

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Романчук Иван Сергеевич

Должность: Ректор

Дата подписания: 20.07.2023 17:18:25

Уникальный программный ключ:

6319edc2b582ffdacea443f01d5779368d0957ac34f5cc074d81131530452479

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
**ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ**

Кафедра моделирования физических процессов и систем

Р.М. Ганопольский, А.Я. Гильманов, Г.А. Малыгин

**ГИДРОДИНАМИЧЕСКОЕ МОДЕЛИРОВАНИЕ  
ПОТОКОВ СЛОЖНОЙ ФОРМЫ**

Учебно-методическое пособие для студентов направления «Техническая  
физика: Физика недр (магистратура)»  
(Материалы для лабораторных и самостоятельных работ)

Тюмень

Издательство

Тюменского государственного университета

2019

УДК 378.147.88  
ББК 30.123  
Г19

**Р.М. Ганопольский, А.Я. Гильманов, Г.А. Малыгин.**  
**Гидродинамическое моделирование потоков сложной формы.**  
Учебно-методическое пособие для студентов направления «Техническая физика: Физика недр (магистратура)» (Материалы для лабораторных и самостоятельных работ). Тюмень: Издательство Тюменского государственного университета, 2019. 69 с.

Учебно-методическое пособие содержит необходимый материал для лабораторных занятий по дисциплине «Гидродинамическое моделирование потоков сложной формы» для студентов 1 курса направления магистратуры «Техническая физика: Физика недр». Пособие даёт необходимые рекомендации по работе в среде OpenFOAM. Освоив данный курс, студент сможет применять полученные знания на практике и при дальнейшем трудоустройстве.

Учебная программа дисциплины опубликована на сайте ТюмГУ: Гидродинамическое моделирование потоков сложной формы. Режим доступа: <http://www.umk3plus.utmn.ru>, свободный.

Рекомендовано к изданию кафедрой моделирования физических процессов и систем. Утверждено первым проректором Тюменского государственного университета.

ОТВЕТСТВЕННЫЙ  
РЕДАКТОР:

**Р.М. Ганопольский**, зав. каф. МФПиС  
ТюмГУ, к.ф.-м.н.

РЕЦЕНЗЕНТЫ:

**Л. П. Пащенко**, к.п.н., доцент кафедры  
ин.яз. и МПК ЭПО  
экономико-правовых отношений  
**И.В. Давыдова**, к.п.н., доцент кафедры  
ин.яз. и МПК ГН

© ФГАОУ ВО Тюменский государственный университет, 2019

© Ганопольский Р.М., Гильманов А.Я., Малыгин Г.А., 2019

Родион Михайлович Ганопольский  
Александр Янович Гильманов  
Георгий Андреевич Малыгин

ГИДРОДИНАМИЧЕСКОЕ МОДЕЛИРОВАНИЕ  
ПОТОКОВ СЛОЖНОЙ ФОРМЫ

Учебно-методическое пособие для студентов направления «Техническая физика: Физика недр (магистратура)»  
(Материалы для лабораторных и самостоятельных работ)

Данные  
выставляет  
Издательство

Подписано в печать \_\_\_\_\_. Тираж \_\_ экз.  
Объем \_\_\_\_\_ усл. печ. л. Формат 60x84/16. Заказ № \_\_\_\_\_.  
Издательство Тюменского государственного университета  
625003, г. Тюмень, Семакова, 10

## Содержание

Лабораторная работа №0.....	5
Самостоятельная работа №0.....	8
Лабораторная работа №1.....	14
Лабораторная работа №2.....	29
Лабораторная работа №3.....	34
Лабораторная работа №4.....	46
Лабораторная работа №5.....	52
Самостоятельная работа №1.....	64

## Лабораторная работа №0

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

**Часть 1.** 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).
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.

**Часть 2.** 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>.

### **Часть 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').

```
$ ls -a
'$Recycle.Bin'      Drivers      Microsoft   'Program Files'      TATIANA
.                  found.000   MSOCache    'Program Files (x86) Users
..                 hiberfil.sys office15client.microsoft.com  Windows
bootmgr           Intel       OFTut       SALOME-8.3.0-WIN64
BOOTNXT          labal      pagefile.sys swapfile.sys
'Documents and Settings' lazarus     PerfLogs    'System Volume Information'
```

Рис. 1. Пример задания команды 'ls -a' в окне BlueCFD – Core Terminal.

Ещё одна команда – **pwd** – показывает текущую папку.

Нажатие стрелочек вверх-вниз на клавиатуре позволяет пролистывать список команд (вверх – вернуться к предыдущей команде, вниз – перейти к следующей, если был осуществлён возврат).

С помощью этих команд перейдите в свою рабочую папку и просмотрите список хранящихся в ней файлов.

## Самостоятельная работа №0

**Тема:** знакомство с интерфейсом ParaView и SALOME.

ParaView – пакет для визуализации результатов и расчётных сеток, поставляется по умолчанию вместе с OpenFOAM.

Меню программы включает в себя набор средств для открытия/сохранения файлов (File), отмены/повторения действия (Edit), переключения видимости различных панелей инструментов (View), создания источников данных различных типов (Sources) и фильтров для обработки данных (Filters), доступа к расширенным функциям, например, управлению плагинами или доступу к оболочке для написания скриптов на языке Python (Tools).

Следить за состоянием программы позволяет дерево исследуемых объектов (Pipeline Browser), менять свойства выбранного объекта можно в блоке Properties (рис. 2). Некоторые из кнопок на приборной панели доступны в зависимости от типа выбранного объекта, другие доступны независимо от выбора объекта.

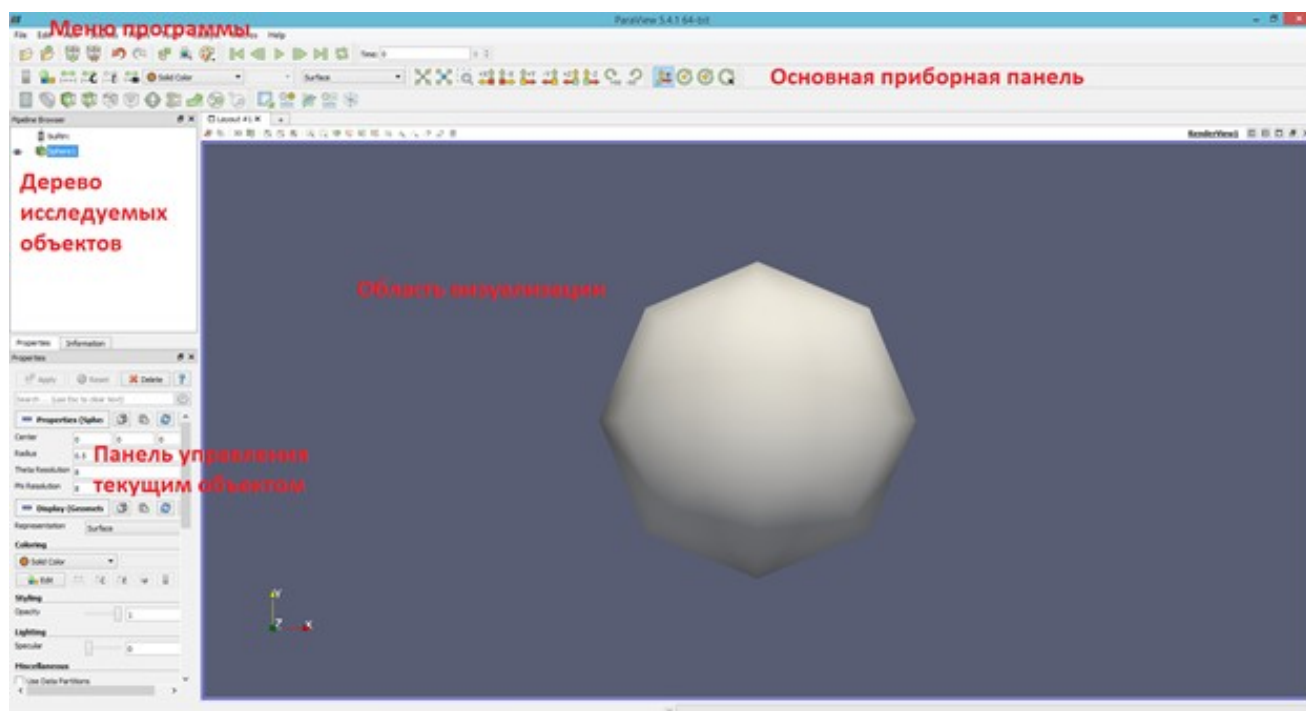
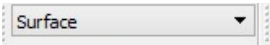


Рис. 2. Интерфейс ParaView.



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).

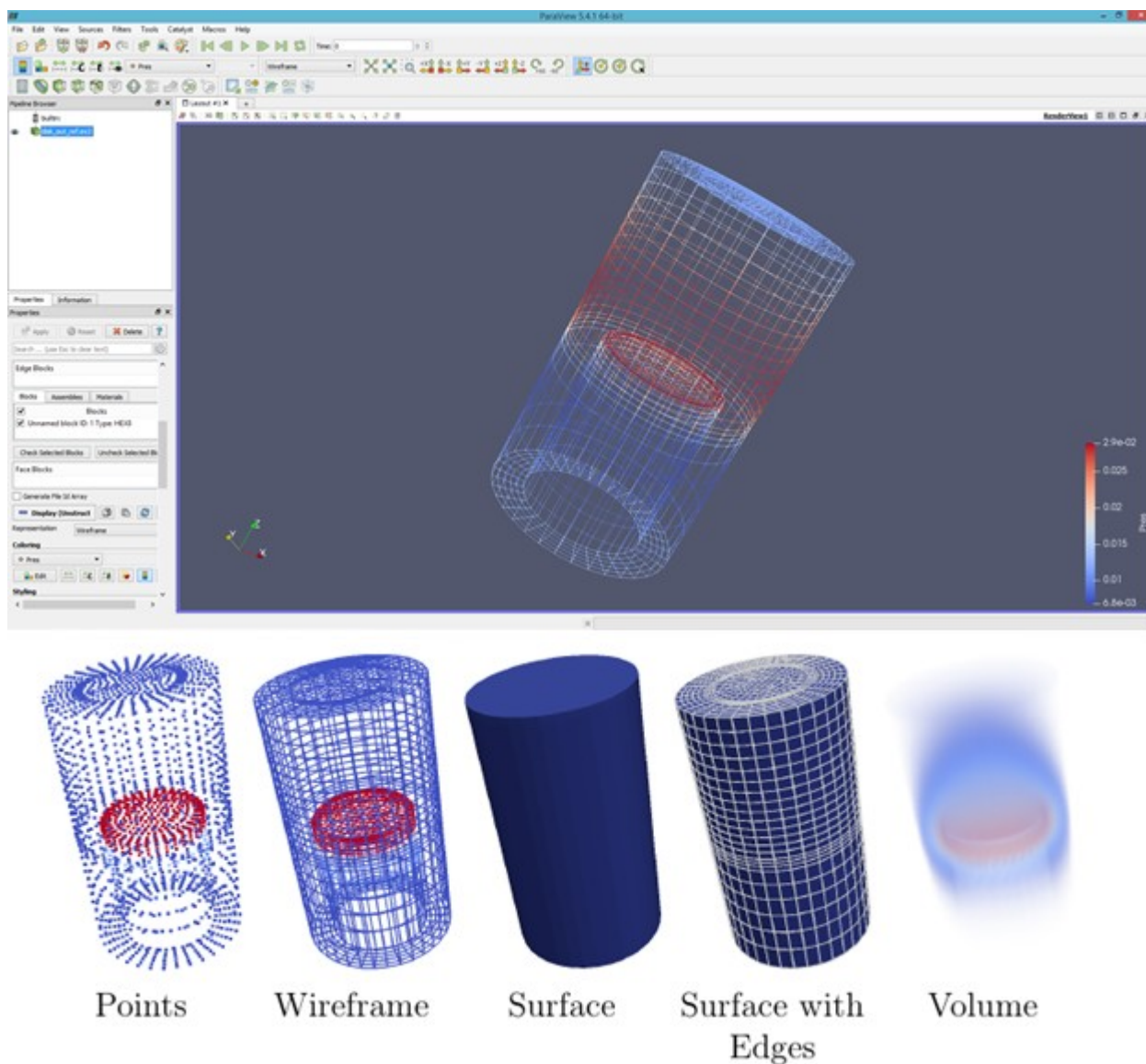



Рис. 3. Отображение объекта в ParaView.

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

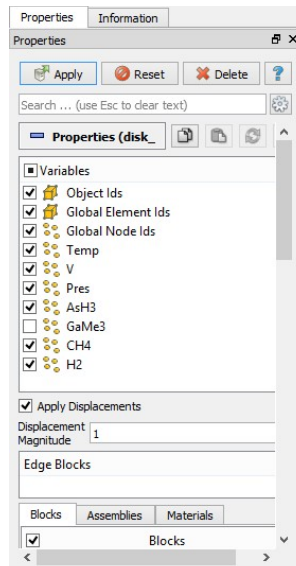


Рис. 4. Блок Properties.

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

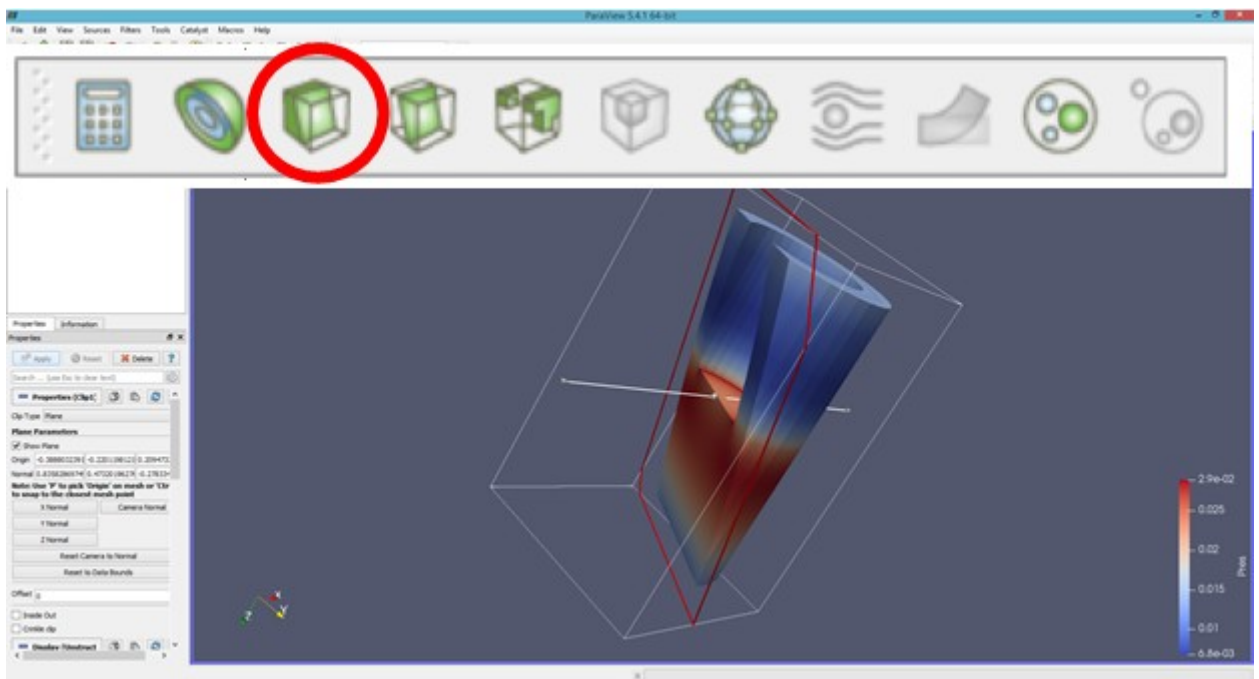


Рис. 5. Применение фильтра Clip.

