

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Романова Ирина Сергеевна

Должность: Ректор

Дата подписания: 20.07.2023 17:18:25

Уникальный программный ключ:

63119ed-2b582ffdacea443f01d5779368d0957ac34f5cd074d81181530452479

**Гидродинамическое
моделирование на OpenFOAM:
учебное пособие
для самостоятельного изучения**

УДК 378.147.88
ББК 30.123
Г19

Р. М. Ганопольский, А. Я. Гильманов. Гидродинамическое моделирование на OpenFOAM: учебное пособие для самостоятельного изучения.

Учебное пособие содержит необходимый материал для самостоятельного изучения платформы OpenFOAM, среды SALOME и программы ParaView. Эти программные продукты помогают осуществить полный цикл гидродинамического моделирования – препроцессинг, расчёт и постпроцессинг. Материал выстроен от самого простого к сложному. После выполнения всех заданий читатель сможет выполнить собственный проект. Учебное пособие предназначено для студентов физико-технических и механико-математических направлений бакалавриата и магистратуры, а также тем, кто хочет самостоятельно освоить принципы гидродинамического моделирования.

Содержание

Предисловие.....	4
Занятие №1.....	6
Занятие №2.....	21
Занятие №3.....	47
Занятие №4.....	57
Занятие №5.....	76
Занятие №6.....	86
Занятие №7.....	94
Занятие №8.....	119
Занятие №9.....	136
Занятие №10.....	142
Список возможных ошибок.....	163
Алгоритмы моделирования.....	168
Перечень возможных тем самостоятельных проектов.....	171
Список литературы.....	174

Предисловие

Главное назначение учебного пособия – знакомство с основными этапами гидродинамического моделирования: подготовка геометрии и расчетной сетки, компьютерная симуляция, обработка полученных результатов. В качестве прикладной платформы моделирования используется комплекс программного обеспечения OpenFOAM. Геометрия и расчетная сетка создаются в среде SALOME, результаты компьютерной симуляции обрабатываются в программе Paraview. Данные программные продукты распространяются бесплатно. В первом занятии приведены инструкции по установке и настройке необходимого программного обеспечения в операционной системе Windows.

Весь учебный материал в пособии представлен в виде занятий. В каждом занятии описаны последовательно все действия для достижения результата. Каждое из действий подробно проиллюстрировано. В последних занятиях даны задания для самостоятельного выполнения: приведена только постановка задачи. Последовательность занятий в пособии выстроена от простейших к сложным. После выполнения всех работ читатель освоит основные принципы и этапы гидродинамического моделирования, познакомится с программным обеспечением для компьютерной симуляции физических процессов и анализа результатов. По окончании всех занятий желательно выполнить собственный проект. Структура пособия позволяет выстроить проектную работу.

Учебное пособие будет полезным студентам бакалавриата и магистратуры различных направлений, связанных с вычислительной физикой, а также для самостоятельного изучения всем желающим.

Занятие №1

Тема: Установка BlueCFD и SALOME, знакомство с командной строкой Linux, знакомство с интерфейсом ParaView и SALOME.

Этап 1. Установка BlueCFD.

Для установки BlueCFD зайдите на <http://bluecfid.github.io/Core/Downloads/> и скачайте последнюю версию. В ходе установки, также будут установлены Notepad и Gnuplot, необходимые для работы.

После установки запустите BlueCFD – Core terminal. При запуске через ярлык на рабочем столе могут возникать ошибки, в таком случае для правильной работы программы необходимо запустить её следующим образом:

1. Создайте папку на диске C: и назовите её, используя только цифры и латинские буквы, без пробелов (допустим только символ нижнего пробела «_») (например, C:/laba_1).

Примечание: при создании файла не используйте пробелы и русские буквы.

2. Кликнув по ней правой кнопкой мыши, выберите «Open in blueCFD-Core terminal».
3. Выполнить команду **icoFoam** для проверки правильности установки.

Примечание: все команды в OpenFOAM начинаются с маленькой буквы, однако каждое последующее слово в названии команды начинается с большой буквы.

4. Запустите пакет для визуализации ParaView с помощью команды **paraFoam**. Если возникает ошибка и вызов ParaView невозможен, найдите самостоятельно на компьютере пакет ParaView и запустите его. В таком случае файлы будут необходимо для визуализации открывать уже в пакете ParaView.

Этап 2. Установка SALOME

Для установки SALOME зайдите на <https://www.salome-platform.org/downloads/previous-versions/salome-v8.3.0#binaries-for-windows>. Скачается самораспаковывающийся архив, после чего

найдите в распакованной папке папку WORK и запустите в ней файл run_salome.bat.

Если при запуске возникнет окно с сообщением о недостающей библиотеке, установите бесплатный распространяемый пакет Microsoft Visual C++ 2010. Скачать его можно здесь: <http://dlltop.ru/soft/46-microsoft-visual-c>.

Этап 3. Основные команды в Linux.

Основными командами являются ls (она же dir) и cd. Чтобы вызвать команду, нужно ввести её в окне BlueCFD – Core Terminal после приветственного символа «\$» и нажать клавишу Enter.

cd – смена рабочей директории:

'cd' (без аргументов) – меняет директорию на домашний каталог

'cd -' – переход в предыдущую директорию и печать ее имени.

'cd c:' – перейти на диск C.

'cd /laba_1' – перейти в /laba_1 (в Linux используется символ «/»).

'cd p' и нажатие клавиши «Tab» без нажатия «Enter» – показывает в командной строке папку на букву «p», если такая папка одна, для последующего перехода в эту директорию после нажатия «Enter». Если папок на букву «p» несколько, то будут выведены названия всех папок на эту букву в следующей строке. Возможно также написание не одной буквы, а сразу нескольких, в таком случае будут выведены все папки, имеющие идентичное начало, либо в командной строке будет показана единственная папка, имеющая такое начало.

'cd ..' – вернуться в предыдущую директорию.

ls – выдаёт список содержимого директории:

'ls -a' – подробный список, включая скрытые файлы (имена которых начинаются с точки) (рис. 1).

'ls constant' – показывает содержимое папки «constant».

Ввод одной или нескольких букв после 'ls' работает аналогично команде cd (см. пример 'cd p').

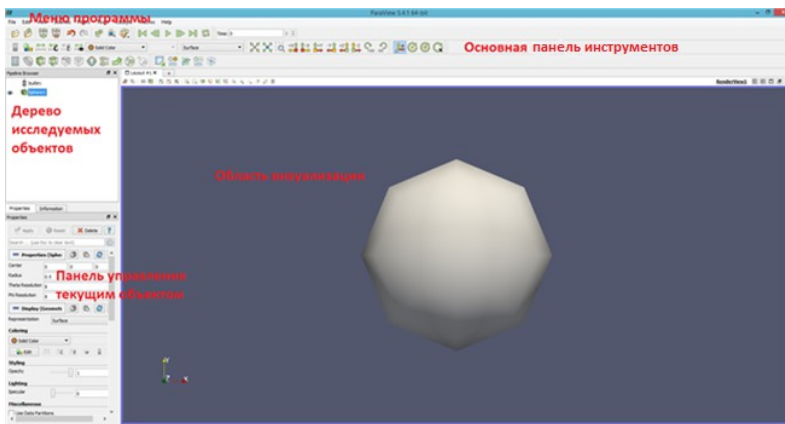


Рис. 2. Интерфейс ParaView.

Запустите пример `disk_out_ref` (File-Open-Examples) и нажмите Apply. Результат визуализации может быть представлен разными способами, выбрать которые можно из раскрывающегося списка на приборной панели. Попробуйте все способы (рис. 3).

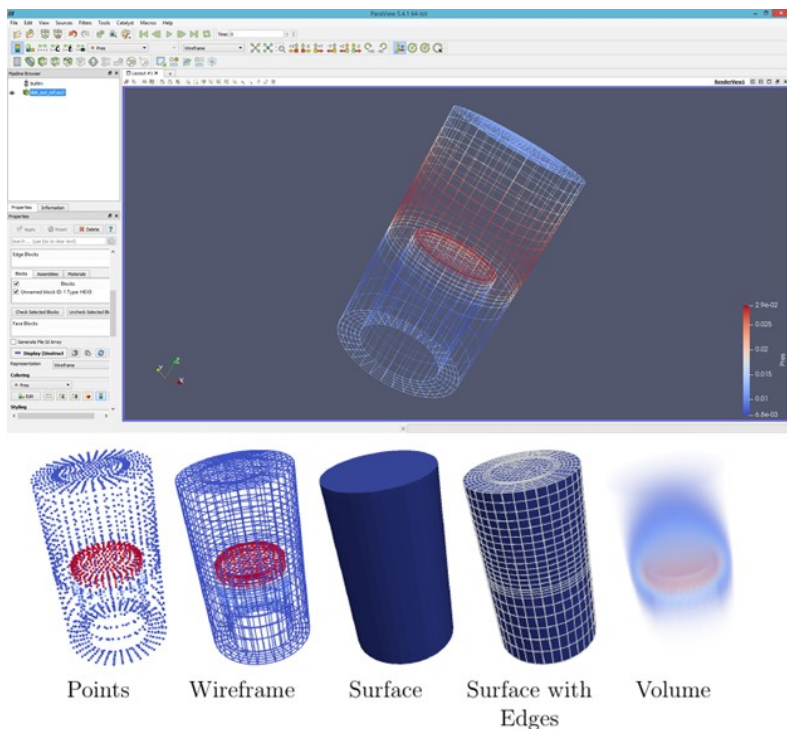
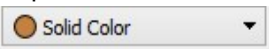


Рис. 3. Отображение объекта в ParaView.

Выбрать отображаемый элемент можно с помощью выпадающего списка , чтобы элемент оказался в списке необходимо установить напротив него флажок в блоке Properties (рис. 4).

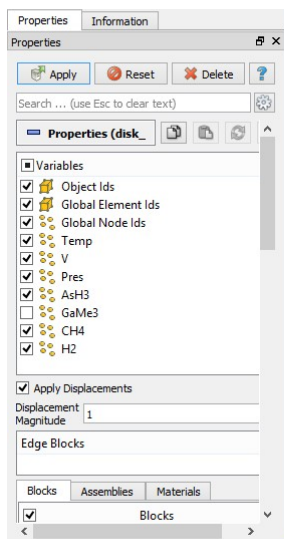


Рис. 4. Блок Properties.

Этап 5. Применение фильтров в ParaView.

Получите изображение объекта в разрезе, воспользовавшись фильтром Clip (рис. 5).

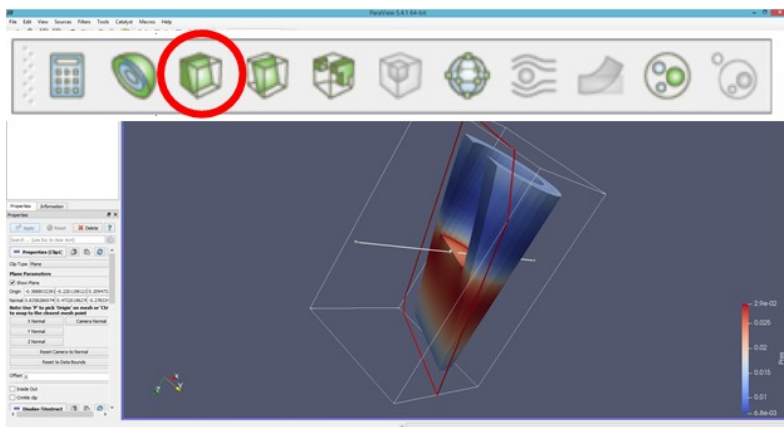


Рис. 5. Применение фильтра Clip.

Получите сечение объекта, воспользовавшись фильтром Slice (рис. 6).

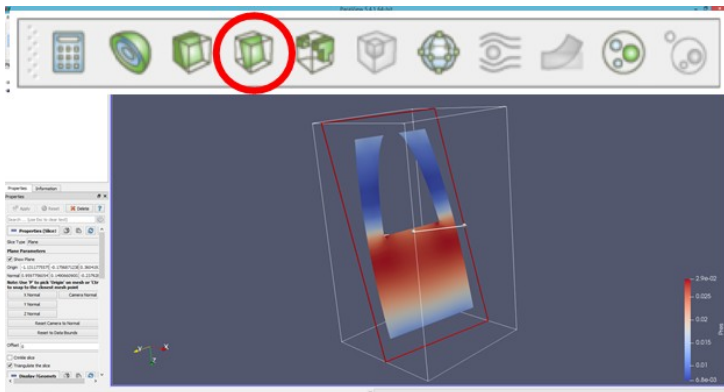


Рис. 6. Применение фильтра Slice.

Фильтры ParaView также предоставляют доступ к дополнительным средствам визуализации данных. Постройте график изменения величины давления вдоль произвольно проведённой оси (Filters – Data Analysis – Plot Over Line) (рис. 7), предварительно выбрав в списке свойств давление (Pres).

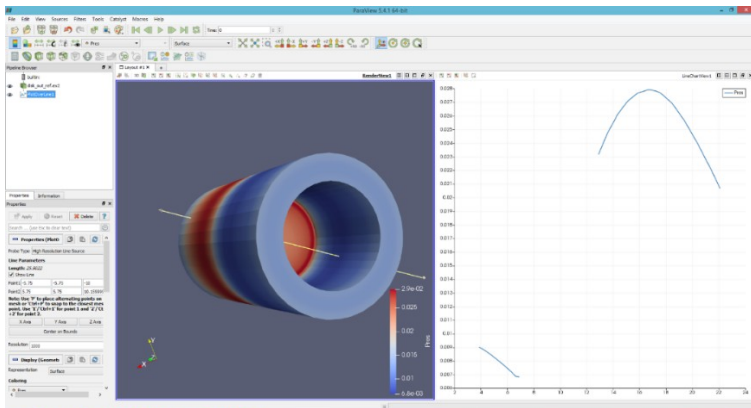


Рис. 7. Построение графика изменения величины давления вдоль произвольно проведённой оси.

Для построения графиков или таблиц в области визуализации можно выделять как отдельные ячейки или точки, так и их группы (Select Cells Through, панель находится над окном визуализации) (рис. 8). График строится с помощью фильтра Plot Data.

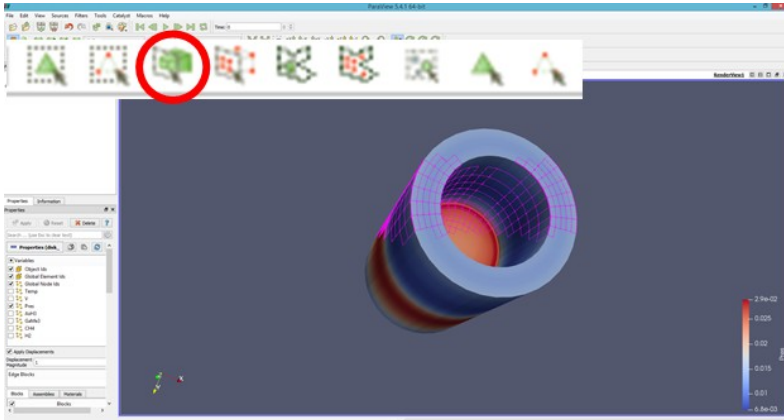


Рис. 8. Выделение группы ячеек.

Все фильтры доступны также в алфавитном порядке в меню Filters-Alphabetical.

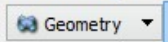
Для отображения объектов и величин, меняющихся во времени, в ParaView предусмотрена возможность анимированной визуализации. Откройте пример cap (File-Open-Examples-cap) и запустите анимацию с помощью кнопки Play на панели инструментов.

Этап 6. Знакомство с программой SALOME.

SALOME – открытая интегрируемая платформа для численного моделирования. Представляет собой набор пре- и постпроцессинга.

Запуск SALOME осуществляется следующим образом:

1. Найдите папку, имеющую в названии слово SALOME (как правило, SALOME-8.3.0-WIN64), и зайдите в неё.
2. Далее зайдите в папку Work.
3. Запустите файл run_salome.bat

Создайте новый файл и перейдите в модуль Geometry, выбрав его из выпадающего списка . SALOME имеет во многом схожий с ParaView интерфейс (рис. 9): область визуализации с кнопками управления отображением, расположенное слева от него дерево объектов и расположенная над ними панель инструментов, дублирующая некоторые пункты основного меню.

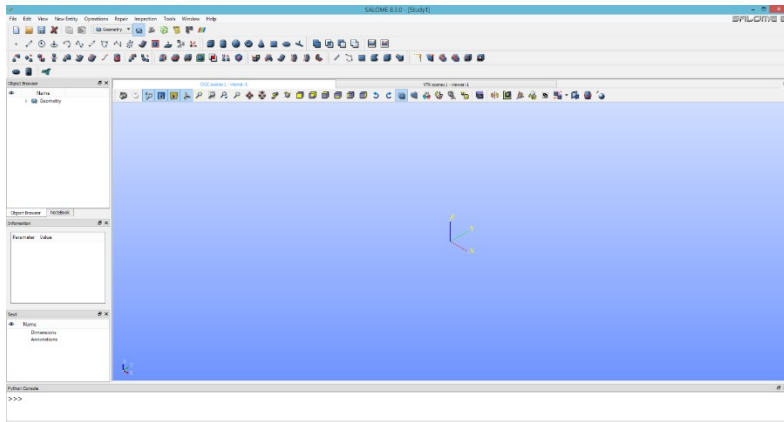


Рис. 9. Интерфейс SALOME.

В данном модуле вы можете создавать различные геометрические объекты, комбинировать и изменять их.

Перейдя в модуль Mesh, вы сможете создать для выбранного объекта расчётную сетку с требуемыми параметрами и алгоритмами её расчёта.

Дальнейшее знакомство с SALOME осуществляется в следующих занятиях.

Таким образом, в данной работе Вы познакомились с интерфейсом программ BlueCFD, ParaView и SALOME.

Занятие №2

Тема: Моделирование течения в каверне.

Цель: Изучение примера, подготовка геометрии и сетки с помощью blockMesh, задание параметров, запуск на счет, анализ результатов в ParaView.

Постановка задачи

Рассмотрим каверну (полость неправильной или округлой формы), расположенную в трубе, по которой течет несжимаемая жидкость с постоянной скоростью (число Re соответствует ламинарному режиму). Найдем скорости и давление (в любой момент времени и в любой точке). Схема с размерами представлена на рис. 10.

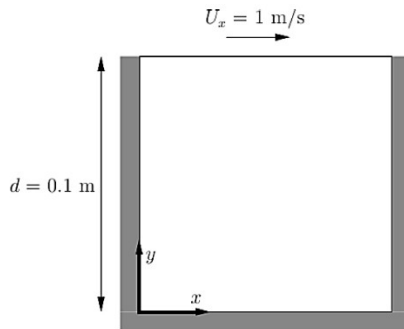


Рис. 10. Схема каверны.

Ход работы

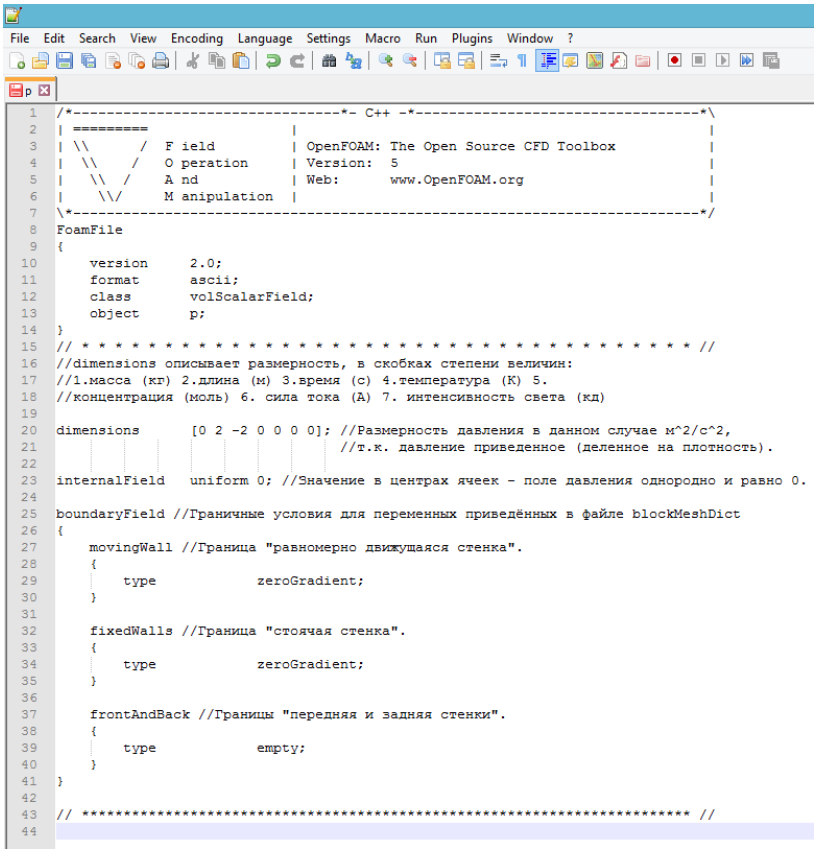
Этап 1. Ознакомление со структурой и кодом

1. Откройте ярлык «Browse blueCFD-Core folder», расположенный в меню «Пуск» или на рабочем столе.
2. Найдите папку «Cavity»: OpenFOAM-5.x > tutorials > incompressible > icoFoam > cavity
3. Скопируйте папку «Cavity» в свою папку.

Теперь рассмотрим содержимое папок (нужно зайти в папку «cavity», расположенную в папке «cavity»).

