

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

А. Ю. АБАЛЯЕВ    А. Б. ЛЮХТЕР

ОСНОВЫ МОДЕЛИРОВАНИЯ  
ГАЗОДИНАМИЧЕСКИХ ПРОЦЕССОВ  
И КОНВЕКТИВНОГО ТЕПЛООБМЕНА  
В ПРОГРАММНОМ КОМПЛЕКСЕ  
*OpenFOAM*

Учебное пособие



Владимир 2023

УДК 536.2  
ББК 31.16  
А12

Рецензенты:

Доктор технических наук, профессор  
начальник отдела «Математическое моделирование и САПР ГТД»  
Центрального института авиационного моторостроения  
имени П. И. Баранова  
*Ю. М. Темис*

Доктор технических наук, доцент  
старший научный сотрудник  
специалист по сертификации ООО «Оружейные мастерские»  
*А. Р. Кульчицкий*

**Абляев, А. Ю.** Основы моделирования газодинамических  
А12 процессов и конвективного теплообмена в программном ком-  
плексе *OpenFOAM* : учеб. пособие / А. Ю. Абляев, А. Б. Люх-  
тер ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир :  
Изд-во ВлГУ, 2023. – 104 с.  
ISBN 978-5-9984-1708-5

Изложены основные сведения по работе с программным комплексом *OpenFOAM*. Приведены рекомендации по установке и настройке комплекса. Детально рассмотрены процессы построения и даны рекомендации по выбору параметров расчётных сеток, обеспечивающих наилучшую точность результатов моделирования гидрогазодинамических процессов в каналах тепловых двигателей и энергетических установок. Рассмотрено решение задач расчета движения сжимаемой и несжимаемой жидкости в двумерной и трёхмерной постановке. Даны рекомендации по решению задач конвективного теплообмена.

Предназначено для студентов вузов, обучающихся по направлению подготовки 13.04.03 – Энергетическое машиностроение.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Табл. 3. Ил. 57. Библиогр.: 3 назв.

УДК 536.2  
ББК 31.16

ISBN 978-5-9984-1708-5

© ВлГУ, 2023  
© Абляев А. Ю.,  
Люхтер А. Б., 2023

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	5
1. СКАЧИВАНИЕ И УСТАНОВКА <i>OPENFOAM</i> .....	7
1.1. Скачивание .....	7
1.2. Установка .....	7
1.3. Первый запуск <i>OpenFOAM</i> .....	24
Задание .....	25
Вопросы для самоконтроля .....	26
2. ТЕЧЕНИЕ В ПОЛОСТИ .....	27
2.1. Постановка задачи .....	27
2.2. Создание проекта .....	28
2.3. Построение расчетной сетки .....	29
2.4. Условия однозначности .....	33
2.5. Управление процессом расчёта .....	36
2.6. Параметры численной схемы .....	37
2.7. Параметры модуля решения систем линейных алгебраических уравнений .....	40
2.8. Генерация расчётной сетки .....	42
2.9. Запуск процесса расчёта .....	43
2.10. Обработка результатов .....	43
Задание .....	46
Вопросы для самоконтроля .....	47
3. ТЕЧЕНИЕ В ТРУБЕ .....	48
3.1. Постановка задачи .....	48

3.2. Расчетная сетка .....	49
3.3. Условия однозначности .....	64
3.4. Свойства жидкости.....	65
3.5. Турбулентность.....	66
3.6. Управление процессом расчёта.....	66
3.7. Параметры численной схемы .....	67
3.8. Параметры модуля решения.....	69
3.9. Генерация расчетной сетки .....	71
3.10. Расчёт .....	71
3.11. Обработка результатов.....	71
Задание.....	75
Вопросы для самоконтроля .....	75
4. ТЕПЛООБМЕН ПРИ ТУРБУЛЕНТНОМ ТЕЧЕНИИ ГАЗА .....	77
4.1. Постановка задачи .....	78
4.2. Расчётная сетка .....	80
4.3. Условия однозначности .....	89
4.4. Параметры численной схемы .....	95
4.5. Параметры модуля решения.....	95
4.6. Обработка результатов.....	96
Задание.....	97
Вопросы для самоконтроля .....	98
ЗАКЛЮЧЕНИЕ.....	99
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	100
ПРИЛОЖЕНИЕ .....	101

## ВВЕДЕНИЕ

Создание и совершенствование современных высокоэффективных энергетических установок, в частности двигателей внутреннего сгорания, невозможно без всестороннего изучения гидро- и газодинамических процессов, протекающих в их каналах и проточных частях. Определяющее влияние на экономические, мощностные и эффективные показатели силовых агрегатов оказывают процессы в системах топливоподачи, смазки, охлаждения, впуска (во впускном трубопроводе, впускных каналах), выпуска (в выпускных каналах и выпускном трубопроводе). Достаточное внимание при форсировании двигателей уделяется также организации движения газов в картере, вызванного перемещением поршней, шатунов и вращением коленчатого вала. Во всех этих процессах участвуют как несжимаемые (топлива, смазочные материалы, охлаждающие жидкости), так и сжимаемые (воздух, топливовоздушная смесь, отработавшие газы) жидкости. Кроме того, движение жидкостей и газов в двигателях внутреннего сгорания сопровождается интенсивным теплообменом, как, например в системах воздушного и жидкостного охлаждения.

Наряду с экспериментальными широкое распространение, особенно при проектировании новых силовых агрегатов, получили методы расчётного исследования гидрогазодинамических процессов в системах двигателей внутреннего сгорания. Наиболее распространённый на сегодня метод математического моделирования стационарных и нестационарных потоков жидкости и газа – метод конечных объёмов, в полной мере реализованный в программном комплексе *OpenFOAM*.

*OpenFOAM* в отличие от большинства других программных комплексов вычислительной гидрогазодинамики является свободно распространяемым программным продуктом, охватывающим широкую область применения – от стационарных ламинарных течений несжимаемой жидкости до турбулентных потоков газов, многофазных течений, течений со свободной поверхностью, конвективного и сопряженного теплообмена, химически реагирующих течений и т. д.

Существенное преимущество *OpenFOAM* – его открытая архитектура, что позволяет создавать решатели для весьма специфичных задач, стоящих перед конкретным пользователем.

В учебном пособии приводятся рекомендации по установке и настройке программного комплекса *OpenFOAM*, описываются базовые принципы решения задач механики жидкости и газа, приводятся рекомендации по построению расчётных сеток, учитывающих особенности течений в каналах теплоэнергетических установок, примеры анализа результатов расчётов.

Материал, изложенный в учебном пособии, рекомендован для изучения дисциплины «Численные методы механики жидкости и газа», а также может быть использован при курсовом проектировании и выполнении ВКР магистра.

# 1. СКАЧИВАНИЕ И УСТАНОВКА *OPENFOAM*

## 1.1. Скачивание

**Особенности лицензии.** *OpenFOAM* является свободным программным обеспечением (ПО) с открытым исходным текстом распространяемым под *GNU General Public License v3*<sup>1</sup>. В соответствии с ней<sup>2</sup>:

На программное обеспечение включающее в свой состав полностью или частично ПО под действием *GPL* также распространяется действие *GPL*.

Для ПО, на которое распространяется действие *GPL*, поставляемое в виде двоичных (исполняемых) файлов, в обязательном порядке должен быть полностью открыт исходный код.

**Скачивание установочного файла.** Путь для скачивания *OpenFOAM* для *Windows*: <https://www.cfdsupport.com/openfoam-for-windows.html>

Для скачивания необходимо ввести адрес своей электронной почты и нажать кнопку [*Download*].

Владельцы сайта предупреждают, что не занимаются рассылкой спама. Отдельно прописана возможность отписаться от рассылки.

Скаченный файл имеет имя вида *OF4Win-20.09v2-win64.exe*, где *20.09* – номер версии *OpenFOAM*, *win64* – обозначение операционной системы и её разрядность. С точки зрения освоения базовых навыков работы с программным обеспечением различия между версиями отсутствуют. Они становятся существенными только при задании некоторых специфических граничных условий.

## 1.2. Установка

**Запуск процесса установки.** Установка и работа *OpenFOAM* для *Windows* возможна только в шестидесятичетырехразрядной (x64) среде. Под управлением тридцатидвухразрядной (x86) операционной системы работа *OpenFOAM* невозможна.

Для установки *OpenFOAM* необходимо запустить ранее скаченный файл (рис. 1.1)

---

<sup>1</sup> <https://www.gnu.org/licenses/gpl.html>

<sup>2</sup> <https://openfoam.org/license>

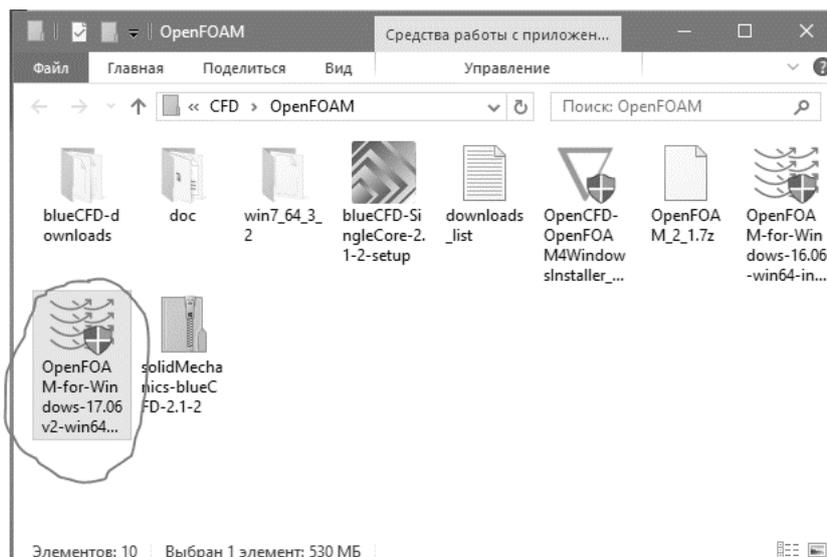


Рис. 1.1. Установочный файл *OpenFOAM*

После запуска, появится диалоговое окно (рис. 1.2).



Рис. 1.2. Запуск установки *OpenFOAM*

Для продолжения установки необходимо нажать кнопку [*Next>*].

**Предварительная настройка системы.** Для корректной работы OpenFOAM необходимо, чтобы операционная система (ОС) могла различать строчные и прописные буквы в именах файлов<sup>3</sup>. Первым этапом

<sup>3</sup> по умолчанию файловая система *NTFS*, используемая в ОС *Windows*, не различает регистр символов в имени файла. Так, следующие имена файлов считаются идентичными: *document.doc*, *DOCUMENT.doc* и *DoCuMeNt.DoC*.

установки (рис. 1.3) оказывается настройка параметров *Windows* таким образом, чтобы имена файлов, записанные в разных регистрах, различались.

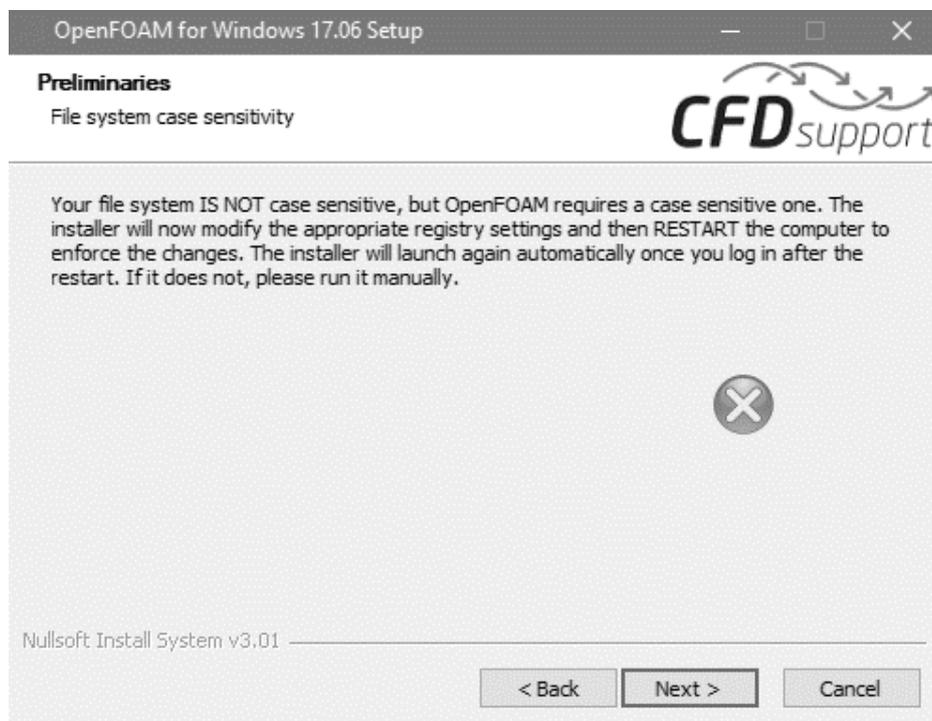


Рис. 1.3. Настройка регистрозависимых имен файлов

После нажатия на кнопку [*Next*>] появится предупреждение о перезагрузке (рис. 1.4).

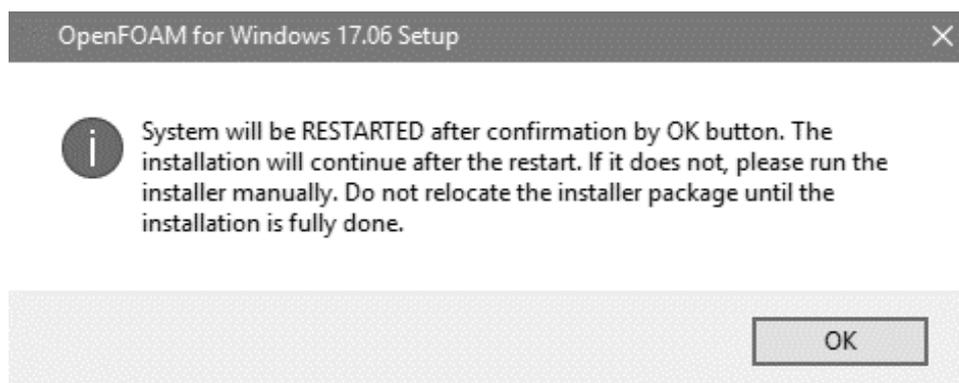


Рис. 1.4. Предупреждение о перезагрузке

Для перезагрузки нужно нажать кнопку [*OK*]. Компьютер перезагрузится, после чего автоматически продолжится процесс установки (рис. 1.5).

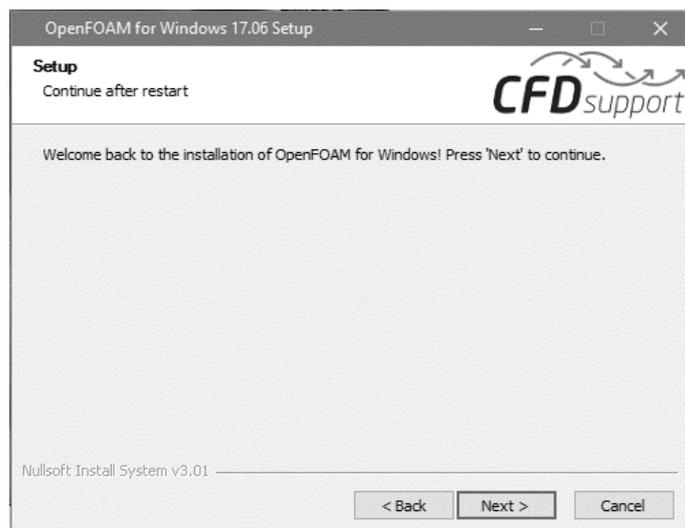


Рис. 1.5. Продолжение процесса установки после перезагрузки

**NB.** Если после перезагрузки компьютера процесс установки не возобновился, т.е. диалоговое окно с рис. 1.5 не появилось, необходимо снова запустить скачанный файл (рис. 1.1), установка продолжится с места перезагрузки.

Для продолжения установки необходимо нажать кнопку [*Next>*]. В появившемся диалоговом окне (рис. 1.6) будет предложено продолжить установку *OpenFOAM*, т. к. система отвечает всем требованиям к его работе.