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

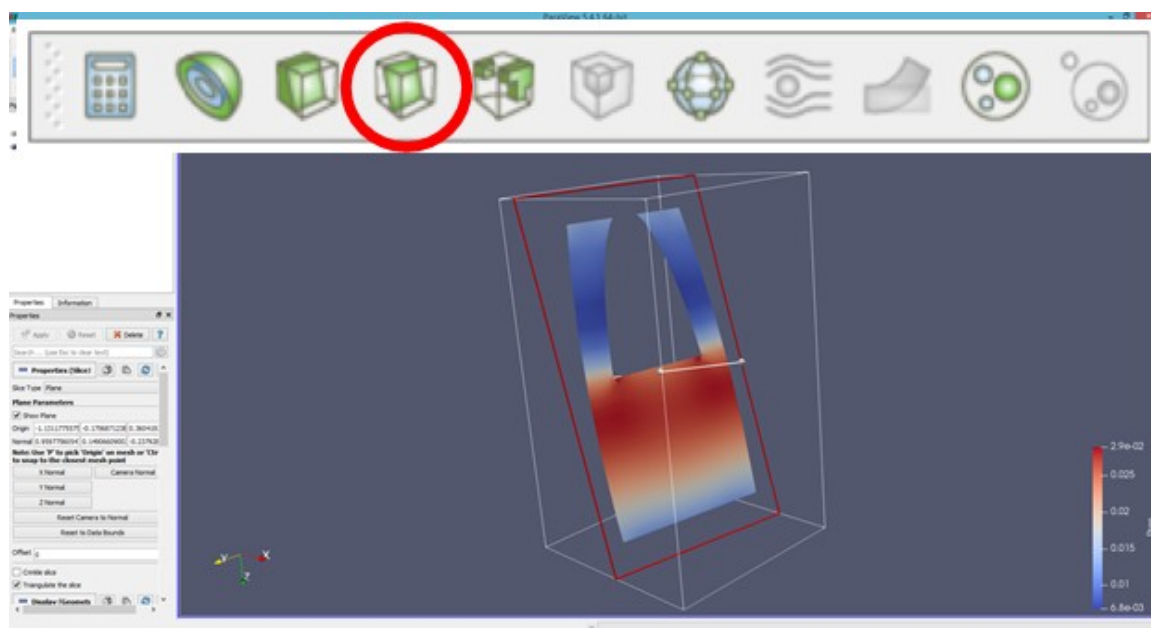


Рис. 6. Применение фильтра Slice.

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).

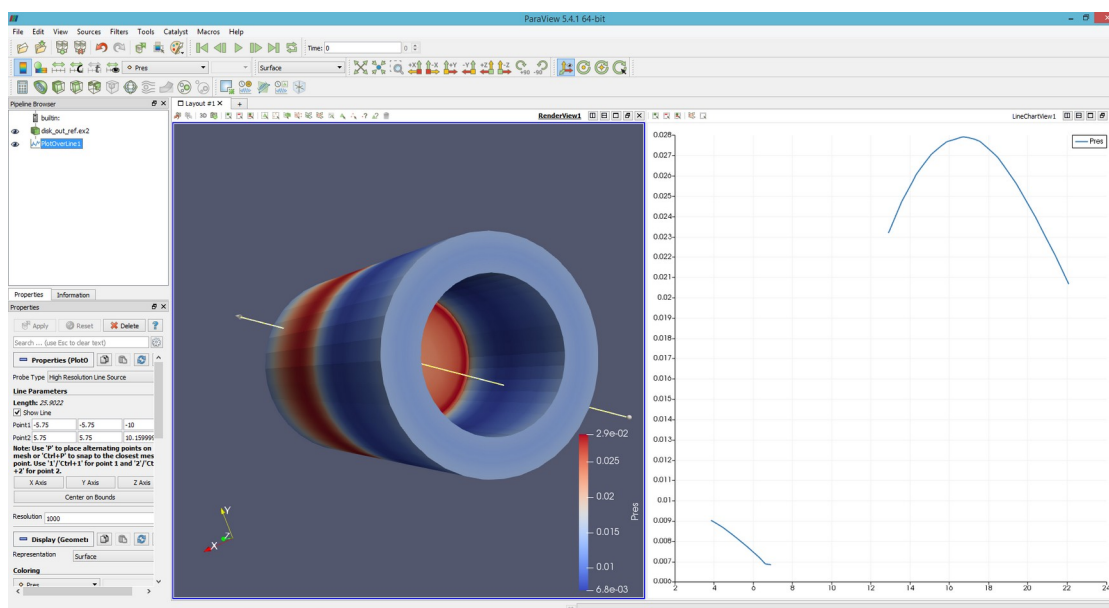


Рис. 7. Построение графика изменения величины давления вдоль произвольно проведённой оси.

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.

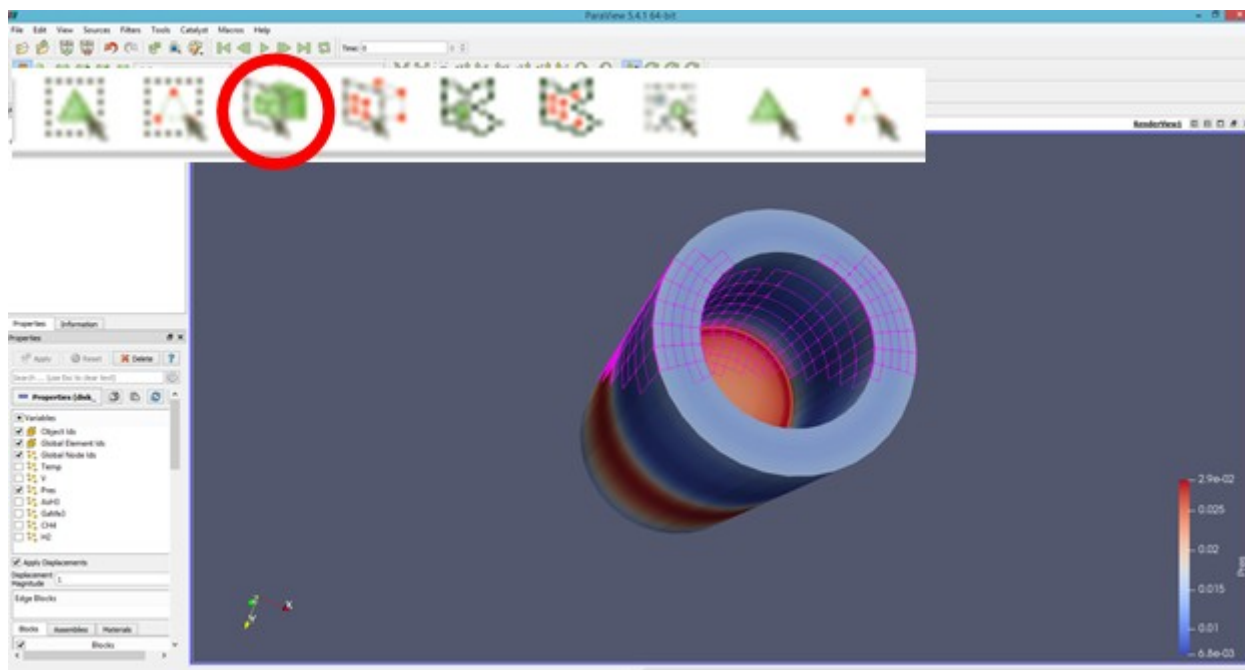


Рис. 8. Выделение группы ячеек.

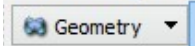
Все фильтры доступны также в алфавитном порядке в меню Filters-Alphabetical.

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

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.

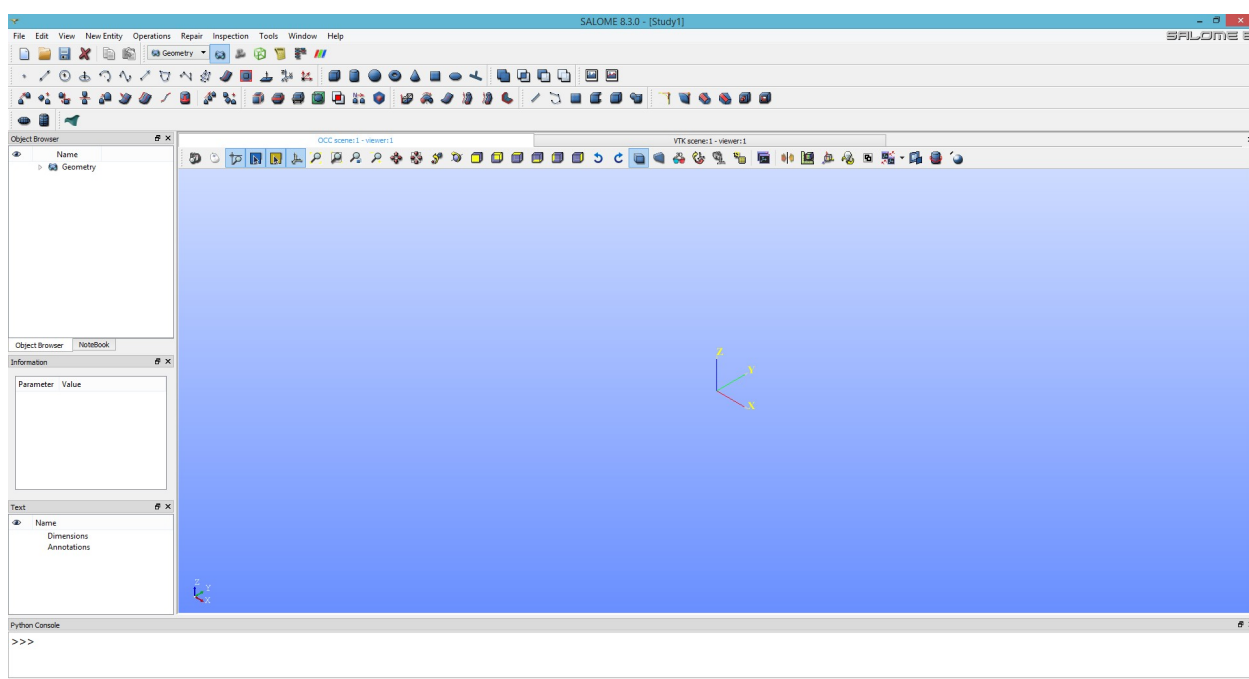


Рис. 9. Интерфейс SALOME.

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 ознакомление с SALOME осуществляется в лабораторных работах.

## Лабораторная работа №1

**Тема:** Демонстрация задачи о течении 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.

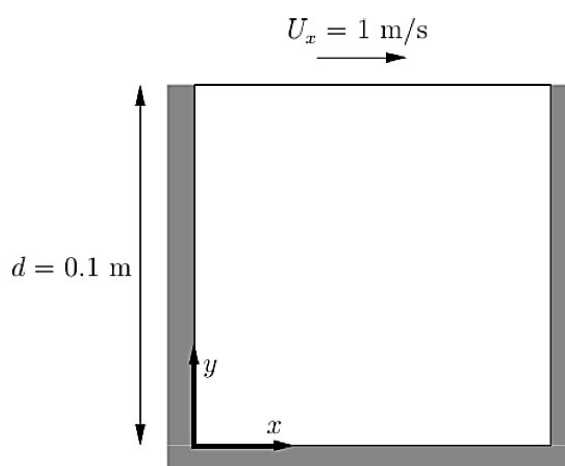


Рис. 10. Схема каверны.

### Ход работы

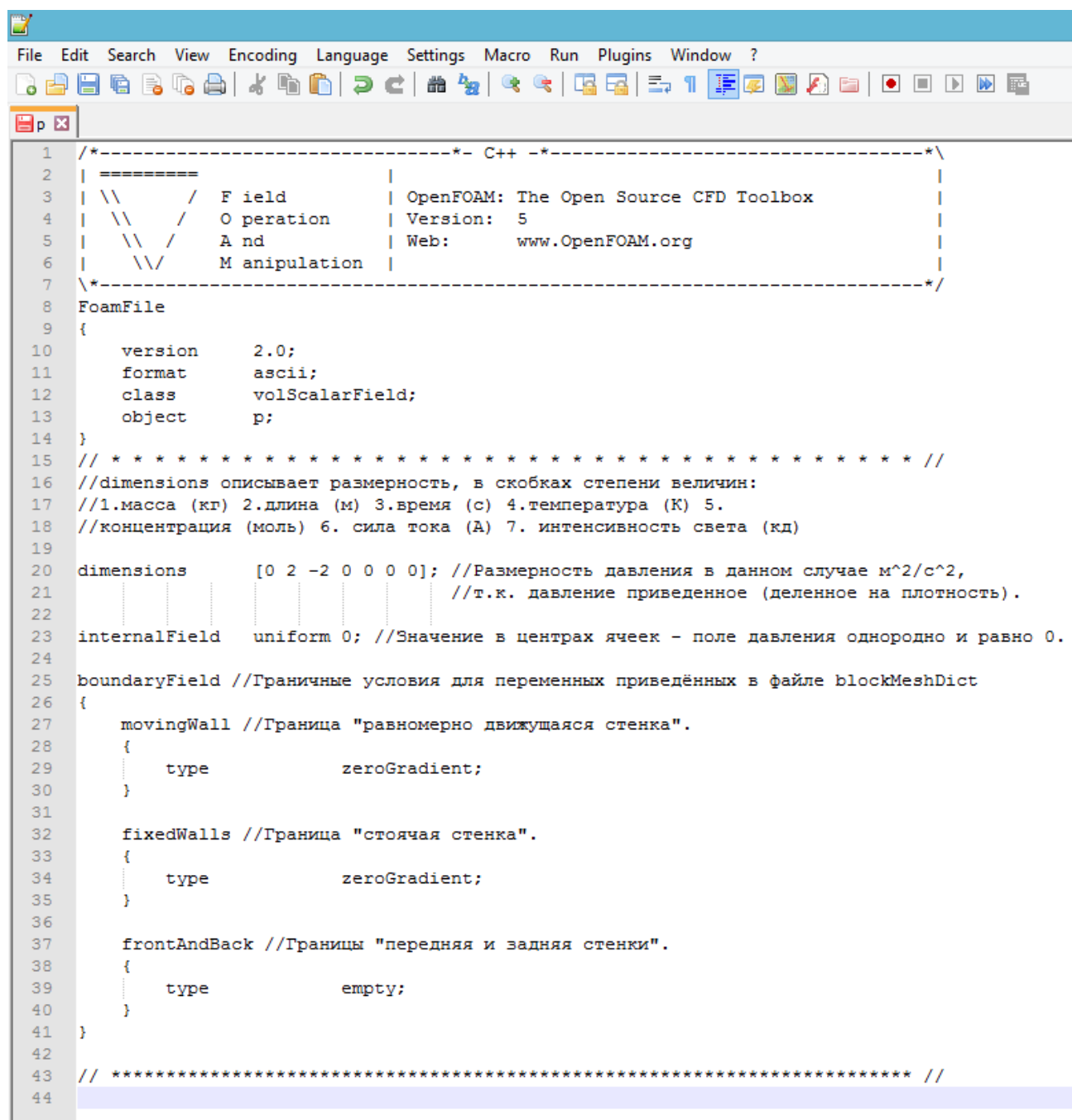
#### Упражнение 1. Ознакомление со структурой и кодом

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

Рис. 11. Файл «p».