«0» – начальные условия

Файл «p» – начальные значения давления (откройте файл с помощью «Notepad 2.0») (рис. 11).



```
1 /*-----*-- C++ -*-----*\
2
3 =====
4 \\      / F i e l d           | OpenFOAM: The Open Source CFD Toolbox
5 \\      / O peration        | Version: 5
6 \\      / A n d              | Web:      www.OpenFOAM.org
7 \\      / M anipulation      |
8 =====\
9
10 FoamFile
11 {
12     version      2.0;
13     format       ascii;
14     class        volScalarField;
15     object       p;
16 }
17 // ***** //
18 //dimensions описывает размерность, в скобках степени величин:
19 //1.масса (кг) 2.длина (м) 3.время (с) 4.температура (К) 5.
20 //концентрация (моль) 6. сила тока (А) 7. интенсивность света (кд)
21
22 dimensions      [0 2 -2 0 0 0 0]; //Размерность давления в данном случае м^2/с^2,
23                                     //т.к. давление приведенное (деленное на плотность).
24
25 internalField    uniform 0; //Значение в центрах ячеек - поле давления однородно и равно 0.
26
27 boundaryField //Граничные условия для переменных приведенных в файле blockMeshDict
28 {
29     movingWall //Граница "равномерно движущаяся стенка".
30     {
31         type          zeroGradient;
32     }
33
34     fixedWalls //Граница "стоячая стенка".
35     {
36         type          zeroGradient;
37     }
38
39     frontAndBack //Границы "передняя и задняя стенки".
40     {
41         type          empty;
42     }
43 }
44 // ***** //
```

Рис. 11. Файл «p».

Файл «U» – начальные значения скорости (рис. 12).

- Список блоков `blocks`
- Список криволинейных рёбер `edges`
- Граничные условия стенок `boundary`
- Список объединяемых граней `mergePatchPairs`

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     object       blockMeshDict;
14 }
15 // * * * * *
16
17 convertToMeters 0.1; //Коэффициент масштабирования
18
19 vertices //Список координат вершин исследуемой области
20 (
21     (0 0 0)
22     (1 0 0)
23     (1 1 0)
24     (0 1 0)
25     (0 0 0.1)
26     (1 0 0.1)
27     (1 1 0.1)
28     (0 1 0.1)
29 );
30
31 blocks //Список блоков, из которых состоит расчетная область
32 (
33     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34     //Первая часть состоит из номеров вершин данного блока. Вторая запись показывает
35     //сколько точек разбиения задается в каждом направлении (направления x, y и z).
36     //Третья запись задаёт коэффициент расширения ячейки для каждого направления в блоке.
37 );
38
39 edges //Края блоков. По умолчанию каждый край считается прямым,
40     | //поэтому в данном случае эта область пустая.
41 (
42 );

```

Рис. 13. Начало файла «blockMeshDict».

```

43
44 boundary //Граничные условия стенок.
45 (
46     movingWall
47     {
48         type wall;
49         faces
50         (
51             (3 7 6 2)
52         );
53     }
54     fixedWalls
55     {
56         type wall;
57         faces
58         (
59             (0 4 7 3)
60             (2 6 5 1)
61             (1 5 4 0)
62         );
63     }
64     frontAndBack
65     {
66         type empty;
67         faces
68         (
69             (0 3 2 1)
70             (4 5 6 7)
71         );
72     }
73 );
74
75 mergePatchPairs //Список объединяемых граней
76 (
77 );

```

Рис. 14. Продолжение файла «blockMeshDict».

В общем случае может содержать также список поверхностей patches.

После определения точек следует сформировать гексаэдрические блоки – шестигранные объёмы с 8 узлами, 12 ребрами (рис. 15). Каждая грань имеет строго 4 ребра и 4 узла. Блоки задаются через перечисление узлов в определенном порядке (например, против часовой стрелки). Также задаются и внешние поверхности, определяющие граничные условия.

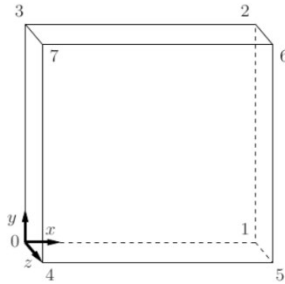


Рис. 15. Гексаэдрический блок.

Настройка численных схем интегрирования производится в файле «fvSchemes» (рис. 16).

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       fvSchemes;
15 }
16 // ***** //
17 //Используется для нахождения производных, которые вычисляются в процессе моделирования.
18 ddtSchemes //Производная первого порядка по времени
19 {
20     default      Euler;
21 }
22
23 gradSchemes //Градиент
24 {
25     default      Gauss linear;
26     grad(p)      Gauss linear;
27 }
28
29 divSchemes //Дивергенция
30 {
31     default      none;
32     div(phi,U)   Gauss linear;
33 }
34
35 laplacianSchemes //Диффузия
36 {
37     default      Gauss linear orthogonal;
38 }
39
40 interpolationSchemes //Интерполяция на грани
41 {
42     default      linear;
43 }
44
45 snGradSchemes //Производная по нормали
46 {
47     default      orthogonal;
48 }
49
50
51 // ***** //
52

```

Рис. 16. Файл «fvSchemes».

Способ нахождения производных указан после её конкретного вида (например, градиент давления «grad(p)» аппроксимирован по линейному способу Гаусса «Gauss linear», рис. 16).

Диффузия описывается уравнением, содержащим лапласиан.

Настройка методов решения систем линейных алгебраических уравнений производится в файле «fvSolution» (рис. 17). Слово «solver» означает решатель, «tolerance» – точность.

В OpenFOAM используется метод расщепления переменных, т.е. для каждой искомой переменной (скаляра или тензора) своя система уравнений, свой метод решения. Каждому полю – свой метод решения системы линейных алгебраических уравнений. Для компонент тензора метод решения один, но процедура – последовательная.

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       fvSolution;
15 }
16 // ***** //
17 //Настройка методов решения систем линейных алгебраических уравнений
18 solvers //Методы решения для скорости и давления
19 {
20     P
21     {
22         solver      PCG;
23         preconditioner DIC;
24         tolerance   1e-06;
25         relTol      0.05;
26     }
27
28     pFinal
29     {
30         Sp;
31         relTol      0;
32     }
33
34     U
35     {
36         solver      smoothSolver;
37         smoother    symGaussSeidel;
38         tolerance   1e-05;
39         relTol      0;
40     }
41 }
42
43 PISO //Параметры алгоритма связывания полей давления и скоростей
44 {
45     nCorrectors      2;
46     nNonOrthogonalCorrectors 0;
47     pRefCell         0;
48     pRefValue        0;
49 }
50
51
52 // ***** //
53

```

Рис. 17. Файл «fvSolution».

Контроль за ходом решения задачи осуществляется в файле «controlDict» (рис. 18). В разделе «Application» указан используемый решатель (в описываемом случае – «icoFoam»).

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       controlDict;
15 }
16 // * * * * *
17
18 application     icoFoam;
19
20 startFrom       startTime;
21
22 startTime      0; //Начальное время
23
24 stopAt         endTime; //Необходимо для окончания счёта
25
26 endTime        0.5; //Конечное время
27
28 deltaT         0.005; //Шаг по времени
29
30 writeControl    timeStep; //Запись поля скоростей и давления в определённые
31                   //промежутки времени - запись будет происходить
32                   //через равное количество итераций, описанное в секции writeInterval
33
34 writeInterval  20;
35
36 purgeWrite     0;
37
38 writeFormat    ascii; //формат записи
39
40 writePrecision 6; //Точность вывода результатов - количество знаков в записи
41
42 writeCompression off; //Сжатие - отключено
43
44 timeFormat     general; //формат времени
45
46 timePrecision  6;
47
48 runTimeModifiable true; //Возможность внесения изменений во время работы программы
49
50
51 // * * * * *
52

```

Рис. 18. Файл «controlDict».

Этап 2. Работа программы

1. Кликнув по папке C:\$ВАША_ПАПКА\$/cavity/cavity правой кнопкой мыши, выберите «Open in blueCFD-Core terminal».
2. Введите blockMesh – появится расчёт сетки объекта (рис. 19). Посмотрите, что в «constant» появилась папка «polyMesh». Изучите её содержимое.

Примечание: при введении команд следите за маленькими и большими буквами! Все команды начинаются с маленькой буквы, последующие слова пишутся с большой буквы.

Примечание: обращайтесь внимание на ошибки и предупреждения.

```

C:\labal1\cavity\cavity
$ blockmesh
-----
      \  /
       V
      /  \
-----
Field      | OpenFOAM: The Open Source CFD Toolbox
Operation  | Version: 5.x
And        | web: www.OpenFOAM.org
Manipulation|
-----
/* Windows 32 and 64 bit porting by blueCAPE: http://www.bluecape.com.pt */
Based on Windows porting (2.0.x v4) by Symscape: http://www.symscape.com */
-----
Build   : 5.x-7ac8a13a3b0
Exec    : C:\PROGRA~1\BLUECF~1\OpenFOAM-5.x\platforms\mingw_w64gccDPInt32Opt\bin\blockmesh.exe
Date    : Feb 25 2019
Time    : 15:18:36
Host    : "pc-05-210a-05"
PID     : 7068
I/O     : uncollated
Case    : C:\labal1\cavity\cavity
nProcs  : 1
SigFpe  : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster (fileModificationSkew 10)
allowSystemOperations : Allowing user-supplied system call operations

// * * * * *
Create time

Deleting polyMesh directory
"C:/labal1/cavity/cavity/constant/polyMesh"
Creating block mesh from
"C:/labal1/cavity/cavity/system/blockMeshDict"
Creating block edges
No non-planar block faces defined
Creating topology blocks
Creating topology patches

Creating block mesh topology

Check topology

Basic statistics
Number of internal faces : 0
Number of boundary faces : 6
Number of defined boundary faces : 6
Number of undefined boundary faces : 0
Checking patch -> block consistency

Creating block offsets
Creating merge list .

Creating polyMesh from blockMesh
Creating patches
Creating cells
Creating points with scale 0.1
Block 0 cell size :
i : 0.005 .. 0.005
j : 0.005 .. 0.005
k : 0.01 .. 0.01

Writing polyMesh

Mesh Information
-----
boundingBox: (0 0 0) (0.1 0.1 0.01)
nPoints: 882
nCells: 400
nFaces: 1640
nInternalFaces: 760
-----
Patches
-----
patch 0 (start: 760 size: 20) name: movingWall
patch 1 (start: 780 size: 60) name: fixedWalls
patch 2 (start: 840 size: 800) name: frontAndBack

End

```

Рис. 19. Расчёт сетки объекта.

3. Введите isoFoam – расчет течения жидкости в каверне. Посмотрите, что изменилось в папке «cavity» (в случае удачного выполнения продолжительное время идёт расчет, иначе выводится Error).

Примечание: обращайтесь внимание на ошибки и предупреждения.

4. Введите paraFoam – запустится программа ParaView, используемая для визуализации решений (рис. 20). Если окно ParaView не открывается, самостоятельно создайте в папке «cavity» файл «cavity.foam», запустите программу и загрузите этот файл.

Этап 3. Знакомство с ParaView

Перейдите к работе с ParaView (рис. 20).

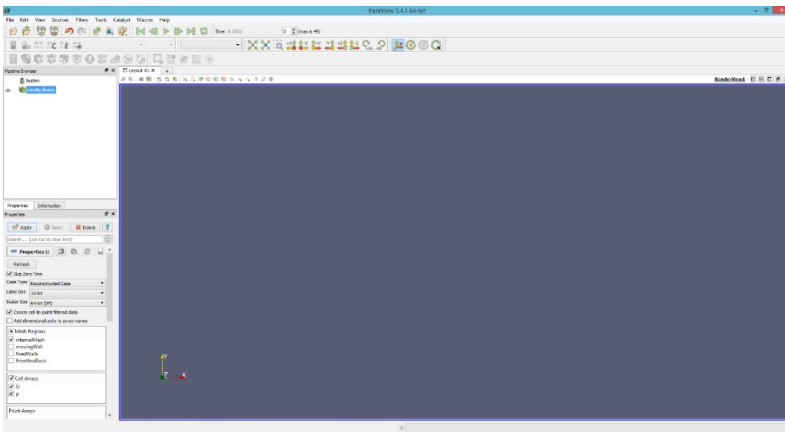


Рис. 20. Окно ParaView.

Нажмите кнопку Apply в меню Properties слева. Появится изображение давления жидкости (p) в камере в начале расчета (0 с) (рис. 21).

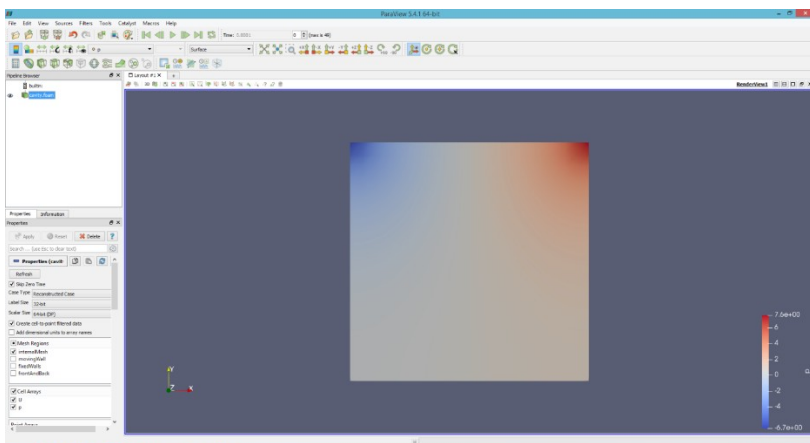
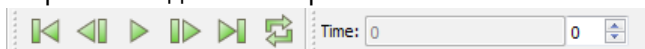









Рис. 21. Изображение давления жидкости в камере в начале расчёта.

Изображение можно вращать, зажав левую кнопку мыши, перемещать по экрану, зажав колесико мыши, и изменять его размер, вращая колесико мыши.

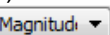
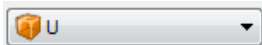
В верхнем меню кнопками можно выбирать время, расчет для которого выводится на экран.



-  – запустить анимацию
-  – следующий шаг
-  – последний шаг
-  – предыдущий шаг
-  – первый шаг
-  – зациклить анимацию

С помощью меню  можно выбирать параметр, решение которого выводится на экран. Например, параметр p (давление) можно изменить на U (скорость жидкости).

А также в этом меню можно задать изображение решения в виде одного цвета для каждого элементарного объема (



С помощью меню можно выбрать, что будет показывать программа: амплитуду величины (magnitude) или ее проекцию на одну из координатных осей (синие участки имеют наименьшее значение величины, красные – наибольшую.).



С помощью меню можно менять способ представления результатов (в виде поверхности, сетчатой поверхности, точек и т.д.).

Чтобы увидеть шкалу, нажмите .

Чтобы построить поле модуля скорости и давления: переходите к последнему моменту времени, извлеките центры ячеек (Filters (в верхнем меню) → Alphabetical → Cell Centers); строите векторное поле в центрах ячеек (Filters → Alphabetical → Glyph), отключив масштабирование вектора скорости (во вкладке Properties для Glyph1 в раскрывающемся меню Scale Mode выберите off) (рис. 22).

Для построения линий тока: выберите объект, соответствующий каверне (cavity.foam) и вызовите фильтр Stream Tracer (Filters → Alphabetical → Stream

Tracer) (рис. 23). Нажмите Apply.

Рис. 22. Выбор off в меню Scale Mode.

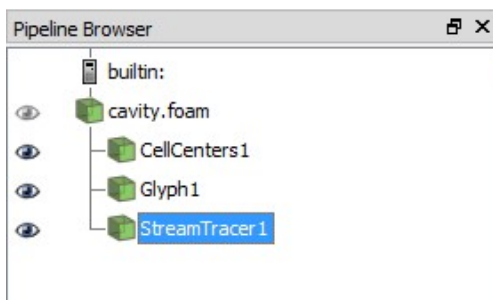


Рис. 23. Вызов фильтра Stream Tracer.

Чтобы увидеть только направления или линии тока, нажмите на значок глаза возле StreamTracer или Glyph1 соответственно (рис. 24).

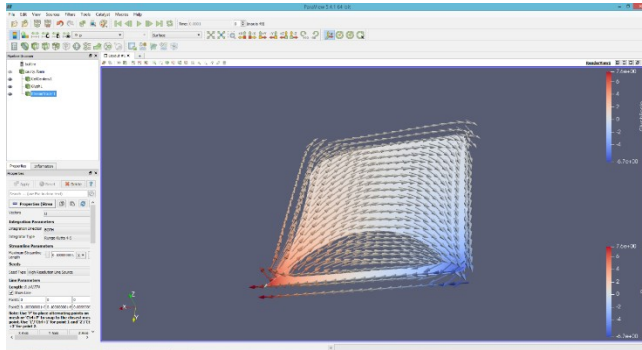


Рис. 24. Просмотр StreamTracer1 и Glyph1.

Этап 4. Изменение параметров задачи

Получите решения, внося следующие изменения:

1. Увеличьте вязкость до 0.1.
2. Измените параметры (см. рис. 25).

Примечание: в качестве разделителя целой и дробной части в десятичных дробях используется точка.

Если требуется произвести расчет еще раз, удалите все папки, имеющие названия-числа, кроме 0 (там заданы начальные условия). Если папки не будут удалены, то ParaView визуализирует старое и новое решение одновременно. Кроме того необходимо перезапустить ParaView. После каждого изменения повторите все действия, начиная с пункта 3 упражнения 2.

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       controlDict;
15 }
16 // * * * * * //
17
18 application     icoFoam;
19
20 startFrom       startTime;
21
22 startTime       0; //Начальное время
23
24 stopAt          endTime; //Необходимо для окончания счёта
25
26 endTime        0.01; //Конечное время
27
28 deltaT          0.0001; //Шаг по времени
29
30 writeControl    timeStep; //Запись поля скоростей и давления в определённые
31                   //промежутки времени - запись будет происходить
32                   //через равное количество итераций, описанное в секции writeInterval
33
34 writeInterval  1;
35
36 purgeWrite     0;
37
38 writeFormat    ascii; //формат записи
39
40 writePrecision 6; //Точность вывода результатов - количество знаков в записи
41
42 writeCompression off; //Сжатие - отключено
43
44 timeFormat     general; //формат времени
45
46 timePrecision  6;
47
48 runTimeModifiable true; //Возможность внесения изменений во время работы программы
49
50
51 // * * * * * //
52

```

Рис. 25. Изменение параметров задачи.

Таким образом, в данной работе было смоделировано течение жидкости в каверне, получены поле скорости и распределение давления с помощью средств OpenFOAM.

Занятие №3

Тема: Моделирование прорыва дамбы.

Цель: Изучение математической модели, задание параметров, запуск на счет, анализ результатов в ParaView.

Постановка задачи

Столб жидкости, расположенный в левой части расчетной области, в момент времени $T=0$ обрушивается на препятствие на дне и создает поток сложной формы (рис. 26).

Следует получить значения концентрации воды, давления и скорости.

Двумерная задача решается с использованием решателя `interFoam`. Особенностью является нестационарное течение двух жидкостей, разделенных интерфейсом, или свободной поверхностью.

Геометрия задачи разбивается на 5 блоков из-за наличия выступа. Подумайте, каким образом было произведено это разбиение.

Переход от единиц геометрии к естественным единицам задан с помощью множителя 0,146 и операции `convertToMeters` в файле `blockMeshDict` (рис. 27).

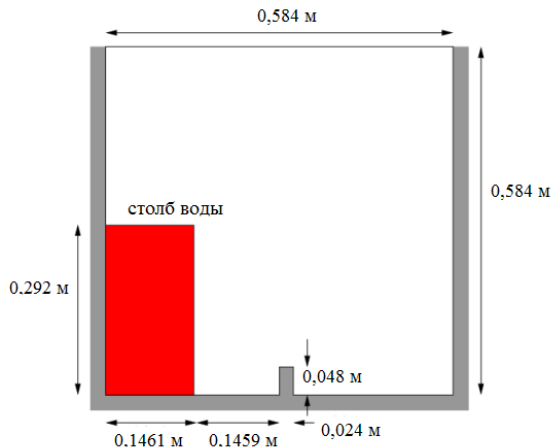


Рис. 26. Геометрия и начальные параметры задачи.

```

convertToMeters 0.146;

vertices
(
    (0 0 0)
    (2 0 0)
    (2.16438 0 0)
    (4 0 0)
    (0 0.32876 0)
    (2 0.32876 0)
)

```

Рис. 27. Переход от единиц геометрии к естественным единицам в файле blockMeshDict.

Ход работы

Этап 1. Запуск программы

1. Скопировать папку damBreak из blueCFD-Core-2017/OpenFOAM-5.x /tutorials/multiphase/interFoam/laminar/dambreak в свою папку.
2. Кликнув по папке **C:\$ВАША_ПАПКА\$/damBreak/damBreak** правой кнопкой мыши, выберите «Open in blueCFD-Core terminal».
3. Построить сетку с помощью команды **blockMesh**.
4. С помощью команды **setFields** заполните часть ячеек водой (рис. 28).

```

Create time
Create mesh for time = 0
Reading setFieldsDict
Setting field default values
  Setting internal values of volScalarField alpha.water
Setting field region values
  Adding cells with center within boxes 1((0 0 -1) (0.1461 0.292 1))
  Setting internal values of volScalarField alpha.water
End

```

Рис. 28. Применение команды setFields.

5. Произвести расчет движения жидкости с помощью команды **interFoam**.
6. Запустить программу для визуализации командой **paraFoam** (если запуск не произошёл, повторить действия, описанные в занятии №1).

При последующих изменениях параметров, необходимо повторять данные действия:

- с пункта 3, если меняется геометрия;
- с пункта 4, если меняется файл `setFieldsDict`.

Этап 2. Изменение начальных условий

Чтобы перезапустить файл `setFieldsDict` удалите содержимое старого файла `alpha.water`, а затем скопируйте содержимое файла `alpha.water.orig` в файл `alpha.water`, затем в файле `setFieldsDict` измените размер и расположение блока воды. Например, можно сделать слой воды шире, сдвинуть его и поднять. Координаты меняются в строке `box`, строящей параллелепипед по координатам начала и конца его диагонали.

Примечание: перед запуском расчёта введите команду `setFields`.

Произведите расчет, проанализируйте изменения.

Разместите ещё один такой же столб воды симметрично первому. Для этого зайдите в `setFieldsDict` и добавьте ещё один блок `boxToCell` с соответствующими координатами (рис. 29). Блок `boxToCell` строится по координатам двух точек диагонали параллелепипеда – задней верхней правой вершины и передней нижней левой вершины (рёбра параллельны осям).

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       setFieldsDict;
15 }
16 // ***** //
17
18 defaultFieldValues
19 {
20     volScalarFieldValue alpha.water 0
21 };
22
23 regions
24 {
25     boxToCell
26     {
27         box (0 0 -1) (0.1461 0.292 1);
28         fieldValues
29         {
30             volScalarFieldValue alpha.water 1
31         };
32     }
33
34     boxToCell
35     {
36         box (0.4379 0 -1) (0.584 0.292 1);
37         fieldValues
38         {
39             volScalarFieldValue alpha.water 1
40         };
41     }
42 };
43
44
45 // ***** //
46

```

Рис. 29. Добавление блока boxToCell.

