

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра Математического и компьютерного моделирования

**Модификация стандартного решателя OpenFOAM
при моделировании течения жидкости в колене трубы**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы

направление 01.03.02 — Прикладная математика и информатика

механико-математического факультета

Салтыкова Никиты Александровича

Научный руководитель
доцент, к.т.н., доцент

И.А. Панкратов

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

Ю.А. Блинков

Саратов 2023

Введение. Данная бакалаврская работа посвящена модификации стандартного решателя OpenFOAM при моделировании течения жидкости в колене трубы. В ней рассматривается решатель isoFoam — решатель для течения несжимаемой вязкой жидкости, находящийся в классе задач incompressible — турбулентное течение несжимаемой жидкости. В работе рассмотрен стандартный пример течения жидкости в колене трубы и то, какие математические формулы и физические законы лежат в основе стандартного решателя isoFoam.

Представлены результаты расчетов после внедрения внутрь колена трубы некоторого тела, построенного в OpenSCAD. Приведено сравнение результатов до и после внедрения тела.

А также проведена модификация стандартного решателя таким образом, что коэффициент вязкости был не постоянным, а переменным. Приведено сравнение результатов модифицированного решателя без фигурки и с фигуркой внутри.

Моделирование течения жидкости в колене трубы, является актуальной задачей в настоящее время. Это связано с тем, что такие конструкции широко используются в различных инженерных приложениях, например, в трубопроводах для нефтяной и газовой промышленности, системах отопления и кондиционирования воздуха, системах охлаждения электроники и т.д.

Модификация решателя течения жидкости показывает, как будет меняться движение жидкости при переменной вязкости. Это может позволить оптимизировать конструкцию и рассчитать параметры течения, такие как расход, давление и скорость потока, что в свою очередь поможет улучшить эффективность и безопасность работы системы, а также снизить затраты на ее эксплуатацию.

Целью работы является моделирование течения жидкости в колене трубы. Для достижения цели были поставлены следующие задачи:

- Изучение работы стандартного решателя.
- Определение математических формулы и физических законов, лежащих в основе стандартного решателя isoFoam.
- Модификация стандартного решателя.
- Анализ и сравнение полученных данных.

Моделирование течения жидкости в колене трубы с модифицированным решателем позволяет увидеть, что будет происходить с конструкцией, и как будут вести себя параметры течения при различных ситуациях, например, когда вязкость жидкости будет увеличиваться с каждой секундой времени.

Структура бакалаврской работы. В данной бакалаврской работе содержится введение, 4 раздела и заключение:

- **Во введении** рассматривается актуальность темы бакалаврской работы и краткое описание самой работы.
- **В первом разделе** работы вводятся используемые уравнения движения жидкости.
- **Второй раздел** посвящен разбору стандартного примера.
- **В третьем разделе** рассматривается модифицированный решатель, и проводится сравнение его работы с работой стандартного решателя.
- **В четвертом разделе** проводится внедрение тела внутрь фигурки.
- **В заключении** подводятся итоги работы.

Основное содержание работы. Решатель isoFoam решает ламинарные уравнения Навье-Стокса для несжимаемой жидкости с использованием алгоритма PISO (неявное давление с разделением операторов). Код по своей сути является переходным, требующим начального условия (например, нулевой скорости) и граничных условий. Код isoFOAM может учитывать неортогональность сетки с помощью последовательных итераций неортогональности. Количество поправок PISO и поправок на неортогональность управляется пользовательским вводом.

Описание движения сплошной среды в декартовой системе координат описывается следующими уравнениями

$$\rho \left(\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} + v_z \frac{\partial v_x}{\partial z} \right) = \frac{\partial p_{xx}}{\partial x} + \frac{\partial p_{xy}}{\partial y} + \frac{\partial p_{xz}}{\partial z} + \rho F_x$$

$$\rho \left(\frac{\partial v_y}{\partial t} + v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} + v_z \frac{\partial v_y}{\partial z} \right) = \frac{\partial p_{yx}}{\partial x} + \frac{\partial p_{yy}}{\partial y} + \frac{\partial p_{yz}}{\partial z} + \rho F_y$$

$$\rho \left(\frac{\partial v_z}{\partial t} + v_x \frac{\partial v_z}{\partial x} + v_y \frac{\partial v_z}{\partial y} + v_z \frac{\partial v_z}{\partial z} \right) = \frac{\partial p_{zx}}{\partial x} + \frac{\partial p_{zy}}{\partial y} + \frac{\partial p_{zz}}{\partial z} + \rho F_z$$

где $\rho(x, y, z, t)$ – плотность сплошной среды $v_x(x, y, z, t)$, $v_y(x, y, z, t)$, $v_z(x, y, z, t)$ – проекции скорости среды, p_{ij} – компоненты тензора напряжений, $F_x(x, y, z, t)$, $F_y(x, y, z, t)$, $F_z(x, y, z, t)$ – компоненты вектора массовой плотности объёмных сил, действующих на сплошную среду.