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

```
1  /*-----*- C++ -*------*/\n2  |=====|\n3  | \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |\n4  | \\ / O p e r a t i o n | Version: 5 |\n5  | \\ / A n d | Web: www.OpenFOAM.org |\n6  | \\ / M a n i p u l a t i o n | |\n7  /*-----*/\n8  FoamFile\n9  {\n10     version      2.0;\n11     format       ascii;\n12     class        volVectorField;\n13     object       U;\n14  }\n15  // ***** //\n16\n17  dimensions      [0 1 -1 0 0 0]; //Размерность скорости м/с\n18\n19  internalField   uniform (0 0 0); //Поле скорости в нулевой момент времени невозмущенное.\n20\n21  boundaryField //Граничные условия для переменных, приведенных в файле blockMeshDict.\n22  {\n23     movingWall //Граница "равномерно движущаяся стенка"\n24     {\n25         type          fixedValue;\n26         value          uniform (1 0 0); //Условие "прилипания"\n27     }\n28\n29     fixedWalls //Граница "стоячая стенка"\n30     {\n31         type          noSlip;\n32     }\n33\n34     frontAndBack\n35     {\n36         type          empty;\n37     }\n38  }\n39\n40  // ***** //\n41
```

Рис. 12. Файл «U».

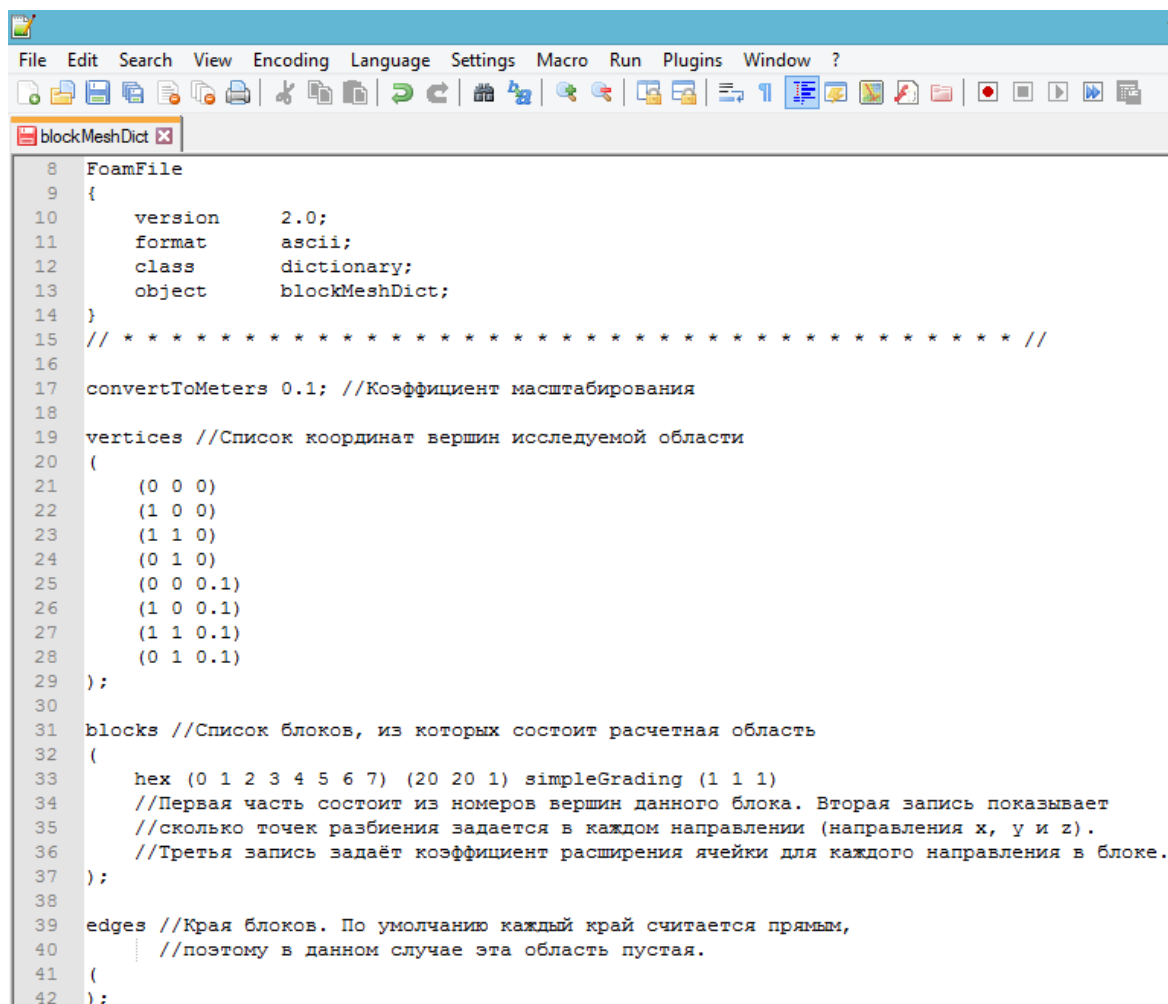
### «Constant» – постоянные значения

Данная задача является несжимаемой и ламинарной, поэтому в ней есть только одна величина, определяющая физические свойства среды – кинематическая вязкость ( $m^2/c$ ). Эта величина задается в файле «transportProperties».

## «System» – метод решения

Файл «blockMeshDict» (блочный генератор сетки, рис. 13 и рис. 14) содержит:

- Параметр масштабирования `convertToMeters`
- Список узлов `vertices`
- 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 гексаэдральные 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.

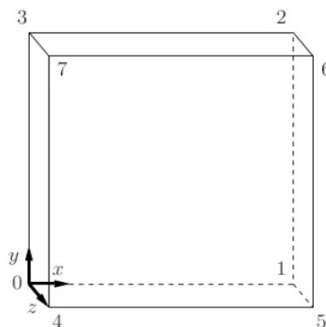


Рис. 15. Гексаэдральный блок.

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         $p;
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».

## Упражнение 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 расчет сетки объекта (рис. 19). Посмотрите, что в «constant» появилась папка «polyMesh». Изучите её содержимое.

```

/c:/lab1/cavity/cavity
$ blockmesh
-----*
|      F ield      |   OpenFOAM: The Open Source CFD Toolbox   |
|      O peration  |   Version: 5.x                                     |
|      A nd        |   Web:      www.OpenFOAM.org                     |
|      M anipulation|   |
|-----*
| * 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-7ac8a133a3b0
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:/lab1/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:/lab1/cavity/cavity/constant/polyMesh"
Creating block mesh from
"C:/lab1/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).
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.

### Упражнение 3. Знакомство с ParaView

Перейдите к работе с ParaView (рис. 20).

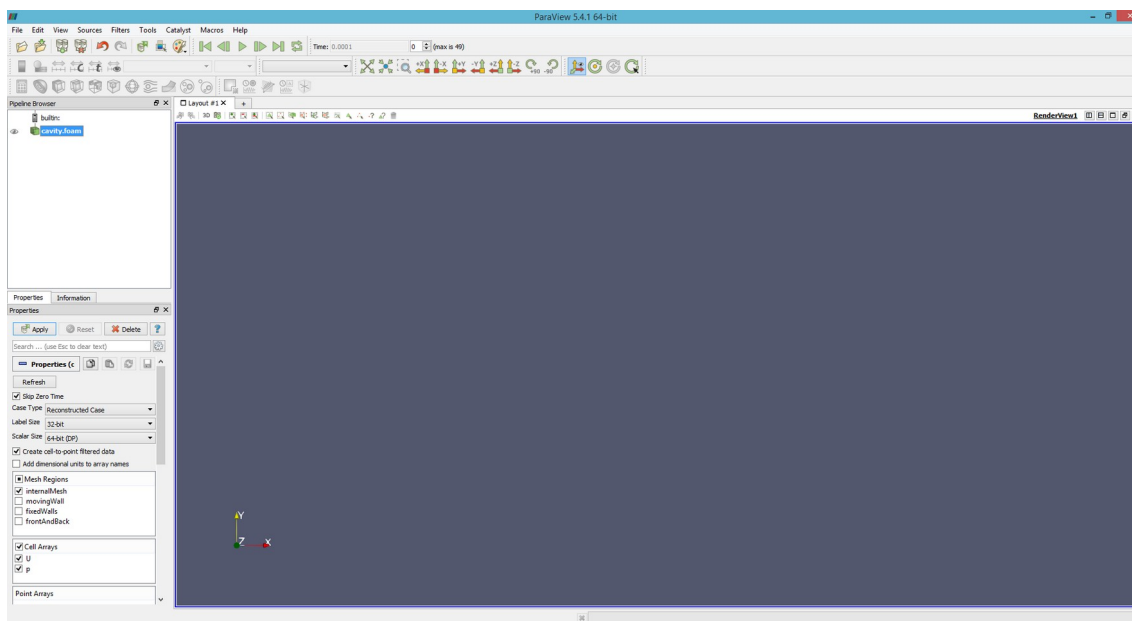


Рис. 20. Окно ParaView.

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



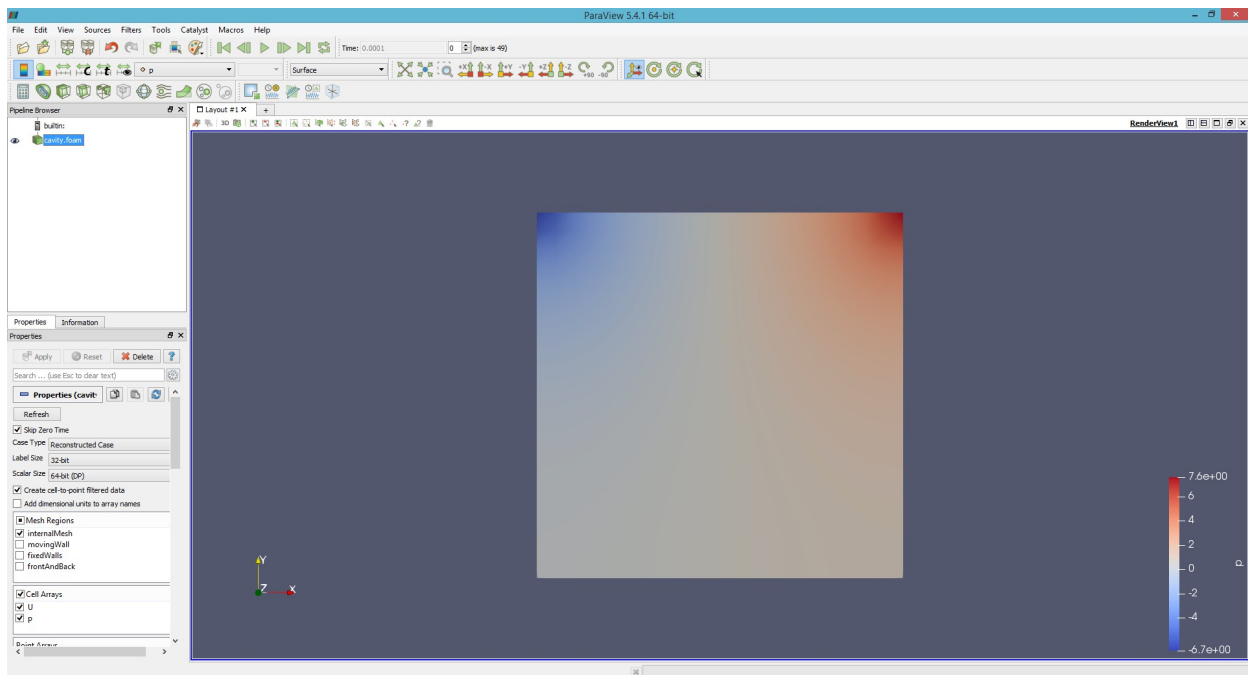
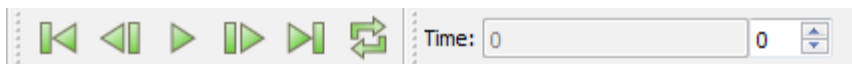








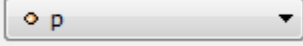
Рис. 21. Изображение давления жидкости в каверне в начале расчёта.


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

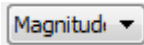
В верхнем меню кнопками можно выбирать время, расчет для которого выводится на экран.





-  – 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  .

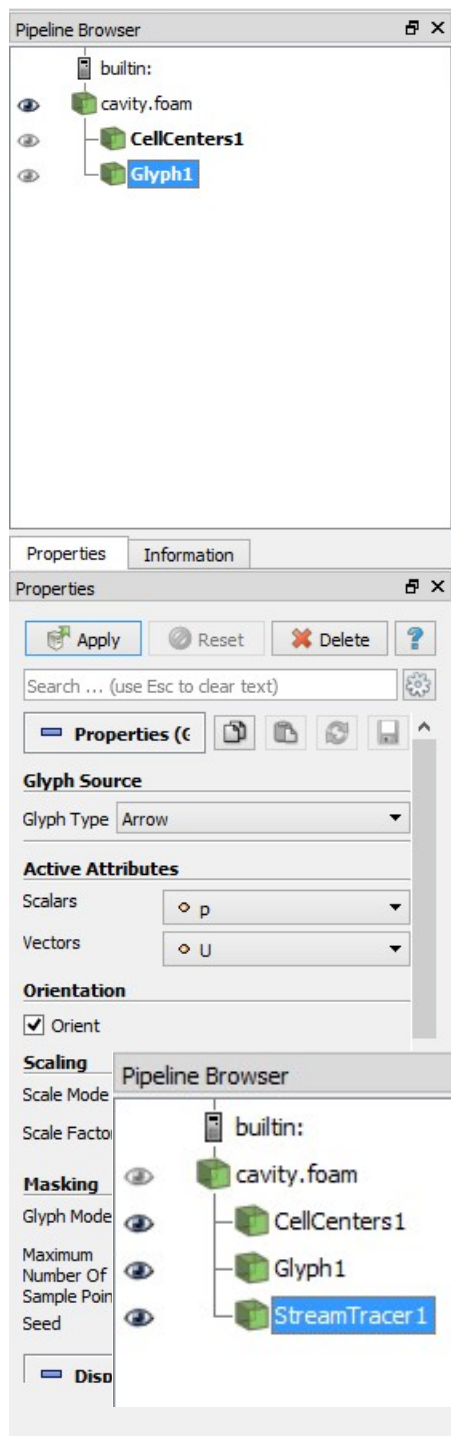


Рис. 22. Выбор off в меню Scale Mode.

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 камере (cavity.foam) и вызовите фильтр Stream Tracer (Filters → Alphabetical → Stream Tracer) (рис. 23). Нажмите Apply.

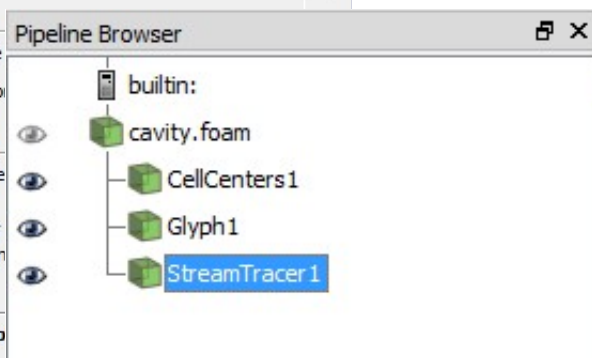


Рис. 23. Вызов фильтра Stream Tracer.

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

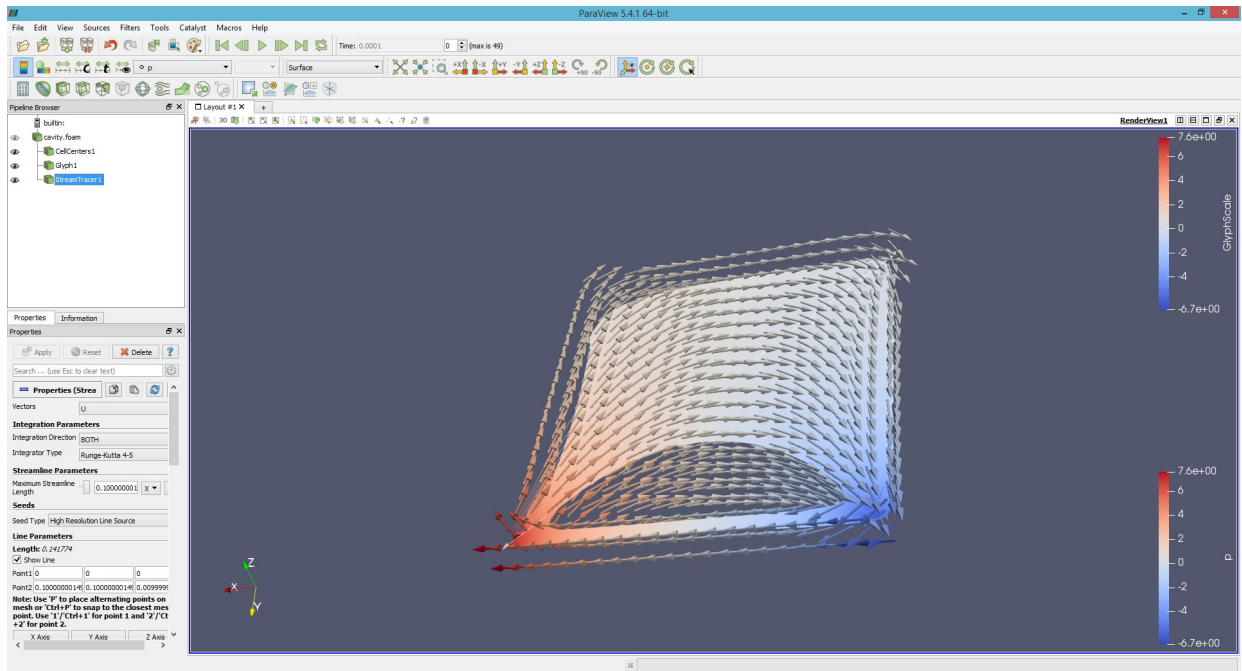


Рис. 24. Просмотр StreamTracer1 и Glyph1.

#### Упражнение 4. Изменение параметров задачи

Получите решения, внося следующие изменения:

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

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.

## Лабораторная работа №2

**Тема:** Демонстрация решения задачи о прорыве 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.

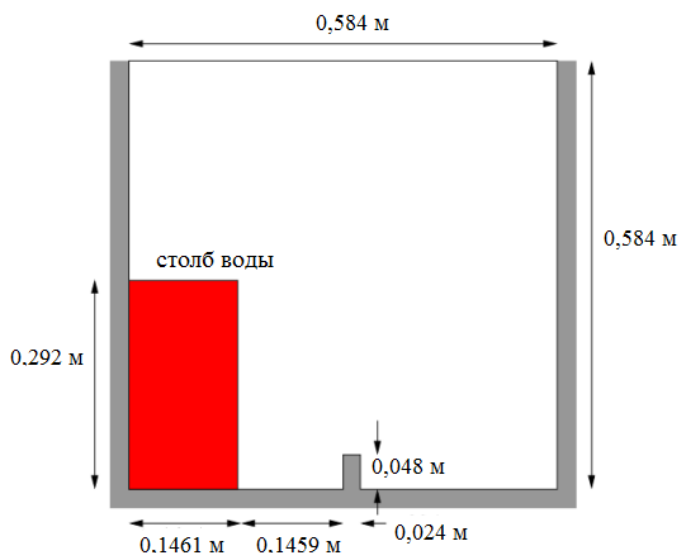


Рис. 26. Геометрия и начальные параметры задачи.

## Ход работы

### Упражнение 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. 27).