Произведите расчет и проанализируйте полученный результат.

Сделайте из двухмерной задачи трёхмерную, увеличив в файле blockMeshDict толщину (масштаб) области по вертикали (для этого в поле vertices поменяйте координаты узлов, где первая координата – x , вторая – y , третья – z , нужно менять координату верхних узлов по оси z , исходно равную 0,1), и в поле hex поменяв число точек разбиения в направлении оси z (рис. 30) (в первой скобке стоят номера узлов, входящих в этот блок, во второй – число точек разбиения по x , y , z , в третьей – множитель масштабирования). Затем вместо параллелепипеда добавьте шар воды. Для этого используйте блок sphereToCell, указав координаты центра (centre) шара и его радиус (radius) (рис. 31). В ParaView посмотрите движение сферы воды с помощью опции

Clip (рис. 32), отключив опцию Show Plane, чтобы сделать невидимой секущую плоскость.

<pre> vertices ((0 0 0) (2 0 0) (2.16438 0 0) (4 0 0) (0 0.32876 0) (2 0.32876 0) (2.16438 0.32876 0) (4 0.32876 0) (0 4 0) (2 4 0) (2.16438 4 0) (4 4 0) (0 0 0.1) (2 0 0.1) (2.16438 0 0.1) (4 0 0.1) (0 0.32876 0.1) (2 0.32876 0.1) (2.16438 0.32876 0.1) (4 0.32876 0.1) (0 4 0.1) (2 4 0.1) (2.16438 4 0.1) (4 4 0.1)); blocks (hex (0 1 5 4 12 13 17 16) (23 8 1) simpleGrading (1 1 1) hex (2 3 7 6 14 15 19 18) (19 8 1) simpleGrading (1 1 1) hex (4 5 9 8 16 17 21 20) (23 42 1) simpleGrading (1 1 1) hex (5 6 10 9 17 18 22 21) (4 42 1) simpleGrading (1 1 1) hex (6 7 11 10 18 19 23 22) (19 42 1) simpleGrading (1 1 1)); Исходный вариант </pre>	<pre> vertices ((0 0 0) (2 0 0) (2.16438 0 0) (4 0 0) (0 0.32876 0) (2 0.32876 0) (2.16438 0.32876 0) (4 0.32876 0) (0 4 0) (2 4 0) (2.16438 4 0) (4 4 0) (0 0 0.1) (2 0 0.1) (2.16438 0 0.1) (4 0 0.1) (0 0.32876 0.1) (2 0.32876 0.1) (2.16438 0.32876 0.1) (4 0.32876 0.1) (0 4 0.1) (2 4 0.1) (2.16438 4 0.1) (4 4 0.1)); blocks (hex (0 1 5 4 12 13 17 16) (23 8 40) simpleGrading (1 1 1) hex (2 3 7 6 14 15 19 18) (19 8 40) simpleGrading (1 1 1) hex (4 5 9 8 16 17 21 20) (23 42 40) simpleGrading (1 1 1) hex (5 6 10 9 17 18 22 21) (4 42 40) simpleGrading (1 1 1) hex (6 7 11 10 18 19 23 22) (19 42 40) simpleGrading (1 1 1)); Вариант с увеличенным масштабом </pre>
--	--

Рис. 30. Пример изменения масштаба области в 40 раз.

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       setFieldsDict;
15 }
16 // ***** //
17
18 defaultFieldValues
19 (
20     volScalarFieldValue alpha.water 0
21 );
22
23 regions
24 (
25     sphereToCell
26     {
27         centre (0.08 0.2 0.07);
28         radius 0.1;
29         fieldValues
30         {
31             volScalarFieldValue alpha.water 1
32         };
33     }
34 );
35
36
37 // ***** //
38

```

Рис. 31. Добавление шара воды.

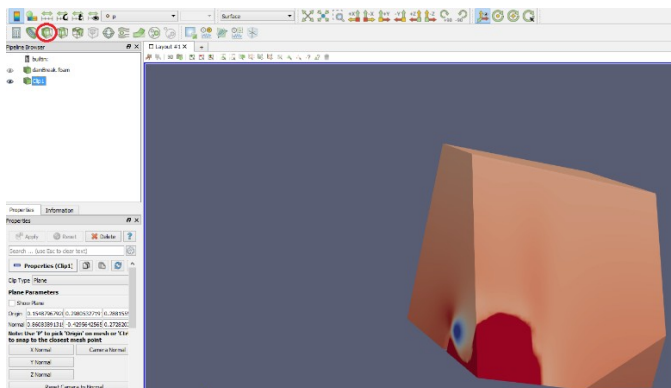


Рис. 32. Просмотр сечения со сферой из воды с помощью опции Clip.

Этап 3. Изменение транспортных свойств

Зайдите по ссылке `C:/$ВАША_ПАПКА$/damBreak/damBreak/constant/transportProperties` и поменяйте вязкость жидкости (μ). Произведите расчет и проанализируйте полученный результат.

Примечание: расчёт проводится с помощью команды `interFoam`.

Этап 4. Изменение параметров расчета

В папке `C:/$ВАША_ПАПКА$/damBreak/damBreak/system` открыть файл `controlDict`. Можно изменить время, когда расчет начинается (`startTime`) и заканчивается (`endTime`), промежуток времени, через который проводятся вычисления (`deltaT`), интервал, через который производится запись расчета (`writeInterval`).

Произведите расчет, проанализируйте изменения.

Справку по различным функциям OpenFoam можно посмотреть здесь: <https://openfoamwiki.net/index.php/TopoSet>

Итак, в ходе данной работы Вы познакомились с заданием областей воды в двумерной области с помощью файла `set-`

FieldsDict и решением двумерной задачи с помощью решателя interFoam.

Занятие №4

Тема: Знакомство с инструментарием программы Salome.

Цель: Построить трехмерную геометрическую модель детали Lego с помощью модуля Geometry программы Salome и создать для него трехмерную сетку.

Ход работы

Этап 1. Построение геометрии

Запустить программу, как было описано в самостоятельной работе №0, нажать New Document и в выпадающем меню модулей (Modules) выбрать Geometry.

В меню **NewEntity** необходимо выбрать **Primitives — Box**. В появившемся окне следует выбрать второй вариант создания параллелепипеда (по размерам) и ввести следующие значения:

Name: TopBox;
Dx: 0.025;
Dy: 0.012;
Dz: 0.016.

Затем нужно построить две точки, определяющие диагональ внутреннего прямоугольника (будущей полости). Для этого выберем **NewEntity — Basic — Point**.

Значения для первой точки:

Name: BottomD;
x: 0.002;
y: 0.002;
z: 0.

Для второй точки:

Name: TopD;
x: 0.023;
y: 0.010;
z: 0.014.

Переключать вид отображения объекта можно, если зайти в **View — Display Mode** и выбрать, например, каркасный режим отображения (Wireframe) или режим отображения трехмерной модели (Shading) (рис. 33).

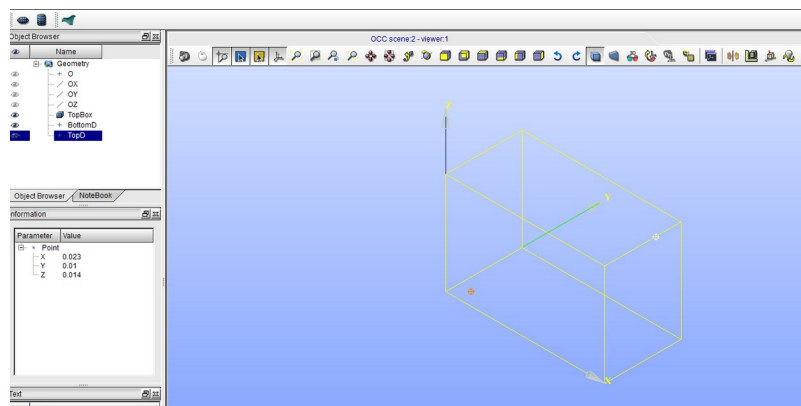


Рис. 33. Каркасный режим отображения.

Далее следует создать внутренний параллелепипед. Необходимо выбрать **NewEntity — Primitives — Box**. Во вкладке Box выбирается первый пункт (построение по точкам диагонали) и задаются значения:

Name: InnerBox;

Point1: BottomD; (следует кликнуть по названию точки в дереве объектов Geometry)

Point2: TopD.

Теперь необходимо вычесть маленький параллелепипед из большего. Это делается с помощью логической операции Cut (**Operations — Boolean — Cut**). Необходимо выбрать значения для новой фигуры:

Name: LegoBase;

Main Object: TopBox;

Tool Object: InnerBox.

Полученную модель можно посмотреть, если перейти из каркасного режима отображения в режим отображения трехмерной модели (рис. 34).

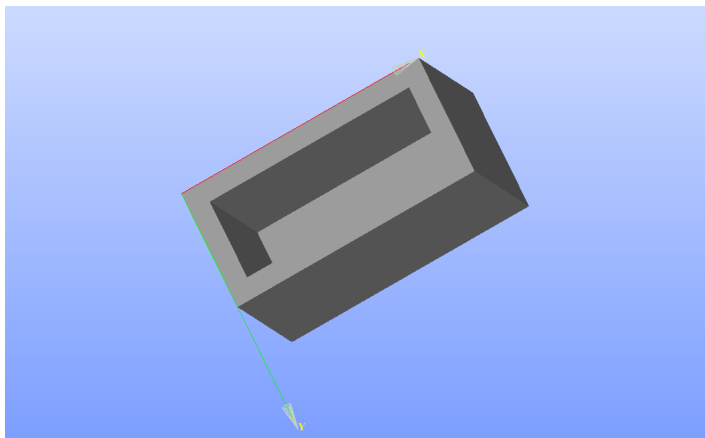


Рис. 34. Режим отображения трёхмерной модели.

Остается создать два цилиндра с указанными параметрами и переместить их на соответствующие позиции. Параметры первого цилиндра:

Name: BumpL;

Radius: 0.004;

Height: 0.002.

Переместим цилиндр с помощью операции Translation (**Operations — Transformation — Translation**), в которой необходимо снять галочку с CreateCopy:

Objects: BumpL;

Dx: 0.006;

Dy: 0.006;

Dz: 0.016.

С помощью этой же операции Translation создается второй цилиндр с установленной галочкой в CreateCopy и с параметрами:

Name: BumpR;

Objects: BumpL;

Dx: 0.013;

Dy: 0;

Dz: 0.

Видимость элементов можно менять нажатием на значок глаза рядом с каждым названием в дереве объектов.

Осталось объединить цилиндры с основой с помощью логической операции Fuse. Выбор объектов осуществляется нажатием по названиям в дереве объектов с нажатой клавишей Ctrl (рис. 35). Объединять можно по 2 объекта сразу, поэтому будет необходимо провести операцию объединения 2 раза.

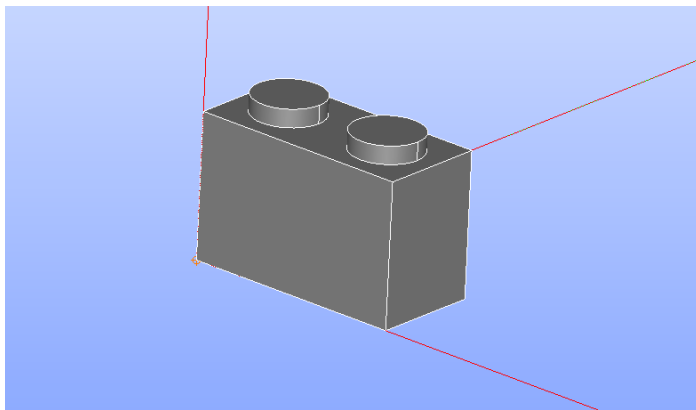


Рис. 35. Выбор объектов для применения логической операции Fuse.

Этап 2. Создание сетки

Описать поведение реального объекта зачастую бывает достаточно сложно. Упростить решение можно с помощью разбиения объекта на конечные элементы. При разбиении объекта на малые элементы сначала необходимо определить точки, которые будут вершинами элементов. Эти точки называют узлами (node) расчетной сетки. Далее, на основе выбранных алгоритмов и геометрии базовых элементов, узлы соединяются прямыми линиями, называемыми ребрами (edge). Область, заключенная между несколькими ребрами и не содержащая ни одного узла или ребра, называется гранью (face). Обычно грани образуются тремя или четырьмя ребрами. Область, заключенная между несколькими гранями и не содержащая ни одного

узла, ребра или грани (части грани) называется объемным объектом (volume object).

Построим трехмерную сетку для детали Lego.

В Salome перейдем из модуля Geometry в модуль Mesh. Для построения сетки воспользуемся автоматическим генератором двумерных и трехмерных сеток NetGen. Для вызова диалогового окна создания сетки выберем **Mesh — Create Mesh**:

Name: Mesh3D_01;

Geometry: Lego;

Algorithm: NetGen 1D-2D-3D.

Hypothesis: NetGen 3D Simple Parameters (иконка с шестеренкой; оставить стандартные значения для всех параметров).

Далее выбрать пункт **Mesh — Compute**.

Примечание: важно сначала создать сетку, потом рассчитать.

Объект оказался разбит более чем на 14000 объемных элементов (рис. 36). Для уменьшения времени расчета необходимо уменьшить количество элементов. Будем уменьшать количество элементов в три раза.

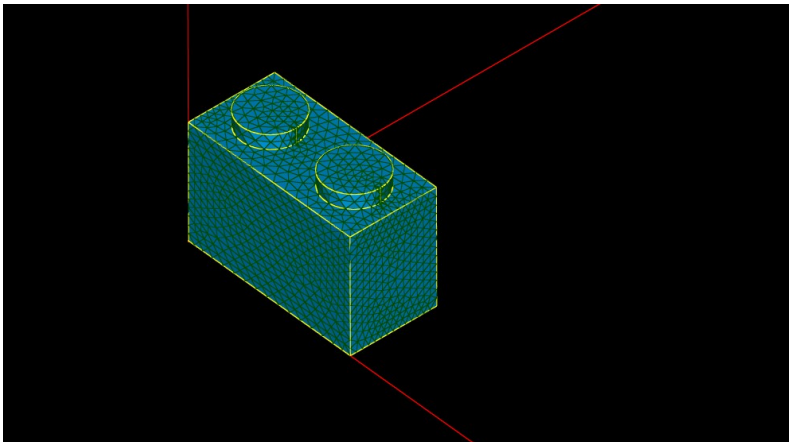


Рис. 36. Разбиение объекта более чем на 14000 элементов.

Попробуем уменьшить количество частей разбиения. Щелкнем правой кнопкой мыши на названии гипотезы Netgen 3D Simple Parameters_1 в дереве объектов и выберем Edit Hypothesis, уменьшим число сегментов (Number of Segment) с 15 до 5.

Пересчитаем сетку: правой кнопкой мыши по Mesh3D_01 — Compute. При этом количество частей снизилось до 850, что не удовлетворяет нашим требованиям.

Для адекватного разбиения следует построить подсетку. Для этого сгруппируем однотипные элементы. Перейдем в модуль Geometry и выделим в дереве объектов объект Lego (щелкнуть левой кнопкой мыши). Выбираем **New Entity — Explode**:

Main Object: Lego;

Sub-shapes Type (тип выделяемых составляющих): Edge.

Нажимаем Apply, затем, не закрывая окна, выбираем Face вместо Edge и нажимаем Apply and Close. В результате в дереве объектов в элементе Lego появится 15 граней и 30 ребер. Далее мы должны построить более мелкую сетку для цилиндрических выступов и прилегающей к ней верхней грани. Создадим две группы объектов: набор ребер и набор граней.

Необходимо выбрать основной объект Lego и создать группу: **New Entity — Group — Create Group**. В результате откроется окно создания группы.

Следует выделить необходимые ребра, как показано на рис. 37, левой кнопкой мыши, при этом необходимо удерживать клавишу Shift, затем нажать кнопку Add (добавить). В результате соответствующем окне отобразится список с номерами выбранных ребер. Нумерация ребер может не совпадать с представленной на рисунке, совпадать будет их количество (8), важно выбрать именно ребра верхней части объекта, лишние ребра можно удалить клавишей Remove.

Примечание: чтобы отобразить объект в виде, как на рис. 37, следует включить опцию View – Display Mode – Wireframe.

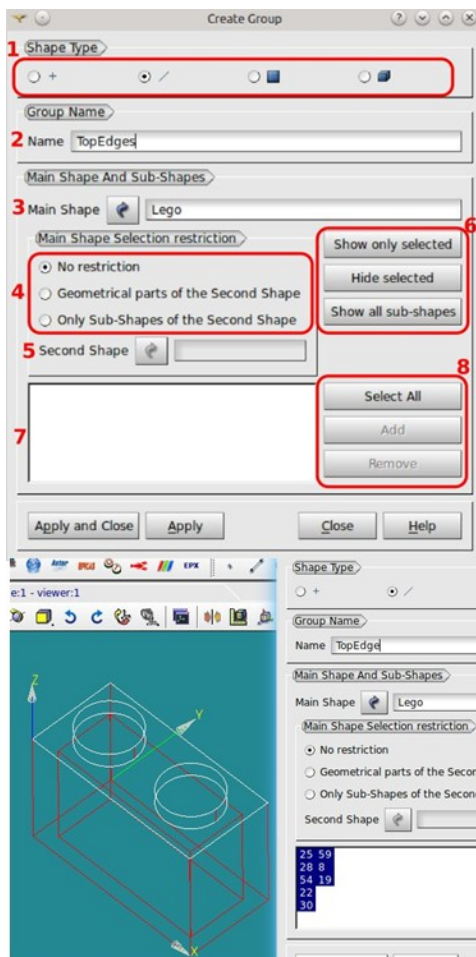


Рис. 37. Создание группы рёбер.

Нажав кнопку Apply и не закрыв окно, перейдем к созданию группы граней. Для этого в верхней строке следует выбрать соответствующий тип элемента (рис. 38), выделить аналогично рёбрам необходимые грани (их будет 5) (рис. 38) и нажать кнопку Add.

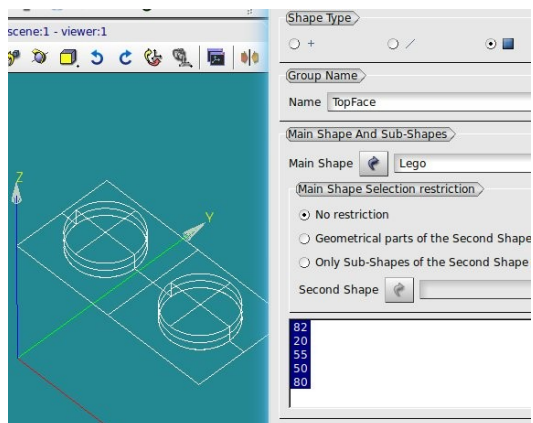


Рис. 38. Создание группы граней.

В результате получится две группы элементов с именами **TopEdge** и **TopFace**. Теперь следует вернуться в модуль **Mesh**. Необходимо заполнить поля диалоговых окон для создания подсетки на основе группы ребер. Для этого нажмем **Mesh — Create Sub-Mesh** и сделаем, как показано на рис. 39, после чего нажмем на кнопку **Apply and Close**.

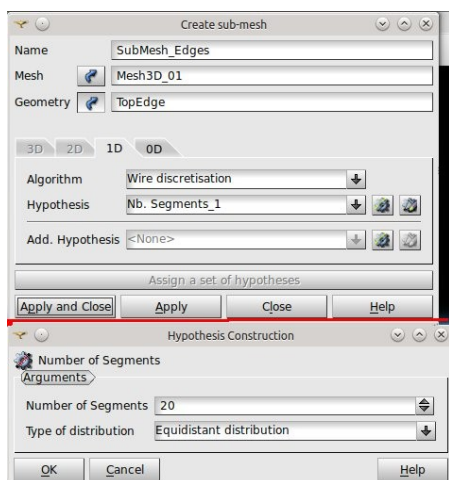


Рис. 39. Создание подсетки для группы рёбер.

Следует рассчитать сетку с помощью команды **Mesh — Compute**. Также по рис. 40 необходимо сделать для граней и нажать на кнопку **Apply and Close**.

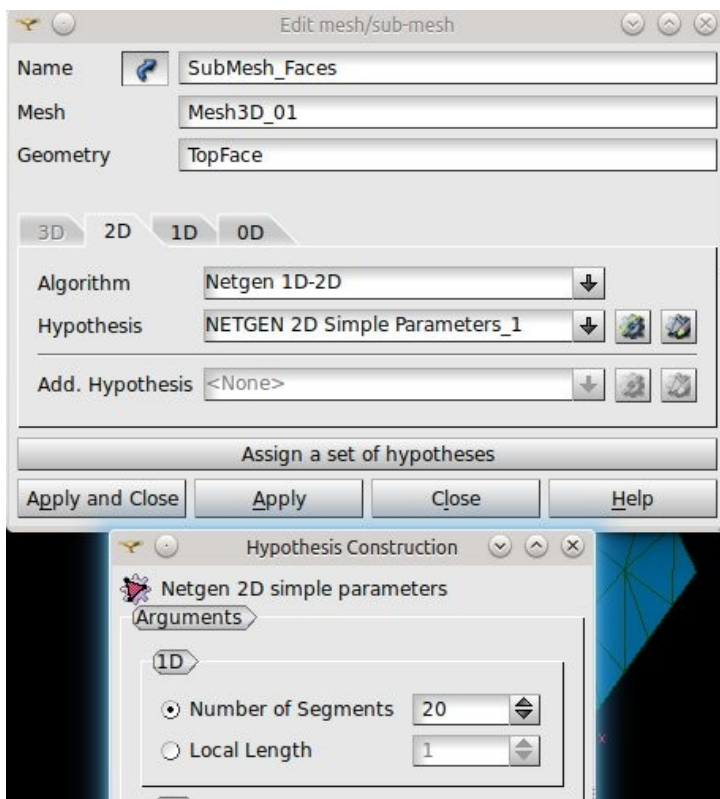


Рис. 40. Создание подсетки для группы граней.

Также произвести пересчёт сетки с помощью команды **Compute**. Переформируем сетку в дереве объектов, изменив гипотезы формирования сетки NETGEN 3D Simple Parameters_1 и подсетки NETGEN 2D Simple Parameters_1, как показано на рис. 41.

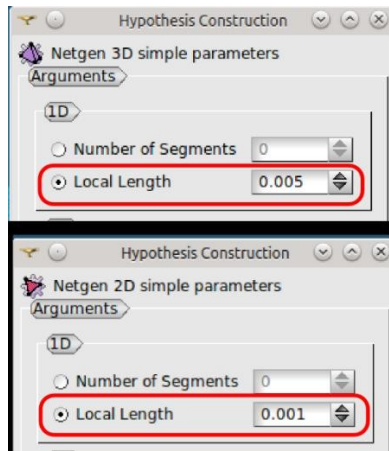


Рис. 41. Изменение гипотезы формирования сетки.

Произвести пересчёт с помощью команды Compute. Получится сетка и подсетка с одинаковыми расстояниями между узлами (рис. 42).

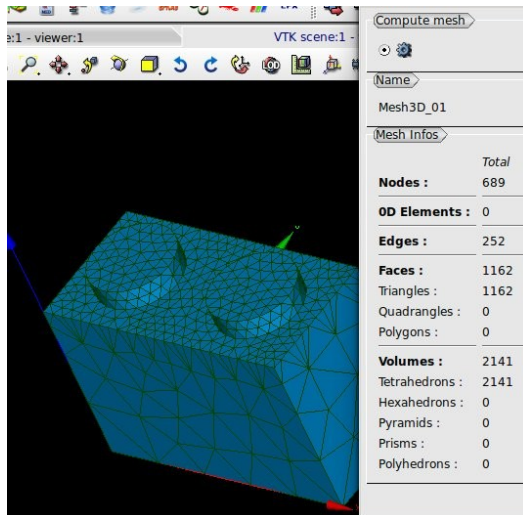


Рис. 42. Пересчитанная сетка.

Этап 3. Экспорт сетки из Salome в OpenFoam

Для начала требуется создать в Salome группу (**Mesh — Create Group**), отметить **Volume** и в строке Mesh выбрать необходимую сетку (например, Mesh3D_01). Далее отметить **Select All** и нажать **Apply and Close**. Далее выбрать саму сетку (именно сетку в списке объектов, иначе проверка checkMesh выдаст ошибку) и сохранить файл (**File — Export — UNV file**) в корневую папку проекта OpenFoam (туда, где находятся папки 0, system, constant). Откройте корневую папку проекта в BlueCFD Core Terminal (папку с названием проекта, а не папку 0). Экспорт сетки в OpenFoam осуществляется с помощью команды **ideasUnvToFoam Mesh3D_01.unv**. Визуализируйте сетку с помощью команды **paraFoam**. Это один из способов проверки корректности сетки (рис. 43). Далее ввести **checkMesh** для проверки правильности импорта сетки.

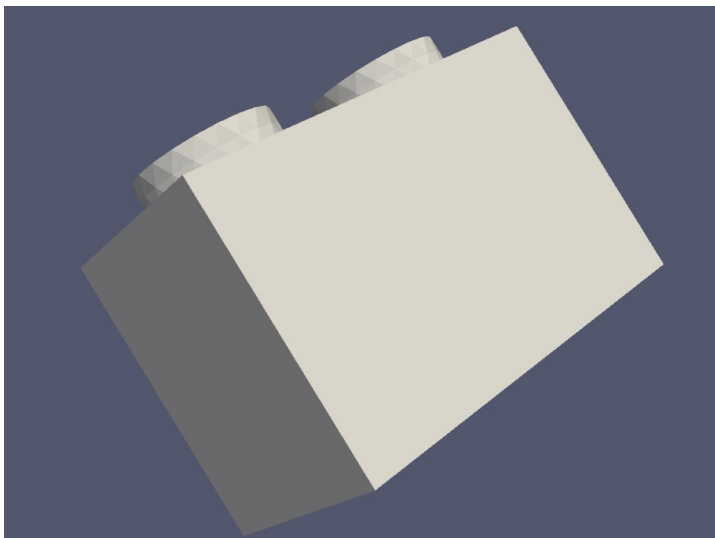


Рис. 43. Открытие Lego в ParaView.

Таким образом, в данной работе Вы познакомились с инструментарием программы Salome и научились создавать объекты и сетку для расчётов.

Занятие №5

Тема работы: Моделирование течения в трубе с обратным уступом и сужением.

Цель: Рассмотреть пример моделирования ламинарного течения невязкого газа в трубе с обратным уступом и сужением, изучить модели жидкостей доступные в OpenFoam.

Постановка задачи

Область имеет 2 измерения, состоит из короткого входа, обращенной назад по потоку ступени и конусообразного сопла на выходе.

Начальные условия:

$$U = 0 \text{ м/с}, p = 0 \text{ Па.}$$

Граничные условия:

- Вход (слева) с фиксированной скоростью $U = (10, 0, 0)$ м/с;
- Выход (справа) с фиксированным давлением $p = 0$ Па;
- Условия прилипания (noSlip) к стенке на остальных границах для скорости.

Главные уравнения:

- Уравнение неразрывности для несжимаемой жидкости

$$\nabla \cdot U = 0.$$

- Уравнение количества движения для устойчивого потока

$$\nabla \cdot (UU) + \nabla \cdot R = -\nabla p,$$

где p – кинематическое давление и (упрощенно) $R = \nu_{eff} \nabla U$ – компонента вязкого напряжения с эффективной кинематической вязкостью ν_{eff} , вычисленная из выбранных моделей переноса и турбулентности.

Ход работы

Этап 1. Запуск программы

1. Из папки C:/Program Files/blueCFD-Core-2017/OpenFOAM-5.x/tutorials/incompressible/simpleFoam/ скопируйте папку pitzDaily в вашу папку.
2. Кликнув по папке C:/\$ВАША_ПАПКА\$/pitzDaily правой кнопкой мыши, выберите «Open in blueCFD-Core terminal».

3. Командой `blockMesh` построить сетку.
4. В папке `C:/ВАША_ПАПКА$/pitzDaily/system` найдите файл `controlDict`. В этом файле есть строка:
`application simpleFoam;`
которая указывает на то, какой решатель используется. Это означает, что, чтобы запустить решение программы, в командной строке надо ввести `simpleFoam`. Решатель `simpleFoam` выполняет расчет для устойчивого несжимаемого потока.
5. Запустите расчет командой `simpleFoam`.

Примечание: первая буква команды маленькая, слово Foam – с большой буквы.

6. Визуализируйте решение.

Этап 2. Изменение граничных условий

В файле `U (C:/pitzDaily/0)` находятся краевые условия для скорости.

Измените только верхнюю границу, например, поставить условие непроницаемости – `zeroGradient`, на выходе из трубы поставьте условие `noSlip`. Снова запустите решатель (предварительно удалив из папки предыдущее решение).

Примечание: команду `blockMesh` вводить уже не нужно.

Сравните получившееся решение с предыдущим.

Этап 3. Изменение транспортной модели

В `OpenFoam` существует библиотека моделей жидкости, каждая модель имеет собственное уравнение, описывающее значение кинематической вязкости и ее связь со скоростью деформации.

Модели жидкостей и приметы ввода значений в `transportModel`:

`Newtonian` – ньютоновская модель, кинематическая вязкость ν – постоянная величина (рис. 44):

$$\nu = const$$