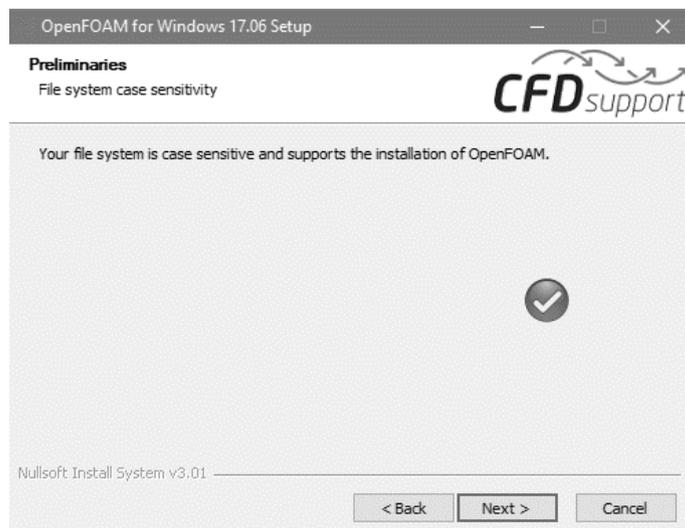


Рис. 1.6. Продолжение установки

Для продолжения установки необходимо нажать кнопку [*Next>*]. **Выбор устанавливаемого ПО.** В появившемся диалоговом окне (рис. 1.7) можно настроить установку *OpenFOAM*.

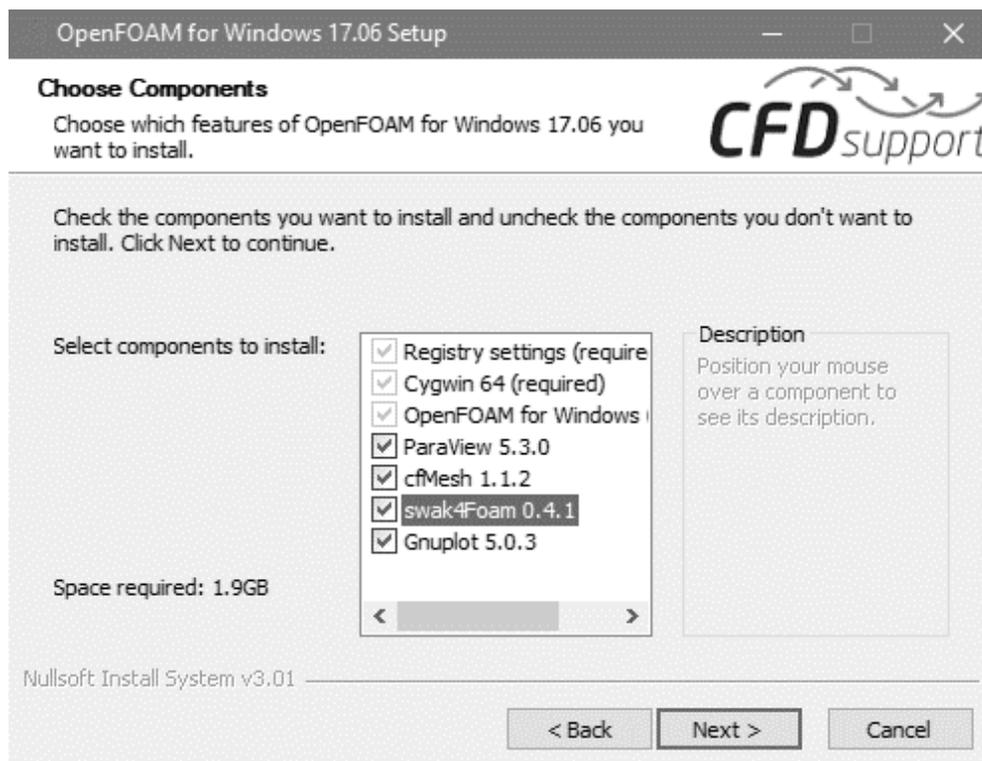


Рис. 1.7. Выбор компонентов для установки

Три компонента являются обязательными (*required*): *Registry settings* – значения, записываемые в реестр *Windows*, *Cygwin64* – среда исполнения и набор утилит командной строки, и *OpenFOAM for Windows* – собственно пакет прикладных программ математического моделирования *OpenFOAM*. Четыре оставшихся компонента можно не устанавливать. Предлагаемые на выбор компоненты включают в себя:

*ParaView* – графический постпроцессор, служит для наглядного представления результатов расчетов.

*cfMesh* – альтернативный стандартному препроцессору, служащий для построения расчётных сеток, с которыми стандартный препроцессор (*blockMesh*) может не справиться.

*swak4Foam* – набор утилит, значительно расширяющих и упрощающих практическое применение *OpenFOAM*.

*Gnuplot* – программа построения и оформления графиков, отличающаяся большой гибкостью в использовании.

Для комфортной работы необходимо установить *ParaView*. Такие компоненты, как *cfMesh* и *swak4Foam* позволят расширить возможности *OpenFOAM* и их также рекомендуется установить на локальный компьютер. Установку *Gnuplot* следует производить только в том случае, если на компьютере еще не установлена эта утилита.

Для продолжения установки необходимо нажать кнопку [*Next>*].

**Выбор места установки.** Следующим действием станет выбор папки, в которую будет установлен *OpenFOAM*. В появившемся диалоговом окне (рис. 1.8) необходимо ввести в строку редактирования или выбрать с помощью диалога, вызываемого кнопкой [*Browse...*], другую папку для установки. Если свободного места на диске *c:* достаточно – лучше оставить путь по умолчанию.

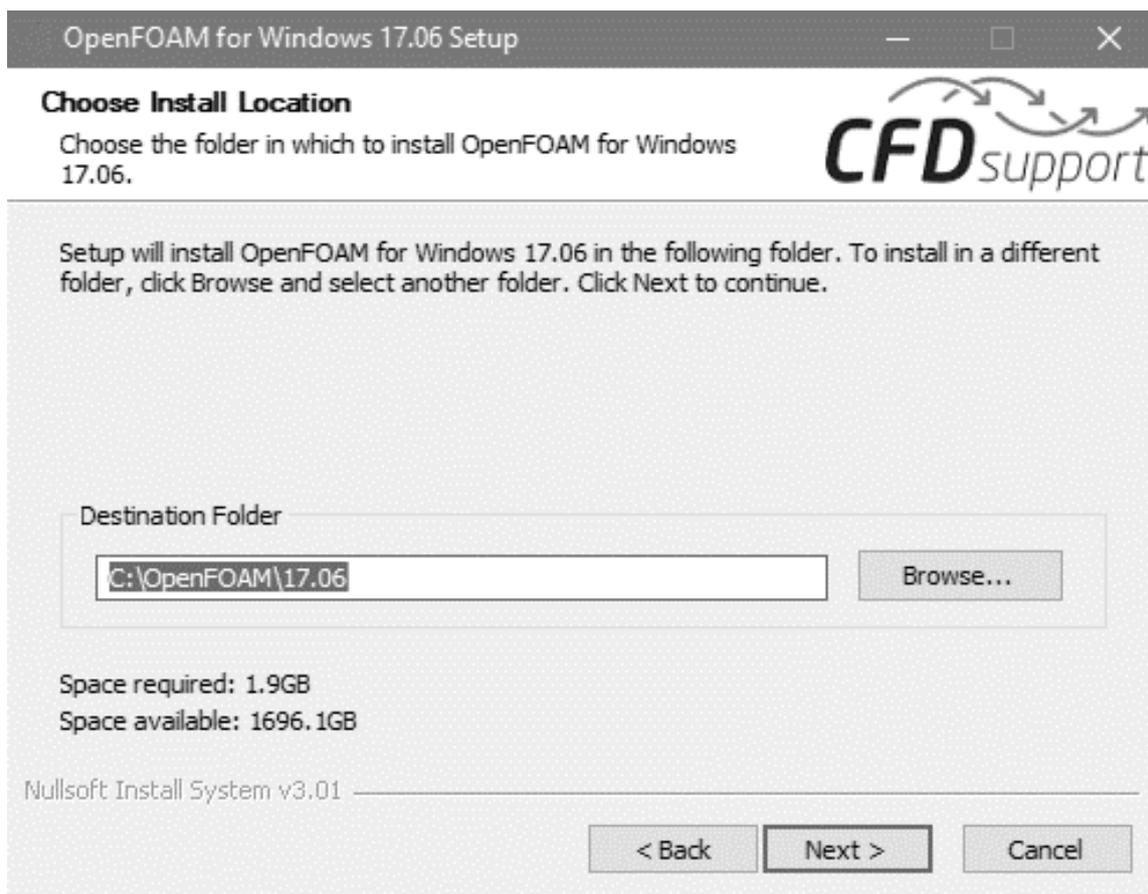


Рис. 1.8. Выбор папки для установки

Для продолжения установки необходимо нажать кнопку [*Next>*].

**Установка *MPI*.** Для распараллеливания вычислений *OpenFOAM* использует технологию *MPI – Message Passing Interface*. В процессе установки, в диалоговом окне (рис. 1.9) будет предложено установить библиотеку *MPI*. Если она уже установлена, то необходимо снять соответствующий флажок (в этом случае можно сразу перейти к следующему подразделу).



Рис. 1.9. Выбор установка *MPI*

Для продолжения установки необходимо нажать кнопку [*Install*], после чего начнется процесс установки *Microsoft MPI* (рис. 1.10).



Рис.1.10. Начало установки *MPI*

Для продолжения установки необходимо нажать кнопку [*Next*].

В появившемся диалоговом окне (рис. 1.11) необходимо ознакомиться с лицензионным соглашением и установить флажок «*I accept the terms in the License Agreement*» в случае согласия. В случае несогласия

с лицензионным соглашением процесс установки *MPI* завершится с ошибкой и соответствующим предупреждением. В результате *OpenFOAM* может начать работать неправильно или полностью потерять свою работоспособность.

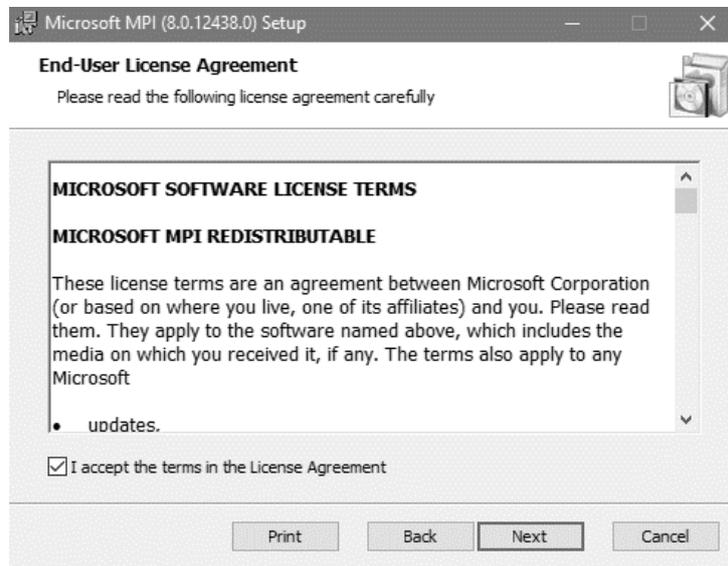


Рис. 1.11. Лицензионное соглашение *MPI*

Для продолжения установки необходимо во всех появляющихся диалоговых окнах, связанных с установкой *MPI*, нажимать кнопку [Next]. Вводить какую-либо дополнительную информацию не требуется.

В случае успешного завершения установки *MPI* появится диалоговое окно рис. 1.12

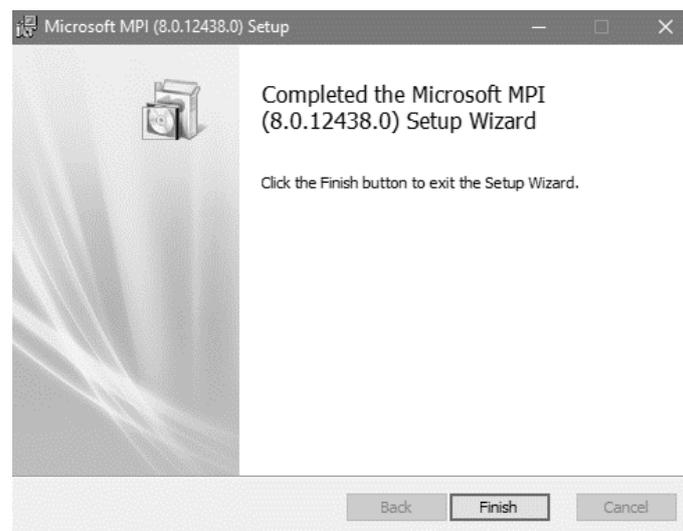


Рис. 1.12. Успешное завершение установки *MPI*

Для продолжения установки *OpenFOAM* необходимо нажать кнопку [*Finish*].

**Установка *Cygwin*.** О начале установки *Cygwin* свидетельствует диалоговое окно рис. 1.13.

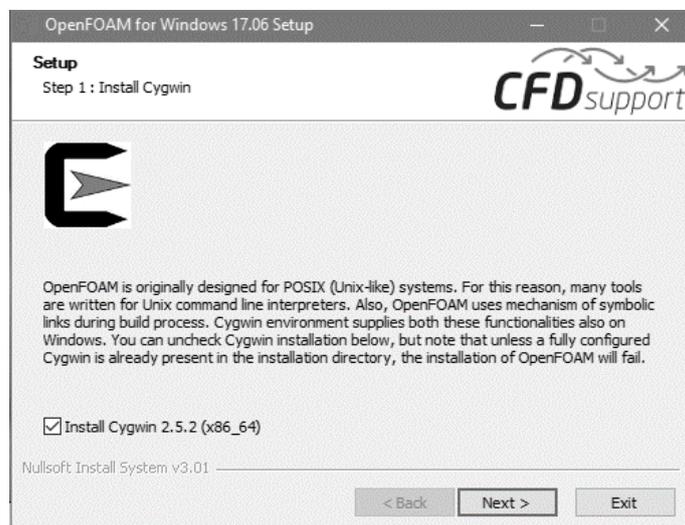


Рис. 1.13. Установка *Cygwin*

***NB.*** *Cygwin* является необходимым компонентом для работы *OpenFOAM* в среде *Windows*.

Для продолжения установки необходимо, ничего не меняя, нажать кнопку [*Next*>].

**Установка *OpenFOAM*.** О начале установки свидетельствует диалоговое окно рис. 1.14.

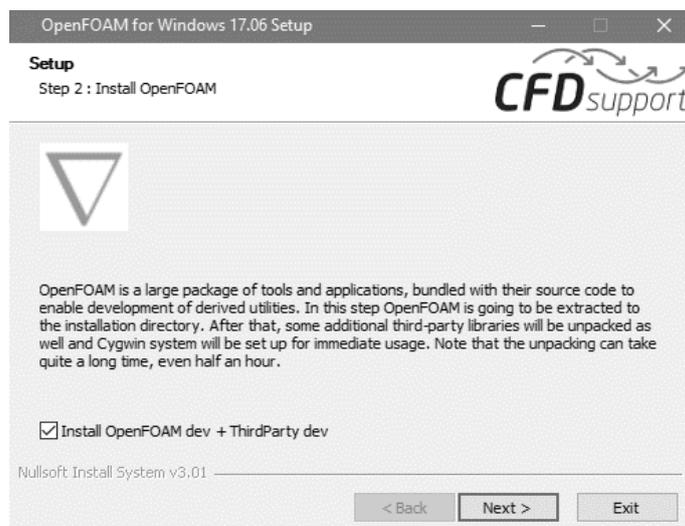


Рис. 1.14. Установка *OpenFOAM*

Для продолжения установки необходимо, ничего не меняя, нажать кнопку [*Next*>].