```
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. 27. 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 лабораторной работе №1).

При последующих изменениях параметров, необходимо повторять данные действия:

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

### Упражнение 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.

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. 28). 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. 28. Добавление блока boxToCell.



Произведите расчет и проанализируйте полученный результат.

Сделайте из двумерной задачи трёхмерную, увеличив в файле blockMesh 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. 29) (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. 30).

<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) ); </pre> <p>Исходный вариант</p>	<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 4)   (2 0 4)   (2.16438 0 4)   (4 0 4)   (0 0.32876 4)   (2 0.32876 4)   (2.16438 0.32876 4)   (4 0.32876 4)   (0 4 4)   (2 4 4)   (2.16438 4 4)   (4 4 4) );  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) ); </pre> <p>Вариант с увеличенным масштабом</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Рис. 29. 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. 30. Добавление шара воды.

### Упражнение 3. Изменение транспортных свойств

Зайдите по ссылке [C:/ВАША\\_ПАПКА\\$/damBreak/damBreak/constant/transportProperties](C:/ВАША_ПАПКА$/damBreak/damBreak/constant/transportProperties) и поменяйте вязкость жидкости ( $\nu$ ). Произведите расчет и проанализируйте полученный результат.

### Упражнение 4. Изменение параметров расчета

В папке [C:/ВАША\\_ПАПКА\\$/damBreak/damBreak/system](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>

## Лабораторная работа №3

**Тема:** Знакомство с программой Salome.

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

### Ход работы

#### Упражнение 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;

Dx: 0.002;

Dy: 0.002;

Dz: 0.

Для второй точки:

Name: TopD;

Dx: 0.023;

Dy: 0.010;

Dz: 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. 31).

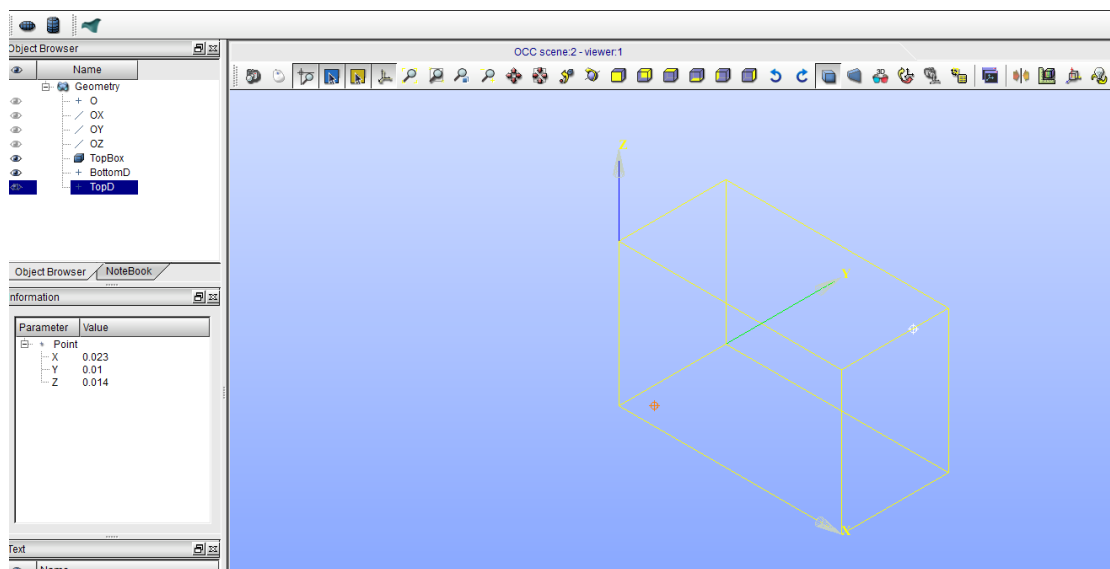


Рис. 31. Каркасный режим отображения.

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. 32).

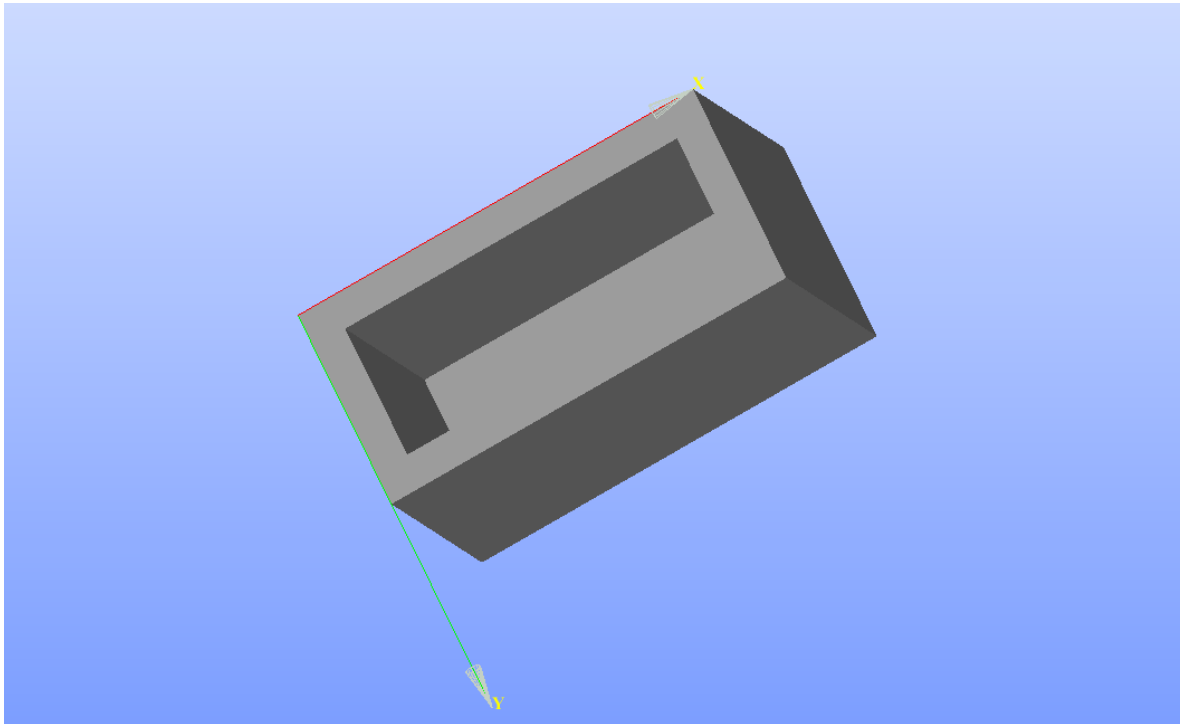


Рис. 32. Режим отображения трёхмерной модели.

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. 33). Объединять можно по 2 объекта сразу, поэтому будет необходимо провести операцию объединения 2 раза.

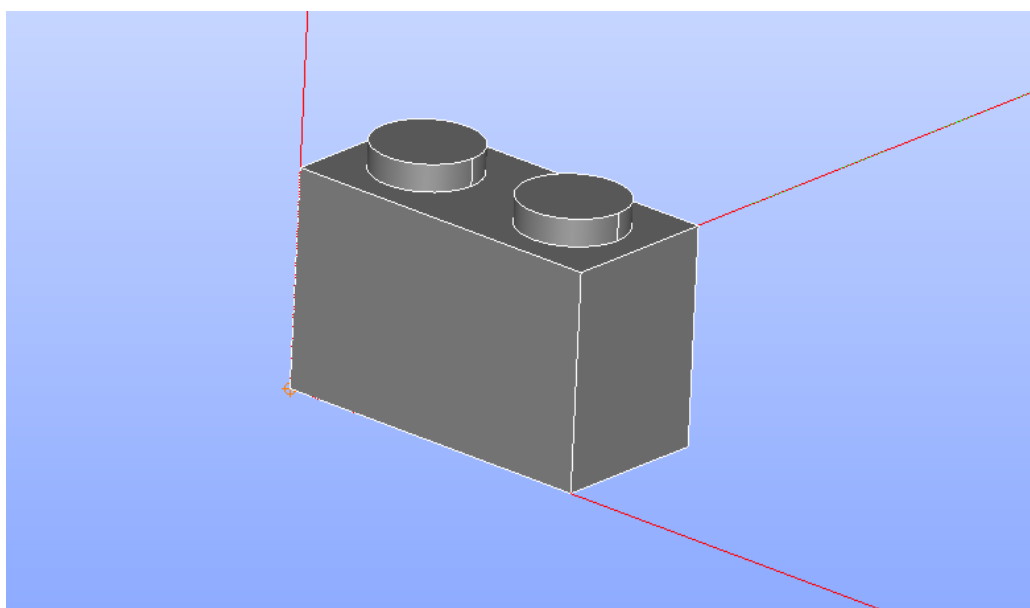


Рис. 33. Выбор объектов для применения логической операции Fuse.