```
transportModel Newtonian;
nu          nu [ 0 2 -1 0 0 0 0 ] 1.5e-05;
```

Рис. 44. Ньютоновская модель кинематической вязкости.

Bird-Carreau model:

$$v = v_{\infty} + (v_0 - v_{\infty}) \left[1 + (k \dot{\gamma})^a \right]^{\frac{n-1}{a}},$$

где v_{∞} и v_0 – кинематическая вязкость при бесконечной и нулевой скорости деформации соответственно, параметр a по умолчанию имеет значение 2, $\dot{\gamma}$ – скорость деформации, n и k – параметры реологических моделей (рис. 45).

```
transportModel BirdCarreau;
BirdCarreauCoeffs
{
    nu0          nu0 [ 0 2 -1 0 0 0 0 ] 1e-03;
    nuInf        nuInf [ 0 2 -1 0 0 0 0 ] 1e-05;
    k            k [ 0 0 1 0 0 0 0 ] 1;
    n            n [ 0 0 0 0 0 0 0 ] 0.5;
}
```

Рис. 45. Модель Bird-Carreau.

Cross Power Law model (рис. 46):

$$v = v_{\infty} + \frac{v_0 - v_{\infty}}{1 + (m \dot{\gamma})^n},$$

где m – параметр реологической модели (рис. 46).

```
transportModel CrossPowerLaw;
CrossPowerLawCoeffs
{
    nu0          nu0 [ 0 2 -1 0 0 0 0 ] 1e-03;
    nuInf        nuInf [ 0 2 -1 0 0 0 0 ] 1e-05;
    m            m [ 0 0 1 0 0 0 0 ] 1;
    n            n [ 0 0 0 0 0 0 0 ] 0.5;
}
```

Рис. 46. Модель Cross Power Law.

Power Law model – степенная модель ограничивает значение вязкости в определённом диапазоне от минимальной ν_{min} до максимальной ν_{max} (рис. 47):

$$\nu = k \dot{\gamma}^{n-1} \nu_{min} \leq \nu \leq \nu_{max}$$

```
transportModel powerLaw;
powerLawCoeffs
{
    nuMax          nuMax [ 0 2 -1 0 0 0 0 ] 1e-03;
    nuMin          nuMin [ 0 2 -1 0 0 0 0 ] 1e-05;
    k              k     [ 0 2 -1 0 0 0 0 ] 1e-05;
    n              n     [ 0 0 0 0 0 0 0 ] 1;
}
```

Рис. 47. Модель Power Law.

Herschel-Bulkley model – модель Хершеля-Балкли

$$\nu = \min\left(\nu_0, \frac{\tau_0}{\dot{\gamma}} + k \dot{\gamma}^{n-1}\right),$$

здесь τ_0 – предел текучести вязкопластической жидкости, для степенной жидкости равный нулю. Ниже определенного предельного значения напряжений среда ведет себя как твердое тело, выше этого предела – как несжимаемая вязкая жидкость (рис. 48).