Уравнение несжимаемости

$$\nabla \cdot \bar{v} = 0. \quad (1)$$

Векторный вид уравнения движения Навье-Стокса для жидкости записывается следующим образом:

$$\frac{\partial \bar{v}}{\partial t} = -(\bar{v} \nabla) \bar{v} + \nu \Delta \bar{v} - \frac{1}{\rho} \nabla p + \bar{f} \quad (2)$$

где ∇ – оператор Набла, Δ – векторный оператор Лапласа, t – время, ν – коэффициент кинематической вязкости ρ – плотность, p – давление, v – скорость, $\bar{v} = (v^1, \dots, v^n)$ – векторное поле скорости, \bar{f} – векторное поле массовых сил. Неизвестные p и \bar{v} являются функциями времени t и координаты $x \in \Omega$, где $\Omega \subset \mathbb{R}^n$, $n = 2, 3$ – плоская или трёхмерная область, в которой движется жидкость.

Для того, чтобы запустить стандартный решатель, необходимо перейти по следующему пути `/opt/openfoam10/tutorials/incompressible/icoFoam/elbow` и желательно скопировать оттуда все файлы в отдельную папку в домашней директории.

В файле `system/controlDict` немного уменьшим `writeInterval`, с 20 до 1, чтобы увеличить количество записанных шагов, сделав отображения изменений более подробным.

Также для большей наглядности, в файле `/0/U` были изменены некоторые значения: значение внутренней скорости по оси x (`velocity-inlet-5`) было изменено с 1 на 3 м/с, внутренняя скорость по оси y (`velocity-inlet-6`) была изменена с 3 на 12 м/с.

В соответствии с рисунком 1, приведен пример работы стандартного решателя `icoFOAM` при течения жидкости в колене трубы.

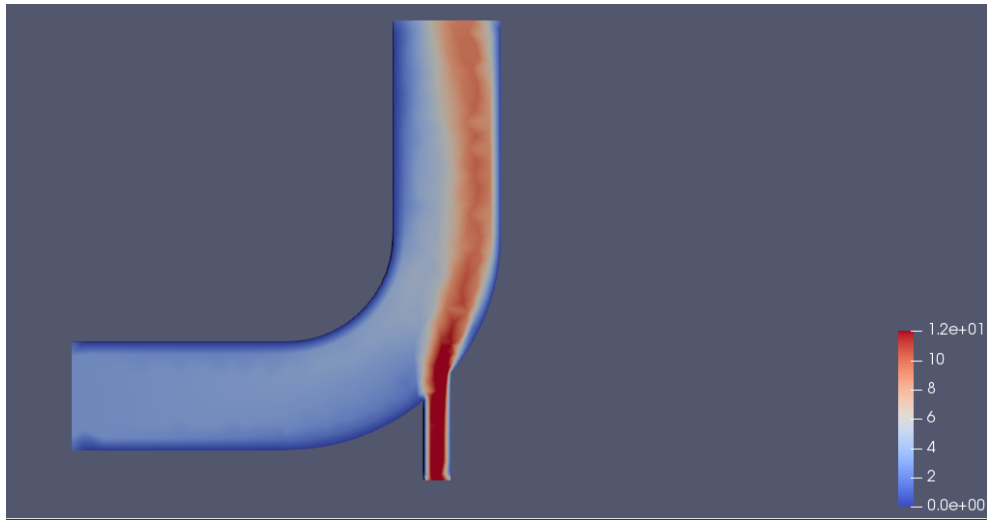


Рисунок 1 — Пример работы стандартного решателя

Для модификации нашего решателя перейдем по пути `/opt/openfoam10/applications/solvers/incompressible/icoFoam`. Скопируем оттуда все файлы и перенесем в папку нашего примера. Отдельно для этих файлов создадим папку `icoFoamTime`, а также внутри этой папки создадим папку `bin`.

Это мы перенесли файл написанный на языке программирования C, который создает стандартный решатель. А изменив его, мы можем создать новый – модифицированный. Для этого перейдем в файл `icoFoam.C` и на 64 строке изменим функцию `s`

```
$fvM::laplacian(nu, U)$
```

на

```
$fvM::laplacian(nu*(1+runTime.elapsedCpuTime()*100), U)$
```

Функция

```
runTime.elapsedCpuTime()
```

запускает отсчет времени использования ЦП и возвращает время, прошедшее с момента запуска. Таким образом, наше значение вязкости теперь будет увеличиваться с каждым шагом, от 0.01 до $0.01 \cdot (1 + 200 \cdot 100) = 200.01$.