**Установка *ParaView*.** О начале установки свидетельствует диалоговое окно рис. 1.15.

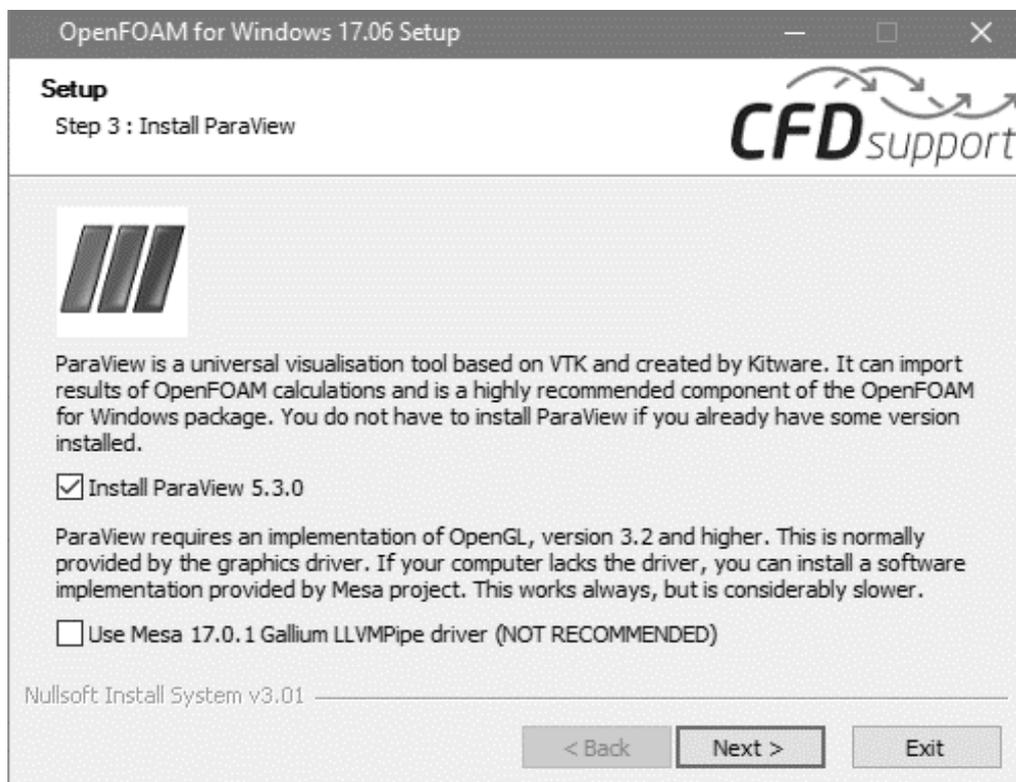


Рис. 1.15. Установка *paraView*

В случае, если компьютер не оснащен видеокартой, поддерживающей *OpenGL* версии 3.2 и выше, или драйвера на видеокарту отсутствуют, необходимо<sup>4</sup> установить флаг «*Use Mesa 17.0.1 Gallium LLVMPipe driver*». При этом скорость работы *paraView* может несколько уменьшиться. Как правило, для современных компьютеров, даже оснащенных встроенной видеокартой, установка этого флажка не требуется.

Для продолжения установки необходимо нажать кнопку [*Next*>].

**Установка *Swak4Foam*.** О начале установки свидетельствует диалоговое окно рис. 1.16.

---

<sup>4</sup> Как показывает практика, для нормальной работы *ParaView* на ноутбуках и планшетах, особенно при питании от аккумулятора, лучше включать эту опцию при первоначальной установке.

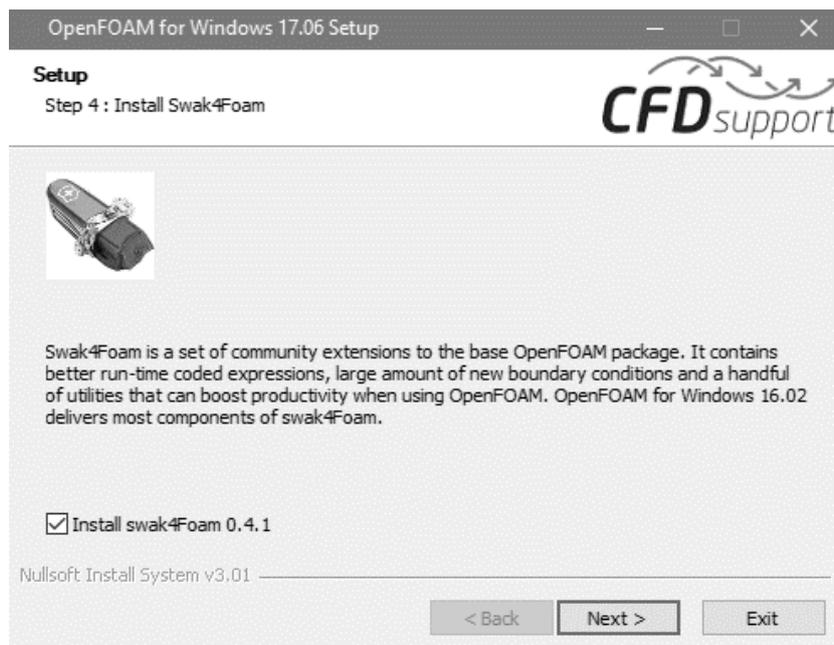


Рис. 1.16. Установка *Swak4Foam*

Для продолжения установки необходимо нажать кнопку [*Next>*].  
**Установка *cfMesh*.** О начале установки свидетельствует диалоговое окно рис. 1.17.

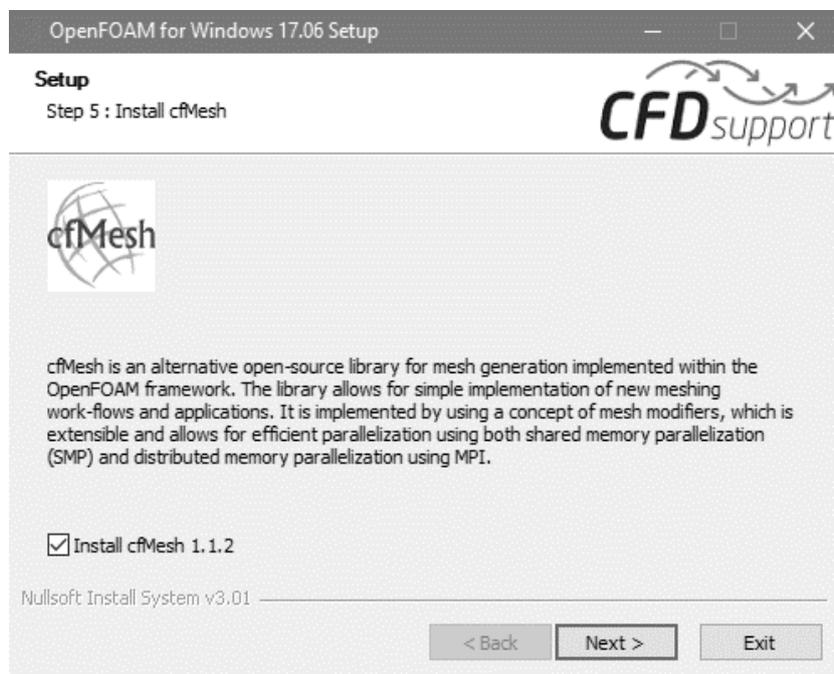


Рис. 1.17. Установка *cfMesh*

Для продолжения установки необходимо нажать кнопку [*Next>*].

**Установка Gnuplot.** О начале установки свидетельствует диалоговое окно рис. 1.18.

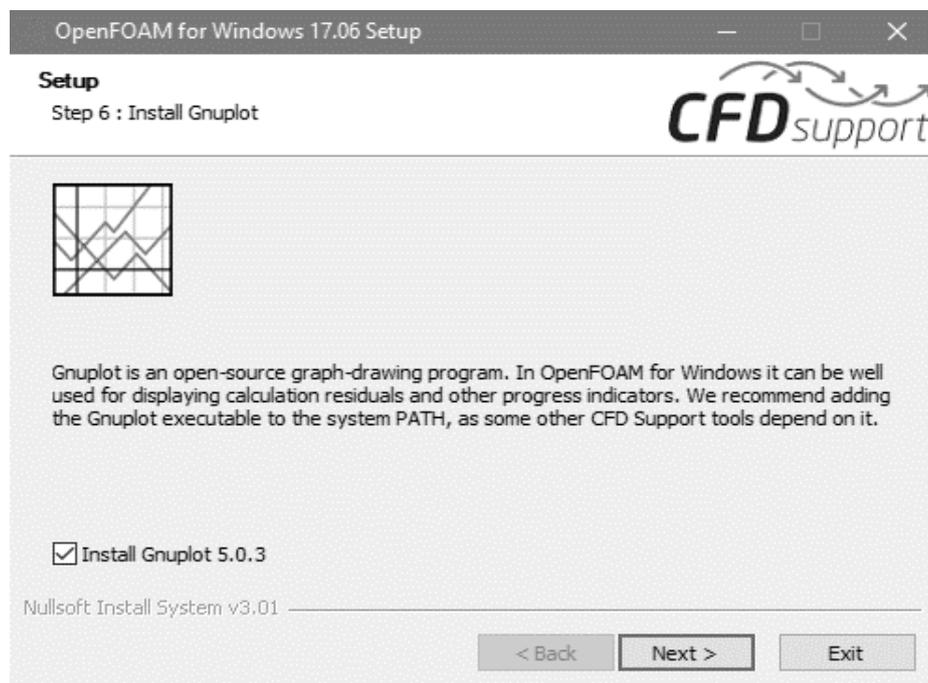


Рис. 1.18. Установка *Gnuplot*

Для продолжения установки необходимо нажать кнопку [*Next>*].

В появившемся диалоговом окне (рис. 1.19) следует выбрать **английский** (*English*) язык и продолжить установку нажатием кнопки [*OK*].

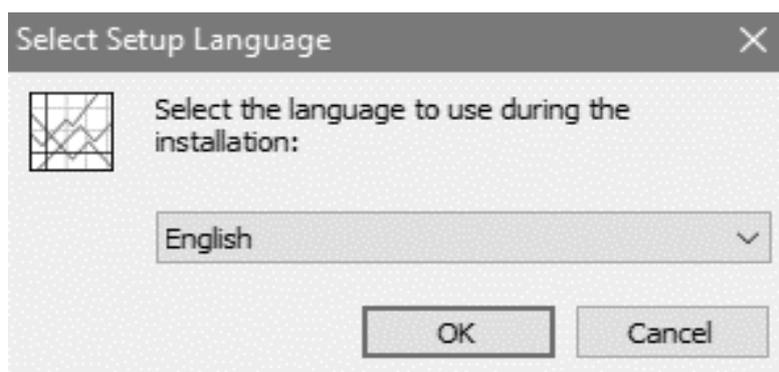


Рис. 1.19. Выбор языка утилиты установки *Gnuplot*

Для продолжения установки (рис. 1.20) необходимо нажать кнопку [*Next>*].

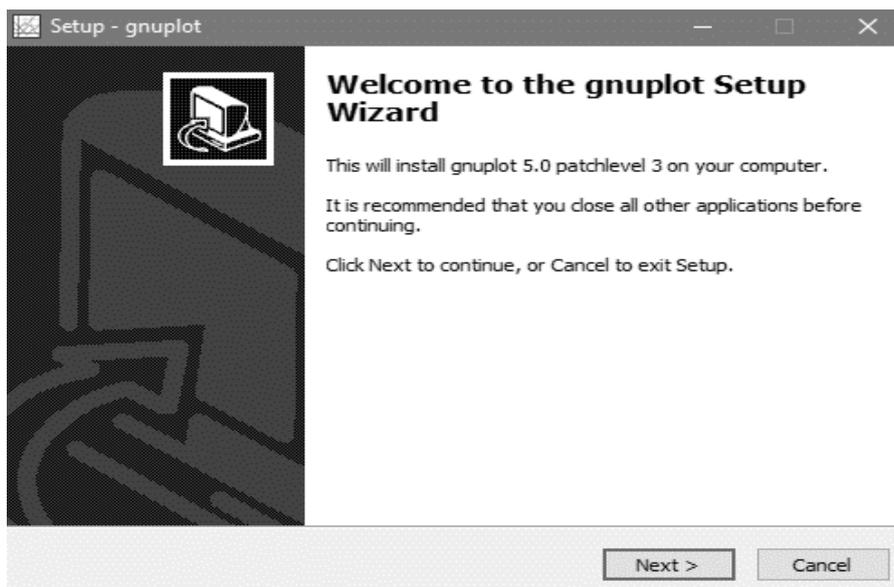


Рис. 1.20. Начало установки *Gnuplot*

В появившемся окне (рис. 1.21) следует ознакомиться и согласиться («*I accept the agreement*»), с лицензионным соглашением.

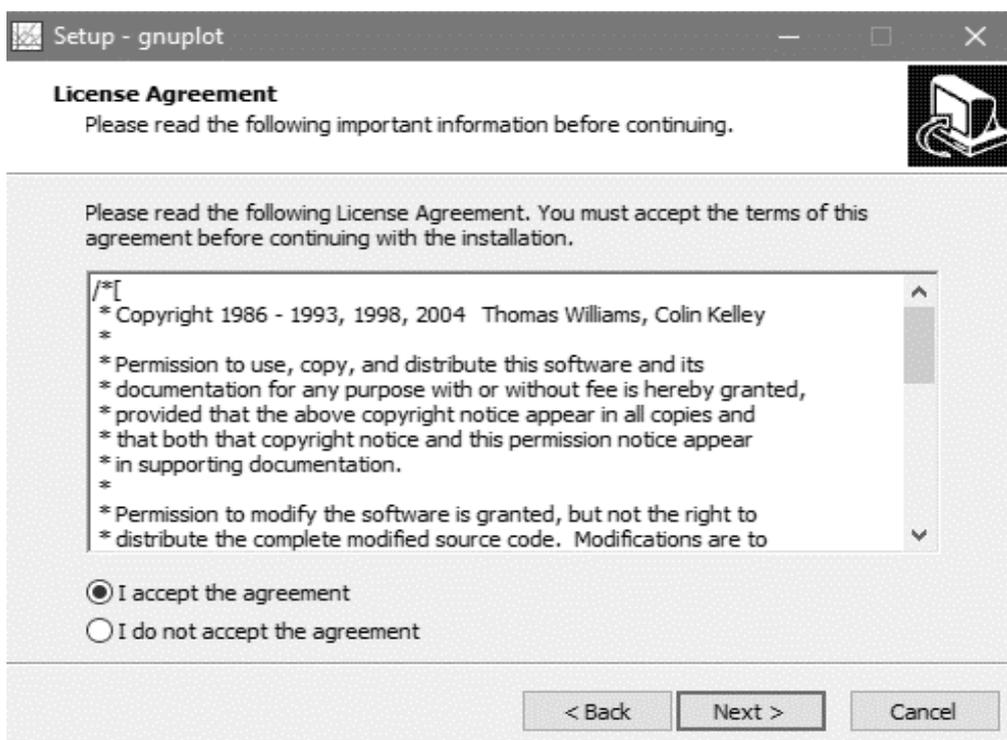


Рис. 1.21. Лицензионное соглашение *Gnuplot*

Для продолжения установки необходимо нажать кнопку [*Next>*].

В появившемся диалоговом окне (рис. 1.22) следует ознакомиться с информацией о *gnuplot*.

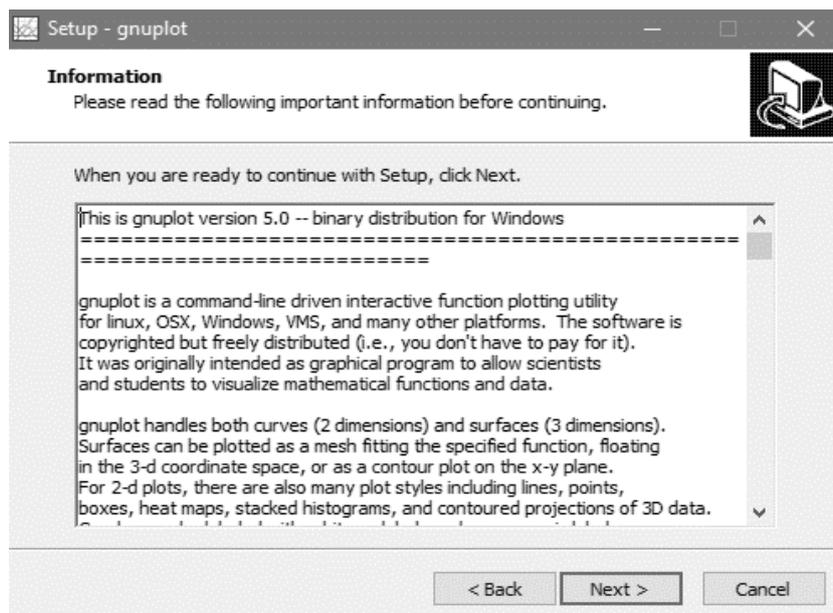


Рис. 1.22. Информация о *Gnuplot*

Для продолжения установки необходимо нажать кнопку [Next>]. В появившемся диалоговом окне (рис. 1.23) необходимо выбрать место, куда будет установлен *Gnuplot*.