## Упражнение 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**.

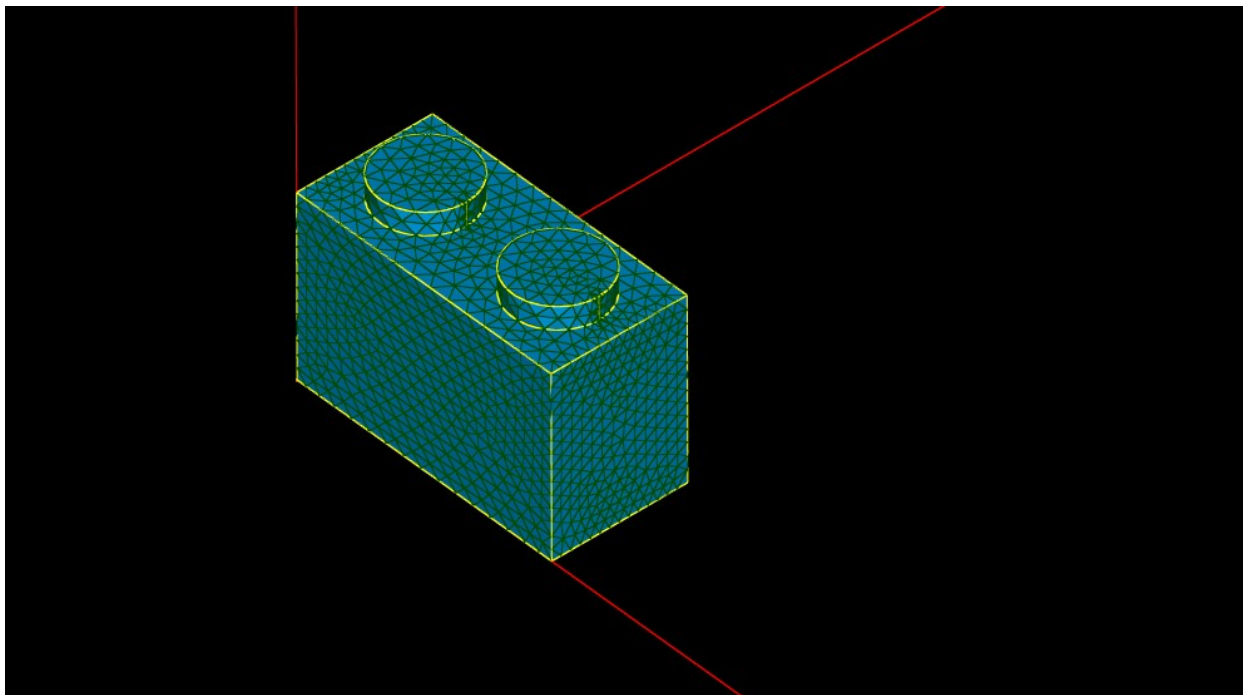


Рис. 34. Разбиение объекта более чем на 14000 элементов.

Объект оказался разбит более чем на 14000 объемных элементов (рис. 34). Для уменьшения времени расчета необходимо уменьшить количество элементов. Будем уменьшать количество элементов в три раза.

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. 35, 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.

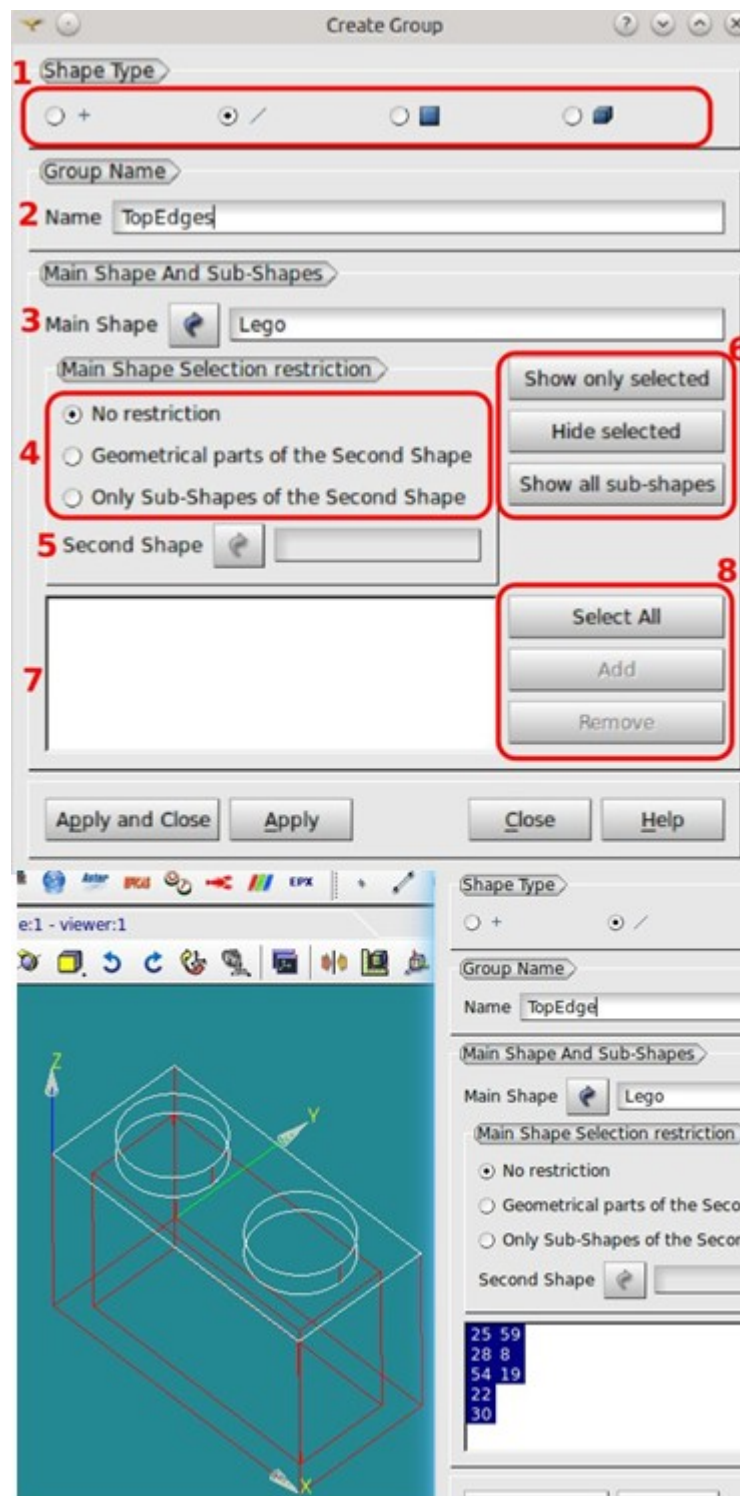


Рис. 35. Создание группы рёбер.

Нажав кнопку Apply и не закрыв окно, перейдем к созданию группы граней. Для этого в верхней строке следует выбрать соответствующий тип элемента (рис. 36), выделить аналогично рёбрам необходимые грани (их будет 5) (рис. 36) и нажать кнопку Add.

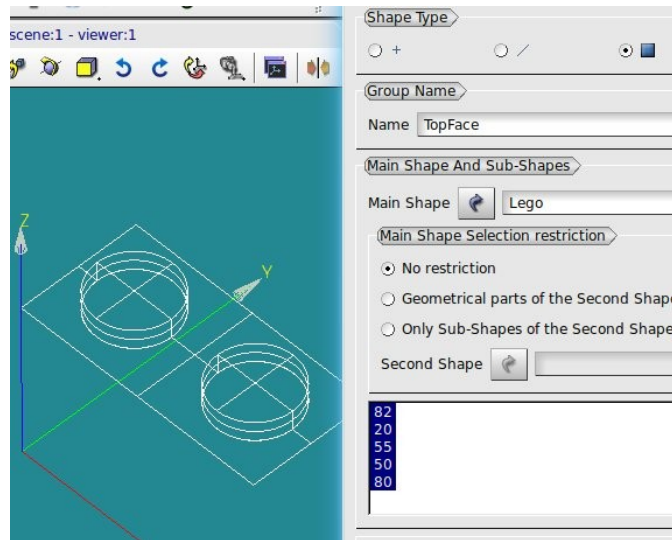


Рис. 36. Создание группы граней.

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. 37, после чего нажмем на кнопку **Apply and Close**.

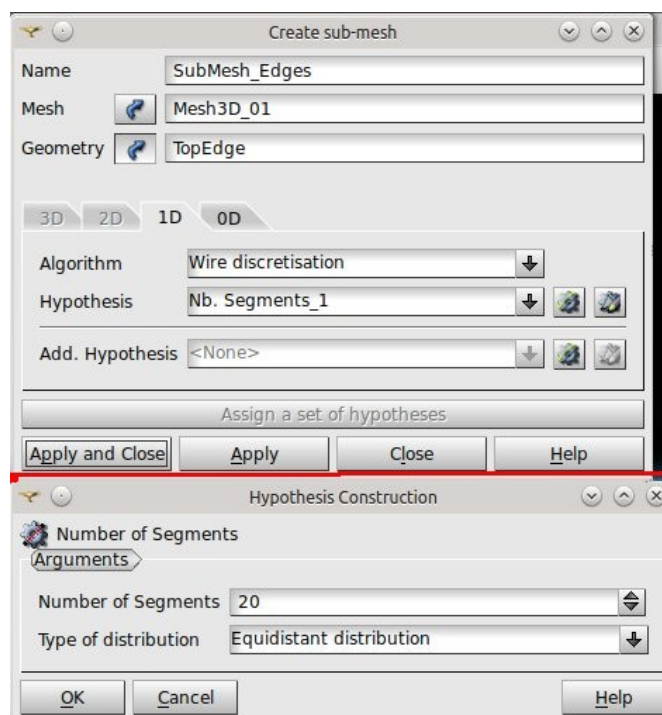


Рис. 37. Создание подсетки для группы ребер.

Следует рассчитать сетку с помощью команды **Mesh — Compute**. Также по рис. 38 необходимо сделать для граней и нажать на кнопку **Apply and Close**.

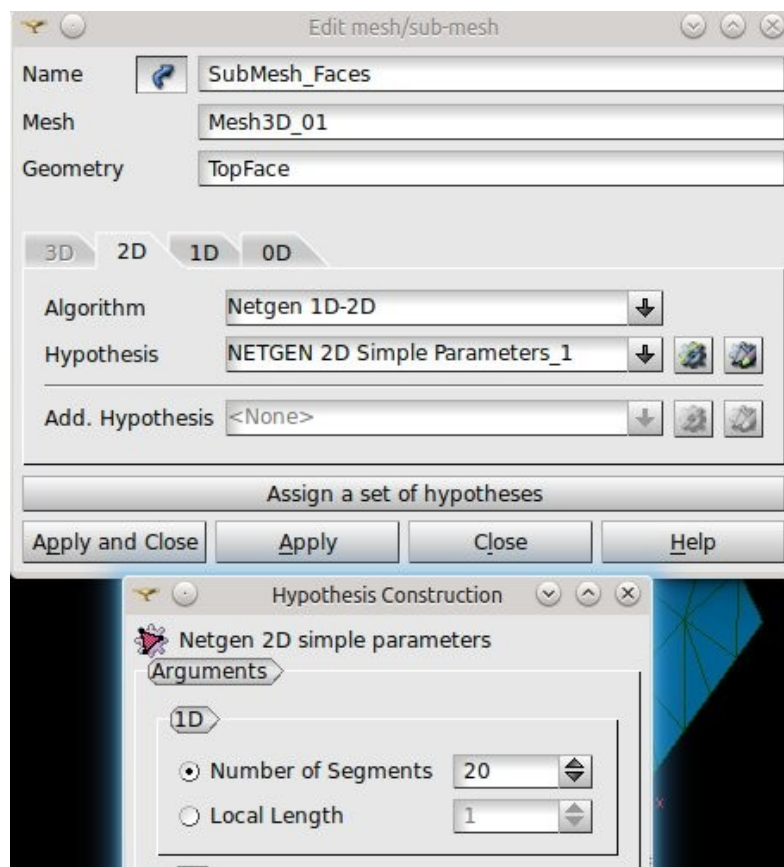


Рис. 38. Создание подсетки для группы граней.

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. 39.

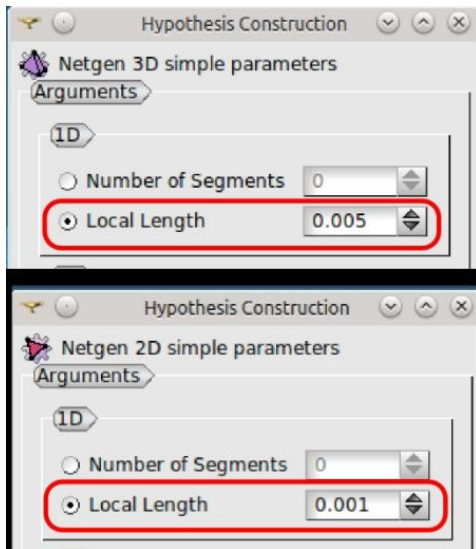


Рис. 39. Изменение гипотезы формирования сетки.

Произвести пересчёт с помощью команды Compute. Получится сетка и подсетка с одинаковыми расстояниями между узлами (рис. 40).

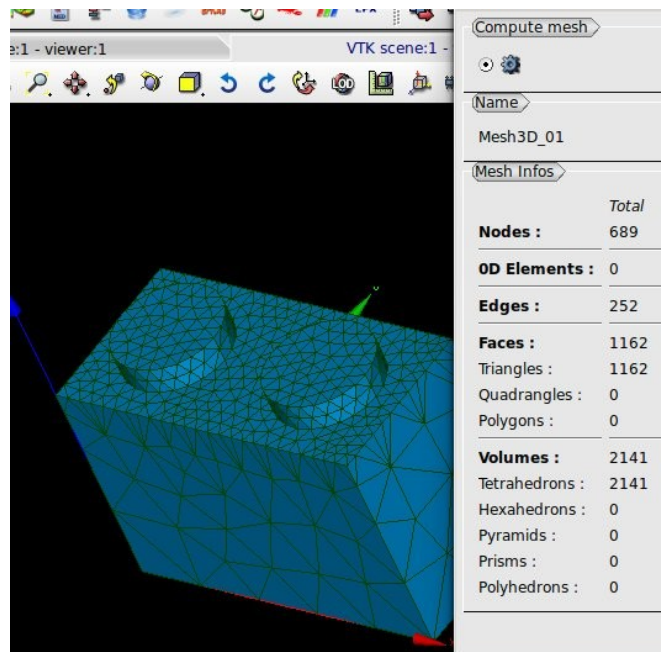


Рис. 40. Пересчитанная сетка.

### Упражнение 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, иначе проверка checkMesh выдаст ошибку) 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**. Далее ввести **checkMesh** для проверки правильности импорта сетки.

## Лабораторная работа №4

**Тема работы:** Течение 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  $U = (10, 0, 0) \text{ м/с}$ ;
- Выход (справа) с фиксированным давлением  $p = 0 \text{ Па}$ ;
- 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  $\nu_{eff}$ , вычисленная из выбранных моделей переноса и турбулентности.

## Ход работы

### Упражнение 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. SimpleFoam: выполняет расчет для устойчивого несжимаемого потока.
5. Запустите счет командой simpleFoam.
6. Визуализируйте решение.

### Упражнение 2. Изменение граничных условий

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

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

Сравните получившееся решение с предыдущим.



### Упражнение 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  $\nu$  – постоянная величина (рис. 41):

$$\nu = const$$

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

Рис. 41. Ньютоновская модель кинематической вязкости.

Bird-Carreau model:

$$\nu = \nu_{\infty} + (\nu_0 - \nu_{\infty}) [1 + (k\dot{\gamma})^n]^{\frac{n-1}{n}},$$

где  $\nu_{\infty}$  и  $\nu_0$  – кинематическая вязкость при бесконечной и нулевой скорости деформации соответственно, параметр  $\alpha$  по умолчанию имеет значение 2,  $\dot{\gamma}$  – скорость деформации,  $n$  и  $k$  – параметры реологических моделей (рис. 42).

```
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. 42. Модель Bird-Carreau.

Cross Power Law model (рис. 43):

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

где  $m$  – параметр реологической модели (рис. 43).

```
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. 43. Модель Cross Power Law.