Значение функции `runTime.elapsedCpuTime()` меняется от 0 до 200 т.к. в файле `controlDict` у нас указано

```
endTime 10;
```

```
deltaT 0.05;
```

Теперь для того, чтобы создать решатель, который мы модифицировали, нужно перейти в папку Make, открыть файл files и изменить значение EXE на путь ~/название папки/icoFoamTime/bin/icoFoamTime.

После того, как мы это сделали, в консоли необходимо необходимо перейти в папку bin по вышеуказанному пути, прописать wmake – команда создаст наш решатель. А после можно запустить его, прописав путь до нашего решателя в папке bin и в дальнейшем добавить его в Allrun.

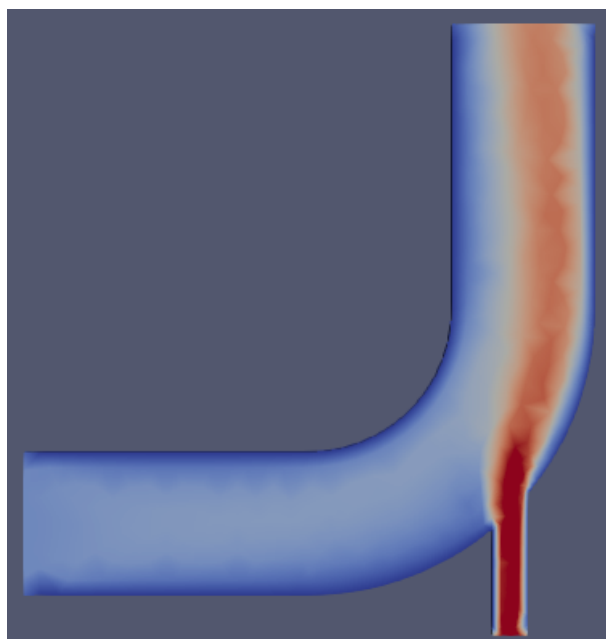


Рисунок 2 — Пример работы модифицированного решателя

В соответствии с рисунком 2, приведен пример работы модифицированного решателя, на котором можно заметить, что жидкость повела себя другим образом. Скорость в пристыкованной трубе постоянная и равна 12 м/с, как мы и задавали. А вот при попадании жидкости из этой трубы в "колесо" можно заметить, что скорость начинает падать. В соответствии с рисунком 1 мы можем заметить, что скорость течения жидкости до модификации при попадании в колесо трубы тоже начинает падать, но не так быстро. Также можно заметить, что в соответствии с рисунком 1, течение жидкость сместилось вправо, ближе к стене, а в соответствии с рисунком 2 оно сместилось чуть меньше.

Для добавления тела в нашу конструкцию, необходимо создать в папке /constant папку geometry, в которой будет находиться наше тело в виде файла формата stl.

Само тело мы строим в программе OpenScad используя данный код

```
translate ([62, 62, 0.]) {  
    sphere (5, $fn=100);  
}
```

translate отвечает за перемещение объекта, т.е. мы его перемещаем на 62 по оси x и y, и на 0 по оси z т.к. у нас двумерный случай. Тело создается командой *sphere(5, \$fn = 100)*; как понятно из названия, создается сфера радиуса 5 и количеством фрагментов для рисования дуги = 100. Чем меньше значение параметра \$fn, тем более квадратная фигура получается.

Также в папке system, нам нужен файл snappyHexMeshDict это генератор сетки, который берет уже существующую сетку и превращает ее в желаемую сетку. Там указывается то, как будет строиться наша фигурка. И вдобавок к нему идет файл meshQualityDict. Он отвечает за общие настройки качества сетки.