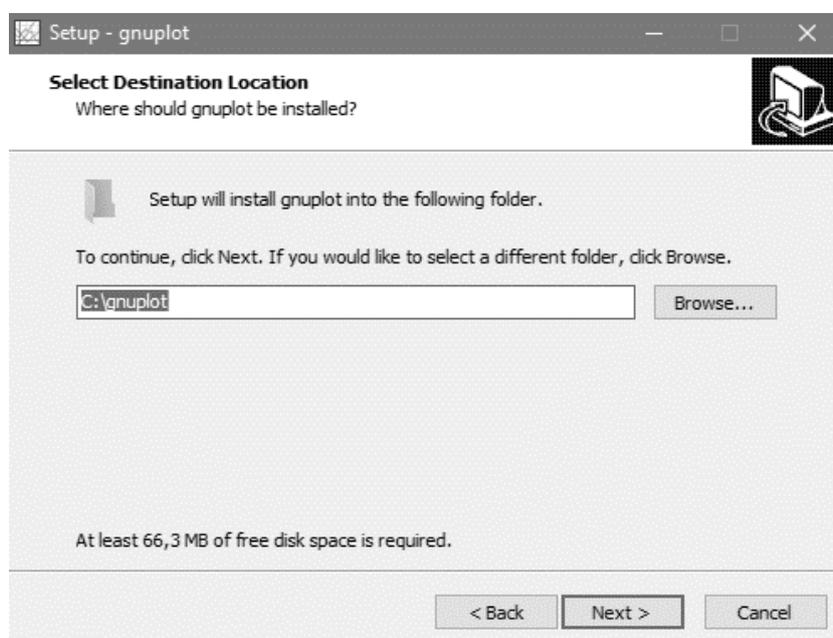


Рис. 1.23. Выбор папки для установки *Gnuplot*

**NB.** Чтобы *Gnuplot* было удобнее использовать, путь по умолчанию (*c:\Program Files (x86)\gnuplot*) лучше заменить на *c:\gnuplot*.

Для продолжения установки необходимо нажать кнопку [Next>].

В появившемся диалоговом окне (рис. 1.24) нужно снять флажки у всех необязательных компонентов. В результате будут установлены только «*gnuplot Core Components*».

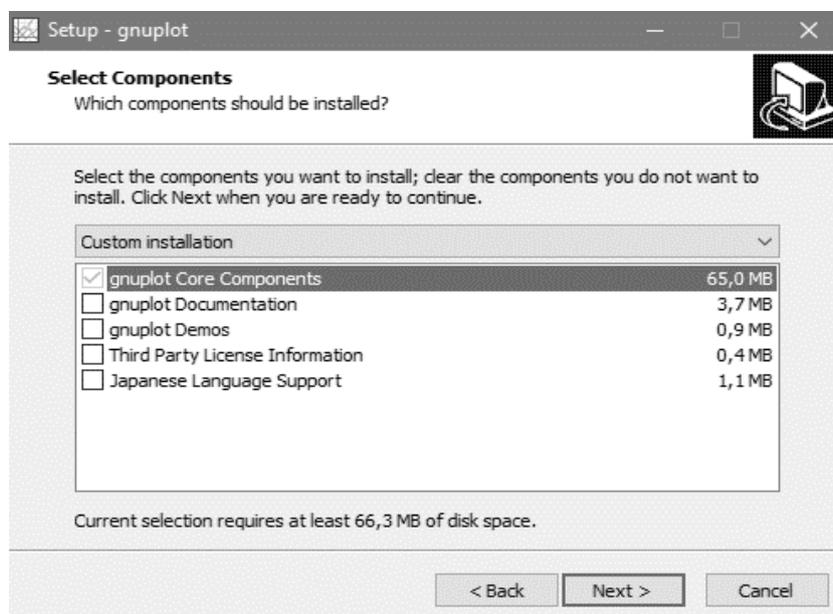


Рис. 1.24. Выбор компонентов *Gnuplot*

Для продолжения установки необходимо нажать кнопку [*Next>*].

*Gnuplot* редко используется в интерактивном режиме, т. к. является утилитой командной строки и выполняется, в основном, в пакетном режиме. Поэтому, в появившемся диалоговом окне (рис. 1.25), необходимо установить флажок «*Don't create a Start Menu folder*».

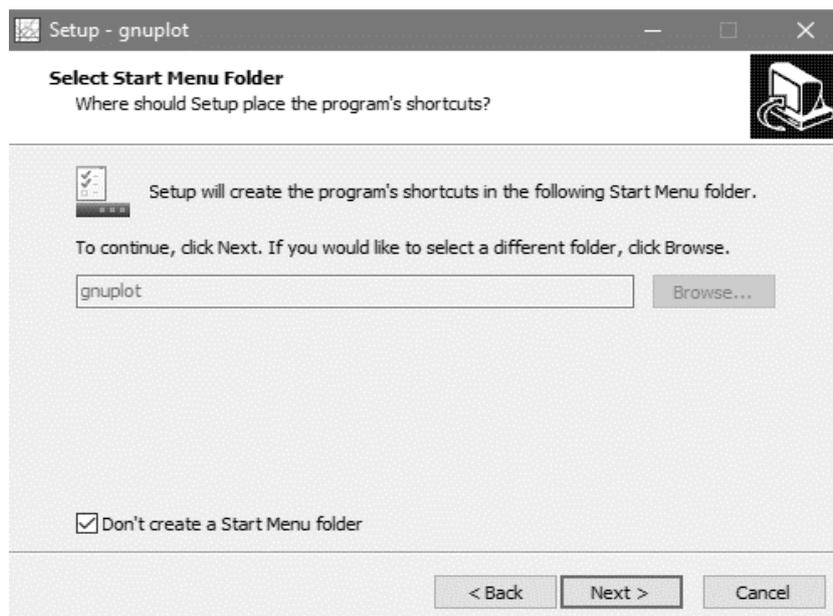


Рис. 1.25. Отказ от создания пункта *Gnuplot* в меню Пуск (*Start*)

Для продолжения установки необходимо нажать кнопку [Next>].

В следующем диалоговом окне необходимо настроить терминал по умолчанию (*windows*, рис. 1.26) и ассоциацию приложения *gnuplot.exe* с файлами имеющими расширения «.plt», «.gp» и «.gpl» (рис. 1.27).

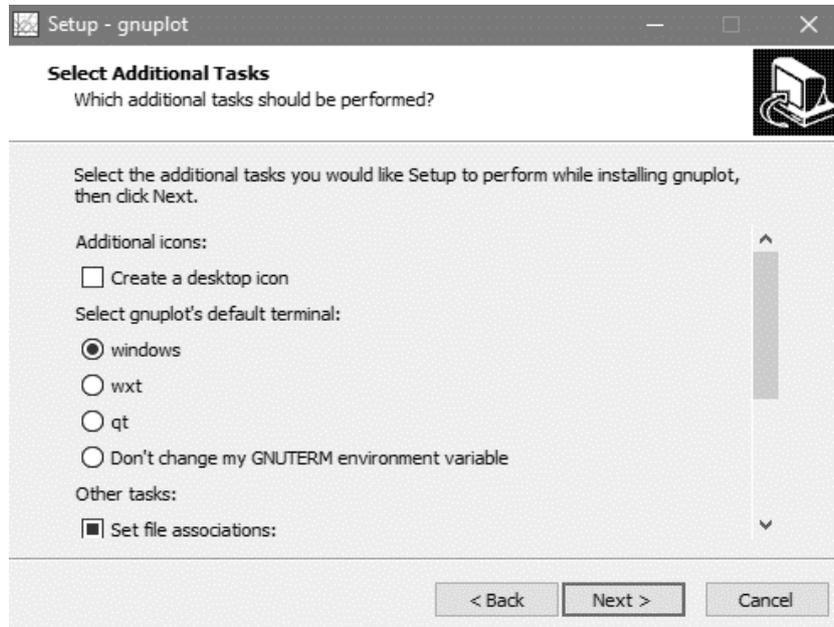


Рис. 1.26. Настройка терминала *Gnuplot*

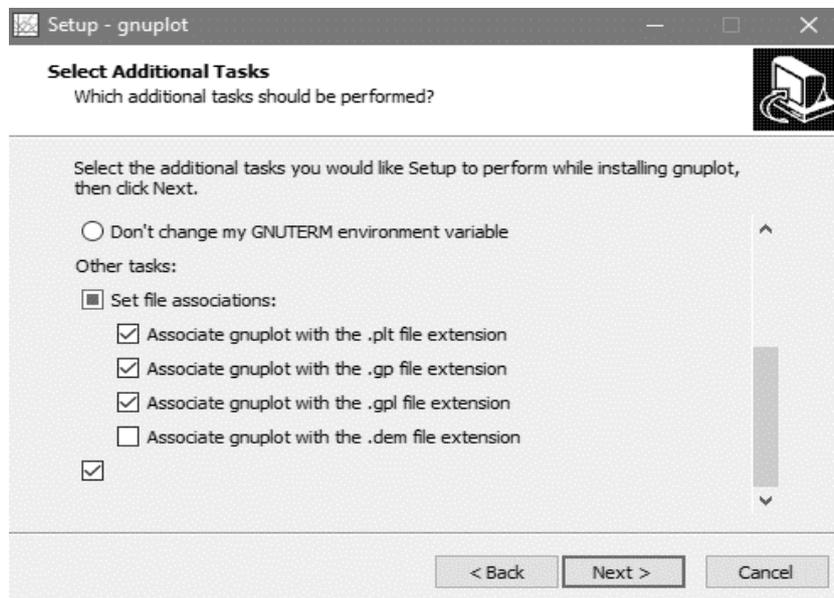


Рис. 1.27. Настройка файловых ассоциаций *Gnuplot*

Для продолжения установки необходимо нажать кнопку [Next>].

В появившемся диалоговом окне (рис. 1.28) следует ознакомиться с информацией о *Gnuplot*.

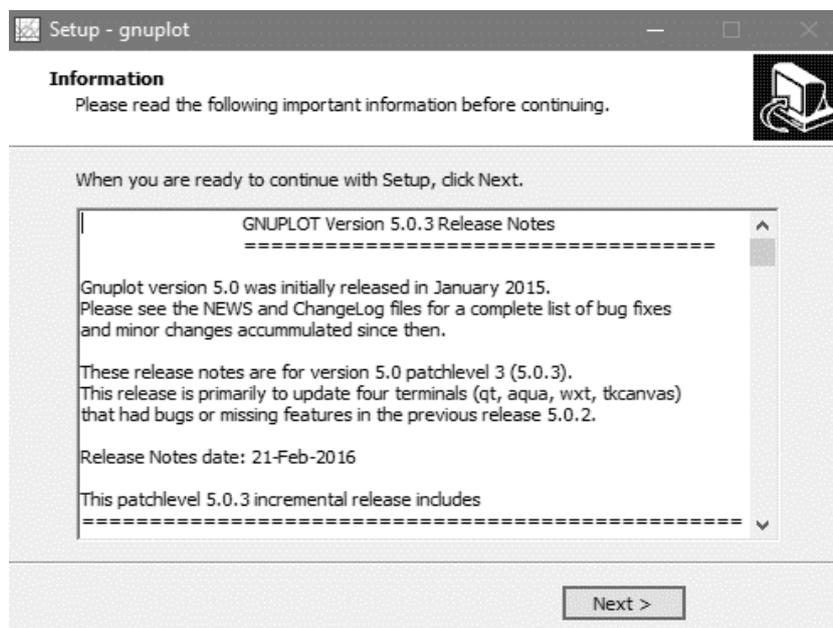


Рис. 1.28. Информация и *Gnuplot*

Для продолжения установки необходимо нажать кнопку [*Next*>]. В диалоговом окне (рис. 1.29) сообщаем об успешной установке *gnuplot* следует снять все флажки и нажать кнопку [*Finish*].

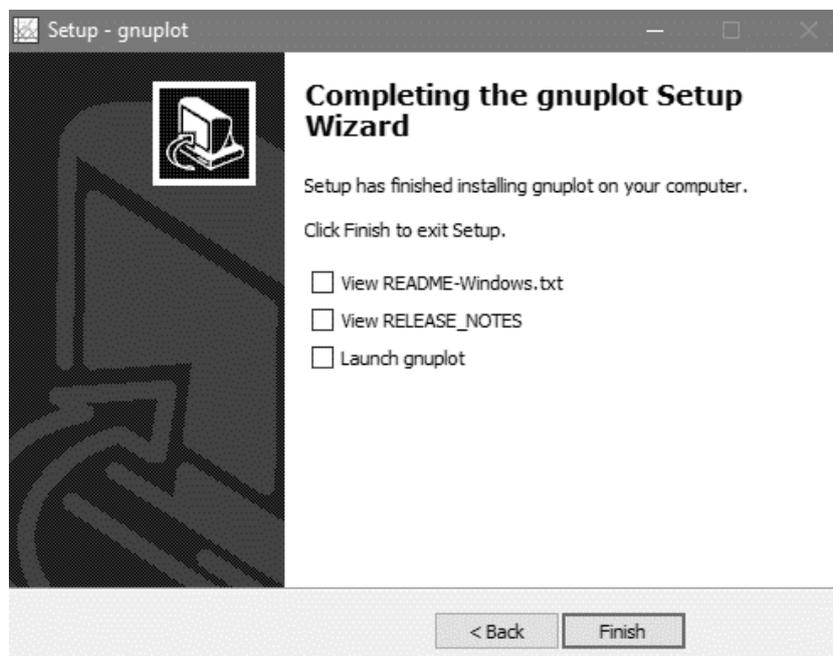


Рис. 1.29. Завершение установки *Gnuplot*

**Завершение установки *OpenFOAM*.** Диалоговое окно рис. 1.30 свидетельствует об успешном завершении установки *OpenFOAM*.

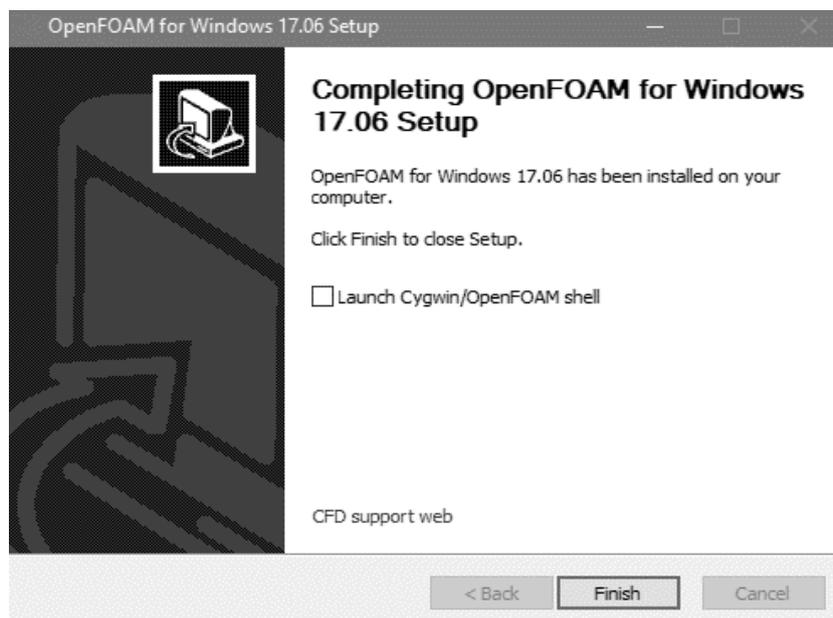


Рис. 1.30. Завершение установки *OpenFOAM*

Для выхода из программы установки необходимо снять флажок «*Launch Cygwin/OpenFOAM shell*» и нажать кнопку [*Finish*].

### 1.3. Первый запуск *OpenFOAM*

Запустить *OpenFOAM* можно с помощью ярлыка (рис. 1.31), расположенного на рабочем столе.



Рис. 1.31. Ярлык для запуска *OpenFOAM* на рабочем столе

В появившемся окне необходимо набрать следующие команды в ответ на приглашение \$:

```
$ mc
$ run
$ cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity/cavity ./
$ cd cavity
$ blockMesh
$ icoFoam
$ paraFoam
```

В результате в папку `[run]` будет скопирован один из примеров, входящих в состав *OpenFOAM*, выполнен препроцессор, запущен расчет, и, по окончании расчета, будет запущен постпроцессор для просмотра результатов. После настройки изображения окно постпроцессора может выглядеть как на рис. 1.32.