Power Law model – степенная модель ограничивает значение вязкости в определённом диапазоне от минимальной  $v_{min}$  до максимальной  $v_{max}$  (рис. 44):

$$v = k\dot{\gamma}^{n-1} \quad v_{min} \leq v \leq v_{max}$$

```
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. 44. Модель Power Law.

Herschel-Bulkley model – модель Хершеля-Балкли

$$v = \min\left(v_0, \frac{\tau_0}{\dot{\gamma}} + k\dot{\gamma}^{n-1}\right),$$

здесь  $\tau_0$  – 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. 45).

```
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. 45. Модель Хершеля-Балкли.

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

### Упражнение 3. 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. 46).

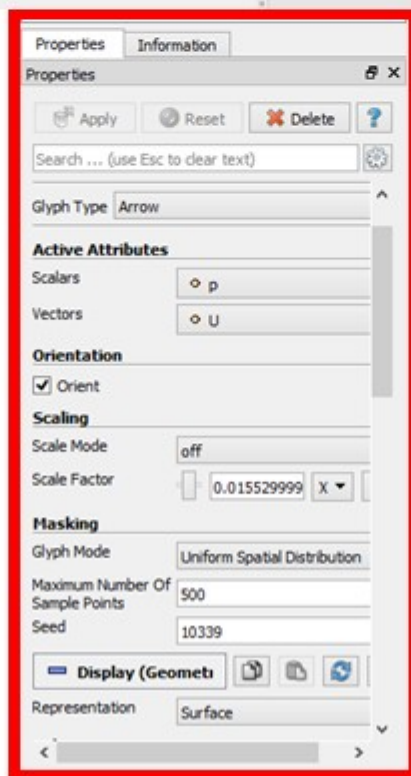
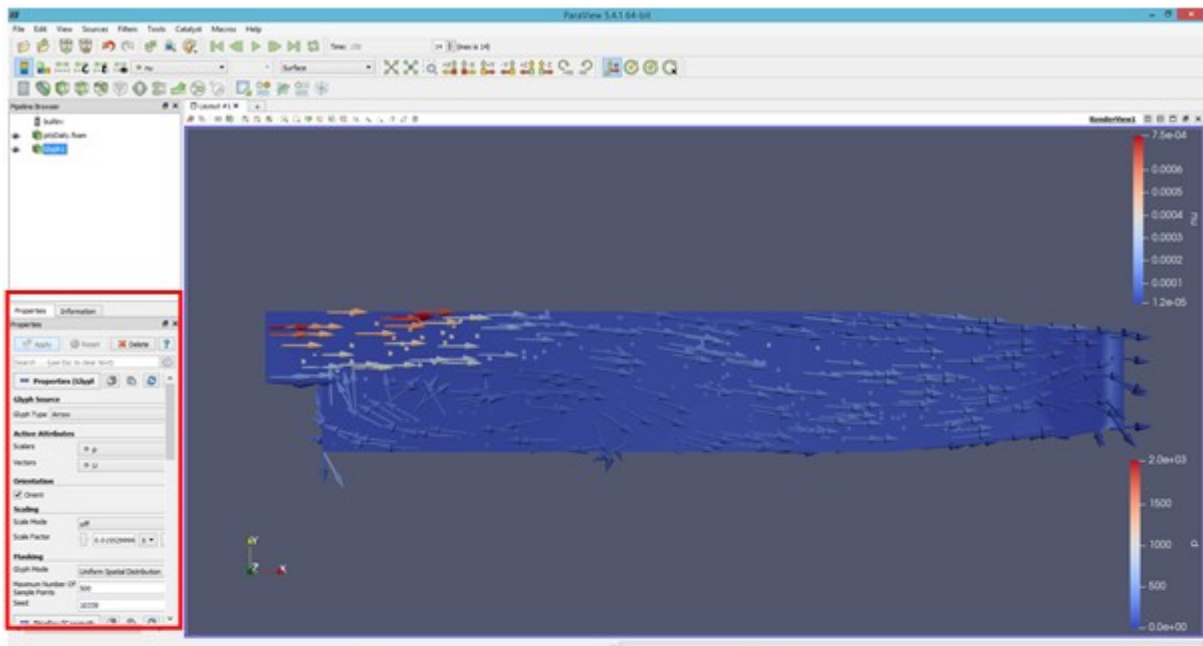


Рис. 46. Изменение количества векторов и их масштаба.

## Лабораторная работа №5

**Тема работы:** Обтекание бруса.

**Цель:** Решить задачу о плоскопараллельном обтекании бруса несжимаемой вязкой жидкостью при различных числах Рейнольдса. Для формирования геометрии используется параметр  $h = 10$  см (0,1м). Использовать модель k-omega SST. Сформировать геометрию расчетной области средствами SALOME (GEOMETRY) в блочно-гексаэдральном виде; определить на геометрии поверхности для задания граничных условий; построить гексаэдральную 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. 47.

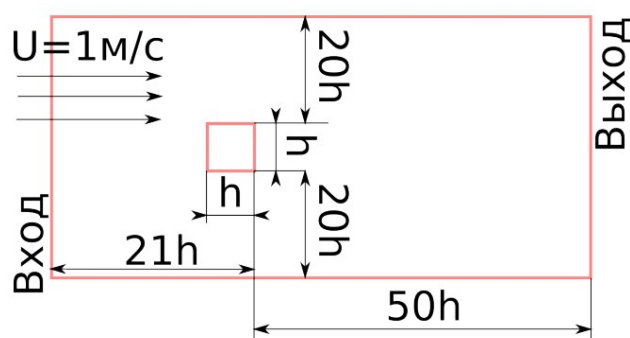


Рис. 47. Условия задачи об обтекании бруса.

## Ход работы:

### 1. Создание блочной геометрии

По схеме расчетной области создадим блочную геометрию. Для этого создайте параллелепипед, нажав **Create a Box** и укажите его размеры (рис. 48).

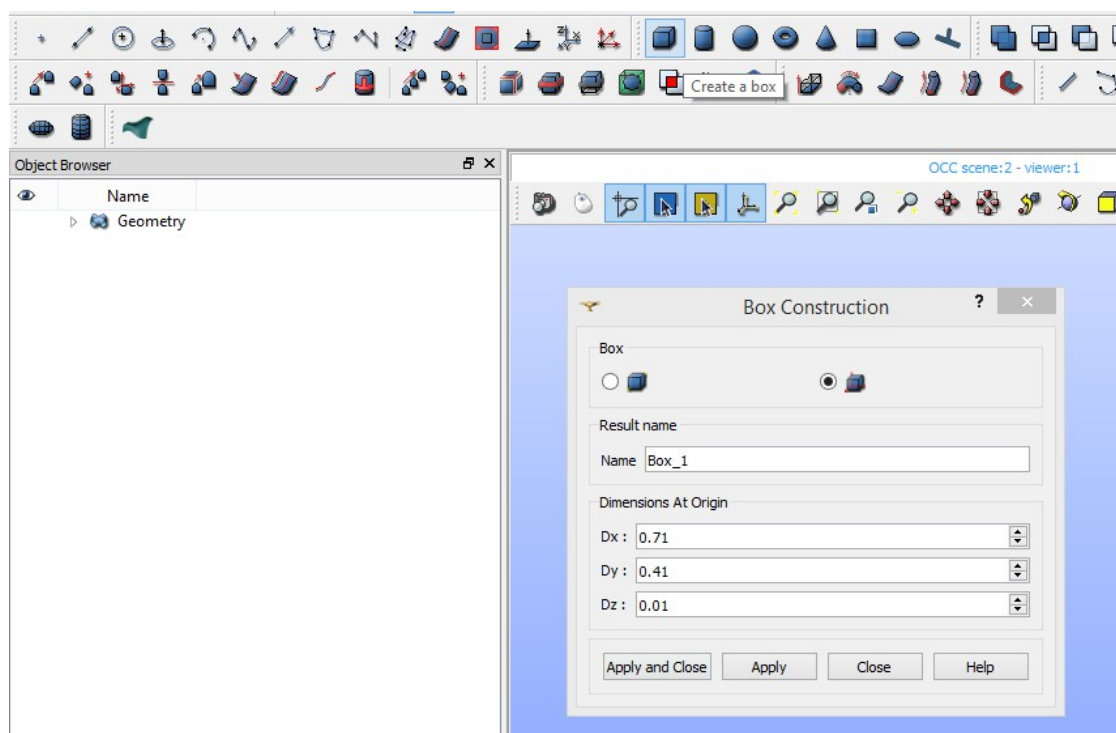


Рис. 48. Создание параллелепипеда.

Нажмите **Apply** и создайте ещё один параллелепипед меньших размеров (рис. 49). Нажмите **Apply and Close**. Переместите его с помощью операции **Translation**, предварительно убрав флажок с **Create a copy** (рис. 50).

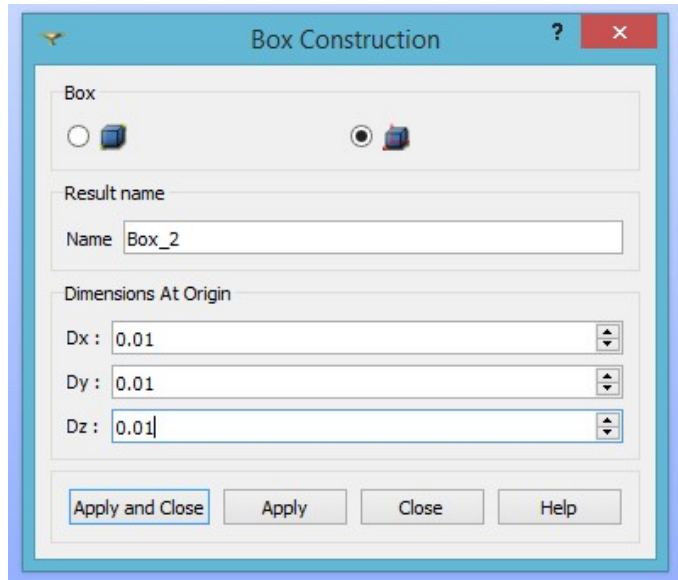


Рис. 49. Создание параллелепипеда меньших размеров.

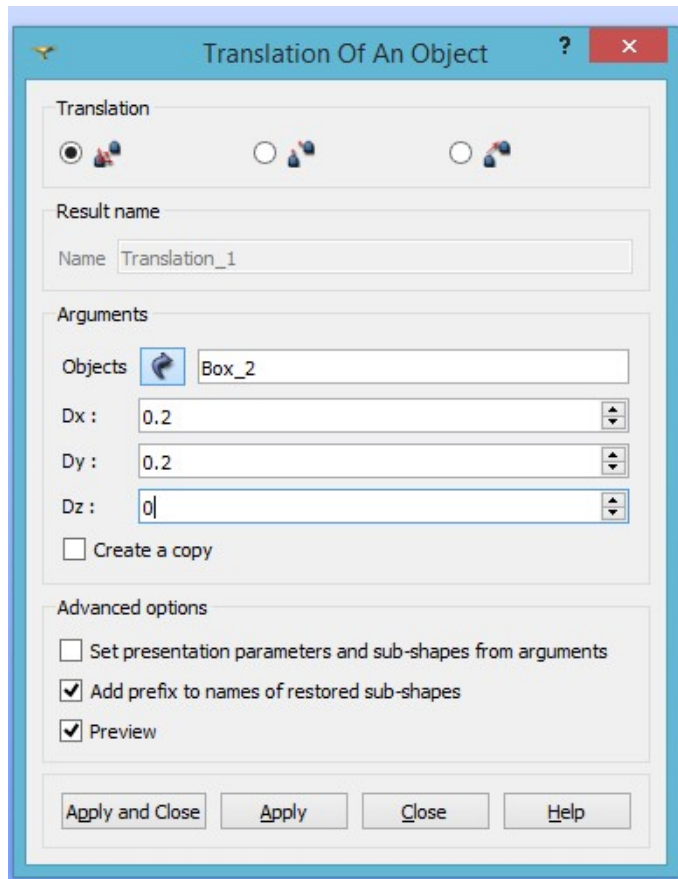


Рис. 50. Перемещение параллелепипеда.

С помощью логической операции **Cut** вырежьте Box\_2 из Box\_1 (рис. 51).

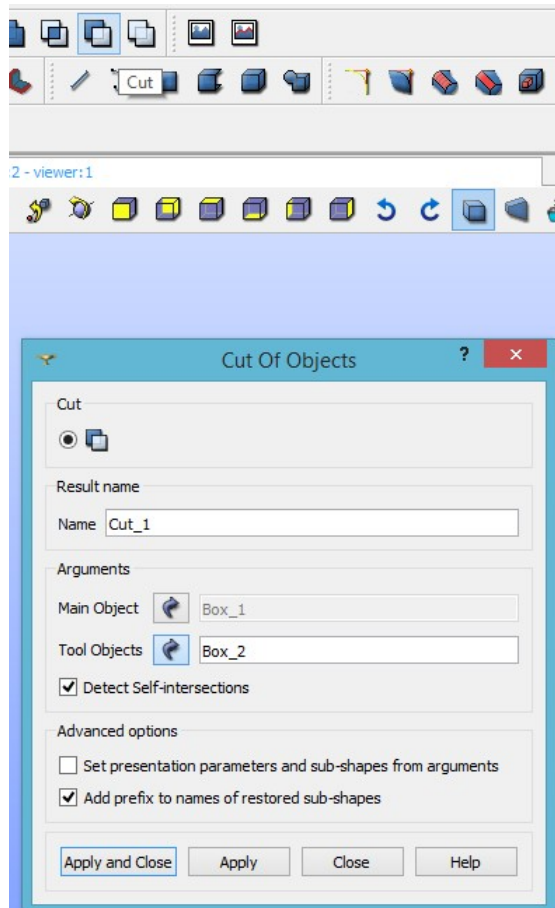


Рис. 51. Применение логической операции Cut.

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. 52). Таким же образом создайте еще три плоскости.