```
transportModel HerschelBulkley;
HerschelBulkleyCoeffs
{
    nu0           nu0 [ 0 2 -1 0 0 0 0 ] 1e-03;
    tau0          tau0 [ 0 2 -2 0 0 0 0 ] 1;
    k             k    [ 0 2 -1 0 0 0 0 ] 1e-05;
    n             n    [ 0 0 0 0 0 0 0 ] 1;
}
```

Рис. 48. Модель Хершеля-Балкли.

В файле `transportProperties` (в папке `constant`), измените тип модели жидкости в строке `transportModel`. Сравните полученные результаты для моделей `Newtonian` и `BirdCarreau`.

Этап 4. Изменение параметров расчета

В файле `controlDict` присвойте значение 150 времени, когда расчет заканчивается (`endTime`), и значение 10 – интервалу, через который производится запись расчета (`writeInterval`).

Произведите расчет и проанализируйте полученный результат. Визуализируйте решение с помощью `ParaView`, добавьте фильтр `Glyph`, чтобы увидеть распределение скоростей по направлению в потоке. Для более удобного восприятия уменьшите количество стрелок до 500 (блок `Properties`, поле `Maximum Number Of Sample Points` – число векторов), а их размер уменьшите в 2 раза (поле `Scale Factor` – масштаб векторов, в выпадающем меню выберите множитель `0.5x`) (рис. 49).

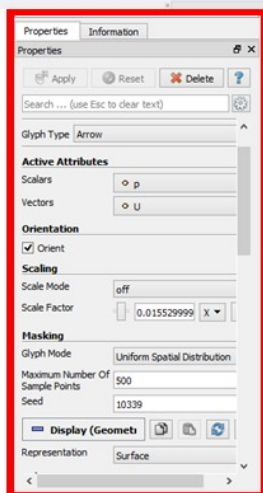
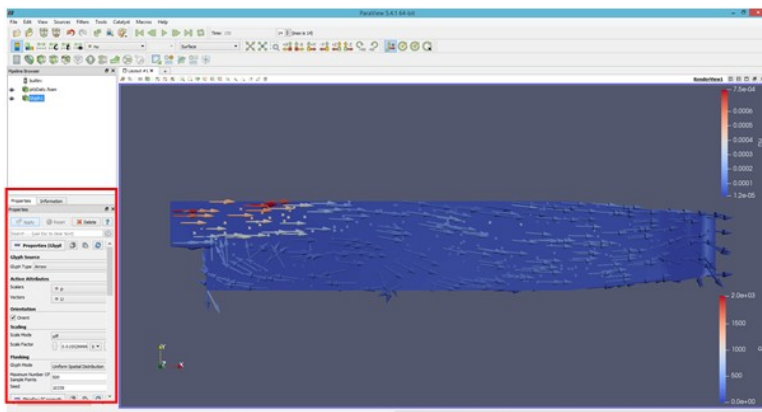


Рис. 49. Изменение количества векторов и их масштаба.

Таким образом, в данной работе было смоделировано течение устойчивой несжимаемой жидкости в трубе с обратным уступом при различных моделях динамической вязкости.

Занятие №6

Тема работы: Моделирование быстрого открытия клапана бака, заполненного водой под давлением.

Цель: Решить задачу о быстром открытии клапана бака, заполненного жидкостью под давлением. Для исследования особенностей распространения волн давления жидкость моделируется как сжимаемая. Использовать решатель sonicLiquidFoam. По результатам расчета в пакете ParaView построить поле давления.

Условия задачи

Условия задачи показаны на рис. 50. Давление p на выходе $p=0$, введена декартова прямоугольная система координат с осями x и y .

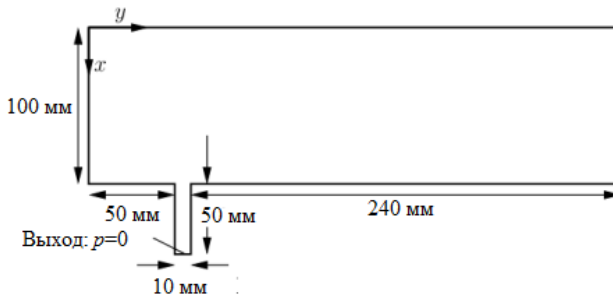


Рис. 50. Условия задачи о быстром открытии клапана бака, заполненного жидкостью под давлением.

Начальные условия:

$$U = 0 \text{ м/с}, p = 100 \text{ бар},$$

где U – скорость.

Граничные условия:

- Внешняя стенка с условием прилипания для скорости и нулевым градиентом давления;
- Выход клапана с фиксированным давлением $p = 0$ Па и нулевым градиентом скорости;
- Симметрия скорости и давления относительно оси x ;

- Отсутствие вычислений (empty) скорости и давления на передней и задней стенках (относительно оси x).

Динамическая вязкость воды $\mu = 1$ мПа·с.

Термодинамические параметры: плотность воды $\rho = 1000$ кг/м³, относительное давление $p_0 = 1$ бар (атмосферное), сжимаемость воды $\beta = 4,54 \cdot 10^{-7}$ с²/м².

Главные уравнения:

- Уравнение неразрывности для сжимаемой жидкости

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = 0,$$

где t – время.

- Зависимость плотности жидкости от давления

$$\rho = \rho_0 + \beta(p - p_0),$$

где ρ_0 – плотность при давлении p_0 .

- Уравнение количества движения для ньютоновской жидкости

$$\frac{\partial (\rho \vec{U})}{\partial t} + \nabla \cdot (\rho \vec{U} \vec{U}) - \nabla \cdot \mu \nabla \vec{U} = -\nabla p.$$

Ход работы:

Этап 1. Запуск программы

1. Из папки C:/Program Files/blueCFD-Core-2017/OpenFOAM-5.x/tutorials/compressible/sonicLiquidFoam/ скопируйте папку decompressionTank в вашу папку.
2. Кликнув по папке C:/\$ВАША_ПАПКА\$/decompressionTank правой кнопкой мыши, выберите «Open in blueCFD-Core terminal».
3. Командой blockMesh построить сетку.
4. В папке C:/\$ВАША_ПАПКА\$/decompressionTank/system найдите файл controlDict. В этом файле есть строка:
application sonicLiquidFoam;
которая указывает на то, какой решатель используется. Это означает, что для запуска решения программы в командной строке надо ввести sonicLiquidFoam. SonicLiquidFoam: выполняет расчет для сжимаемой жидкости.

5. Запустите счет командой sonicLiquidFoam.

Этап 2. Визуализация решения

Визуализируйте решение с помощью команды paraFoam. Посмотрите распределение давления в момент времени $9 \cdot 10^{-5}$ с (рис. 51) и значение модуля скорости в конечный момент времени (рис. 52). Проследите за распространением волны давления.

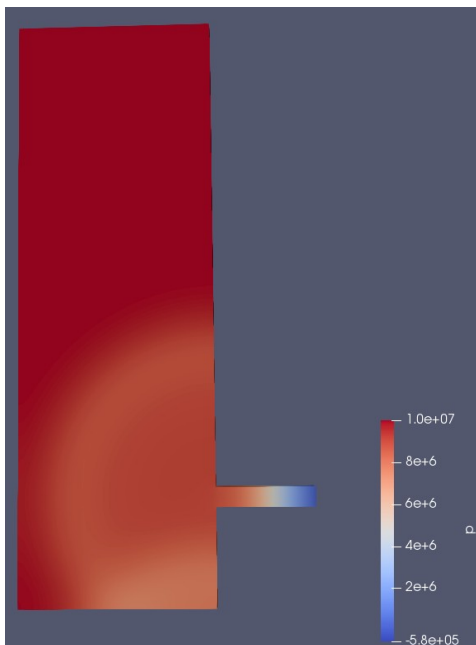


Рис. 51. Распределение давления в момент времени $9 \cdot 10^{-5}$ с.

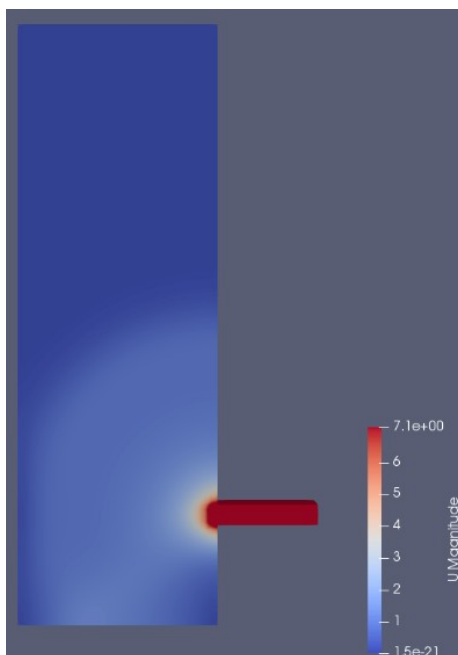


Рис. 52. Значение модуля скорости в конечный момент времени.

Этап 3. Изменение термодинамических свойств

Измените сжимаемость жидкости (psi) на $10^{-7} \text{ c}^2/\text{m}^2$. Посмотрите распределение давления в момент времени $9 \cdot 10^{-5} \text{ c}$ (рис. 53) и значение модуля скорости в конечный момент времени (рис. 54). Сравните получившееся решение с предыдущим.

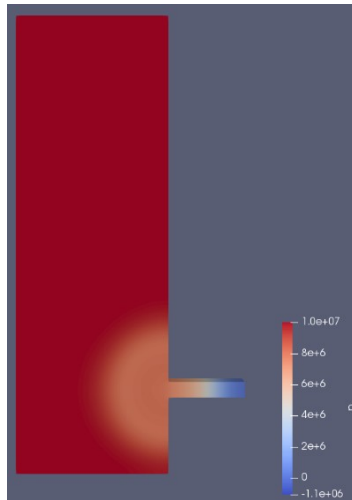


Рис. 53. Распределение давления в момент времени $9 \cdot 10^{-5}$ с при сжимаемости жидкости $10^{-7} \text{ с}^2/\text{м}^2$.

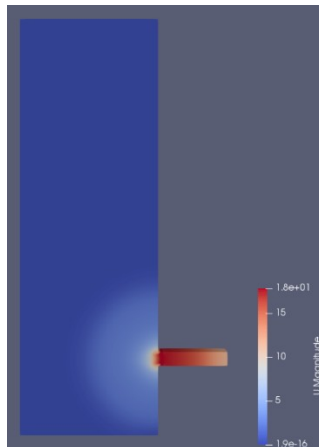


Рис. 54. Значение модуля скорости в конечный момент времени при сжимаемости жидкости $10^{-7} \text{ с}^2/\text{м}^2$.

Таким образом, в данной работе было смоделировано распространение волн давления при быстром открытии клапана бака.

Занятие №7

Тема работы: Расчёт обтекания бруса.

Цель: Решить задачу о плоскопараллельном обтекании бруса несжимаемой вязкой жидкостью при различных числах Рейнольдса. Для формирования геометрии используется параметр $h = 1$ см (0,01 м). Использовать модель k-omega SST. Сформировать геометрию расчетной области средствами SALOME (GEOMETRY) в блочно-гексаэдрическом виде; определить на геометрии поверхности для задания граничных условий; построить гексаэдрическую расчетную сетку средствами SALOME (MESH); в пакете OpenFOAM провести расчет стационарного состояния для одного из следующих чисел Re: 100, 1000, 10000, 100000; по результатам расчета в пакете ParaView построить поля скорости и давления; построить распределение давления по периметру бруса.

Условия задачи

Условия задачи показаны на рис. 55.

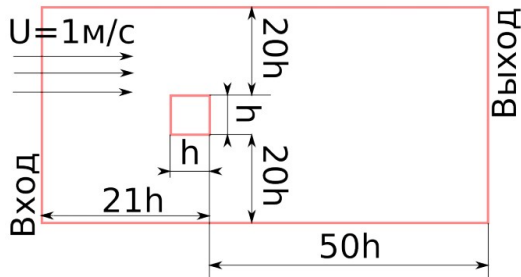


Рис. 55. Условия задачи об обтекании бруса.

Ход работы:

Этап 1. Создание блочной геометрии

По схеме расчетной области создадим блочную геометрию. Для этого создайте параллелепипед, нажав **Create a Box** и укажите его размеры (рис. 56).

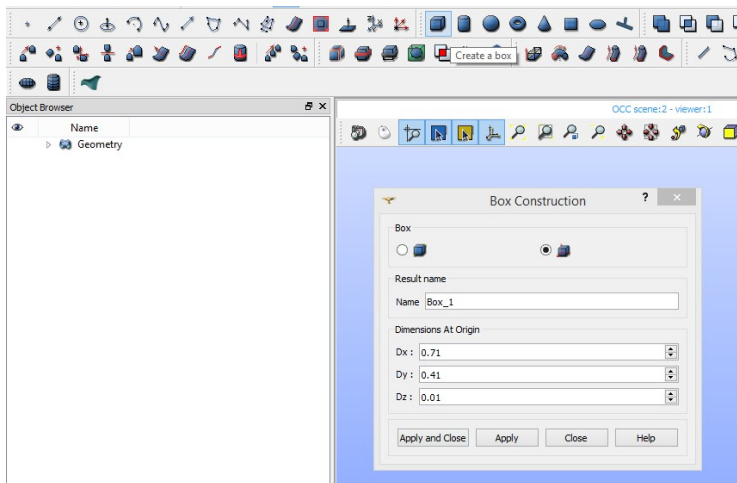


Рис. 56. Создание параллелепипеда.

Нажмите **Apply** и создайте ещё один параллелепипед меньших размеров (рис. 57). Нажмите **Apply and Close**. Переместите его с помощью операции **Translation**, предварительно убрав флажок с **Create a copy** (рис. 58).

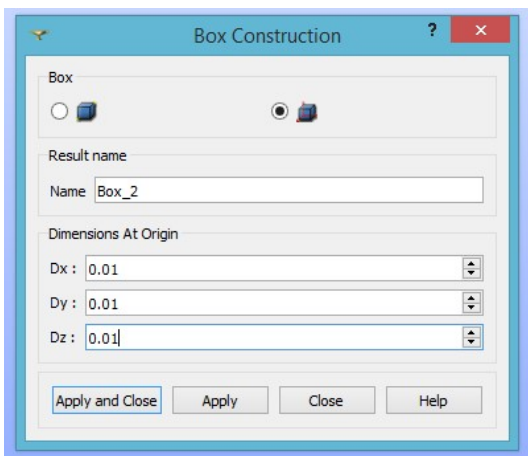


Рис. 57. Создание параллелепипеда меньших размеров.

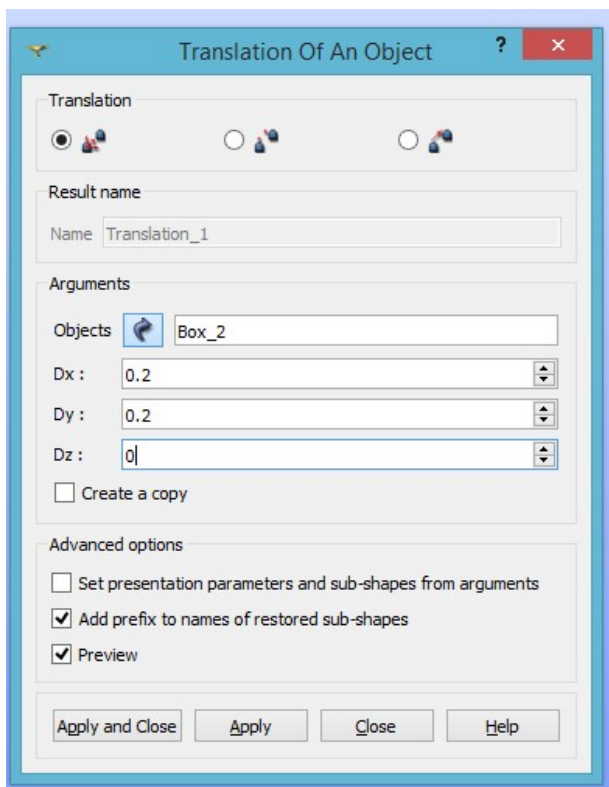


Рис. 58. Перемещение параллелепипеда.

С помощью логической операции **Cut** вырежьте Box_2 из Box_1 (рис. 59).

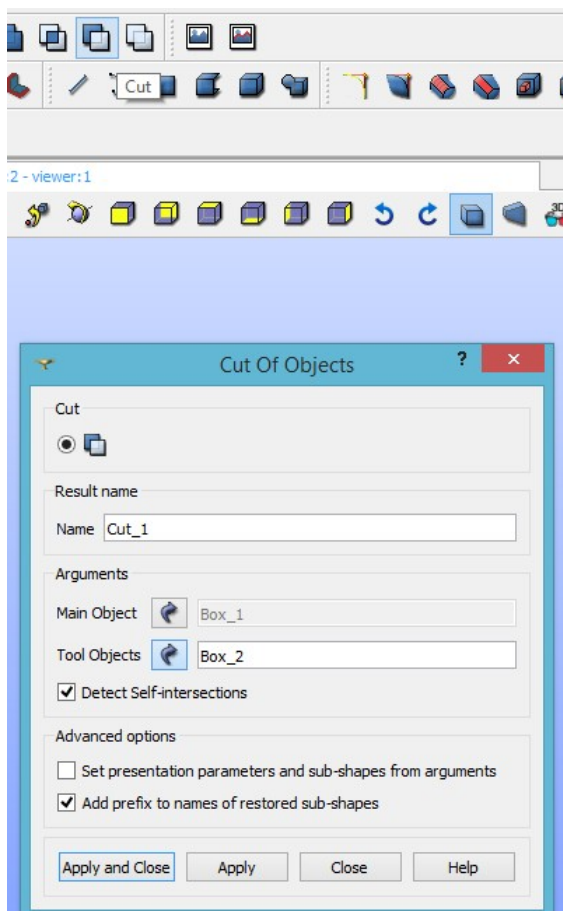


Рис. 59. Применение логической операции Cut.

Теперь необходимо построить 4 секущие плоскости, которые поделят получившийся объект на 8 блоков. Для этого нажмите **Create a Plane** и постройте плоскость по принадлежащей ей точке (в качестве точки укажите вершину Box_2) и вектору нормали к этой плоскости (направьте его вдоль ребра Box_2) (рис. 60). Таким же образом создайте еще три плоскости.

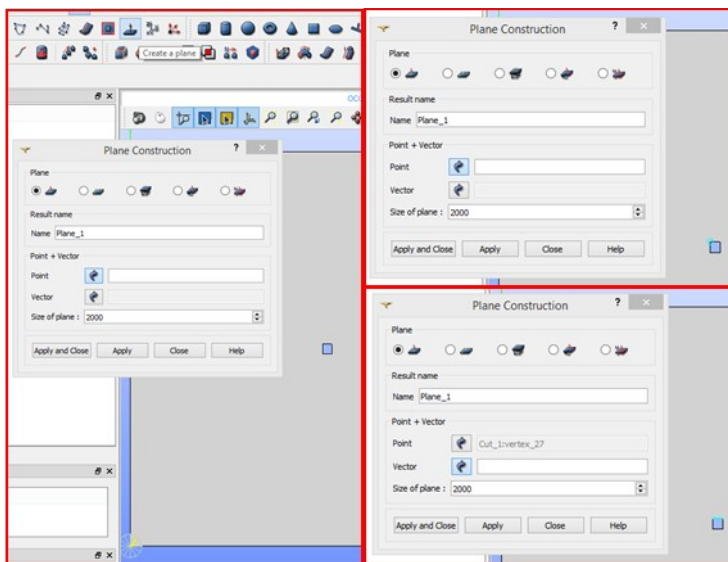


Рис. 60. Построение плоскости.

Далее нажмите **Partition** и разделите Cut_1 с помощью полученных 4 плоскостей на отдельные блоки (Solid) (рис. 61). Таким способом получается единый объект, состоящий из блоков.

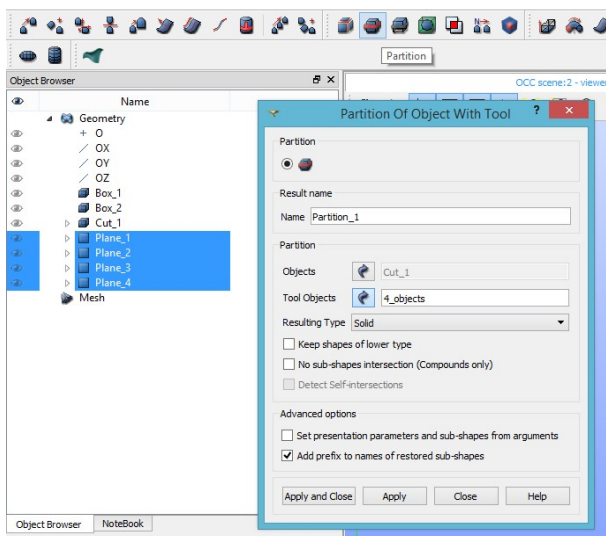


Рис. 61. Применение операции Partition.

Этап 2. Построение расчетной сетки

Далее необходимо перейти в модуль Mesh и создать на блочной геометрии расчетную сетку. Для этого нажать **Create Mesh**, выбрать в качестве геометрии для расчета **Partition_1**, в качестве алгоритма – **Hexahedron (i,j,k)** (гексаэдрические ячейки) (рис. 62).

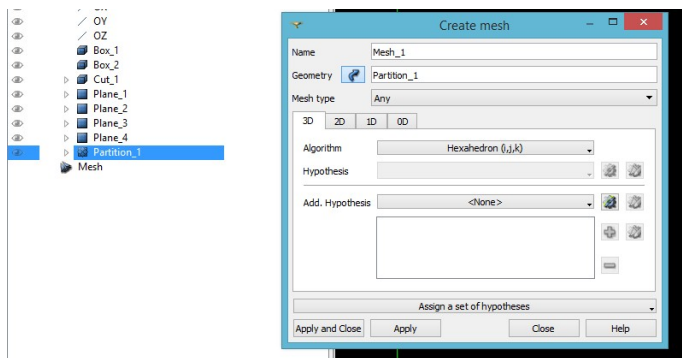


Рис. 62. Создание сетки с гексаэдрическими ячейками.

Также необходимо назначить гипотезы для расчета с помощью нажатия **Assign a set of hypotheses**. В выпадающем списке выбрать **3D: Automatic Hexahedralization** и в строке **Number of Segments** указать 10 (рис. 63).

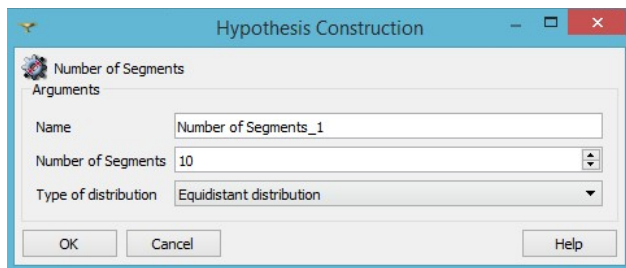


Рис. 63. Указание числа сегментов для гипотезы построения сетки.

Рассчитаем сетку с помощью команды Compute (рис. 64).

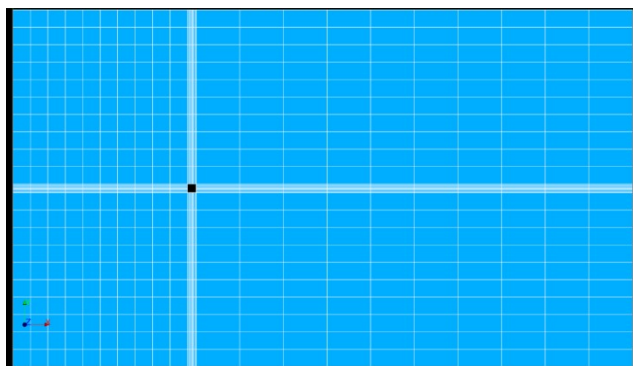


Рис. 64. Получившаяся сетка для задачи об обтекании бруса.

Далее необходимо перейти в модуль Geometry и с помощью операции **Explode** разбить Partition_1 на отдельные поверхности (рис. 65).

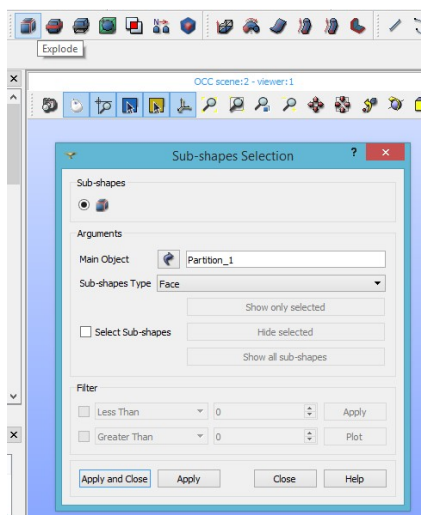


Рис. 65. Разбиение Partition_1 на отдельные поверхности.

Теперь необходимо объединить поверхности, принадлежащие внешним стенкам канала, в соответствующие группы. Для более удобной работы переименуйте поверхности, составляющие входную (inlet) стенку в i1, i2, i3, верхнюю (upper) u1, u2, u3 и т.д. (рис. 66). Стенки бруса (cubeWalls) также следует объединить в группу.

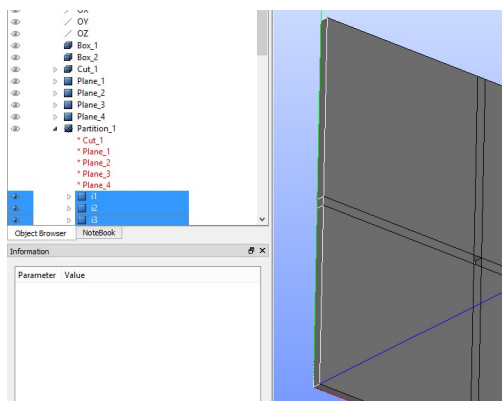


Рис. 66. Переименование поверхностей.

Далее следует перейти в модуль Mesh и сгруппировать поверхности по их принадлежности стенкам. Нажмем **Create Group**, в появившемся окне выберем для сетки **Mesh_1** тип элемента **Face**, тип группы **Group on Geometry**, назовем группу «inlet», в поле **Geometry** выберем **Direct geometry selection** и добавим, удерживая клавишу Ctrl, поверхности i1, i2, i3. Нажмем **Apply** (рис. 67). Аналогичным образом создадим еще 6 групп для оставшихся стенок.

Примечание: если не сгруппировать какие-либо грани, они автоматически попадут в группу **defaultFaces** при экспорте сетки в OpenFoam (см. список возможных ошибок).

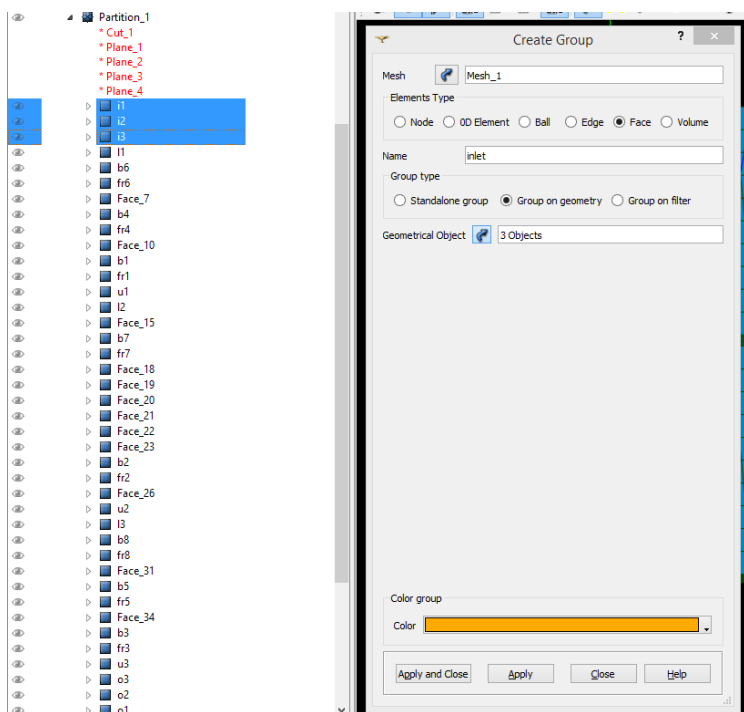


Рис. 67. Группировка поверхностей для создания сетки.

Выберем Mesh_1 и сохраним UNV файл в корневую папку проекта OpenFoam. В качестве корневой папки используйте копию папки pitzDaily из предыдущей работы.

Примените **ideasUnvToFoam** для импорта сетки (при этом не применяйте blockMesh).

В файле «boundary» (из папки polyMesh) замените типы стенок на wall (кроме inlet и outlet, у них остается patch).

Примечание: тип wall означает непроницаемую стенку, patch – вход или выход.

Этап 3. Начальные условия

После создания сетки следует задать начальные условия: невозмущенные поля давления (p), скорости (U), кинетической энергии турбулентности (k), частоты диссипации кинетической энергии турбулентности (ω), турбулентной вязкости (ν_t). Во всех начальных условиях названия границ должны соответствовать названиям в файле boundary. Файлы остальных величин в папке «0» необходимо удалить.

В нулевой момент времени $p=0$, $U=(0 \ 0 \ 0)$, $k=1e-3$, $\omega=1$, $\nu_t=0$.

Этап 4. Граничные условия

- В файле «U» задаются граничные условия скорости: на входе (inlet) $U_x=1$ м/с (т.е. (1 0 0)), тип fixedValue (фиксированная величина), на выходе (outlet) задаётся нулевой градиент (zeroGradient) для U , на стенках канала (upper, lower, front, back) – условие проскальзывания (slip), на стенках бруса (cubeWalls) – условие прилипания (noSlip).

Примечание: slip пишется с маленькой буквы, в случае noSlip – с большой.

- В файле «k»: на входе (inlet) $k=1e-3$, тип fixedValue (фиксированная величина), на выходе (outlet) задаётся нулевой градиент (zeroGradient) для k , на стенках канала (upper, lower, front, back) – условие проскальзывания (slip), на стенках бруса (cubeWalls) – пристеночная функция для k (type kqRWallFunction;

value uniform 0.375, первая строка в файле – тип (пристеночная функция), вторая – её величина).

- В файле «omega»: на входе (inlet) omega=1, тип fixedValue (фиксированная величина), на выходе (outlet) задаётся нулевой градиент (zeroGradient) для omega, на стенках канала (upper, lower, front, back) – условие проскальзывания (slip), на стенках бруса (cubeWalls) – пристеночная функция для omega (type omegaWallFunction; value \$internalField).

- В файле «nut»: на входе (inlet) nut=0, тип calculated, на выходе (outlet) задаётся nut=0 (тип calculated), на передней и задней стенках канала (front, back), условие проскальзывания (slip), на верхней upper и нижней lower границах задаётся пристеночная функция (type nutkWallFunction; value uniform 0), на стенках бруса (cubeWalls) – пристеночная функция (type nutkWallFunction; value uniform 0).

- В файле «p» задаются граничные условия для давления: на входе (inlet) – нулевой градиент (zeroGradient), тип fixedValue (фиксированная величина), на выходе (outlet) задаётся давление p=0 (тип fixedValue), на стенках канала (upper, lower, front, back) – условие проскальзывания (slip), на стенках бруса (cubeWalls) – условие непроницаемости (zeroGradient).

Этап 5. Константы модели

Исходя из заданных h (1 см по умолчанию) и скорости на входе (1 м/с по умолчанию) выбираем такую вязкость, чтобы соответствовала заданному числу Re:

- 1) Re=100, вязкость — $1e-4$ (при $h=1$ см, $U_x=1$ м/с)
- 2) Re=1000, вязкость — $1e-5$ (при $h=1$ см, $U_x=1$ м/с)
- 3) Re=10000, вязкость — $1e-6$ (при $h=1$ см, $U_x=1$ м/с)
- 4) Re=100000, вязкость — $1e-7$ (при $h=1$ см, $U_x=1$ м/с)

Этап 6. Параметры счёта

Для первых двух случаев можно не использовать моделирование турбулентности (turbulence off в файле turbulenceProperties), для третьего и четвертого обязательно (turbulence on в

файле turbulenceProperties). Модель турбулентности – kOmegaSST (задана после слова RASModel в том же файле).

Задав константы модели, переходим к параметрам счёта. Расчёт проводится по итерациям (шаг по времени = 1), вывод через каждые 10 шагов. Общее число итераций – 150 или 1000. Коэффициент релаксации для скорости: $\gamma_U = 0.9$ (рис. 68), задан в файле fvSolution.