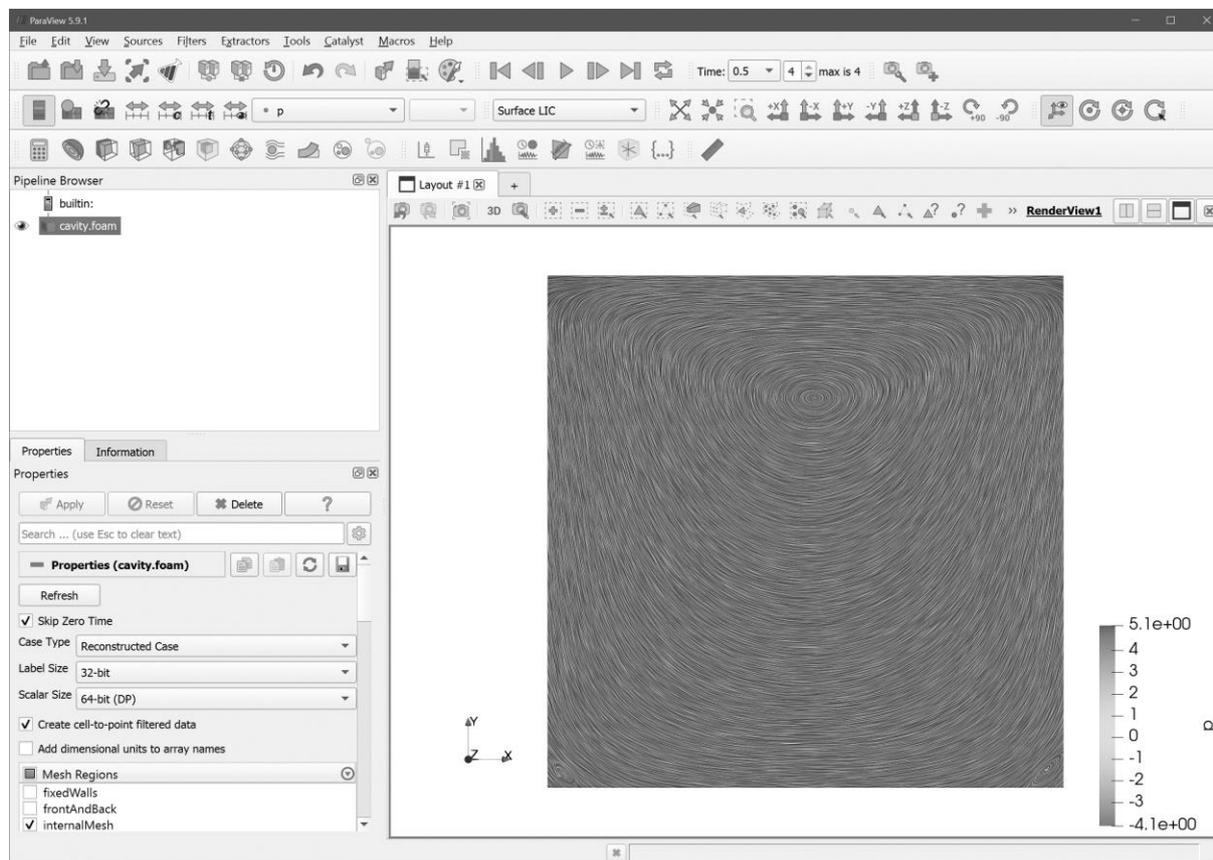


Рис. 1.32. Результат первого запуска *OpenFOAM*

Выход из файлового менеджера `mc` осуществляется по нажатию клавиши `<F10>` или последовательном нажатии клавиш `<Esc>` и `<0>` на клавиатуре. Выход из окна терминала необходимо осуществлять нажатием комбинации клавиш `<Ctrl> + <D>` (одновременно).

### Задание

1. Скачайте последнюю версию *OpenFOAM*.
2. Установите *OpenFOAM*. При выборе компонентов и настройке *OpenFOAM* руководствуйтесь уже имеющимся на компьютере программным обеспечением и конфигурацией оборудования.

3. Запустите среду для выполнения *OpenFOAM*.
4. Скопируйте в папку [*run*] проект *cavity* и, для проверки правильности установки, сформируйте расчетную сетку и выполните расчет.
5. Запустите *paraFoam* и убедитесь в корректном завершении расчета.

### Вопросы для самоконтроля

1. В чем заключается особенность лицензии *OpenFOAM*?
2. Перечислите основные компоненты *OpenFOAM*, обязательные для установки.
3. Перечислите необязательные для установки компоненты *OpenFOAM*.
4. Как называется постпроцессор, используемый в *OpenFOAM* для отображения результатов расчетов?
5. Какую настройку необходимо выполнить при установке *ParaView* для работы на устаревших или маломощных видеокартах?
6. Для чего используется надстройка *Swak4Foam*?
7. Для чего используется *cfMesh*?
8. В каких случаях можно не устанавливать *Gnuplot*?
9. Какая библиотека должна быть установлена для запуска *OpenFOAM* в параллельном режиме?
10. Как проверить правильность установки *OpenFOAM*?

## 2. ТЕЧЕНИЕ В ПОЛОСТИ

### 2.1. Постановка задачи

Одной из достаточно простых по формулировке задач гидрогазодинамики с наглядным решением служит течение в полости квадратного сечения, расположенной на границе невозмущенного потока (рис. 2.1).

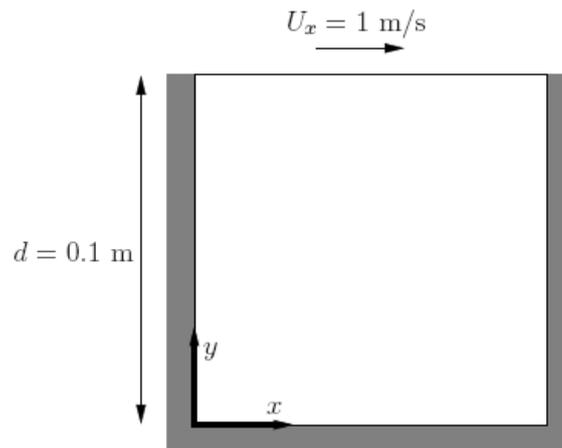


Рис. 2.1. Расчетная схема для решения задачи течения в полости

Анализ расчетной схемы показывает, что полость, квадратного сечения, со стороной, длина которой равна  $d = d_x = d_y = 10 \text{ см} = 0,1 \text{ м}$ , ограничена с трех сторон (слева, снизу и справа) неподвижными твердыми стенками, а с четвертой стороны граничит с потоком жидкости, движущимся в направлении оси  $x$  со скоростью  $U_x = 1 \text{ м/с}$ . В направлении оси  $z$  полость бесконечна, поэтому рассматриваемую задачу можно свести к двумерному случаю (рис. 2.2).

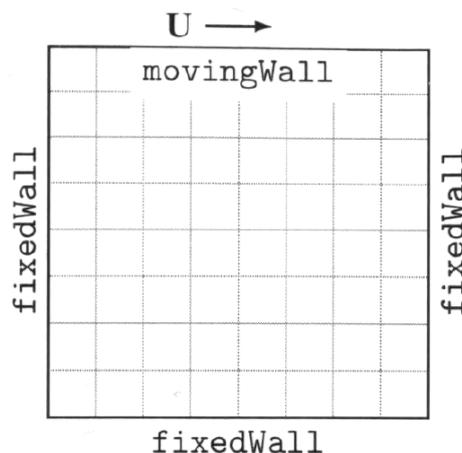


Рис. 2.2. Расчетная сетка с обозначением граничных условий

Для несжимаемой жидкости уравнение неразрывности записывается в виде

$$\nabla \vec{U} = 0$$

В качестве уравнений сохранения момента количества движения используются уравнения Навье-Стокса для вязкой несжимаемой жидкости

$$\frac{\partial U}{\partial t} + \nabla(\vec{u}\vec{u}) - \nabla(\nu\nabla\vec{u}) = -\nabla p^*,$$

где  $p^* = \frac{p}{\rho}$  – кинематическое давление,  $\text{м}^2/\text{с}^2$ ;  $U$  – модуль вектора скорости,  $\text{м}/\text{с}$ ;  $\vec{u}$  – составляющая вектора скорости в направлении оси координат (векторное произведение вектора скорости  $\vec{U}$  на орт  $\vec{i}$ ,  $\vec{j}$  или  $\vec{k}$ );  $\nu$  – кинематическая вязкость,  $\text{м}^2/\text{с}$ .

В процессе моделирования будем полагать, что исследуется течение вязкой несжимаемой жидкости при  $Re = 10$ . Т. к.  $Re = \frac{d \cdot U}{\nu}$ , а  $d = 0,1$  м и  $U = 1$  м/с, то<sup>5</sup>  $\nu = 0,01$   $\text{м}^2/\text{с}$ . Для примера, кинематическая вязкость воды при  $20^\circ\text{C}$  равняется  $\nu = 1,006 \cdot 10^{-6}$   $\text{м}^2/\text{с}$ .

В качестве граничных условий на неподвижных стенках используются условия прилипания, а на подвижной стенке – значение скорости основного потока в направлении оси  $x$ .

В качестве начальных условий заданы т. н. «однородные» поля скорости и давления во всей расчётной области.

## 2.2. Создание проекта

Исходные данные и результаты расчёта хранятся на диске в папке, имеющей специальную структуру подпапок и файлов (рис. 2.3), называемую проектом. По умолчанию проекты размещаются в папке [run]. Для того чтобы попасть в неё в командной строке нужно набрать **run** и нажать клавишу <Enter>.

---

<sup>5</sup> предполагается, что используется некоторая модельная жидкость с набором свойств, таких, чтобы соответствовать условиям задачи. К реальным жидкостям она не имеет никакого отношения.

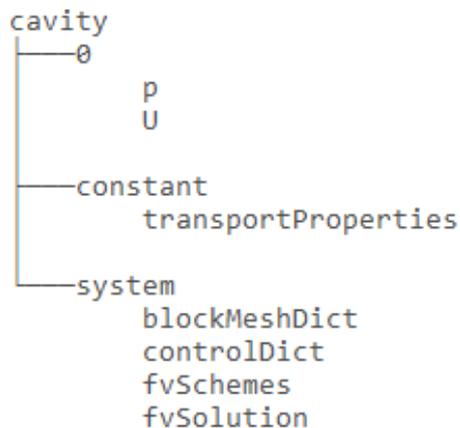


Рис. 2.3. Структура папок и файлов проекта

Для нашего проекта необходимо создать папку проекта `[run\cavity]` и папки `[run\cavity\0]`, `[run\cavity\constant]` и `[run\cavity\system]`, в которых будут размещаться файлы, описывающие геометрию расчетной области, граничные и начальные условия. Запуск препроцессора *blockMesh*, расчетного модуля *icoFoam* и постпроцессора *paraFoam* осуществляется из командной строки в папке проекта `[run\cavity]`.

### 2.3. Построение расчетной сетки

Расчетная сетка описывается с помощью т. н. блоков. Вид блока для построения расчетной сетки приведен на рис. 2.4.

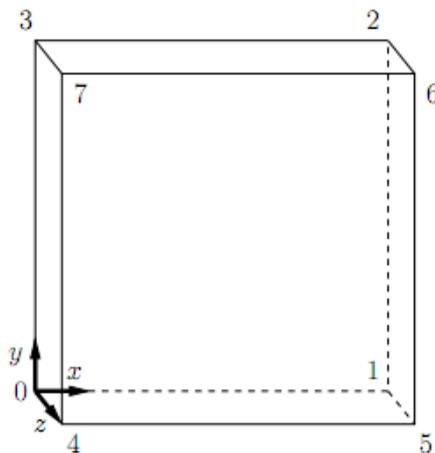


Рис. 2.4. Блок для построения расчетной сетки

Расчетная сетка<sup>6</sup> генерируется с помощью утилиты *blockMesh* на основании информации, содержащейся в файле

---

<sup>6</sup> т. н., блочная

[*run\cavity\system\blockMeshDict*]. Это обычный текстовый файл, имеющий специальную структуру, называемую словарём. Файл *blockMeshDict* содержит несколько разделов, порядок следования которых лучше сохранять в соответствии с приведённым ниже.

**FoamFile.** Этот раздел служит для описания файла данных и содержит следующую информацию: `version 2.0` – версия формата файла; `format ascii` – информация в текстовом формате; `class dictionary` – информация в файле представлена в виде словаря; `object blockMeshDict` – в файле находится описание области течения для построения блочной<sup>7</sup> расчетной сетки.

```
FoamFile
{
version 2.0;
format ascii;
class dictionary;
object blockMeshDict;
}
```

**convertToMeters.** Достаточно удобно использовать наиболее подходящую для описания области течения единицу длины. В СИ таковой является метр. Однако, зачастую, геометрию расчетной области лучше описывать в мм, см или других единицах<sup>8</sup>. Для перехода к СИ, которая является внутренней системой единиц в *OpenFOAM*, используется коэффициент, заданный в этой строке

```
convertToMeters 0.1;
```

Такая запись означает, что координаты в *blockMeshDict* вводятся в *дм*.

**vertices.** «Строительным материалом» блочной расчетной сетки в *OpenFOAM* является шестигранный блок, задаваемый в декартовой системе координат с помощью набора из 8 вершин. Можно создавать блоки с меньшим числом граней просто дублируя записи о совпадающих вершинах. Вершина однозначно определяется местом в списке, нумерация вершин начинается с 0. Нумерация вершин блока производится по часовой стрелке при взгляде в направлении оси *z*. Каждая вершина задаётся тройкой декартовых координат, разделённых пробелами в круглых скобках: (*x y z*). Т. к. решаемая задача носит двумерный характер, а *OpenFOAM* предназначен, в первую очередь, для расчёта

<sup>7</sup> разбитой на блоки

<sup>8</sup> американцам и англичанам – в дюймах и футах

трёхмерных течений, размер расчетной сетки в направлении оси  $z$  принимается достаточно малым и равным 0,1 дм. С учётом сделанных замечаний список вершин будет выглядеть как

```
vertices
(
  (0 0 0)
  (1 0 0)
  (1 1 0)
  (0 1 0)
  (0 0 0.1)
  (1 0 0.1)
  (1 1 0.1)
  (0 1 0.1)
);
```

**blocks.** В этом разделе описываются блоки. Описание каждого блока состоит из определения формы блока<sup>9</sup> – hex, списка вершин (0 1 2 3 4 5 6 7), числа ячеек расчетной сетки в направлении каждой из осей координат (20 20 1) и закона изменения размера (измельчения или укрупнения) ячеек в направлении каждой из осей simpleGrading (1 2 3). Т. к. расчетная область в рассматриваемой задаче состоит из одного блока, то данный раздел будет выглядеть как

```
blocks
(
  hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
);
```

**NB.** Т. к. решаемая задача носит двумерный характер, расчетная область в направлении оси  $z$  состоит только из одной ячейки.

**edges.** В этом разделе находится описание ребер блока, имеющих форму, отличную от отрезка прямой: в виде дуги окружности или сплайна. Т. к. используемый блок не имеет криволинейных границ – этот список пуст

```
edges
(
);
```

**patches.** В этом списке хранится описание граней блока и граничных условий. Каждой грани соответствует определенный тип, напри-

<sup>9</sup> возможные формы: hex, wedge, prism, pyr, tet, tetWedge

мер, `wall`, граничное условие, которое задается в виде имени из словаря<sup>10</sup> граничных условий и список вершин, определяющих грань. Для верхней грани, соприкасающейся с движущейся жидкостью, необходимо установить тип `wall`, граничные условия<sup>11</sup> `movingWall` и задать список вершин, составляющих грань (3 7 6 2). Список следует начинать с вершины находящейся в начале локальной системы координат и перечислять в нём вершины по часовой стрелке при взгляде в положительном направлении оси  $z$ . Если указанному типу и граничным условиям соответствуют несколько граней, то списки определяющих их вершин следуют друг за другом внутри круглых скобок. Каждая грань должна быть описана в разделе `patches` только один раз.

Описание граней блока для рассматриваемой задачи будет выглядеть как

```
patches
(
  wall movingWall
  (
    (3 7 6 2)
  )
  wall fixedWalls
  (
    (0 4 7 3)
    (2 6 5 1)
    (1 5 4 0)
  )
  empty frontAndBack
  (
    (0 3 2 1)
    (4 5 6 7)
  )
);
```

Как можно заметить, верхняя грань (рис. 2.4) соответствует движущейся стенке, представляющей поток жидкости. Левая, нижняя и правая грани блока – неподвижным твёрдым стенкам полости. Передняя и задняя – фиктивными стенкам без определённых граничных условий.