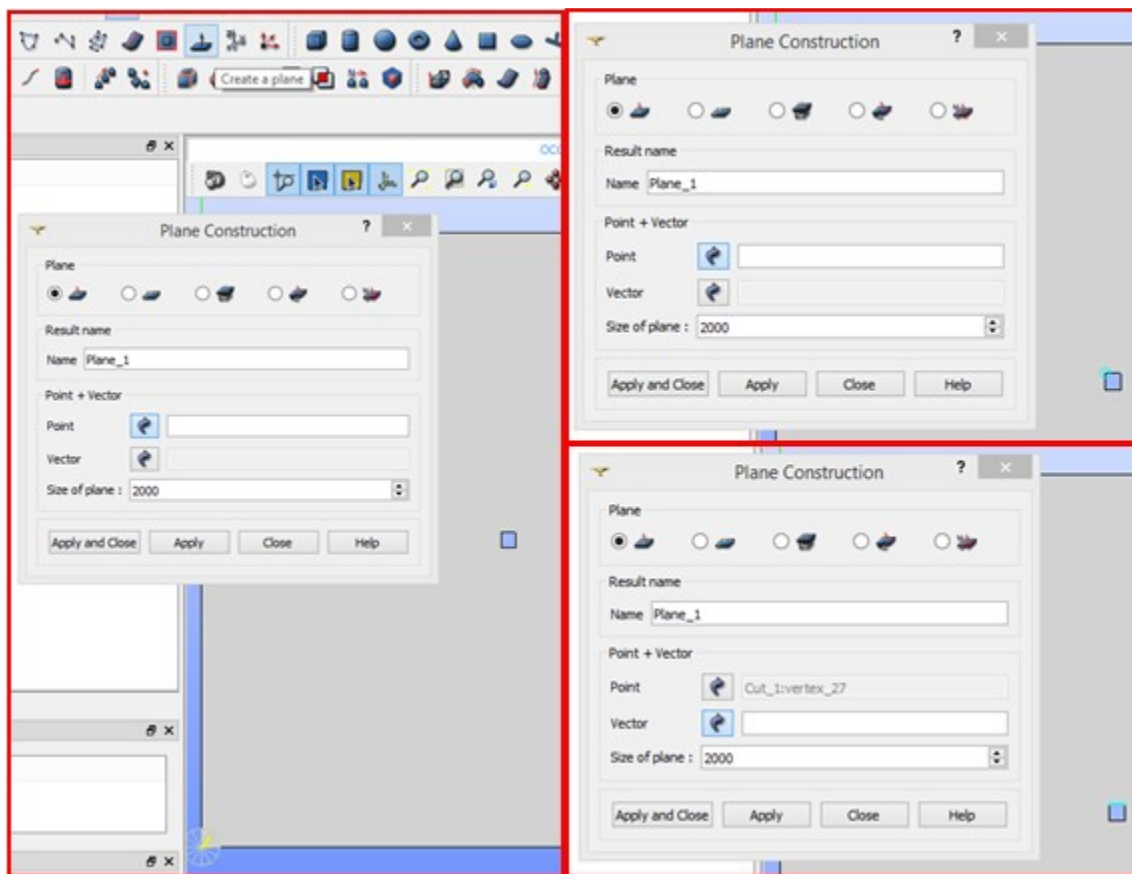


Рис. 52. Построение плоскости.

Далее нажмите **Partition** и разделите Cut\_1 с помощью полученных 4 плоскостей на отдельные блоки (Solid) (рис. 53). Таким способом получается единый объект, состоящий из блоков.

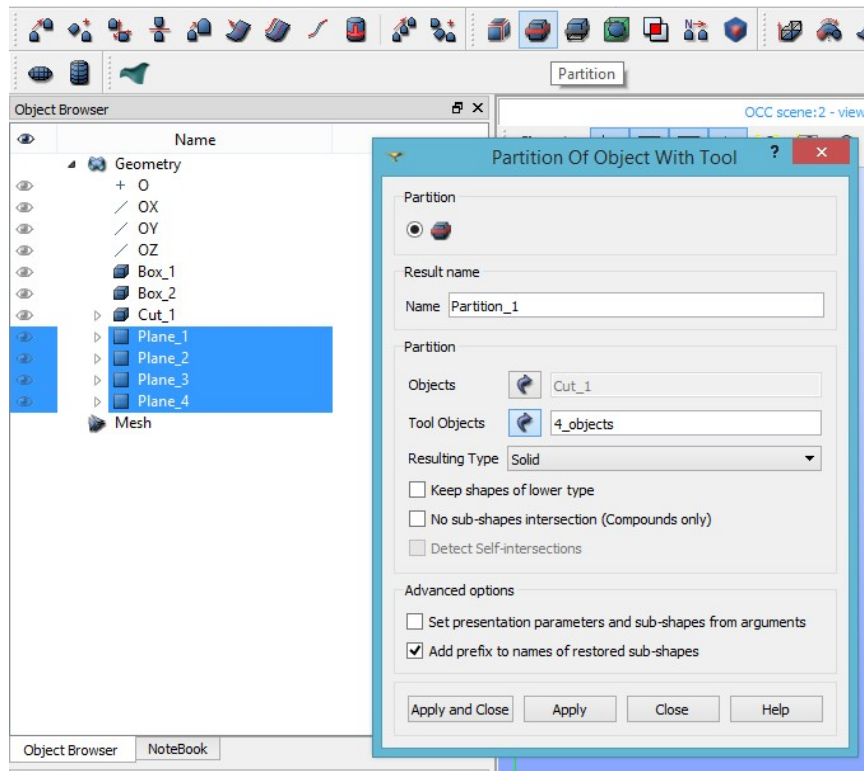


Рис. 53. Применение операции Partition.

## 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)** (гексаэдральные ячейки) (рис. 54).

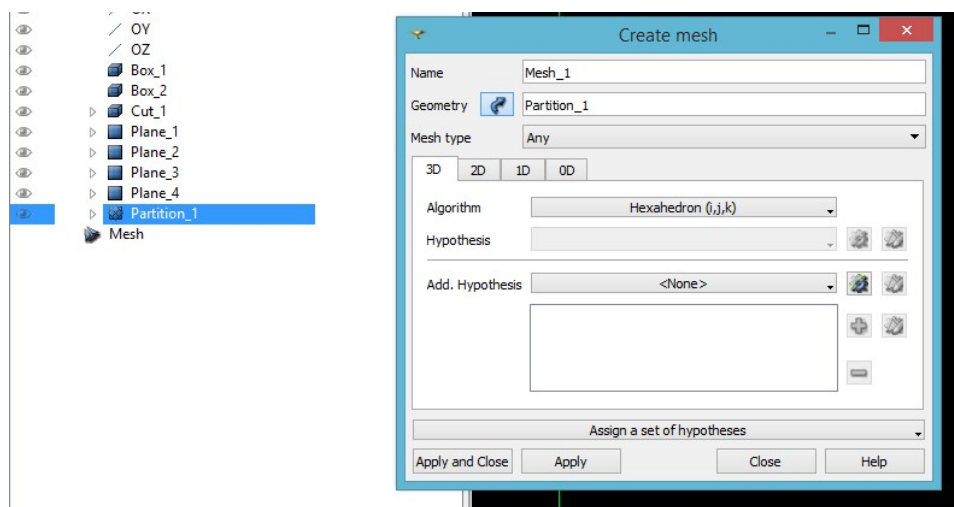


Рис. 54. Создание сетки с гексаэдральными ячейками.

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. 55).

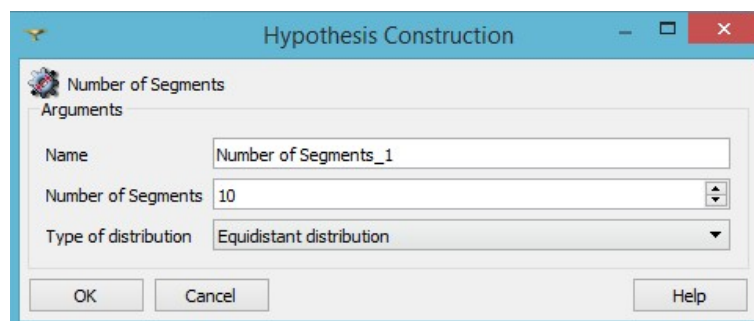


Рис. 55. Указание числа сегментов для гипотезы построения сетки.

Рассчитаем сетку с помощью команды **Compute** (рис. 56).

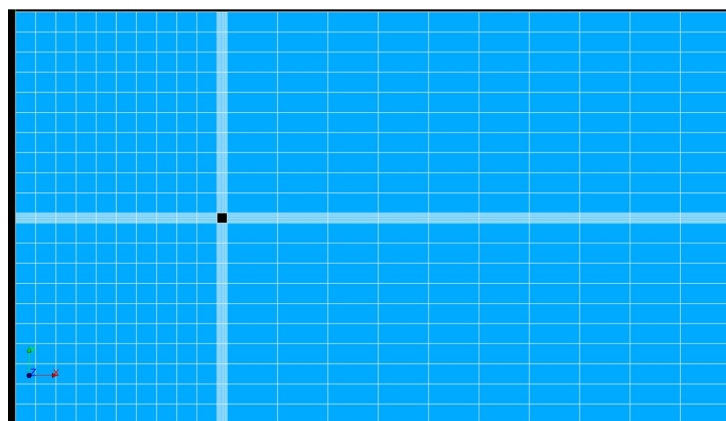


Рис. 56. Получившаяся сетка для задачи об обтекании бруса.

Далее необходимо перейти в модуль **Geometry** и с помощью операции **Explode** разбить **Partition\_1** на отдельные поверхности (рис. 57).

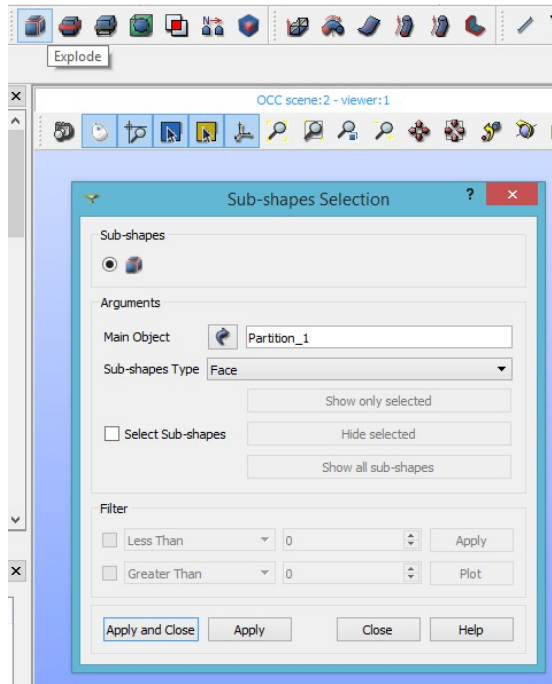


Рис. 57. Разбиение Partition\_1 на отдельные поверхности.

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. 58). Стенки бруса (cubeWalls) также следует объединить в группу.

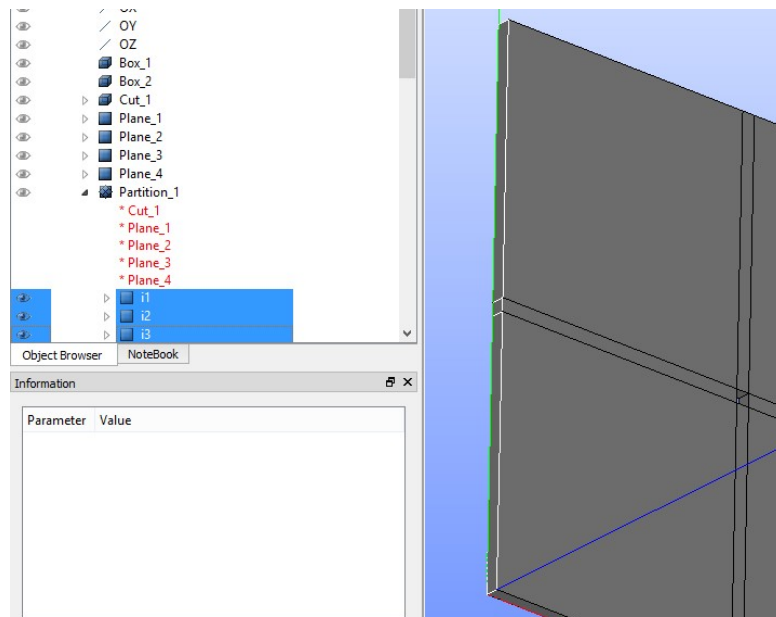


Рис. 58. Переименование поверхностей.

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. 59). Аналогичным образом создадим еще 6 групп для оставшихся стенок.

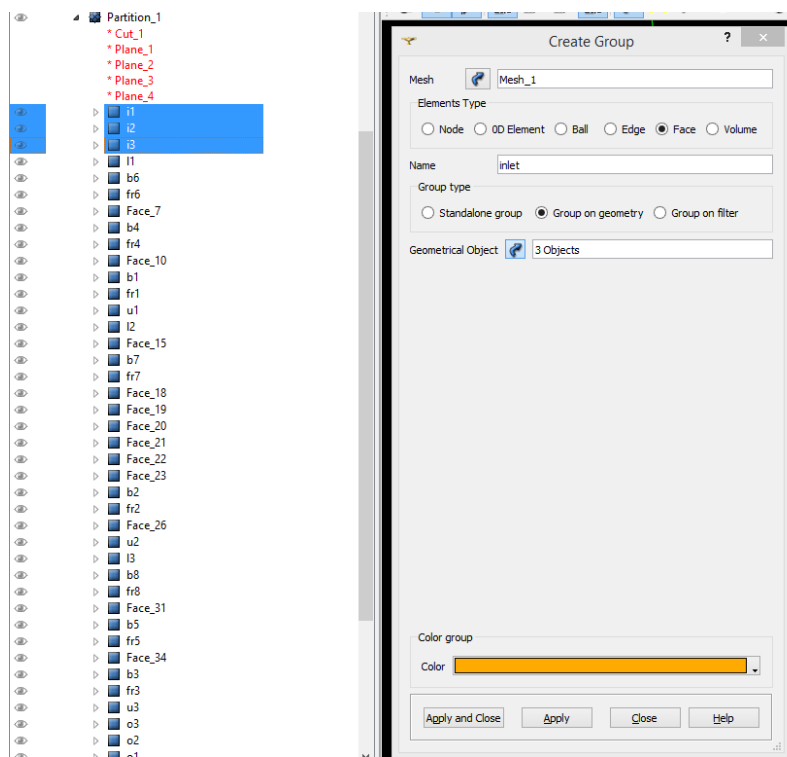


Рис. 59. Группировка поверхностей для создания сетки.

Выберем Mesh\_1 и сохраним UNV файл в корневую папку проекта OpenFoam. В качестве корневой папки используйте копию папки pitzDaily из предыдущей работы.

### 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 ( $\omega$ ), турбулентной вязкости ( $\nu_t$ ). Файлы остальных величин в папке «0» необходимо удалить.

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

#### 4. Граничные условия

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

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

- На стенках канала (upper, lower, front, back) — условие проскальзывания (slip) для всех величин, кроме  $\nu_t$ , для которой 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 бруса (cubeWalls) — условие прилипания (noSlip) для скорости, пристеночные функции для  $k$  (type kqRWallFunction; value uniform 0.375),  $\omega$  (type omegaWallFunction; value \$internalField;) и  $\nu_t$  (как для upper и lower), условие непроницаемости (zeroGradient) для  $p$ .

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

#### 5. Константы модели

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

1)  $Re=100$ , вязкость —  $1e-4$  (при  $h=10\text{см}$ ,  $U_x=1\text{м/с}$ )

- 2)  $Re=1000$ , вязкость —  $1e-5$  (при  $h=10\text{см}$ ,  $U_x=1\text{м/с}$ )
- 3)  $Re=10000$ , вязкость —  $1e-6$  (при  $h=10\text{см}$ ,  $U_x=1\text{м/с}$ )
- 4)  $Re=100000$ , вязкость —  $1e-7$  (при  $h=10\text{см}$ ,  $U_x=1\text{м/с}$ )

## 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. Расчет проводится по итерациям (шаг по времени = 1), вывод через каждые 10 или 100 шагов. Общее число итераций – 150 или 1000. Коэффициент релаксации для скорости:  $r_U = 0.9$  (рис. 60), 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. 60. Коэффициент релаксации для скорости.