```
relaxationFactors
{
    equations
    {
        U           0.9; // 0.9 is more stable but 0.95 more convergent
        ".*"       0.9; // 0.9 is more stable but 0.95 more convergent
    }
}
```

Рис. 68. Коэффициент релаксации для скорости.

Решения

Решения отобразить с помощью ParaView, смотреть скорость. Убедиться, что при заданной сетке не наблюдаются дорожки скорости.

Примечание: используемый решатель записан в файле controlDict.

Чтобы наблюдать дорожки скорости, необходимо изменить расчётную сетку в SALOME. Для этого необходимо удалить старую сетку и создать новую. Следует нажать **Create Mesh**, выбрать в качестве геометрии для расчета **Partition_1**, в качестве алгоритма 3D – **Hexahedron (i,j,k)** (гексаэдрические ячейки), в качестве алгоритма 2D – **Quadrangle: Mapping**, в качестве алгоритма 1D – **Wire Discretisation**, гипотеза 1D – **Local Length** с длиной (**Length**) 0,005 (м). Рассчитайте сетку с помощью команды Compute. Получится сетка, изображённая на рис. 69. У такой сетки будет только два сегмента вблизи бруса, но более детально будет разбита область за брусом. Повторите все последующие действия для этой сетки (создание групп поверхностей по их принадлежности стенкам, экспорт файла сетки, импорт сетки в OpenFOAM, исправление типов стенок, задание

начальных и граничных условий, констант модели и параметров счёта). Проведите снова расчёт средствами OpenFOAM, отобразите с помощью ParaView решения (рис. 70). Пронаблюдайте дорожки скорости.

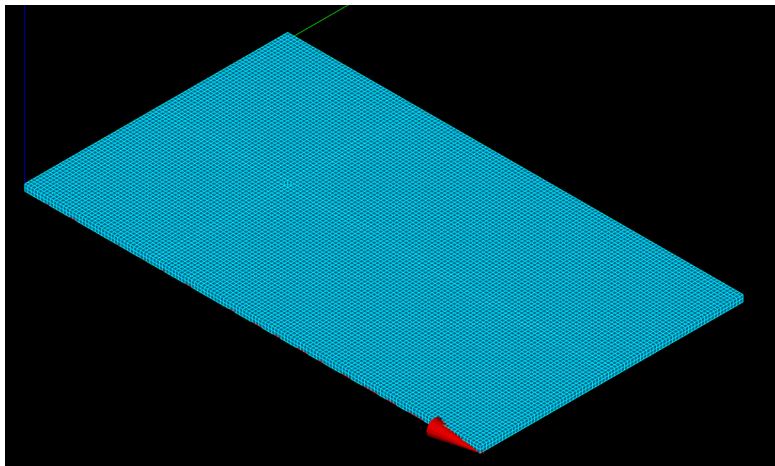


Рис. 69. Получившаяся сетка с длиной ячейки 0,005 м.

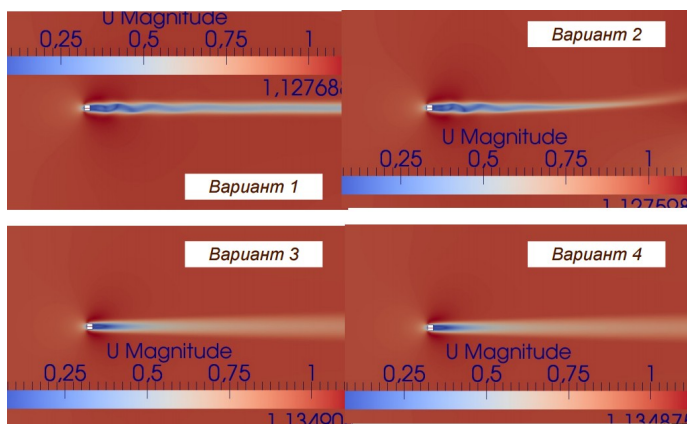


Рис. 70. Примерный вид решения задачи об обтекании бруса.

Отобразите также векторы скорости с помощью фильтра Glyph (рис. 71). В качестве типа стрелок (Glyph Mode) выберите отображение каждой N -ой точки (Every Nth Point), $N=10$. Scale Factor поставьте в 4 раза меньше первоначального. Отображаемое скалярное свойство – давление, вектор – скорость. Понаблюдайте за процессом в динамике.

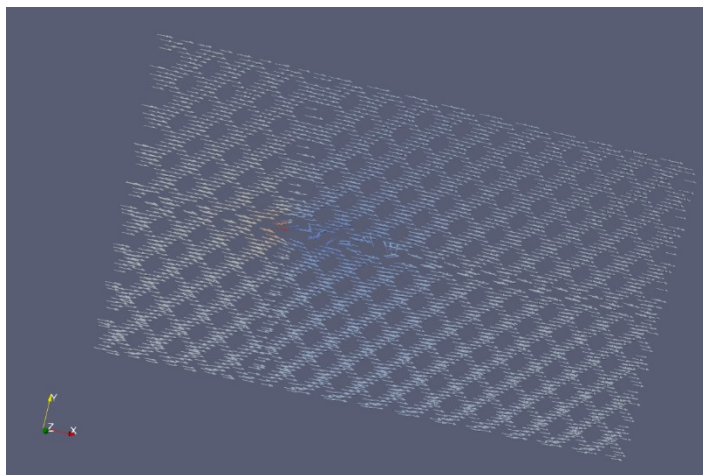


Рис. 71. Применение фильтра Glyph для наблюдения дорожки скорости.

Проведите для заданной геометрии средствами SALOME создание тетраэдрической сетки и её расчёт. Для этого следует нажать **Create Mesh**, выбрать в качестве геометрии для расчета **Partition_1**, в качестве алгоритма 3D – **NETGEN 1D-2D-3D** (тетраэдрические ячейки), гипотеза – **NETGEN 3D Simple Parameters** с числом сегментов 1D (**Number of Segments**) 5. Рассчитайте сетку с помощью команды Compute. Получится сетка, изображённая на рис. 72, имеющая 6880 ячеек, что меньше, чем у гексаэдрической с числом сегментов 10, содержащей 8000 ячеек.

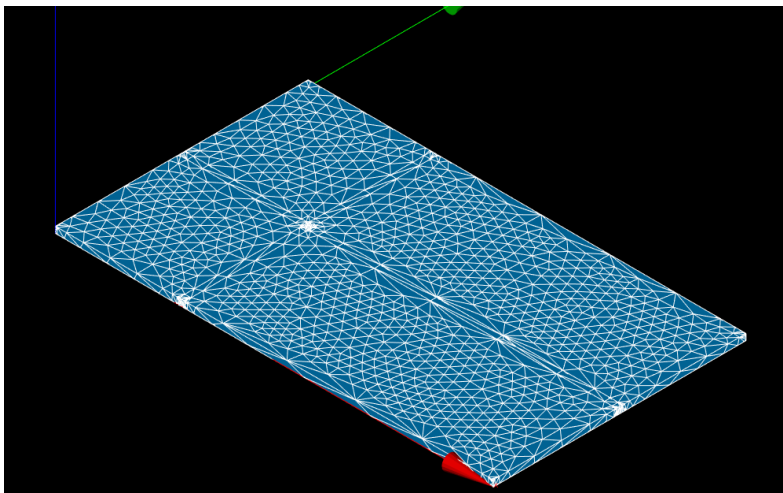


Рис. 72. Тетраэдрическая сетка для задачи об обтекании бруса.

Повторите все последующие действия для этой сетки (создание групп поверхностей по их принадлежности стенкам, экспорт файла сетки, импорт сетки в OpenFOAM, исправление типов стенок, задание начальных и граничных условий, констант модели и параметров счёта). Проведите снова расчёт средствами OpenFOAM. Обратите внимание, что в момент времени 26 с скорость резко возрастает, программа выдаёт ошибку и прекращает расчёт (рис. 73).

```
Time = 26
smoothSolver: Solving for Ux, Initial residual = 0.483151, Final residual = 2.27259e+009, No Iterations 1000
smoothSolver: Solving for Uy, Initial residual = 0.512616, Final residual = 3.61782e+009, No Iterations 1000
smoothSolver: Solving for Uz, Initial residual = 0.570056, Final residual = 2.561e+009, No Iterations 1000
Generating stack trace...