<sup>10</sup> от англ. *dict[ionary]* – словарь

<sup>11</sup> постановка граничных условий будет описана чуть ниже

**mergePatchPairs.** Для задач гидрогазодинамики со сложной геометрией расчетной области не всегда возможно обойтись одним блоком расчетной сетки. Если таких блоков несколько, то в разделе mergePatchPairs описываются способы объединения блоков в единую расчетную область. Т. к. для рассматриваемой задачи достаточно одного блока расчетной сетки, этот раздел остается пустым

```
mergePatchPairs
(
);
```

## 2.4. Условия однозначности

**Давление.** Граничные и начальные условия (условия однозначности<sup>12</sup>), для давления описаны в файле [0\p].

В заголовке файла:

```
FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    object p;
}
```

указывается тип данных, описываемых в файле: class volScalarField, соответствующий полю скалярной переменной ScalarField для контрольного объема vol и, собственно, зависимая переменная, для которой заданы соответствующие условия однозначности: object p.

Непосредственно в файле сначала указывается размерность переменной заданием соответствующих показателей степени

$$\text{кг}^{n_1} \cdot \text{м}^{n_2} \cdot \text{с}^{n_3} \cdot \text{К}^{n_4} \cdot \text{моль}^{n_5} \cdot \text{А}^{n_6} \cdot \text{кД}^{n_7}$$

в массиве [n<sub>1</sub> n<sub>2</sub> n<sub>3</sub> n<sub>4</sub> n<sub>5</sub> n<sub>6</sub> n<sub>7</sub>]. Т. к. используется кинематическое давление, то соответствующая строка будет выглядеть как

```
dimensions [0 2 -2 0 0 0 0];
```

Следующая строка позволяет задать вид начальных условий и начальное приближение для поля (internalField) давления

```
internalField uniform 0;
```

<sup>12</sup> Условия однозначности выделяют конкретную задачу из целого класса задач. Они включают в себя граничные и начальные условия, а также геометрию и размеры расчетной области. Иногда, как в нашем случае, условия однозначности понимаются уже – только как граничные и начальные условия.

Здесь используется «однородное»<sup>13</sup> – uniform, поле давления на всей расчетной сетке, значение относительного<sup>14</sup> кинематического давления равняется  $\pi = p^* - p_0^* = \frac{p-p_0}{\rho} = 0$ .

Затем, в словаре boundaryField перечисляются граничные условия для давления. Всего в рассматриваемой задаче встречается три<sup>15</sup> типа граничных условий: movingWall – на подвижной стенке, в области контакта жидкости в полости с жидкостью в канале; fixedWall – на неподвижных твёрдых стенках; frontAndBack – на фронтальной и тыльной гранях блока расчётной области. Для стенок задано граничное условие zeroGradient, означающее равенство нулю градиента давления на границе расчетной области  $\frac{\partial p}{\partial \vec{n}} = 0$ , где  $\vec{n}$  – вектор нормали к границе. А для фронтальной и тыльной поверхностей расчетной области «пустые» – empty, граничные условия.

Таким образом, словарь граничных условий будет иметь вид:

```
boundaryField
{
    movingWall
    {
        type zeroGradient;
    }
    fixedWalls
    {
        type zeroGradient;
    }
    frontAndBack
    {
        type empty;
    }
}
```

**Скорость.** В отличие от давления, скорость – векторная величина, поэтому в заголовке файла [0\U], тип данных указан как volVectorField, а зависимая переменная обозначена как U.

Для скорости задана размерность [м/с]:

```
dimensions [0 1 -1 0 0 0 0];
```

В качестве начального приближения используется однородное поле скоростей равное нулю во всей расчётной области:

<sup>13</sup> значение зависимой переменной одинаково во всех точках расчетной области

<sup>14</sup> избыточного давления или разрежения

<sup>15</sup> два из них описаны в соответствующем разделе

```
internalField uniform (0 0 0);
```

Т. к. скорость является векторной величиной, то в декартовой системе координат она задаётся вектором из трёх составляющих (0 0 0) в направлении осей координат  $x$ ,  $y$  и  $z$ , соответственно.

В словаре граничных условий для `movingWall` задана постоянная величина – `fixedValue`, составляющей скорости направлении оси  $x$ , равная 1, и равные нулю две другие составляющие для всех точек границы – `uniform (1 0 0)`. На неподвижных стенках задано условие прилипания<sup>16</sup> `noSlip`. Для фронтальной и тыльной поверхностей расчетной области – `empty`. Т. о. словарь граничных условий выглядит как

```
boundaryField
{
    movingWall
    {
        type fixedValue;
        value uniform (1 0 0);
    }
    fixedWalls
    {
        type noSlip;
    }
    frontAndBack
    {
        type empty;
    }
}
```

**Свойства жидкости.** Единственным свойством жидкости, которое необходимо задать, является кинематическая вязкость  $\nu = 0,01 \text{ м}^2/\text{с}$ . Оно должно быть сохранено в файле [`constant\transportProperties`]. В заголовке файла

```
FoamFile
{
    version    2.0;
    format     ascii;
```

---

<sup>16</sup> Граничные условия на твёрдой стенке для скорости разделяются на т. н. условия *непротекания* и *прилипания*. Они различаются заданием вектора скорости фиктивного потока жидкости, «движущегося» за поверхностью твёрдой стенки. Такой приём используется для моделирования пристеночных эффектов. Условия непротекания – это такие граничные условия, при которых только нормальная к стенке составляющая скорости меняет свой знак, а тангенциальная – нет. Условия прилипания заключаются в том, что на стенке свой знак меняет как нормальная, так и тангенциальная составляющие вектора скорости.

```
class    dictionary;
location "constant";
object   transportProperties;
}
```

указано, что это словарь, расположенный (*location*), в папке "constant", содержащий сведения (*object*), о физических свойствах жидкости – *transportProperties*.

Кинематическая вязкость задаётся с помощью одной единственной записи

```
nu [0 2 -1 0 0 0 0] 0.01;
```

в которой свойству  $\nu$  – *nu*, размерностью  $\text{м}^2 \cdot \text{с}^{-1}$  сопоставлено числовое значение 0,01.

## 2.5. Управление процессом расчёта

Для определения параметров процесса моделирования используется файл [*system\controlDict*]. Заголовок файла

```
FoamFile
{
    version    2.0;
    format     ascii;
    class      dictionary;
    location   "system";
    object     controlDict;
}
```

определяет его местоположение и кратко описывает содержимое.

Сам словарь содержит информацию об используемом солвере

```
application icoFoam;
```

В данном случае применяется специальный модуль для расчёта несжимаемых (*incompressible*) течений – *icoFoam*.

Задаётся начальное

```
startFrom startTime;
startTime 0;
```

и конечное

```
stopAt endTime;
endTime 0.5;
```

время для расчёта (время модели). Определяется значение шага по времени. Для этого используется т. н. критерий Куранта<sup>17</sup>

---

<sup>17</sup> физический смысл критерия Куранта заключается в том, чтобы возмущения потока «не проскакивали» через расчётную ячейку, не внося изменений в параметры жидкости в ней.

$$Co = U \frac{\Delta t}{\Delta x},$$

где  $\Delta t$  – шаг по времени, с;  $\Delta x$  – шаг дискретизации по пространству, м.

Для обеспечения устойчивости численной схемы, необходимо выполнение условия  $Co \leq 1$ .

Учитывая, что число ячеек расчётной сетки в направлении осей координат  $x$  или  $y$  равняется  $n = 20$ , а размер расчетной области  $d_x = d_y = d = 0,1$  м и, следовательно,  $\Delta x = \frac{d}{n} = \frac{0,1\text{ м}}{20} = 0,005$  м получаем, что  $\Delta t \leq 0,005$  с.

Т. о. величина шага по времени задаётся как

```
deltaT 0.005;
```

Для того, чтобы изучить поведение жидкости с течением времени, необходимо указать периодичность записи результатов расчёта, например, каждый 20-й шаг по времени:

```
writeControl timeStep;
writeInterval 20;
```

При этом для каждой записи будет создана новая папка, с именем текущего шага по времени

```
purgeWrite 0;
```

А запись значений будет производиться в текстовом формате без сжатия:

```
writeFormat ascii;
writePrecision 6;
writeCompression off;
```

Формат времени при записи результатов задается следующим образом

```
timeFormat general;
timePrecision 6;
runTimeModifiable true;
```

## 2.6. Параметры численной схемы

Настройка параметров используемой для моделирования численной схемы осуществляется путём их сохранения в файле `[system\fvSchemes]`. В заголовке файла указано его местоположение и дано описание содержащейся в нём информации.

```
FoamFile
{
    version 2.0;
```

```

format      ascii;
class       dictionary;
location    "system";
object      fvSchemes;
}

```

Значения, содержащиеся в этом словаре, описывают различные аспекты реализации численной схемы. В *OpenFOAM*, для записи уравнений в частных производных используется специальное надмножество<sup>18</sup> языка C++, реализованное в виде объектных библиотек(-и). Решение укороченных уравнений Навье-Стокса для ламинарного движения вязкой несжимаемой жидкости<sup>19</sup> выглядит следующим образом:

```

fvVectorMatrix UEqn
(
    fvm::ddt(U)
  + fvm::div(phi, U)
  - fvm::laplacian(nu, U)
);
solve(UEqn == -fvc::grad(p))

```

Соответственно, для данной конкретной математической модели, заданной дифференциальным уравнением в частных производных (ДУЧП), или системой ДУЧП, необходимо описать параметры численной схемы, используемой для её дискретизации.

**Дискретизация по времени.** Для решения нестационарных задач гидрогазодинамики, в которых, как и в рассматриваемом случае,  $\frac{\partial U}{\partial t} \neq 0$ , необходимо определить схему дискретизации уравнений в частных производных по времени. Для использования доступны следующие схемы:

- Euler – Эйлера, или двухточечную с разностями назад;
- CoEuler – Эйлера с переменным шагом по времени;
- CrankNicolson – Кранка–Николсона;
- backward – трёхточечная несимметричная с разностями назад;

Для решаемой<sup>20</sup> задачи достаточно задать двухточечную с разностями назад схему дискретизации<sup>21</sup> по умолчанию

```

ddtSchemes
{

```

<sup>18</sup> DSL – Domain Specific Language

<sup>19</sup> приведен фрагмент исходного текста *icoFoam*

<sup>20</sup> как и для большинства других

<sup>21</sup> т. н. схема *вперёд по времени* – BB

```
default Euler;
}
```

**Вычисление градиентов.** Члены, связанные с диффузией, например  $\nabla p^*$ , определяются градиентами соответствующих величин. Известно, что способ вычисления градиента влияет на точность результатов математического моделирования. Поэтому *OpenFOAM* позволяет выбрать наиболее подходящую для этого схему. Всего доступны несколько вариантов:

- Gauss – трёхточечная симметричная, центральными разностями;
- cellLimited
- cellMDLimited
- edgeCellsLeastSquares
- faceLimited
- faceMDLimited
- fourth
- leastSquares
- pointCellsLeastSquares

По умолчанию задана трёхточечная аппроксимация производных центральными разностями. Такая же схема применяется для вычисления градиента давления:

```
gradSchemes
{
    default Gauss linear;
    grad(p) Gauss linear;
}
```

**Вычисление дивергенции.** Дивергенция определяет влияние конвективной<sup>22</sup> составляющей, такой как  $\nabla(\vec{u}\vec{u})$ , связанной с полем скоростей. По умолчанию численная схема вычисления дивергенции не используется, а для вычисления поля скоростей используется трёхточечная аппроксимация центральными разностями<sup>23</sup>:

```
divSchemes
{
    default none;
    div(phi,U) Gauss linear;
}
```

<sup>22</sup> иногда называемой адвективной

<sup>23</sup> т. н. схема *центр по пространству* – ЦП. Т. о. общее название схемы – ВВЦП (*вперёд по времени, центр по пространству*)

Правильный выбор способа вычисления дивергенции может способствовать ускорению сходимости численной схемы.

**Вычисление Лапласиана.** В качестве схемы для вычисления Лапласиана  $-\nabla(\nu\nabla\vec{u})$ , базовым является выбор трёхточечной центрально-разностной аппроксимации:

```
laplacianSchemes
{
    default Gauss linear orthogonal;
}
```

**Интерполяция значений на гранях ячеек.** Для интерполяции значений на гранях<sup>24</sup> *OpenFOAM* предоставляет широкий спектр методов. Однако для решаемой задачи можно использовать линейную интерполяцию:

```
interpolationSchemes
{
    default linear;
}
```

**Вычисление нормальной к грани ячейки составляющей вектора градиента.** В общем случае грани ячейки могут не совпадать с плоскостями  $xu$ ,  $xz$  и  $uz$  декартовой системы координат. Поэтому, для определения потока через грань необходимо, в процессе моделирования, вычислять нормальную составляющую вектора градиента. Т. к. используемая расчётная сетка является ортогональной и, по своим свойствам, не отличается от регулярных расчётных сеток, для вычисления нормальной составляющей можно использовать ортогональную схему:

```
snGradSchemes
{
    default orthogonal;
}
```

## 2.7. Параметры модуля решения систем линейных алгебраических уравнений

Результатом дискретизации уравнений в частных производных является система линейных алгебраических уравнений (СЛАУ). Для решения больших СЛАУ, заданных в матричной форме  $\mathbf{Ax} = \mathbf{B}$  с раз-

---

<sup>24</sup> необходима для вычисления потоков через грани ячеек

реженной матрицей коэффициентов левых частей  $A$  широко используются разного рода итеративные методы. Используемые методы решения СЛАУ и их параметры сохраняются в файле [*system\fvSolutions*].

Заголовок файла:

```
FoamFile
{
    version    2.0;
    format     ascii;
    class      dictionary;
    location   "system";
    object     fvSolution;
}
```

кратко описывает его содержимое и местоположение в папках проекта.

Параметры итеративных методов решения систем уравнений для поля давлений и поля скоростей приводятся в словаре *solvers*. Здесь, для различных зависимых переменных могут быть использованы разные методы решения СЛАУ (*solver*), с разными параметрами, такими как чувствительность – *tolerance*, используемая в качестве критерия сходимости итеративного процесса, относительная чувствительность – *relTol*, используемая в качестве критерия сходимости в случае, если её значение отлично от 0, максимальное количество итераций – *maxIter*, (по умолчанию 1000) используемое для прерывания излишне долгих вычислений с медленной сходимостью и др. Для задачи течения несжимаемой вязкой жидкости в полости эти параметры равны:

```
solvers
{
    p
    {
        solver    PCG;
        preconditioner    DIC;
        tolerance 1e-06;
        relTol    0.05;
    }
    pFinal
    {
        $p;
        relTol 0;
    }
    U
    {
        solver    smoothSolver;
        smoother  symGaussSeidel;
        tolerance 1e-05;
    }
}
```

```
        relTol 0;
    }
}
```

Получаемое в процессе моделирования решение должно соответствовать законам сохранения, заданным в виде исходной системы ДУЧП. В нашем случае это означает, что поля давления и скорости должны в процессе решения корректироваться таким образом, чтобы они одновременно удовлетворяли и уравнению неразрывности, и уравнению движения. Алгоритм корректировки полей скорости и давления, используемый в *icoFoam*, называется *PISO* – *pressure-implicit split operator* (неявный по давлению оператор разделения), соответственно, в словаре *PISO* задаются его параметры `nCorrectors` – число циклов для корректировки поля давлений, `nNonOrthogonalCorrectors` – число внутренних циклов для коррекции неортогональности неструктурированной тетраэдральной расчётной сетки, `pRefPoint` или `pRefCell` – координаты точки в системе координат расчётной сетки или метка ячейки расчётной сетки в которой задано референсное давление. Этот параметр необходим в случае, если давление не задано ни одним граничным условием.

`pRefValue` – непосредственно заданное референсное давление, в большинстве случаев должно быть равно 0.

Для рассматриваемой задачи словарь *PISO* имеет вид:

```
PISO
{
    nCorrectors 2;
    nNonOrthogonalCorrectors 0;
    pRefCell 0;
    pRefValue 0;
}
```

## 2.8. Генерация расчётной сетки

Для генерации расчётной сетки необходимо в командной строке ввести команду *blockMesh* и нажать клавишу *<Enter>*.

Для проверки геометрии расчётной сетки можно в командной строке ввести команду *paraFoam* и нажать клавишу *<Enter>*. Результат должен быть подобен показанному на рис. 2.5.