## Решения

Решения отобразить с помощью `ParaView` (рис. 61), смотреть скорость.

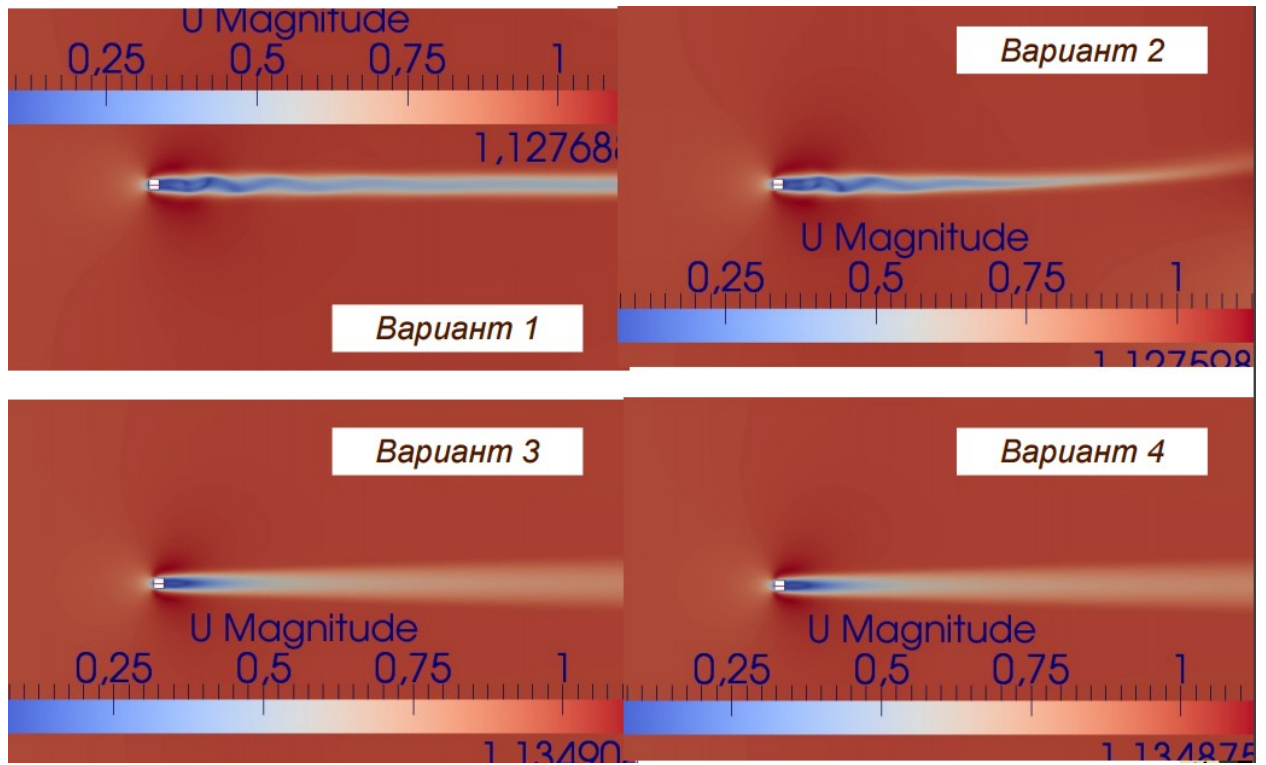


Рис. 61. Примерный вид решения задачи об обтекании бруса.



## Самостоятельная работа №1

**Тема работы:** Обтекание 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. 62, диаметр сферы  $d=10$  см, основная фигура – куб.

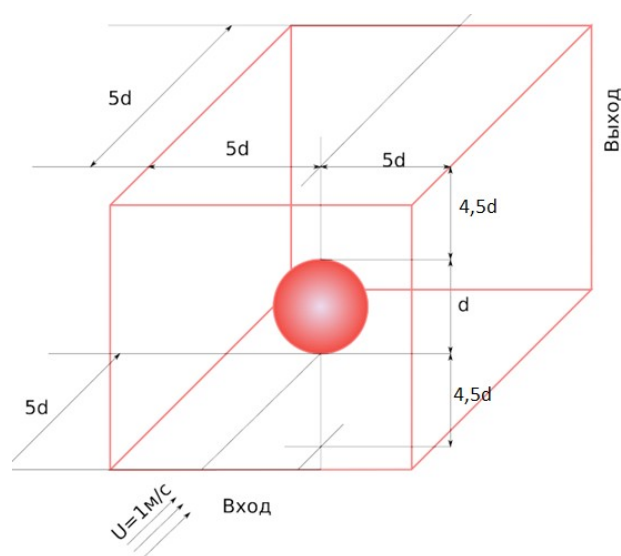


Рис. 62. Условия задачи об обтекании сферы.

### Ход работы:

#### 1. Построение геометрии и расчетной сетки

Средствами Salome создаётся геометрия, как на рис. 63, и расчётная сетка, как на рис. 64. Сфера задаётся с помощью объекта Sphere и параметра радиус (Radius).

Сначала рекомендуется создать куб, составляющий одну четвертую часть от основной фигуры, поместив левую нижнюю его вершину в начала координат. Туда же поместить центр сферы и вырезать её из куба (рис. 65) с помощью операции Cut.

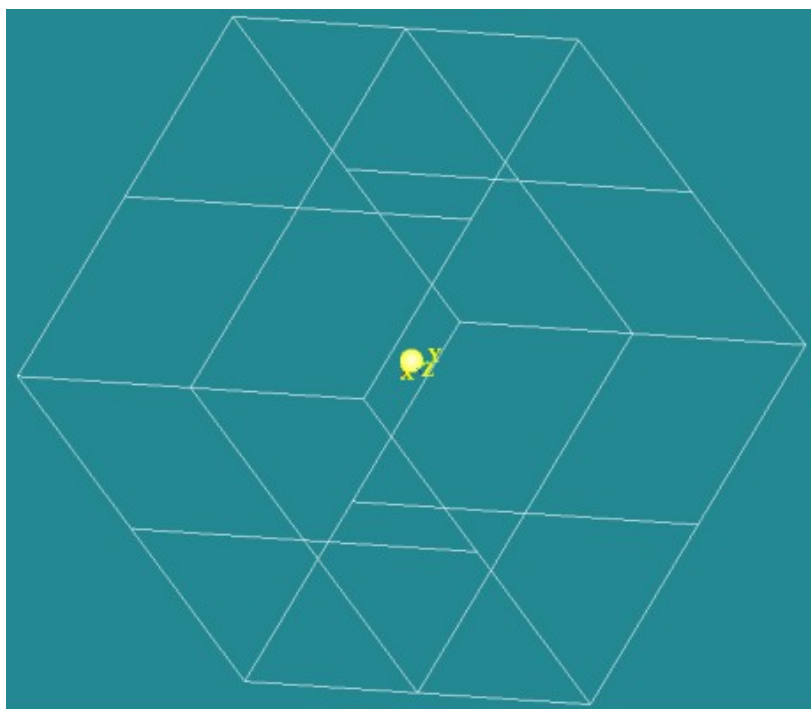


Рис. 63. Геометрия задачи об обтекании сферы.

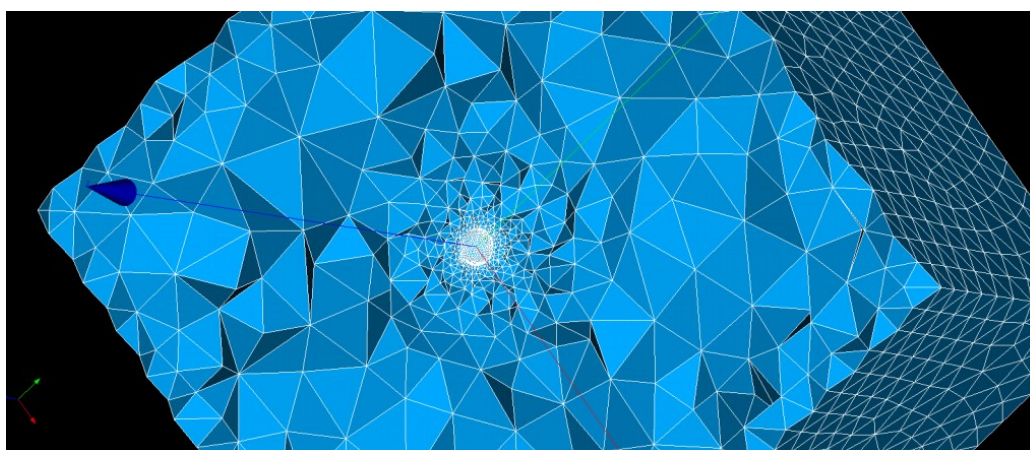


Рис. 64. Расчётная сетка задачи об обтекании сферы.

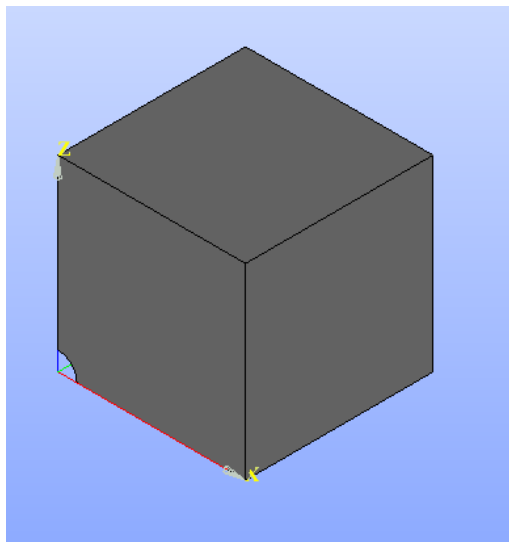


Рис. 65. Куб с вырезанной сферой.

Далее с помощью операции Rotation (вращение, необходимо поворачивать первый куб вокруг оси  $x$  на углы  $90^\circ$ ,  $-90^\circ$  и  $180^\circ$ ) получить половину исходной фигуры, показанную на рис. 66. 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. 66, 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.

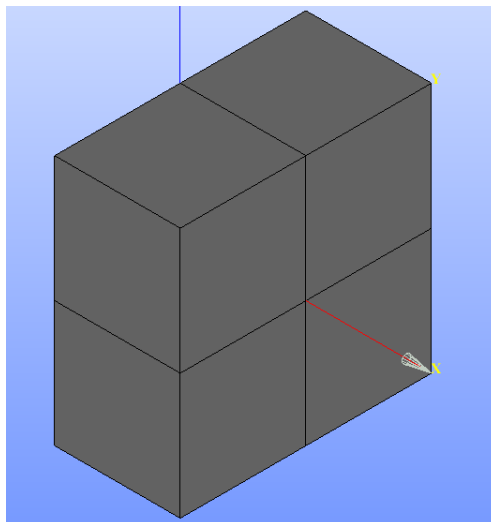


Рис. 66. Половина исходной фигуры.

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.

## 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 ( $\omega$ ), турбулентной вязкости ( $\nu_{\text{t}}$ ). Файлы остальных величин в папке «0» необходимо удалить.

В нулевой момент времени  $p=0$ ,  $U=(0\ 0\ 0)$ ,  $k=1\text{e-}3$ ,  $\omega=1$ ,  $\nu_{\text{t}}=0$ .

Граничные условия:

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

- На выходе (outlet) — задается давление  $p=0$  (тип fixedValue),  $\nu=0$  (тип calculated), нулевые градиенты для  $U$ ,  $k$ ,  $\omega$ .
- На стенках канала (upper, lower, front, back) — условие проскальзывания (slip) для всех величин, кроме  $\nu$ , для которой на верхней upper и нижней lower границах задается пристеночная функция (type nutkWallFunction; value uniform 0, первая строка в файле – тип (пристеночная функция), вторая – её величина).
- На стенках сферы (sphere-walls) — условие прилипания (noSlip) для скорости, пристеночные функции для  $k$  (type kqRWallFunction; value uniform 0.375),  $\omega$  (type omegaWallFunction; value \$internalField;) и  $\nu$  (как для upper и lower), условие непроницаемости (zeroGradient) для  $p$ .

### 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  $d=10\text{см}$ ,  $U_x=1\text{м/с}$ )

### 4. Параметры счёта

Расчет 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.

### Решения

Решения отобразить с помощью ParaView (рис. 67).

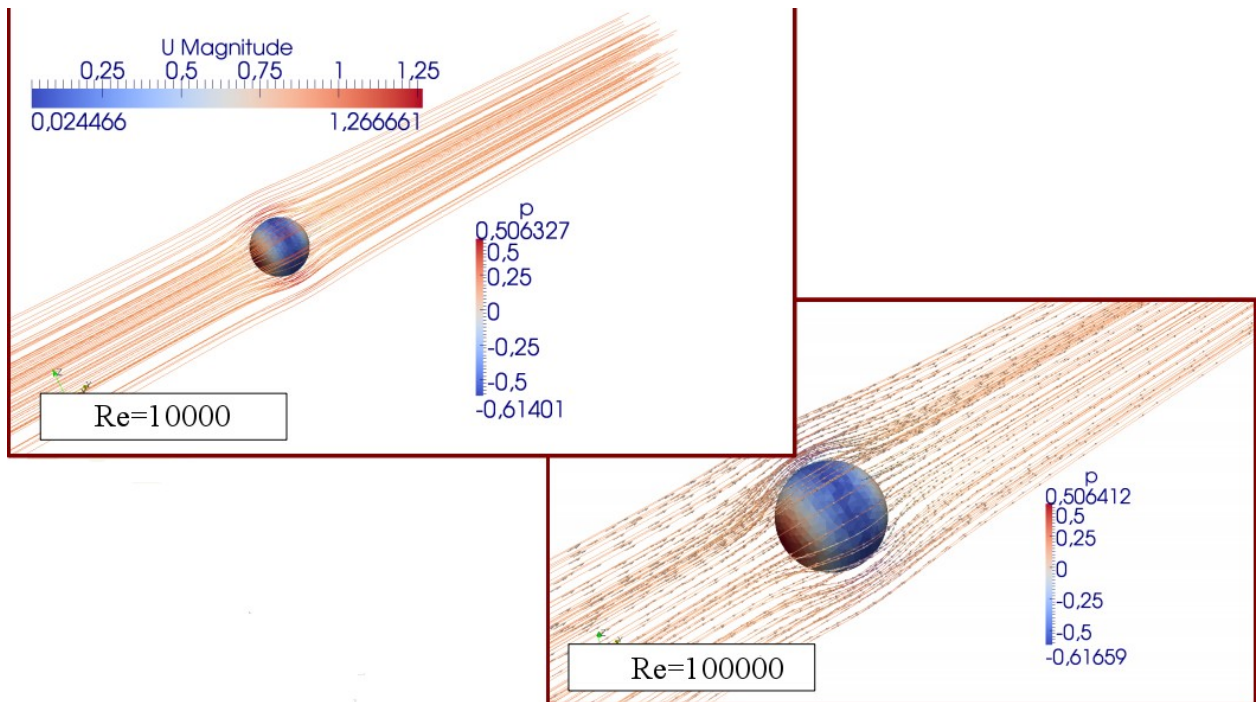


Рис. 67. Примерный вид решения задачи об обтекании сферы.