В папке 0/p нужно указать граничные условия для нашего тела.

```
body  
{  
    type          zeroGradient;  
}  
}
```

И точно так же в файле 0/U. Финальным штрихом в папке constant создаем файл boundary и прописываем там следующий код.

```
FoamFile  
{  
    format        ascii;  
    class         polyBoundaryMesh;  
    location      "constant/polyMesh";  
    object        boundary;  
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   * * * * * * * * * * //

(
  wall-4
  {
    type          wall;
    inGroups      List<word> 1(wall);
    nFaces        100;
    startFace     1300;
  }
  velocity-inlet -5
  {
    type          patch;
    nFaces        8;
    startFace     1400;
  }
  velocity-inlet -6
  {
    type          patch;
    nFaces        4;
    startFace     1408;
  }
  pressure-outlet -7
  {
    type          patch;
    nFaces        8;
    startFace     1412;
  }
  wall-8
  {
    type          wall;
    inGroups      List<word> 1(wall);
  }
)
```



```

        nFaces          34;
        startFace       1420;
    }
    frontAndBackPlanes
    {
        type             wall;
        inGroups         List<word> 1(wall);
        nFaces           1836;
        startFace        1454;
    }
)

```

Далее, нужно прописать в файле Allrun после построения геометрии такую команду: `cp -fv constant/boundary constant/polyMesh/boundary`

Она заменяет файл в папке `constant/polyMesh/boundary` на файл `constant/boundary`. Это сделано для того, чтобы мы могли добавить тело в колено трубы.

Файл Allrun будет выглядеть следующим образом

```

#!/bin/sh
cd ${0%/*} || exit 1    # Run from this directory
./Allclean
# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
# Get application directory
application=$(getApplication)

runApplication fluentMeshToFoam elbow.msh
cp -fv constant/boundary constant/polyMesh/boundary
snappyHexMesh -overwrite
~/Diplom_fig/icoFoamTemp/bin/icoFoamTime
runApplication foamMeshToFluent
runApplication foamDataToFluent

```

Запускать программу через Allrun необязательно, можно прописывать каждый пункт вручную. В качестве текстового редактора использовалось приложение "Kate", т.к в нем есть встроенный терминал.

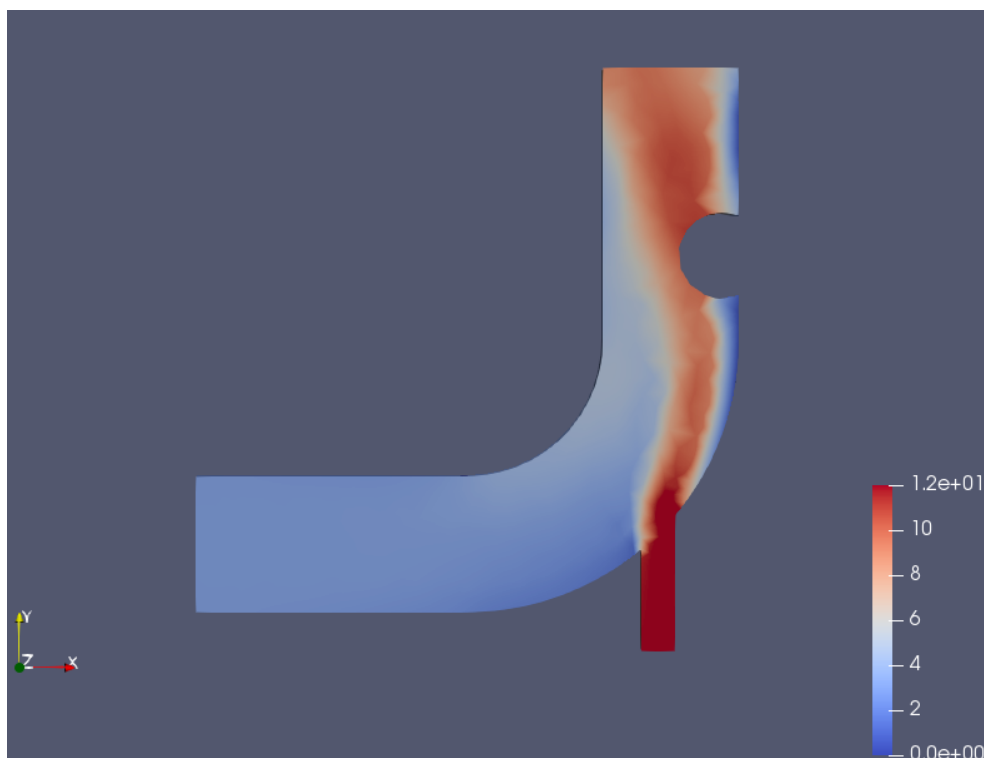


Рисунок 3 — Пример работы стандартного решателя с телом внутри.

В соответствии с рисунком 3 приведен пример работы стандартного решателя с телом внутри. Можно заметить, что после внедрения тела жидкость также замедляется по началу, а после того, как встретит фигурку, обходит её, немного ускоряется и снова замедляется.

Заключение. В данной работе был рассмотрен процесс моделирования течения жидкости в колене трубы в OpenFOAM. В процессе решения также был модифицирован стандартный решатель и проведено небольшое сравнение полученных данных для разных случаев. Получены графики движения течения.