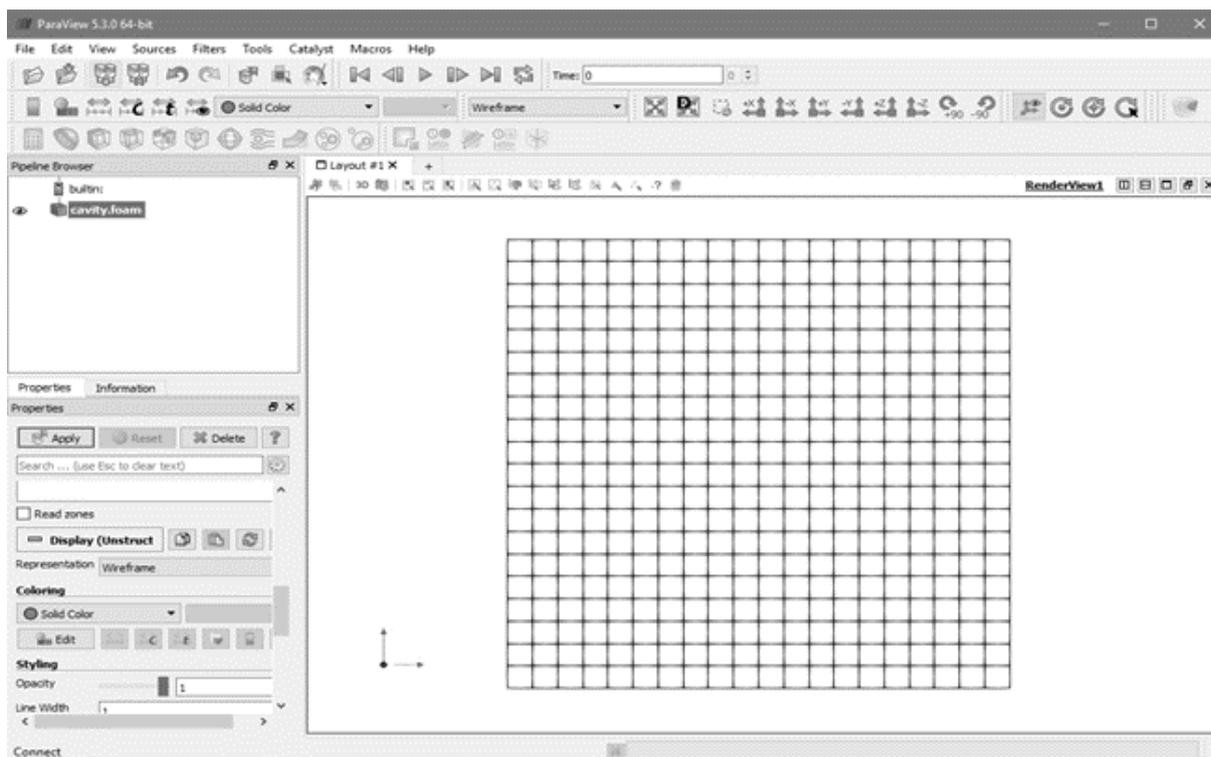


Рис. 2.5. Расчетная сетка в *paraFoam*

## 2.9. Запуск процесса расчёта

Для запуска процесса расчёта численного решения задачи течения вязкой несжимаемой жидкости необходимо в командной строке набрать *icoFoam* и нажать клавишу *<Enter>*.

## 2.10. Обработка результатов

Для запуска графического постпроцессора необходимо набрать в командной строке *paraFoam* и нажать клавишу *<Enter>*.

**Значения зависимых переменных.** Для поиска максимальных или минимальных значений необходимо выбрать пункт меню *Edit>Find Data...* и, в появившемся диалоговом окне (рис. 2.6), выбрать переменную и искомое значение: минимум, максимум, меньше среднего, больше среднего, равные среднему с некоторой погрешностью.

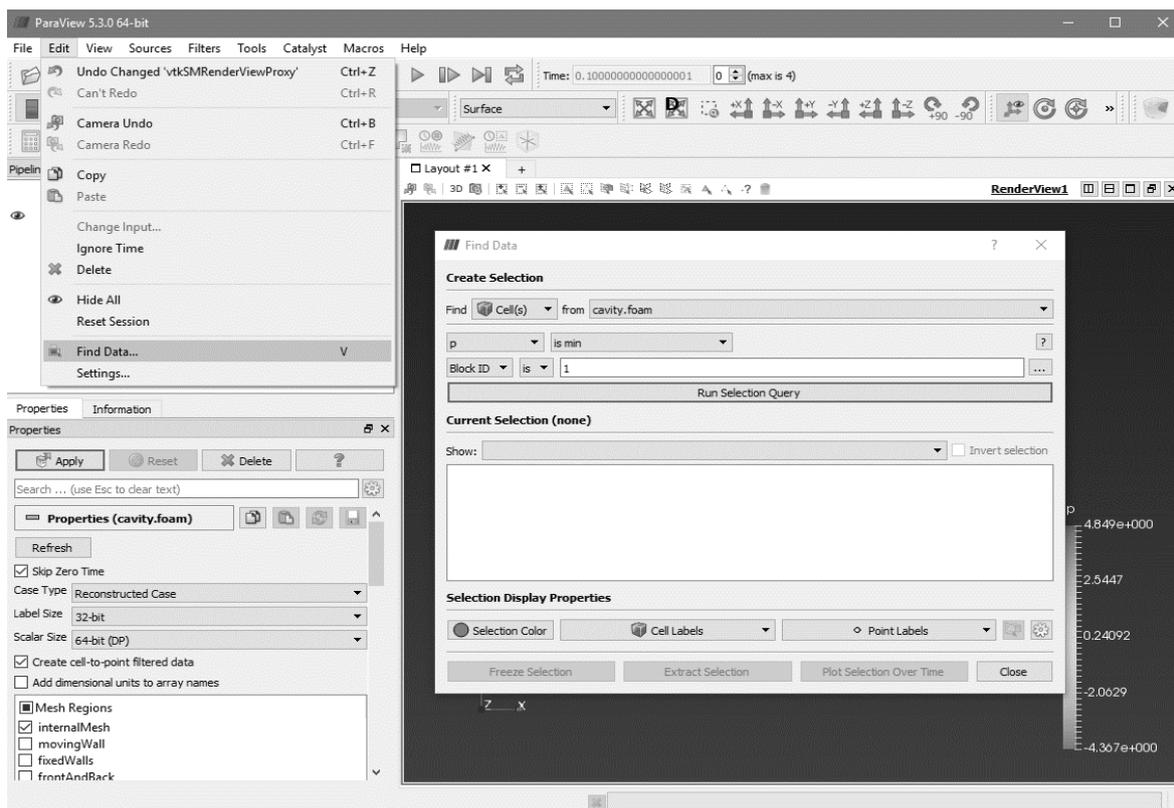


Рис. 2.6. Поиск минимального значения  $p$

Найденные<sup>25</sup> значение затем можно отобразить на расчетной сетке.

**Линии тока.** Для построения линий тока необходимо нажать на пиктограмму *Stream Tracer*  на панели инструментов. Это приведет к появлению элемента *StreamTracer1* в *Pipeline Browser*. Затем необходимо установить максимальную длину линии тока (рис. 2.7) и параметры для построения линий тока (рис. 2.8).



Рис. 2.7. Длина линии тока

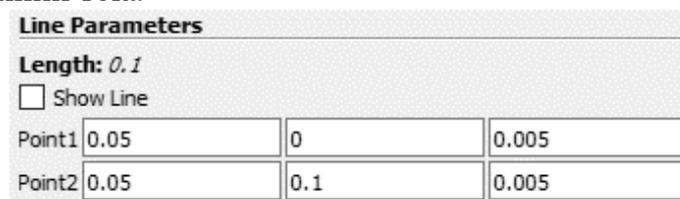


Рис. 2.8. Параметры для построения линий тока

<sup>25</sup> Для поиска значений необходим установленный модуль *Numpy*.

После установки параметров можно увидеть картину линий тока в полости, подобную показанной на рис. 2.9.

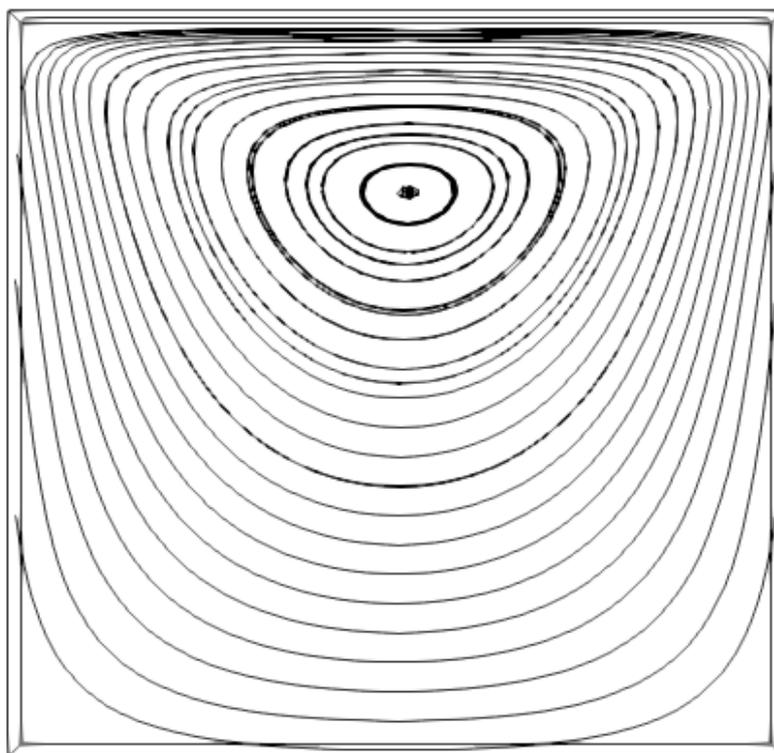


Рис. 2.9. Линии тока

**Векторные поля.** Для построения поля скоростей необходимо нажать на пиктограмму *Glyph*  на панели инструментов. Это приведет к появлению элемента *Glyph1* в *Pipeline Browser*. Затем необходимо установить тип меток (рис. 2.10).

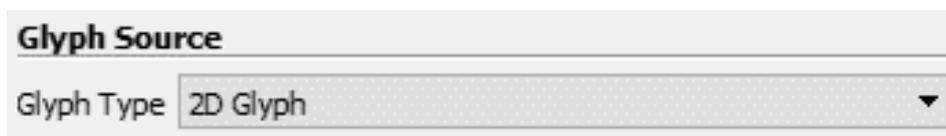


Рис. 2.10. Тип меток векторного поля

Векторное поле можно построить с постоянной длиной вектора, как на рис. 2.11.а, или применив масштабирование по модулю вектора скорости (рис. 2.11.б). Второй способ, как правило, позволяет получить более наглядное представление поля скоростей.

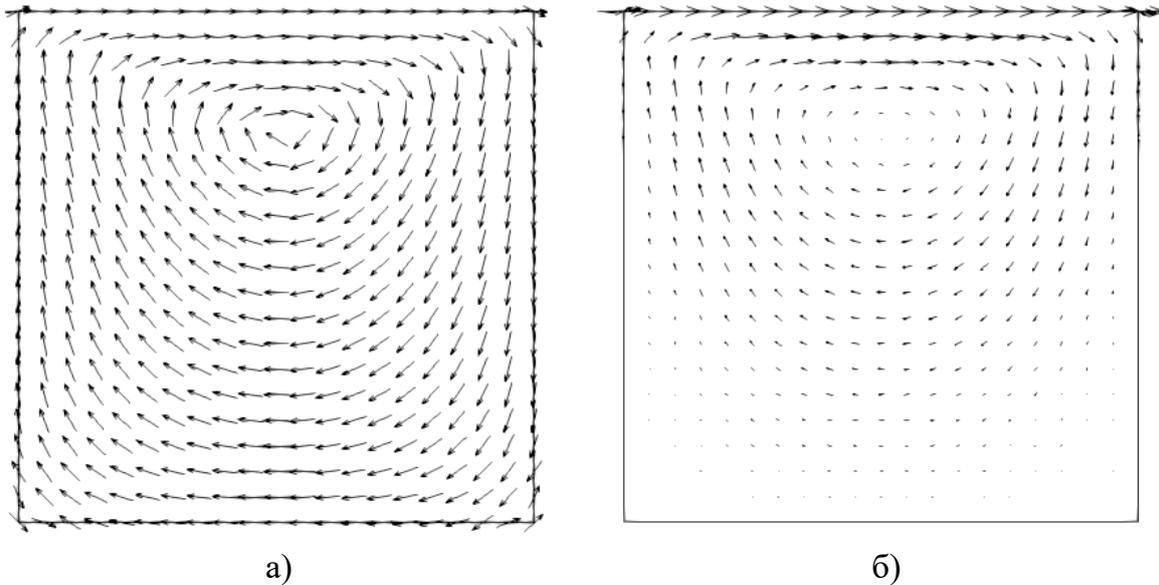


Рис. 2.11. Поле скоростей: а) немасштабированное; б) масштабированное

### Задание

1. Создайте проект «*cavity*» и введите все необходимые исходные данные. Сделайте предположения о планируемых результатах. Рассчитайте параметры течения в полости, обработайте результаты расчёта. Проанализируйте результаты расчёта и сделайте выводы об обоснованности прогноза.

2. Для проекта из п. 1 измените граничные условия с прилипания (*noSlip*) на непротекание (*slip*). Проведите расчёт, сравните его результаты с результатами, полученными в п. 1. Сделайте выводы.

3. Создайте (можно копированием) проект «*waterc*» для исследования течения воды при температуре 20,  $Re = 10$  и  $p = 100$  кПа в области, аналогичной области течения проекта «*cavity*». Задайте начальные и граничные условия. Установите, на сколько нужно изменить время окончания расчёта, чтобы получить стационарное решение, при этом устанавливайте разумное значение периодичности записи промежуточных результатов.

4. Для проекта из п. 3 установите  $p^* = 0$ . Проведите расчёт, сравните его результаты с полученными в п. 3, сделайте выводы.

5. Оформите отчёт по проделанной работе. Сформулируйте общие выводы.

### Вопросы для самоконтроля

1. Запишите систему уравнений, описывающую ламинарное течение вязкой, несжимаемой жидкости.

2. Запишите выражение для определения критерия Рейнольдса.

3. Запишите выражение для определения критерия Куранта.

4. Какие значения должен принимать критерий Куранта для обеспечения устойчивости численной схемы?

5. Дайте определение условий непротекания.

6. Дайте определение условий прилипания.

7. Что входит в понятие условий однозначности?

8. Что подразумевается под единообразным полем давления/скорости?

9. Какой член уравнения Навье-Стокса для несжимаемой вязкой жидкости вычисляется как градиент?

10. Какой член уравнения Навье-Стокса для несжимаемой вязкой жидкости вычисляется как дивергенция?

11. Какой член уравнения Навье-Стокса для несжимаемой вязкой жидкости вычисляется как лапласиан?

12. Опишите структуру проекта *OpenFOAM*.

### 3. ТЕЧЕНИЕ В ТРУБЕ

#### 3.1. Постановка задачи

Рассмотрим задачу по расчёту профиля скоростей потока в поперечном сечении и изменению скорости стационарного ламинарного потока воздуха в продольном сечении на оси прямого цилиндрического канала [4]. Т. к. давление в потоке будет изменяться незначительно, жидкость для данной задачи можно считать несжимаемой. Несмотря на то, что задачу можно рассматривать как двумерное осесимметричное течение в цилиндрических<sup>26</sup> координатах  $(r, z)$ , моделирование будем проводить в трёхмерных декартовых координатах  $(x, y, z)$ , т. к. это позволит получить навыки построения блочных расчётных сеток для областей сложной формы.

В качестве объекта исследования будем использовать прямую трубу (рис. 3,1) круглого сечения диаметром  $d = 15$  мм и длиной  $l = 140$  мм.