Backtrace:
ZN10StackTraceCIEv [0x705c1465+0x25]
module: C:\PROGRA~1\BLUECF-1\ThirdParty-5.x\platforms\mingw_w64GccDPInt32\lib\libstack_trace.dll
ZN4FoamSerror10printStackERNS_7ostreamE [0x7a1d88+0x218]
module: C:\PROGRA~1\BLUECF-1\OpenFOAM-5.x\platforms\mingw_w64GccDPInt32opt\lib\libOpenFOAM.dll
ZN4Foam6sigPep13sigPepHandlerE1 [0x7a2bf3+0x33]
module: C:\PROGRA~1\BLUECF-1\OpenFOAM-5.x\platforms\mingw_w64GccDPInt32opt\lib\libOpenFOAM.dll
(No symbol) [0x403cad]
module: C:\PROGRA~1\BLUECF-1\OpenFOAM-5.x\platforms\mingw_w64GccDPInt32opt\bin\simpleFoam.exe
_C_specific_handler [0x7ffca43c18a6+0x96]
module: C:\windows\SYSTEM32\ntdll.dll
0_chkstk [0x7ffca432f2d+0x9d]
module: C:\windows\SYSTEM32\ntdll.dll
RtlRaiseException [0x7ffca4394547+0xf67]
```

Рис. 73. Ошибка, вызванная резким ростом скорости.

Указанная ошибка вызвана тем, что в тетраэдрической сетке, несмотря на меньшее число ячеек, вблизи бруса имеются очень маленькие ячейки треугольного сечения, сходящиеся в одном узле (рис. 74). Скорость прохождения таких ячеек из-за их малой длины вблизи узла стремительно возрастает. В гексаэдрической сетке количество ячеек даже больше при разбиении с числом сегментов 10, но таких узлов с ячейками малой длины у неё нет, что позволяет избежать указанной ошибки.

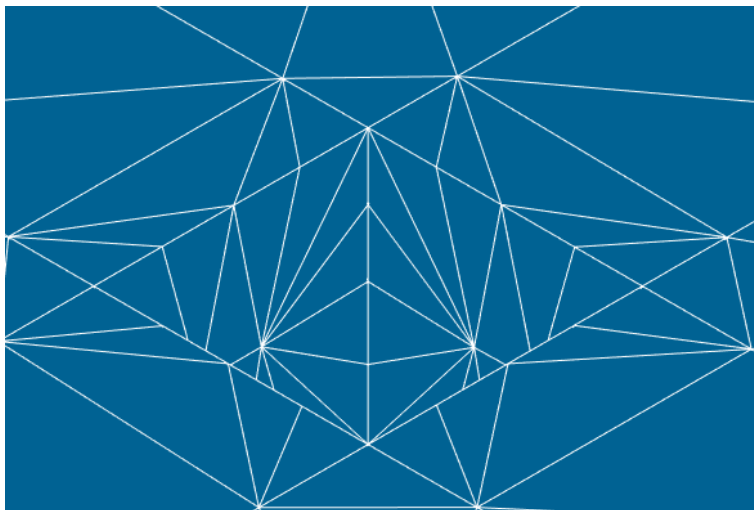


Рис. 74. Мелкое разбиение тетраэдрической сетки вблизи бруса.

Таким образом, в данной работе Вы познакомились с созданием геометрии задачи об обтекании бруса и расчётной сетки, экспортом сетки в OpenFoam, заданием начальных и граничных условий. Был проведён расчёт и отображены дорожки скорости.

Занятие №8

Тема работы: Моделирование движения двухфазной жидкости в цилиндрической трубе.

Цель: Решить задачу о течении двухфазной жидкости в цилиндрической трубе: создать геометрию и сетку (гексаэдрическую) средствами SALOME; задать граничные условия; средствами пакета OpenFOAM выполнить расчет полей скорости и давления; выполнить визуализацию полученных данных средствами пакета ParaView.

Условия задачи

Условия задачи показаны на рис. 75, радиус цилиндра $r=10$ см, длина $l=2$ м.



Рис. 75. Условие задачи о движении двухфазной жидкости в трубе.

Ход работы:

Этап 1. Построение геометрии и расчетной сетки

Средствами SALOME создаётся геометрия, как на рис. 76, и расчётная сетка, как на рис. 77. Цилиндрическая труба задаётся с помощью объекта **Divided Cylinder**, требующего для создания значений радиуса и длины образующей цилиндра (направ-

ленной вдоль оси y). Задайте эти параметры в соответствии с условиями задачи.

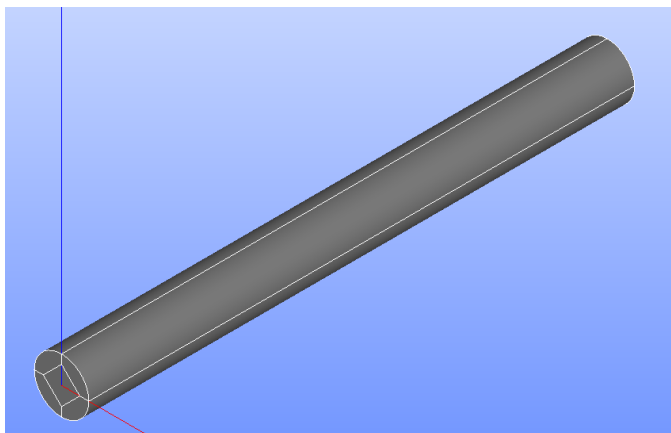


Рис. 76. Геометрия задачи о движении двухфазной жидкости в трубе.

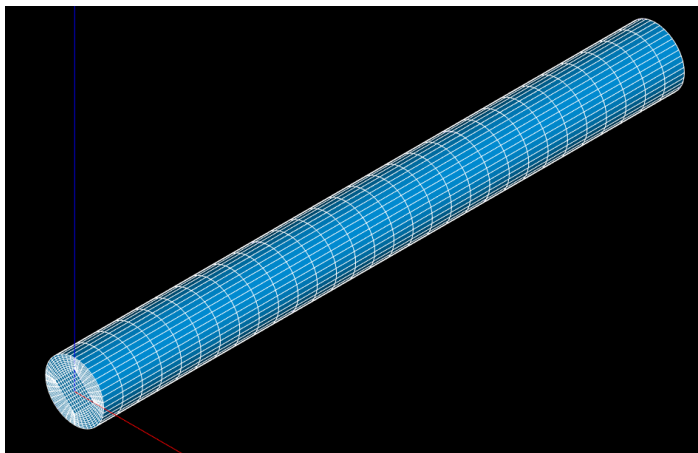


Рис. 77. Расчётная сетка для задачи о движении двухфазной жидкости в трубе.

Добавьте две точки с координатами (0, 0,67, 0) и (0, 1,33, 0). Создайте плоскости, проходящие через эти точки перпендикулярно оси u . У Вас должна получиться геометрия, показанная на рис. 78.

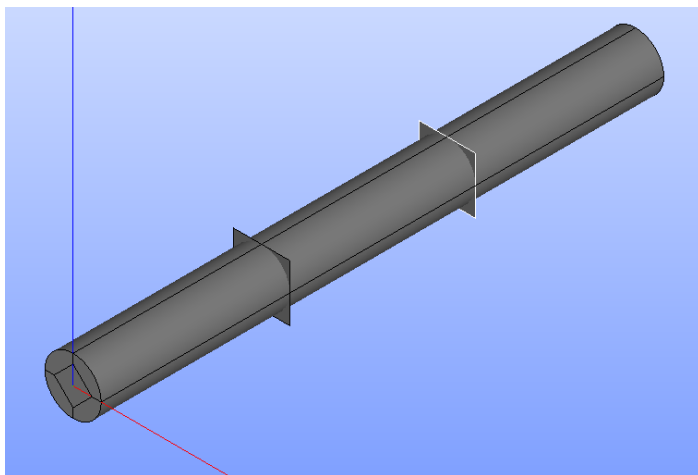


Рис. 78. Цилиндрическая труба с двумя секущими плоскостями.

Далее нажмите **Partition** и разделите цилиндр с помощью проведённых 2 плоскостей на отдельные блоки (Solid). Таким способом получается единый объект, состоящий из блоков (рис. 76).

С помощью операции **Explode** выделите поверхности (Face), из которых состоит получившийся объект. Измените для удобства название поверхностей, образующих вход (input) в трубу, на $i1, i2, i3, i4, i5$, аналогично сделайте для выхода (output) с названиями поверхностей $o1, o2, o3, o4, o5$ и боковой стенки (wall) с названиями $w1, w2, w3, w4, w5, w6, w7, w8, w9, w10, w11, w12$.

Перейдите в модуль Mesh, создайте расчётную сетку, выбрав на вкладке 3D алгоритм **Hexahedron (i, j, k)**, на вкладке 2D – алгоритм **Quadrangle Mapping**, на вкладке 1D – алгоритм **Wire Discretisation**, гипотезу **Number of Segments**, выбрать 10 сегментов. Рассчитайте получившуюся сетку. У Вас должна по-

лучиться сетка, как на рис. 68. Проверьте, что у Вас не обнаружено ошибок при расчёте сетки. Выберите сетку, нажмите правой кнопкой мыши на сетку, выберите опцию **Clipping**, посмотрите получившееся сечение расчётной сетки, проверьте его корректность. Закройте **Clipping**.

Выберите операцию **Create Group**, создайте группы сеток по соответствующим поверхностям (Face), выберите **Group on Geometry**, в качестве объектов – соответствующие переименованные ранее Вами в модуле Geometry поверхности.

Создайте корневую папку проекта OpenFoam, скопируйте туда файлы из исходного проекта damBreak. Далее выбрать в SALOME саму сетку и сохранить файл (**File — Export — UNV file**) в корневую папку проекта OpenFoam (туда, где находятся папки 0, system, constant). Экспорт сетки в OpenFoam осуществляется с помощью команды **ideasUnvToFoam filename.unv** (не применяйте blockMesh).

Этап 2. Начальные и граничные условия

В папке «0» проекта OpenFoam для задания начальных и граничных условий у Вас должно быть 3 файла для величин alpha.water (создаётся вместо alpha.water.orig из проекта damBreak), U, p_rgh. В файле «alpha.water» после строки class volScalarField; пропишите location "0"; (указание, что файл находится в папке «0»).

В нулевой момент времени все величины являются нулевыми.

Граничные условия следующие:

- В файле «U» задаются граничные условия скорости: на входе (inlet) величина $U_y=0,5$ м/с (т.е. inletValue uniform (0 0.5 0);), тип inletOutlet (условие на входе-выходе), на выходе (outlet) прописывается аналогичное условие, но следующей строкой добавляется value (0 0 0); (величина на входе прежняя, на выходе – нулевая), на стенках трубы (side) – условие прилипания (noSlip).

Примечание: slip пишется с маленькой буквы, в случае noSlip – с большой.

- В файле «p_rgh»: на входе (inlet) $p=0$, тип fixedFluxPressure (фиксированное давление в потоке), на выходе (outlet) задаётся полное давление (type totalPressure;) ниже строкой $p0$ uniform 10e5; (константа полного давления $p_0=10^5$) ещё ниже строкой value uniform 0; (нулевая величина давления на выходе), на стенках трубы (side) – такое же условие, как на входе.
- В файле «alpha.water»: на входе (inlet) alpha.water=0.3, тип fixedValue (фиксированная величина), на выходе (outlet) и на стенках трубы (side) задаётся нулевой градиент (zeroGradient).
Файл «setFieldsDict» удалите.

Этап 3. Константы модели

В файле «g» задайте ускорение свободного падения (0 0 – 9.81).




В файле «transportProperties» измените фазы на воду и нефть (phases (water oil);), для воды задайте кинематическую вязкость 10^{-4} м²/с, плотность 1000 кг/м³, для нефти – кинематическую вязкость 10^{-4} м²/с, плотность 600 кг/м³. Поставьте значение sigma, равное 0.05.

В файле «boundary» в качестве типа боковой стенки трубы выберите wall.

Этап 4. Параметры счёта

Задайте конечное время 30 с, шаг по времени – 0,1 с, вывод результатов через каждые 0,1 с. Остальные файлы оставьте неизменными.

Решения

Решения отобразить с помощью ParaView (рис. 79), смотреть alpha.water. Чтобы получить отображение решения, как на рис. 79, необходимо на главный экран вывести всю трубу, затем нажать кнопку Split Horizontal  и отобразить там часть трубы с помощью операции **Clip**, начало (0; 1; 0), нормаль (–1; 0; 0) (рис. 80), уберите галочку **Show Plane**. После этого с помощью кнопок Split Vertical  и Split Horizontal  настройте изображение, как

на рис. 79, на котором заданы сечения трубы (**Slice**), соответствующие координатам по оси y 0,5 м и 1,5 м, нормаль (0; 1; 0).

Примечание: не забывайте убрать галочку Show Plane.

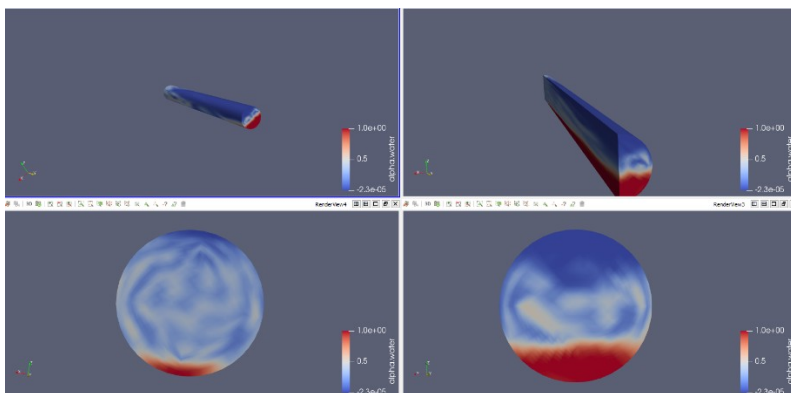


Рис. 79. Отображение решения о двухфазном течении жидкости в трубе в ParaView.

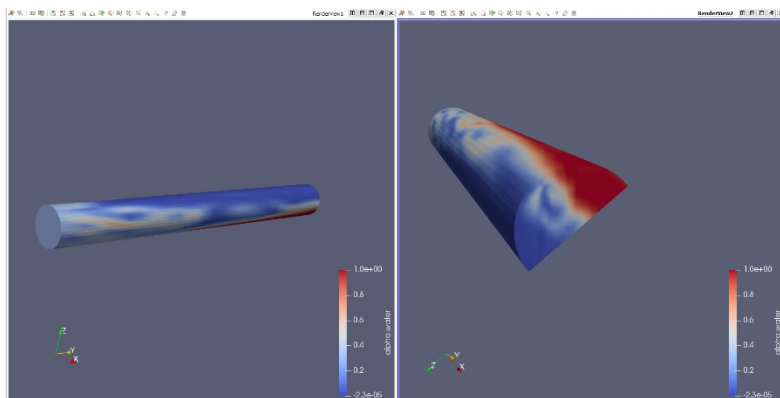



Рис. 80. Разделение экрана на 2 части в ParaView.

Нажмите на кнопку настройки палитры . Отметьте галочку «Enable opacity mapping for surfaces». Сделайте скриншоты в момент достижения потоком воды первого (рис. 81)

и второго (рис. 82) сечений, а также в момент окончания расчётов (рис. 83).

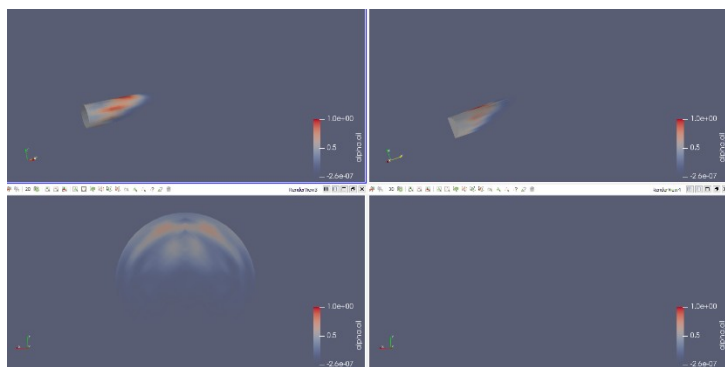


Рис. 81. Скриншот в момент достижения потоком воды первого сечения.

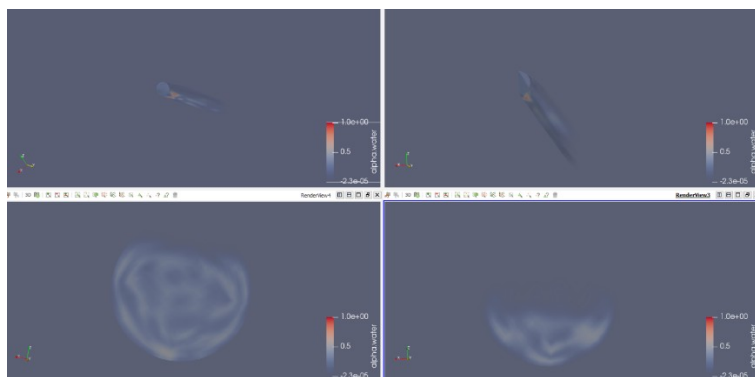


Рис. 82. Скриншот в момент достижения потоком воды второго сечения.

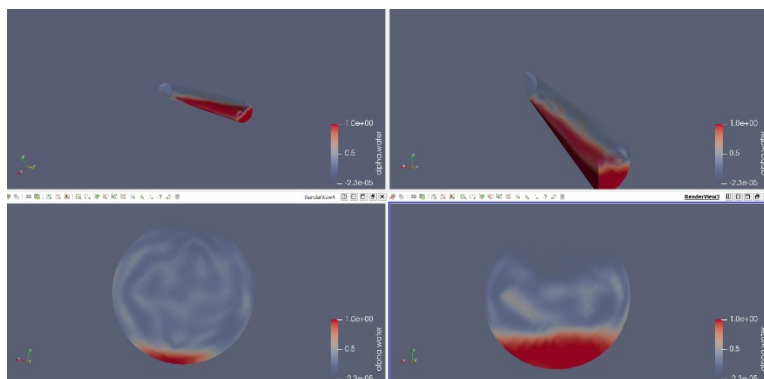


Рис. 83. Скриншот в момент окончания расчётов.

Добавьте фильтр калькулятор (Filters – Alphabetical – Calculator). С помощью кнопок и функций калькулятора введите $\text{mag}(U)$ для расчёта модуля вектора U , U выбирается в списке Vectors. Примените фильтр (рис. 84) и сравните рассчитанное значение с величиной скорости при исходном отображении (на верхней панели выбирается U и Magnitude). Нажмите на основной вариант отображение трубы. Добавьте фильтр статистики по всем временным шагам (Filters – Alphabetical – Temporal Statistics). Выберите расчёт среднего значения (Compute Average) и примените фильтр (рис. 85).

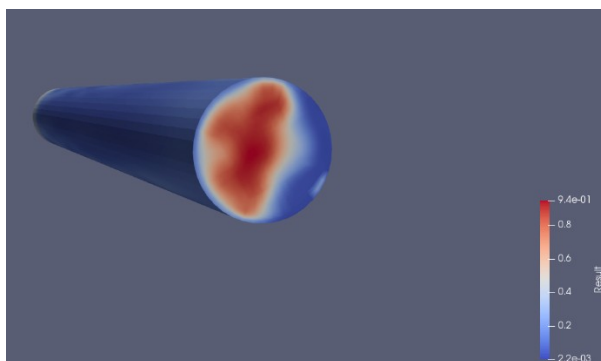


Рис. 84. Применение фильтра калькулятор для отображения модуля вектора скорости.

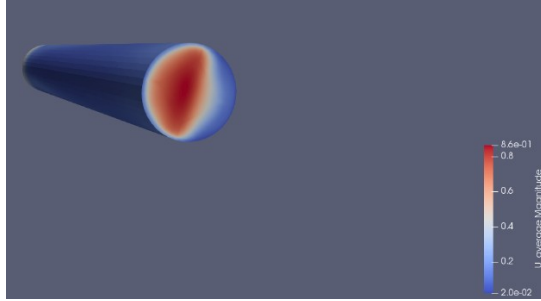


Рис. 85. Отображение среднего значения скорости.

Задание для самостоятельной работы

Повторите расчёт в OpenFOAM для случая, когда на боковой стенке трубы граничное условие у скорости – условие проскальзывания. Рассчитайте среднюю скорость (рис. 86). Сравните значения с предыдущим случаем. Объясните различия.

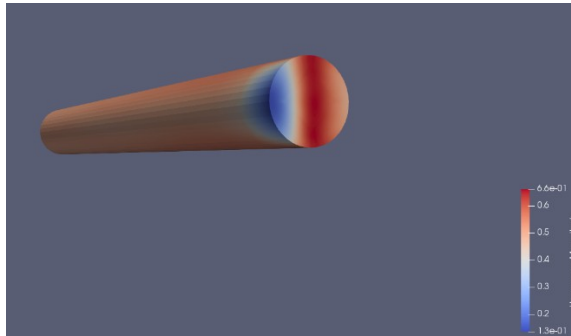


Рис. 86. Отображение среднего значения скорости в случае с условием проскальзывания.

Таким образом, в данной работе Вы познакомились с созданием геометрии и расчётной сетки для моделирования движения двухфазной жидкости в цилиндрической трубе, а также провели расчёт содержания фазы в смеси и скорости, отобразили полученные результаты.

Занятие №9

Тема работы: Создание расчётной сетки для колена трубы.

Цель: Создать геометрию и сетку средствами SALOME для колена трубы.

Ход работы

Этап 1. Создание геометрии колена трубы

Создать в SALOME геометрию колена трубы (рис. 87) и создать расчётную сетку (рис. 88).

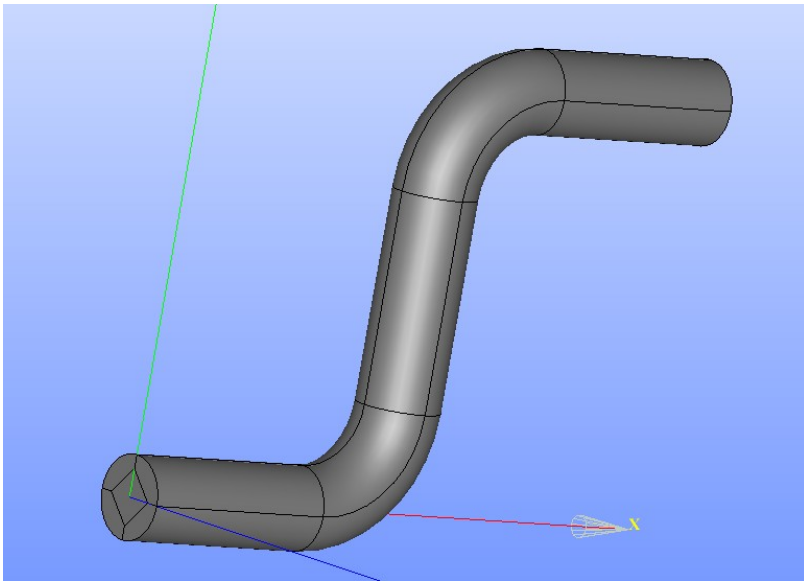


Рис. 87. Колено трубы.

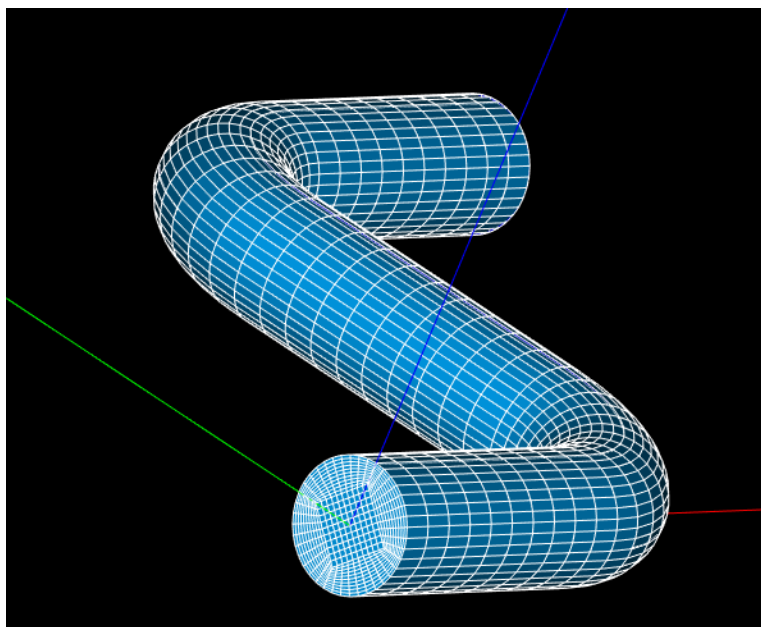


Рис. 88. Расчётная сетка для колена трубы.

Для этого следует:

1. Создать объект 2D sketch (2D траектория) в модуле Geometry, выбрать тип элемента стрелочку, тип места назначения (Destination) – Point (точка), значение – Absolute (абсолютная), координаты (0, 0), нажать применить (Apply). Так создаётся начальная точка траектории.
2. Далее выбрать тип элемента стрелочку, тип места назначения (Destination) – Point (точка), значение – Relative (относительная), $dx=100$, $dy=0$, нажать применить (Apply). Так создаётся первый прямой отрезок.
3. Теперь следует выбрать тип элемента дугу, тип места назначения (Destination) – Direction (направление), значение – Angle (угол), первый угол (Angle) 0° (начальный угол поворота), радиус (Radius) кривизны 50, конечный угол поворота (Angle) 90° , нажать применить (Apply). Так создаётся первый изгиб траектории.

4. После этого нужно выбрать тип элемента стрелочку, тип места назначения (Destination) – Point (точка), значение – Relative (относительная), $dx=0$, $dy=100$, нажать применить (Apply). Так создаётся второй прямой отрезок.
5. Далее выбрать тип элемента дугу, тип места назначения (Destination) – Direction (направление), значение – Angle (угол), первый угол (Angle) 0° (начальный угол поворота), радиус (Radius) кривизны -50 , конечный угол поворота (Angle) 90° , нажать применить (Apply). Так создаётся второй изгиб траектории в противоположную сторону первому изгибу.
6. После этого нужно выбрать тип элемента стрелочку, тип места назначения (Destination) – Point (точка), значение – Relative (относительная), $dx=100$, $dy=0$, нажать применить (Apply). Так заканчивается создание траектории для колена трубы с двумя изгибами. В окне создания траектории можно нажать Close (закрыть).

Теперь необходимо создать объект разделённый диск (Divided Disk), в качестве плоскости для ориентации выберите Oyz (диск будет расположен перпендикулярно оси x), радиус диска 20. Примените изменения. После этого выполните операцию Extrusion along path (вытягивание вдоль траектории), в качестве базового объекта (Base Object) выберите диск, в качестве траектории (Path Object) – созданный 2D sketch. Примените изменения. У вас должен получиться объект, показанный на рис. 87.

Этап 2. Создание расчётной сетки для колена трубы

Перейдите в модуль Mesh для создания расчётной сетки. В качестве объекта для создания сетки выберите получившееся колено трубы. Алгоритм 3D – Hexahedron (i,j,k), алгоритм 2D – Quadrangle: Mapping, алгоритм 1D – Wire Discretisation, гипотеза – Number of Segments, число сегментов 10. Рассчитайте сетку. У вас должна получиться сетка, показанная на рис. 88.

Задание для самостоятельной работы

Средствами SALOME создайте геометрию колена трубы и расчётную сетку для этого случая (рис. 89).

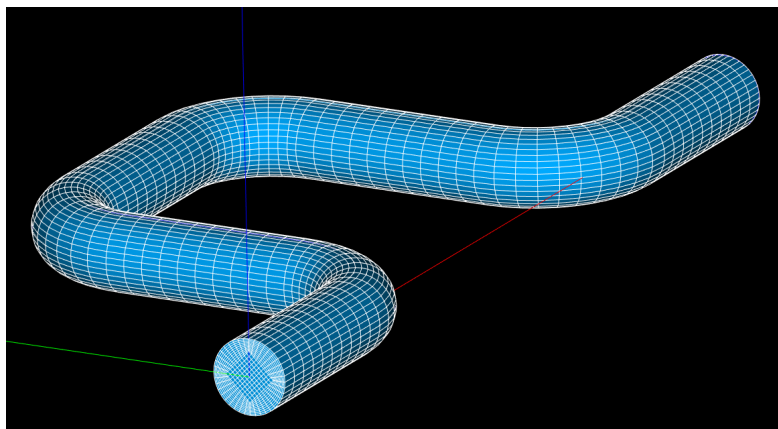


Рис. 89. Расчётная сетка для колена трубы.

Таким образом, в данной работе Вы познакомились с созданием геометрии и расчётной сетки для труб сложной геометрии.

Занятие №10

Тема работы: Расчёт обтекания сферы.

Цель: Решить задачу о течении вокруг сферы несжимаемой вязкой жидкости при числах Рейнольдса 10000 и 100000: создать геометрию и сетку (тетраэдральную) средствами SALOME; задать граничные условия; средствами пакета OpenFOAM выполнить расчет стационарных полей скорости и давления для одного из предложенных чисел Re ; выполнить визуализацию полученных данных средствами пакета ParaView.

Условия задачи

Условия задачи показаны на рис. 90, диаметр сферы $d=10$ см, основная фигура – куб.

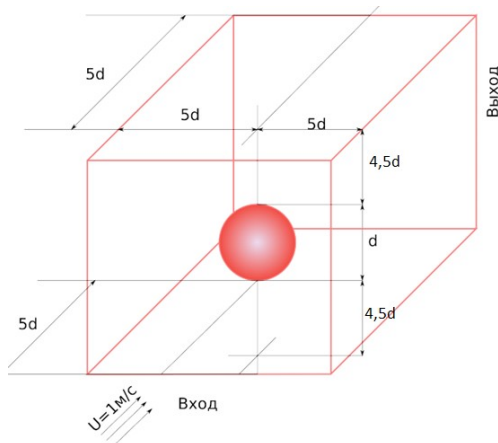


Рис. 90. Условия задачи об обтекании сферы.

Ход работы:

Этап 1. Построение геометрии и расчетной сетки

Средствами SALOME создаётся геометрия, как на рис. 91, и расчётная сетка, как на рис. 92. Сфера задаётся с помощью объекта Sphere и параметра радиус (Radius).

Сначала рекомендуется создать куб, составляющий одну восьмую часть от основной фигуры, поместив левую нижнюю

его вершину в начало координат. Туда же поместить центр сферы и вырезать её из куба (рис. 93) с помощью операции Cut.

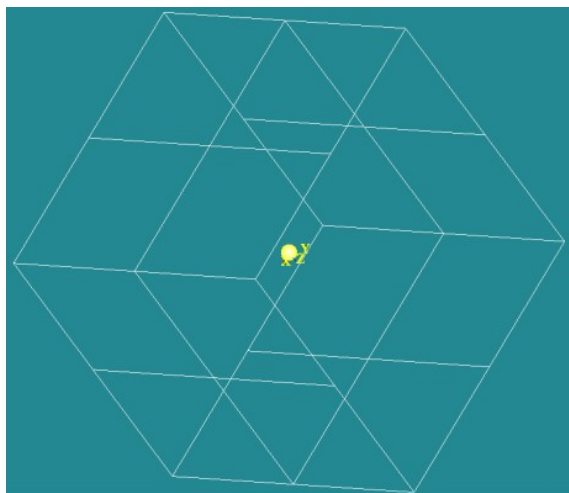


Рис. 91. Геометрия задачи об обтекании сферы.

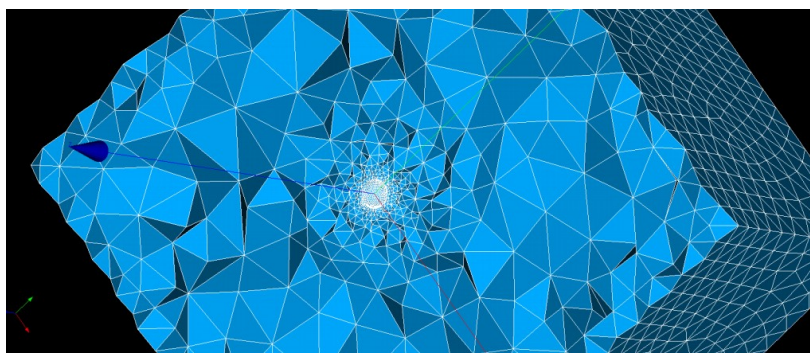


Рис. 92. Расчётная сетка задачи об обтекании сферы.

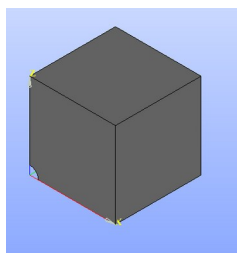


Рис. 93. Куб с вырезанной сферой.

Далее с помощью операции Rotation (вращение, необходимо поворачивать первый куб вокруг оси x на углы 90° , -90° и 180°) получить половину исходной фигуры, показанную на рис. 94. После этого с помощью операции Mirror Image (зеркальное изображение, отражать относительно точки O) отразить каждый из четырёх кубиков, получится исходная фигура, состоящая из частей. Далее объединить часть, показанную на рис. 94, в одну фигуру (объединять сразу по 4 части), проделать то же самое для зеркальной части, получившиеся 2 фигуры объединить в одну. После этого необходимо с помощью операции Explode разбить полученную фигуру на отдельные поверхности, которые следует переименовать в соответствующие граничные стенки, включая внутреннюю сферическую границу.

Примечание: другая последовательность действий приведёт к ошибке (рис. 95, рис. 96). Эта ошибка вызвана тем, что поворот связан с дробным числом π , если поворачивать первый объект во второй, второй в третий и так далее, то может накопиться ошибка округления, поэтому поворачивать всегда нужно исходную фигуру!

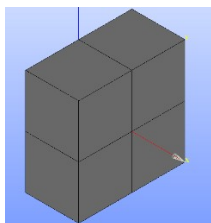


Рис. 94. Половина исходной фигуры.



Рис. 95. Ошибка создания сетки.

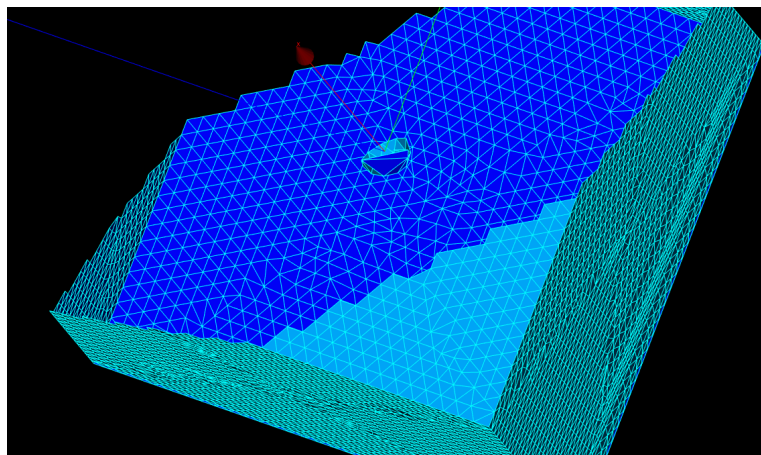


Рис. 96. Некорректно созданная сетка.

Далее в модуле Mesh создайте расчётную сетку для итоговой фигуры. В качестве алгоритма выберите NETGEN 1D-2D-3D, гипотезы – NETGEN 3D Simple Parameters, в пункте 1D выберите Local Length и задайте её равной числу пи (3,14159), в остальных пунктах расставьте галочки. После этого сгруппируйте поверхности по их принадлежности стенкам и экспортируйте файл с сеткой в папку проекта с помощью команды **ideasUnvToFoam**.

Этап 2. Краевые условия

После создания сетки следует задать начальные условия: невозмущенные поля давления (p), скорости (U), кинетической энергии турбулентности (k), частоты диссипации кинетической энергии турбулентности (ω), турбулентной вязкости (ν_t). Файлы остальных величин в папке «0» необходимо удалить.

В нулевой момент времени $p=0$, $U=(0\ 0\ 0)$, $k=1e-3$, $\omega=1$, $\nu_t=0$.

Граничные условия:

- На входе (inlet) — задается U ($U_x=1\text{м/с}$), $k=1e-3$, $\omega=1$, $\nu_t=0$ (тип fixedValue для U , k , ω и тип calculated для ν_t). Нулевой градиент для давления.

- На выходе (outlet) — задается давление $p=0$ (тип fixedValue), $\nu_t=0$ (тип calculated), нулевые градиенты для U , k , ω .

- На стенках канала (upper, lower, front, back) — условие проскальзывания (slip) для всех величин, но для ν_t на верхней upper и нижней lower границах задаётся пристеночная функция (type nutkWallFunction; value uniform 0, первая строка в файле – тип (пристеночная функция), вторая – её величина).

- На стенках сферы (sphereWall) — условие прилипания (noSlip) для скорости, пристеночные функции для k (type kqRWallFunction; value uniform 0.375), ω (type omegaWallFunction; value \$internalField;) и ν_t (как для upper и lower), условие непроницаемости (zeroGradient) для p .

Этап 3. Константы модели

Исходя из заданных d (10 см по умолчанию) и скорости на входе (1м/с по умолчанию), выбираем такую вязкость, чтобы соответствовала заданному числу Re :

- 1) $Re=10000$, вязкость – $1e-6$ (при $d=10\text{см}$, $U_x=1\text{м/с}$)
- 2) $Re=100000$, вязкость – $1e-7$ (при $d=10\text{см}$, $U_x=1\text{м/с}$)

Этап 4. Параметры счёта

Расчёт проводится по итерациям (шаг по времени = 1), вывод через каждые 10 или 100 шагов. Общее число итераций – 150 или 1000. Коэффициент релаксации для скорости: $r_U = 0.9$, задан в файле fvSolution.

Этап 5. Решения

Решения отобразить с помощью ParaView (рис. 97). Чтобы получить такой рисунок, необходимо включить отображение трассеров.

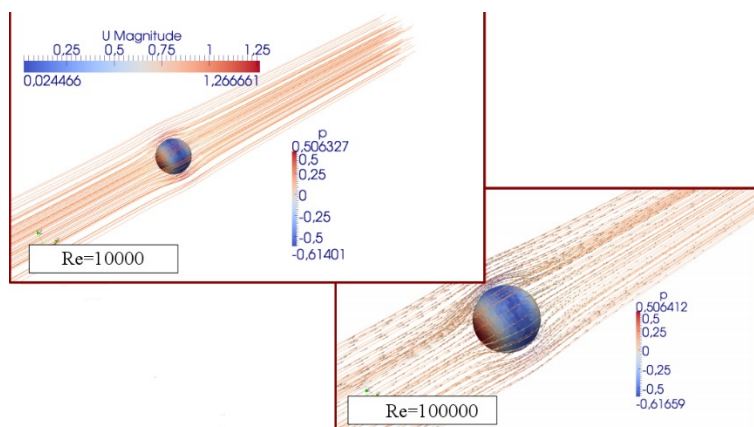


Рис. 97. Примерный вид решения задачи об обтекании сферы.

Этап 6. Создание гексаэдрической сетки

Теперь необходимо создать гексаэдрическую сетку для этой же задачи. Создайте весь куб сразу единым объектом, с помощью операции **Translation** переместите его центр в начало координат. Создайте сферу в начале координат аналогично

предыдущему случаю. С помощью операции **Explode** создайте вершины (**Vertex**) и рёбра (**Edge**) куба, выбрав его в качестве Main Object. Соедините каждую из вершин с помощью линии с началом координат (выберите создание линии по двум точкам) (рис. 98).

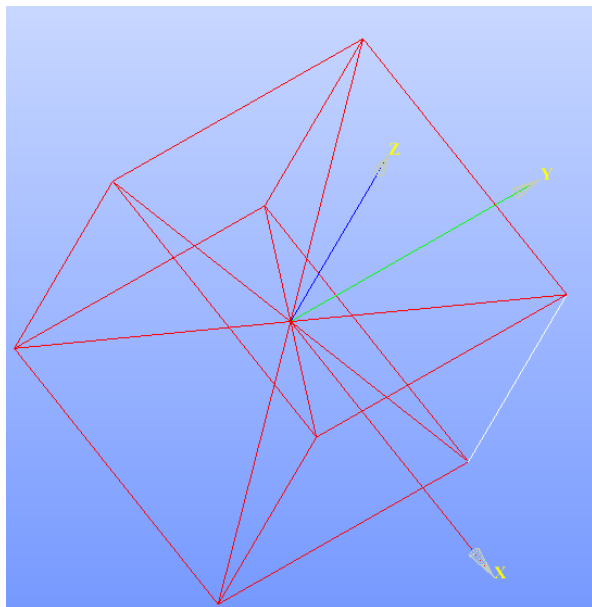


Рис. 98. Линии, соединяющие каждую из вершин куба с началом координат.

Создайте 12 треугольных поверхностей с помощью операции **Build face** (рис. 99). В качестве объектов для создания поверхностей выберите одно из рёбер куба и две линии, соединяющие крайние точки этого ребра с началом координат. Поставьте галочку с указанием о том, чтобы создавались плоские поверхности (Try to create a planar face). Повторите эту операцию для каждого оставшегося ребра. У Вас должно получиться 12 треугольных поверхностей, как на рис. 100.

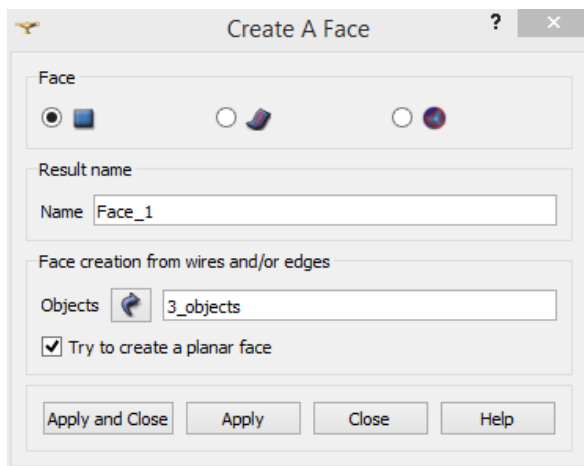


Рис. 99. Окно создания поверхности.

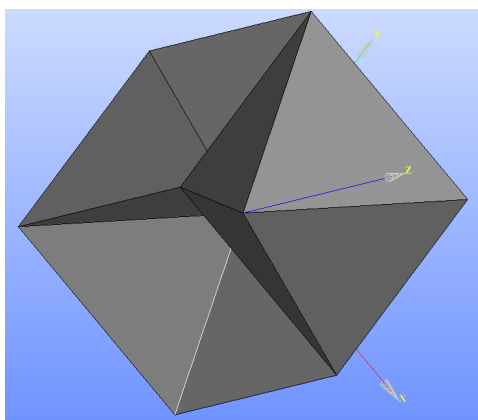


Рис. 100. 12 треугольных поверхностей, содержащих рёбра куба.

Создайте 3 плоскости, проходящие через начало координат, в качестве нормалей выберите поочерёдно оси Ox , Oy , Oz . Должна получиться фигура из 12 треугольных поверхностей и 3 плоскостей, как на рис. 101.

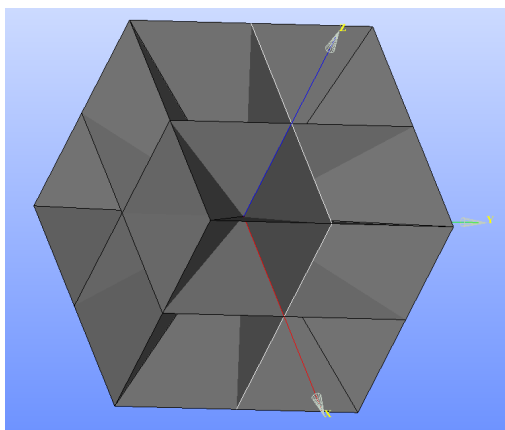


Рис. 101. 3 плоскости и 12 треугольных поверхностей.

Далее вырежьте сферу из куба с помощью операции **Cut**. Осуществите операцию **Partition**, выберите в качестве объекта куб с вырезанной сферой, в качестве вспомогательных объектов – 12 треугольных поверхностей и 3 плоскости. Далее необходимо с помощью операции **Explode** разбить полученную фигуру на отдельные поверхности, которые следует переименовать в соответствующие граничные стенки, включая внутреннюю сферическую границу.

С помощью операции **Inspection – Basic Properties** рассчитайте объём (Volume) исходного куба без сферы, объём фигуры с вырезанной сферой. Самостоятельно рассчитайте пористость фигуры.

После этого перейдите в модуль **Mesh** и создайте расчётную сетку, в качестве алгоритма 3D выберите Hexahedron (i,j,k), алгоритма 2D – Quadrangle: Mapping, алгоритма 1D – Wire Discretisation, гипотезы 1D – Number of Segments, число сегментов – 20. Тип сетки можете оставить Any (любой, SALOME с помощью выбранных алгоритмов сам создаст нужный гексаэдрический тип сетки). Рассчитайте сетку. Нажмите на отображение сетки на экране правой кнопкой мыши, выберите операцию **Clipping**, в качестве базовой точки выберите точку (0; 0; 0), направление (0; -1; 0). Примените **Clipping**. У Вас должна получиться сетка,

изображённая на рис. 102 полностью и на рис. 103 в увеличенном виде. Сгруппируйте поверхности по их принадлежности стенкам и экспортируйте файл с сеткой в папку проекта с помощью команды **ideasUnvToFoam**.

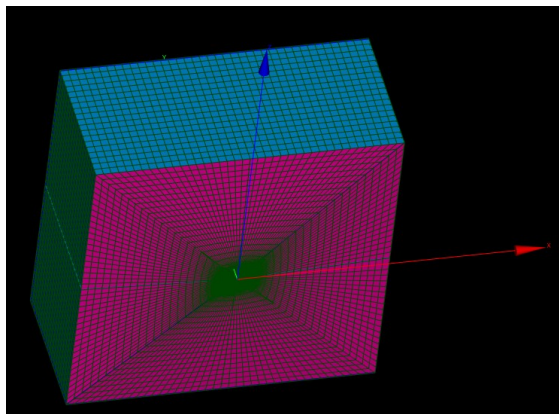


Рис. 102. Расчётная гексаэдрическая сетка для куба с вырезанной сферой после применения операции Clipping.

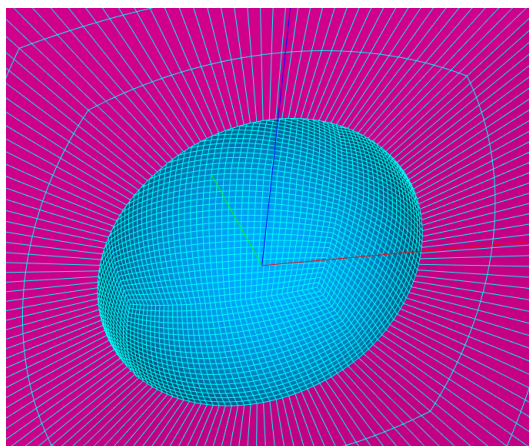


Рис. 103. Расчётная гексаэдрическая сетка для куба с вырезанной сферой в увеличенном виде.

Задайте начальные и граничные условия, параметры расчёта из предыдущего случая. В файле «boundary» смените тип всех стенок, кроме входа и выхода, на wall. Запустите расчёт, после окончания расчёта визуализируйте результаты по скорости (рис. 104). Сравните результаты расчётов с использованием гексаэдрической сетки с исходным случаем.

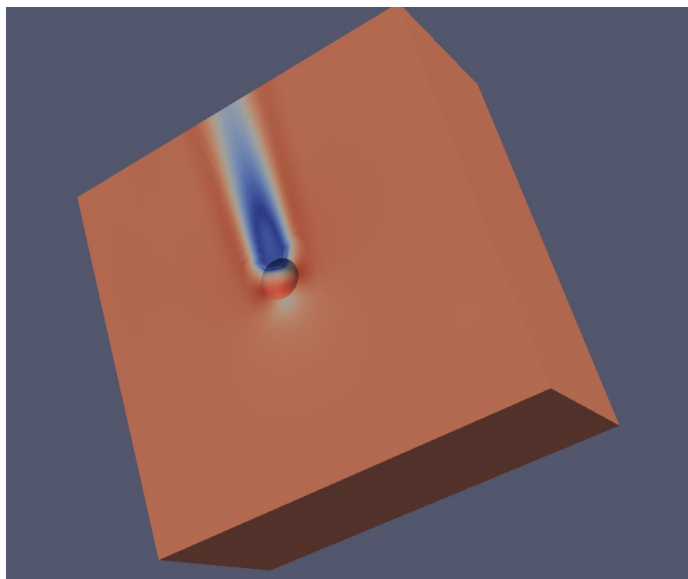


Рис. 104. Результаты расчётов обтекания сферы по гексаэдрической сетке.

Таким образом, в данной работе Вы познакомились с созданием геометрии и расчётной сетки для моделирования обтекания сферы.

Список возможных ошибок

1. Если запустить расчёт проекта одного решателя (например, simpleFOAM) с другим решателем (например, icoFoam, interFoam), то программа выдаст ошибку, что не может найти ключевое слово какое-нибудь ключевое слово (например, PISO, см. рис. 105).

