотеку в каталог пользователя.

Затем используем wclean для подготовки к дальнейшим шагам.

В исходном случае объект является только твёрдым телом, и легко установить его ограничения. Но для случая с несколькими объектами важно идентифицировать главный объект и другие связанные с ним объекты. Давая им имена тел, легче указать, какие тела связаны. Для двух связанных плавающих объектов этот процесс можно выполнить, добавив данные элемента rigidBodyName и connectedBodyName.

Модификацию можно запустить с помощью sixDoFRigidBodyMotion.H.

Изменим данные и добавим два имени.

Перейдем к разделу sixDoFRigidBodyMotionI.H с помощью

```
cd $WM_PROJECT_USER_DIR/src/mysixDoFRigidBodyMotion/sixDoFRigidBodyMotion
vi sixDoFRigidBodyMotionI.H
```

и добавим информацию о названиях тел.

Введем тип vi sixDoFRigidBodyMtion.C, чтобы изменить файл sixDoFRigidBodyMotion.C. Конструкторы SixDoFRigidBodyMotion.C должны содержать атрибуты имен тела.

Функция status () из класса sixDoFRigidBodyMotion требует небольшого обновления для вывода имени rigidBody. Это помогает идентифицировать информацию о твердом теле в файлах.

Информация о связи должна быть добавлена к функции write () в sixDoFRigidBodyMotionIO.C.C, чтобы включить обновление словарей в каждом временном цикле. Используем vi sixDoFRigidBodyMotionIO.C для редактирования файла.

SixDoFRigidBodyMotion использует член функции addRestraints для применения ограничения движения. Тем не менее, ограничения используют дополнительный словарь для поиска коэффициентов. Это означает, что информация об ограничениях может быть задействована только в одном объекте.

В этом случае привязка первого объекта происходит из положения ограничения второго объекта, положение ограничения первого объекта становится привязкой второго объекта. Данные необходимо обменивать между двумя модулями. Для решения проблемы могут применяться OFstream и IFstream. Следует отметить, что OFstream и IFstream - это не лучший способ решить проблему. Должны быть способы обмена информацией между модулями в файле словаря ограничений. Из-за ограниченного времени работы, поток OFstream и IFstream является только альтернативой. Существует множество ограничений в отношении этого метода.

Перейдем к предварительно установленному соединению класса LinearSpring.C.

Некоторые классы Foam и стандартные классы должны быть включены в заголовок.

После restraintMoment = vector :: zero; где вычисление силы удерживания завершено, три компонента положения удерживания выводятся в файл restrainPositionOutput *. В отчете о движении информация о закрепленной позиции может быть записана для проверки того, перемещается ли закрепление.

Идея использования трех файлов для хранения положения ограничения заключается в том, чтобы избежать сложного преобразования данных. Точечный тип, который получается из векторного типа, может быть легко экспортирован, но его сложно построить при чтении с использованием типа строки. Общий способ построения векторного типа состоит в том, чтобы собрать три элемента из стандартных типов, таких как float и double или из скалярного типа Foam. Функция strtod используется для получения числа с плавающей точкой из строки.

Когда выполняются модификации, перейдем в каталог, в котором находится папка Make. Используем wmake libso для компиляции новой библиотеки.

Скопируем предыдущий проект и переименуем. Очистим предыдущий проект перед установкой новой конфигурации командой ./Allclean.

Основная модификация находится в 0.org/pointDisplacement, атрибуты плавающих объектов должны быть добавлены с именами и ограничениями тела.

Значения привязки определены, но при выполнении программы они будут заменены. Необходимо отметить, что жесткость, амортизатор и длина покоя двух ограничений на объектах должны быть точно такими же.

В библиотеки constant/dynamicMeshDict разрешений движения должны быть обновлены.

Чтобы дать начальные значения двух закреплений и избежать ошибок чтения, файл Allrun добавляется с начальными позициями после приложения = 'getApplication' и перед runApplication blockMesh.

В Allclean информация о закреплении может быть удалена, чтобы сделать следующий прогон более удобным.

Моделирование динамики связанных плавающих тел в жидкости: жидкости: результаты, анализ. Движение для связанного случая

показано в соответствии с рисунками в работе. Результаты отображаются с проекцией Y-направления.

По сравнению с несвязанным случаем при t = 1,0 с и 1,5 с можно видеть, что ограничения лимитируют амплитуду движения. Взглянем на файл журнала interDyMFoam, информация привязки изменяется во время итераций.

Согласно журналу, удерживающие силы двух объектов различны, но физически они всегда должны быть одинаковыми. Причина в том, что программа не вычисляет движение модулей одновременно. Позиция привязки первого объекта относится к предыдущему циклу PIMPLE или циклу времени, тогда как второй объект использует положение удерживания первого объекта в текущем цикле PIMPLE или в цикле времени.

Другим фактором, который может вызвать несбалансированную силу, является затухание. Поскольку модификация в этом отчете не включает ввод / вывод информации о скорости, часть силы демпфирования будет отличаться от каждого объекта. Предлагается полагать коэффициент затухания как 0.

Заключение. Работа направлена на реализацию несвязанного или связанного движения двух плавающих объектов. В модификации связанного случая движения IFstream и стандартный поток C++ используются для достижения обмена данными между различными модулями.