```
--> FOAM FATAL IO ERROR:  
keyword PISO is undefined in dictionary "C:/pitzDaily/system/fvSolution"
```

Рис. 105. Ошибка, связанная с выбором неправильного решателя.

2. Неправильная последовательность действий по созданию геометрии в задаче об обтекании сферы приведёт к ошибке (рис. 95, рис. 96). Эта ошибка вызвана тем, что поворот связан с дробным числом π , если поворачивать первый объект во второй, второй в третий и так далее, то может накопиться ошибка округления, поэтому поворачивать всегда нужно исходную фигуру.
3. Если ввести команду с опечаткой, т.е. команду, которую не знает программа blueCFD Core Terminal, то будет выдана ошибка о том, что команда не найдена (рис. 106).

```
$ cr  
bash: cr: команда не найдена
```

Рис. 106. Ошибка введения названия команды.

4. Если не учесть все грани при их объединении в группы в ходе создания сетки, то остаток попадёт в defaultFaces в файле boundary (рис. 107).

```

(
  inlet
  {
    type          patch;
    nFaces        300;
    startFace     22400;
  }
  front
  {
    type          patch;
    nFaces        800;
    startFace     22700;
  }
  defaultFaces
  {
    type          patch;
    nFaces        2100;
    startFace     23500;
  }
)

```

Рис. 107. Необъединённые в группы грани в файле boundary, обозначенные как defaultFaces.

5. Если ввести команду blockMesh для геометрии задачи, в которой сетка создаётся средствами SALOME и отсутствует файл blockMeshDict, то будет выдана ошибка, что указанный файл не обнаружен (рис. 108).

```

--> FOAM FATAL ERROR:
"C:/spherehexlosh/system/blockMeshDict"

From function int main(int, char**)
in file blockMesh.C at line 202.

FOAM exiting

```

Рис. 108. Ошибка обнаружения файла blockMeshDict при использовании команды blockMesh.

6. Если ввести команду ideasUnvToFoam и название файла сетки, но не скопировать файл в папку проекта, то Open-

FOAM выдаст ошибку, что не может обнаружить файл (рис. 109).

```
--> FOAM FATAL ERROR:
cannot find file "C:/damBreakosh/system/controlDict"

From function virtual Foam::autoPtr<Foam::IStream> Foam::fileOperations::uncollatedFileOperation::readStream(Foam::regIOobject&, const Foam::fileName&, const Foam::word&, bool) const
in file global/fileOperations/uncollatedFileOperation/uncollatedFileOperation.C at line 522.

FOAM exiting
```

Рис. 109. Ошибка обнаружения файла сетки, созданной средствами SALOME, при введении команды `ideasUnvToFoam`.

7. Если ввести команду `checkMesh` до создания сетки, то программа выдаст ошибку, что не обнаружит какой-либо элемент геометрии, например, точки (points) (рис. 110).

```
Create time
Create polyMesh for time = 0

--> FOAM FATAL ERROR:
Cannot find file "points" in directory "polyMesh" in times 0 down to constant

From function Foam::word Foam::Time::findInstance(const Foam::fileName&, const Foam::word&, Foam::IOobject::readOpti
on, const Foam::word&) const
in file db/Time/findInstance.C at line 225.

FOAM exiting
```

Рис. 110. Ошибка введения команды `checkMesh` до создания сетки.

8. Если ввести команду `setFields` в проекте, в котором отсутствует файл `setFieldsDict`, OpenFOAM выдаст ошибку, что не может найти соответствующий файл (рис. 111).

```
--> FOAM FATAL ERROR:
cannot find file "C:/cavityosh/cavity/system/setFieldsDict"

From function virtual Foam::autoPtr<Foam::IStream> Foam::fileOperations::uncollatedFileOperation::readStream(Foam::regIOobject&, const Foam::fileName&, const Foam::word&, bool) const
in file global/fileOperations/uncollatedFileOperation/uncollatedFileOperation.C at line 522.

FOAM exiting
```

Рис. 111. Ошибка обнаружения файла `setFieldsDict` при введении команды `setFields`.

Алгоритмы моделирования

Алгоритм 1. Моделирование при создании сетки с помощью команды `blockMesh` (имеется в наличии файл `blockMeshDict`)

1. Откройте `blueCFD Core Terminal` в Вашей папке.
2. Введите команду **`blockMesh`**.
3. Проверьте сетку с помощью команды **`checkMesh`**.
4. Если в задаче имеются области с какой-либо фазой, заполните эти области с помощью команды **`setFields`**, если нет, пропустите этот шаг.
5. Запустите необходимый решатель (в файле `controlDict` написано название решателя) и дождитесь окончания расчёта.
6. Введите команду **`paraFoam`** для визуализации результатов.

Если в задаче изменилась геометрия (файл `blockMeshDict`), то повторить все шаги с пункта 2.

Если изменяется область, занятая какой-либо фазой, например, занятая водой, то удалите файл с координатами области воды (фазы) из прошлых расчётов `alpha.water`, скопируйте оригинальный файл `alpha.water.orig`, назовите его `alpha.water`, внесите необходимые изменения и повторите действия с пункта 4.

Если в задаче изменились параметры в файлах `controlDict`, `turbulenceProperties`, `transportProperties`, то повторите расчёты с пункта 5 включительно.

Алгоритм 2. Моделирование при создании сетки с помощью программы SALOME

1. Запустить SALOME, открыть модуль `Geometry`, сделать необходимую геометрию.
2. Перейти в модуль `Mesh`, создать расчётную сетку.
3. Сгруппируйте поверхности по их принадлежности стенкам.
4. Выгрузите файл сетки в формате `.unv` в папку проекта.
5. Введите команду **`ideasUnvToFoam filename.unv`**, где вместо `filename` должно быть имя экспортируемого файла. **Не применяйте команду `blockMesh`!**

6. Поменяйте типы стенок в файле `boundary` на требуемые по условию моделируемой задачи.
7. В файлах папки «0» убедитесь, что граничные условия для стенок те же самые, что и файле `boundary`.
8. Откройте `blueCFD Core Terminal` в Вашей папке.
9. Если в задаче имеются области с какой-либо фазой, заполните эти области с помощью команды **setFields**.
10. Запустите необходимый решатель (в файле `controlDict` написано название решателя) и дождитесь окончания расчёта.
11. Введите команду **paraFoam** для визуализации результатов.

Если в задаче изменилась геометрия, то повторить всё с пункта 4.

Если изменяется область, занятая какой-либо фазой, например, занятая водой, то удалите файл с координатами области воды (фазы) из прошлых расчётов `alpha.water`, скопируйте оригинальный файл `alpha.water.orig`, назовите его `alpha.water`, внесите необходимые изменения и повторите действия с пункта 9.

Если в задаче изменились параметры в файлах `controlDict`, `turbulenceProperties`, `transportProperties`, то повторите расчёты с пункта 10 включительно.

Перечень возможных тем самостоятельных проектов

1. Моделирование течения жидкости в разветвляющейся трубе (рис. 112).

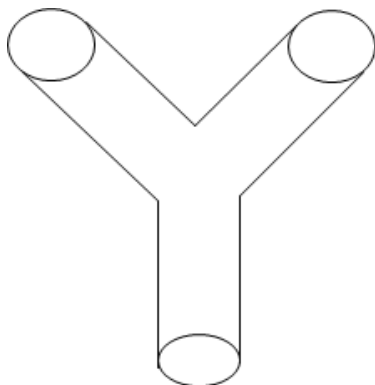


Рис. 112. Задача о течении жидкости в разветвляющейся трубе.

2. Решение задачи о всплывании масла в воде (рис. 113).

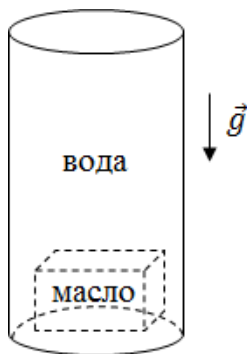


Рис. 113. Задача о всплывании масла в воде.

3. Моделирование обтекания двух шаров в цилиндрической трубе (рис. 114).

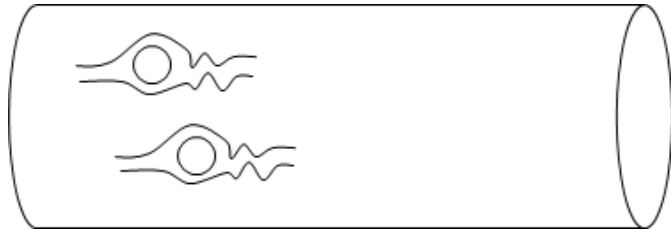


Рис. 114. Задача об обтекании двух шаров в цилиндрической трубе.

4. Решение задачи о прорыве дамбы в многофазном случае.
5. Моделирование течения трёхфазной жидкости в трубе.
6. Решение задачи о прорыве дамбы с несколькими областями проникновения жидкости.
7. Моделирование турбулентного течения многофазной жидкости в трубе.
8. Расчёт обтекания нескольких брусов.

Список литературы

1. Роуч П. Вычислительная гидродинамика / П. Роуч; пер. с англ. – М.: Мир, 1980. – 618 с.
2. Флетчер К. Вычислительные методы в динамике жидкости. Том 1 / К. Флетчер; пер. с англ. – М.: Мир, 1991. – 502 с.
3. Флетчер К. Вычислительные методы в динамике жидкости. Том 2 / К. Флетчер; пер. с англ. – М.: Мир, 1991. – 552 с.
4. Сайт OpenFOAM Tutorial Guide. URL: <https://www.openfoam.com/documentation/tutorial-guide>
5. Сайт OpenFOAM wiki. URL: https://openfoamwiki.net/index.php/Main_Page
6. Форум CFD Online. URL: <https://www.cfd-online.com/Forums/>
7. Официальный сайт SALOME. URL: <https://www.salome-platform.org/>