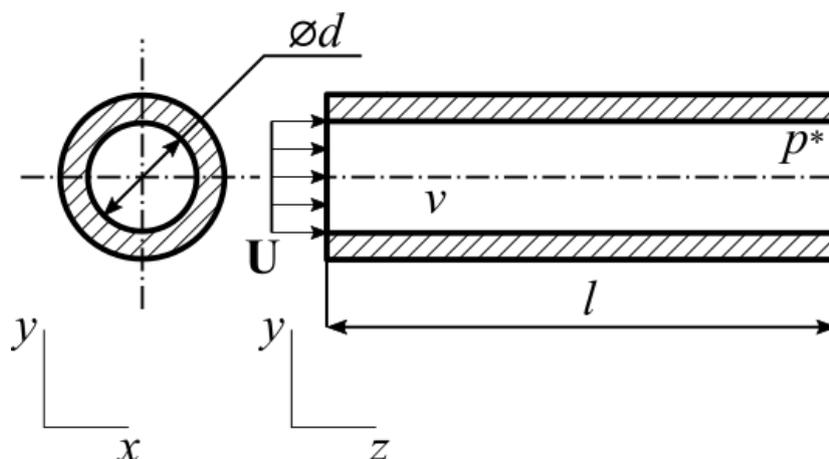


Рис. 3.1. Расчётная схема (эскиз расчётной области)

Течение несжимаемой жидкости описывается уравнением неразрывности

$$\nabla \mathbf{U} = 0$$

и уравнением движения, в виде укороченного уравнения Навье-Стокса для ламинарного течения вязкой несжимаемой жидкости

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}^T \mathbf{U}) - \nabla (v \nabla \mathbf{U}) = -\nabla p^*.$$

<sup>26</sup> подразумевается, что течение не зависит от угловой координаты  $\varphi$

В качестве рабочего тела используется воздух, обладающий следующими свойствами: кинематическая вязкость при  $20^{\circ}\text{C}$  –  $\nu = 1,5 \cdot 10^{-5} \text{ м}^2/\text{с}$ .

На стенке канала заданы условия прилипания, во входном сечении – составляющая скорости в направлении оси  $z$  равная  $0,1 \text{ м/с}$ , в направлении осей  $x$  и  $y$  – равная нулю, эпюра распределения скорости – прямоугольная. В выходном сечении заданы условия бесконечности. Для давления во входном сечении заданы условия бесконечности, а в выходном – величина кинематического давления, равная  $0 \text{ м}^2/\text{с}^2$ .

В качестве начальных условий выберем однородное поле скорости с составляющими вектора в направлении осей  $x$  и  $y$  равными  $0$ , а оси  $z$  – скорости невозмущенного потока,  $u_z = 0,1 \text{ м/с}$ , и однородное поле кинематического давления, равного  $p^* = 0 \text{ м}^2/\text{с}^2$ .

Начальные значение кинематического давления и кинематическое давление в выходном сечении выбраны равными нулю исходя из того соображения, что влияние на картину течения, в рассматриваемой задаче, будут оказывать градиенты давления, а не его абсолютная величина. Если возникнет необходимость, то, имея в виду что на расчётной сетке задано, фактически, относительное давление

$$p_{\text{отн}}^* = p^* - p_0^* = \frac{p - p_0}{\rho},$$

вычислить величину абсолютного давления можно как

$$p = p_{\text{отн}}^* \rho + p_0.$$

### 3.2. Расчетная сетка

При построении расчетной сетки для течения в цилиндрическом канале следует руководствоваться следующими соображениями:

1) во избежание внесения искажений в профиль скоростей по оси течения должны быть расположены расчётные ячейки, т. к. максимальное значение скорости потока наблюдается именно по оси канала;

2) необходимо выполнить измельчение расчетной сетки вблизи стенки, т. к. именно в этой области наблюдаются максимальные градиенты скорости в поперечном направлении.

Поскольку в основе препроцессора *blockMesh* лежит работа с шестигранными блоками прямоугольной<sup>27</sup> формы с возможностью зада-

---

<sup>27</sup> кубы, параллелепипеды и т. д.

ния криволинейных границ, то, для упрощения геометрических построений, необходимо использовать параллелепипед не с гранями, параллельными координатным плоскостям (рис. 3.2.а), а повернутый на  $45^\circ$ , как на рис. 3.2.б

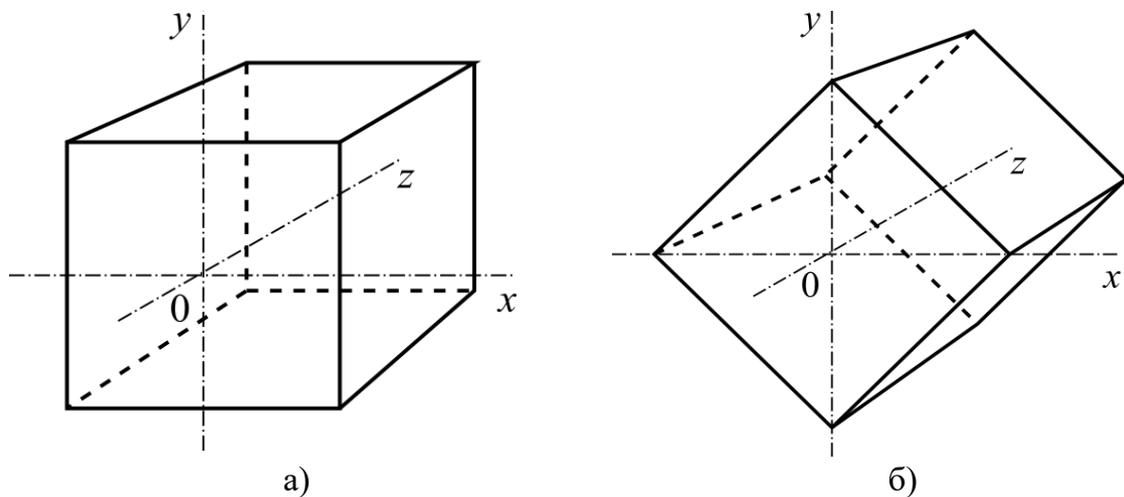


Рис. 3.2. Блок расчётной сетки: а) с гранями, параллельными координатным плоскостям; б) повернутый на  $45^\circ$

**Простейшая расчётная сетка.** В первую очередь построим простейшую расчётную сетку из одного блока с постоянным размером ячеек в радиальном направлении. Схема для его построения приведена на рис. 3.3.

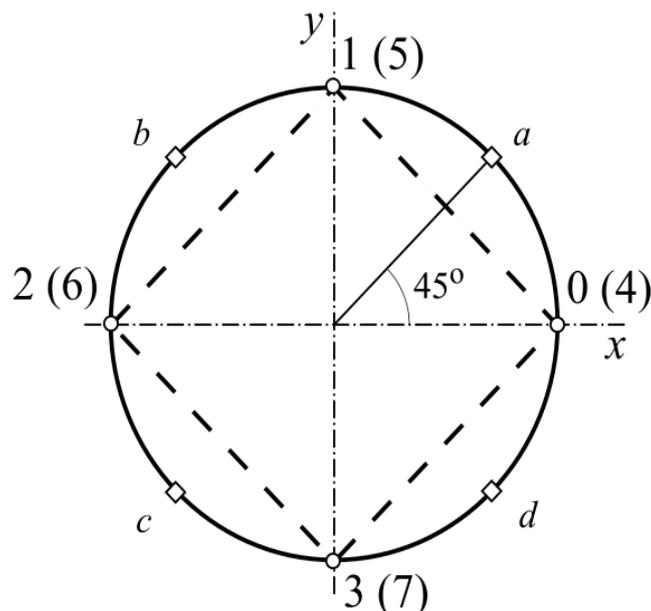


Рис. 3.3. Схема блока для построения простейшей расчётной сетки

Анализ рис. 3.3 показывает, что блок строится с помощью набора вершин, отмеченных цифрами, причем вершины обозначенные цифрами в скобках лежат в плоскости  $z = 140$  мм. Соответственно, список вершин для построения блока будет выглядеть следующим образом<sup>28</sup>:

```
vertices
(
  ( 7.5    0    0)
  ( 0     7.5   0)
  (-7.5   0    0)
  ( 0    -7.5   0)
  ( 7.5   0   140)
  ( 0     7.5  140)
  (-7.5  0   140)
  ( 0    -7.5  140)
);
```

Единственный блок, состоящий из 50 ячеек в направлении координатных осей  $x$ ,  $y$  и  $z$  описывается следующим образом:

```
blocks
(
  hex (0 1 2 3 4 5 6 7) (50 50 50) simpleGrading (1 1 1)
);
```

Для того чтобы задать ребро блока в виде дуги окружности, необходимо вычислить координаты промежуточной точки, через которую проходит окружность.

**NB.** Вычисления нужно проводить с максимально возможной точностью, поскольку, даже небольшая, погрешность в результате может привести к неожиданным, и, порой, весьма существенным, искажениям расчётной сетки.

Точки начала и конца дуги задаются вершинами блока, соединенными ребром. Точки, через которые проходят дуги окружностей на рис. 3.4 помечены латинскими буквами « $a$ »...« $d$ ». В приведённом случае достаточно произвести вычисление всего один раз, по известной из школьного курса тригонометрии формуле

$$x_a = y_a = R \cdot \cos 45^\circ = R \frac{\sqrt{2}}{2}, \quad (3.1)$$

<sup>28</sup> здесь и далее все размеры даны в мм

и подставить результат в список рёбер выбирая знаки координат точек в соответствии с квадрантом системы координат  $xu$ . Для нашего случая  $R = 7,5$  мм и результат вычислений по (3.1), пригодный для использования при генерации расчётной сетки, равен

$$x_a = 5,30330085889910643300 \text{ мм.}$$

Команда, создающая ребро в виде дуги окружности, называется `arc`. Её аргументами являются вершины блока  $n_1$  и  $n_2$ , соединяемые дугой и координаты точки  $(x, y, z)$ , через которую проходит дуга окружности:

```
arc n1 n2 (x y z)
```

Так как все поперечные рёбра блока являются дугами окружностей оснований цилиндра, получим

```
edges
(
  arc 0 1 ( 5.303300858899106433 5.303300858899106433 0)
  arc 1 2 (-5.303300858899106433 5.303300858899106433 0)
  arc 2 3 (-5.303300858899106433 -5.303300858899106433 0)
  arc 3 0 ( 5.303300858899106433 -5.303300858899106433 0)
  arc 4 5 ( 5.303300858899106433 5.303300858899106433 140)
  arc 5 6 (-5.303300858899106433 5.303300858899106433 140)
  arc 6 7 (-5.303300858899106433 -5.303300858899106433 140)
  arc 7 4 ( 5.303300858899106433 -5.303300858899106433 140)
);
```

Удобной представляется возможность использования макроподстановки значений переменных в словарях *OpenFOAM*. В нашем случае для этого необходимо перед списком `edges` добавить две строки:

```
a 5.30330085889910643300;
_a -5.30330085889910643300;
```

которые описывают вводимые переменные  $a = x_a$  и  $\_a = -x_a$ . Далее в файле `[system\blockMeshDict]` можно ссылаться на значения этих переменных просто подставляя их имя после знака `$`, так: `$a`, или так: `$_a`. В результате список рёбер блока приобретёт вид:

```
edges
(
  arc 0 1 ( $a $a 0)
  arc 1 2 ( $_a $a 0)
  arc 2 3 ( $_a $_a 0)
```

```
arc 3 0 ( $a $_a 0)
arc 4 5 ( $a $a 140)
arc 5 6 ($_a $a 140)
arc 6 7 ($_a $_a 140)
arc 7 4 ( $a $_a 140)
);
```

***NB.*** Использование макроподстановок позволяет существенно снизить объём вводимой информации и уменьшить вероятность опечаток.

Осталось определить, какие граничные условия действуют на гранях блока. Нижнее основание блока, образуемое вершинами 0, 1, 2 и 3 будем считать входом канала, и сопоставим ему граничные условия `inlet`. Верхнее основание из вершин 4, 5, 6 и 7 – выход (`outlet`). Четыре боковые грани, состоящие из вершин 0, 1, 5, 4; 1, 2, 6, 5; 2, 3, 7, 6 и 3, 0, 4, 7 относятся к стенке канала, поэтому им соответствуют граничные условия `cylinderWall`. В результате список граничных условий будет выглядеть следующим образом:

```
boundary
(
  inlet
  {
    type patch;
    faces
    (
      (0 1 2 3)
    );
  }
  cylinderWall
  {
    type wall;
    faces
    (
      (0 1 5 4)
      (1 2 6 5)
      (2 3 7 6)
      (3 0 4 7)
    );
  }
  outlet
  {
    type patch;
```

```
        faces
        (
            (4 5 6 7)
        );
    }
);
```

На этом формирование файла [*system\blockMeshDict*] можно считать завершённым. Получившаяся в результате его обработки расчётная сетка (вид в направлении оси *z*) показана на рис. 3.4.

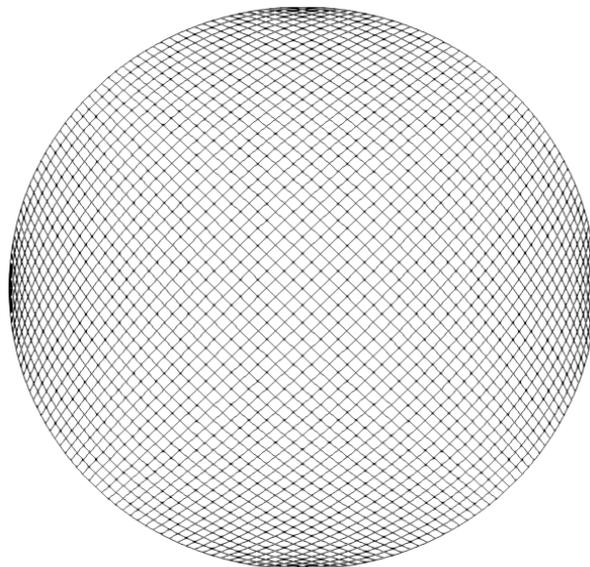


Рис. 3.4. Простейшая расчётная сетка

Анализируя построенную расчётную сетку, можно сделать несколько важных выводов. Во-первых, по оси канала расположен ряд расчётных ячеек, что позволит уменьшить погрешность при расчёте максимальной скорости потока. Во-вторых, размеры ячеек в направлении от центра канала к периферии остаются неизменными или изменяются слабо. В-третьих, вблизи продольных рёбер канала наблюдается сильное искажение формы ячеек, что может привести к снижению точности результатов расчёта.

**Локальное измельчение расчётной сетки.** При построении блока расчётной сетки можно указать не только количество ячеек в направлении каждой из осей локальных координат блока, но и закон изменения их размера вдоль соответствующей оси координат — *simpleGrading*, или вдоль каждого из рёбер — *edgeGrading*. После ключ-

чевого слова размещается список коэффициентов измельчения, каждый из которых равен отношению размера ячейки в конце ребра к размеру ячейки в начале ребра. Таким образом, величина коэффициента измельчения равная единице приводит к формированию расчётной сетки из одинаковых ячеек, больше единицы – из увеличивающихся, меньше – из уменьшающихся. Например `simpleGrading(1 3 0.3)` приведёт к формированию расчётной сетки с ячейками, сохраняющими свой размер в направлении оси  $x$ , увеличивающимися в направлении оси  $y$  в 3 раза и уменьшающимися в направлении оси  $z$  в 3 раза, почти как на рис. 3.5.

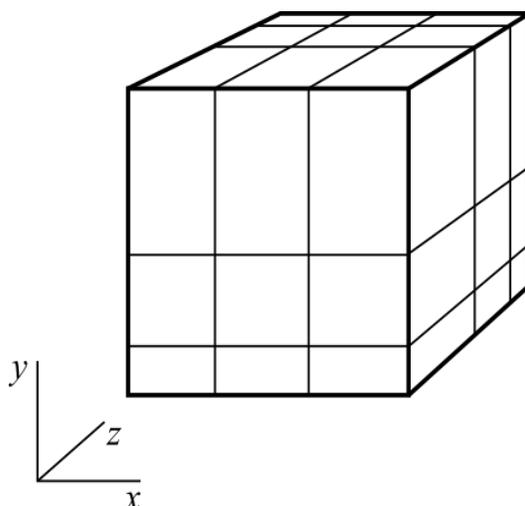


Рис. 3.5. Изменение размера ячеек расчётной сетки

В списке коэффициентов измельчения можно задавать список подобластей для локального измельчения расчётной сетки в виде:  $(\delta_e, \delta_c, \epsilon)$ , где  $\delta_e$  – участок ребра для измельчения, доля или % длины ребра;  $\delta_c$  – относительное число ячеек на участке, доля или % от количества ячеек вдоль ребра;  $\epsilon$  – коэффициент измельчения.

Для рассматриваемого случая каждое ребро в направлении локальных осей координат  $x$  и  $y$  разбивается на три участка:

- 1) начальная 1/3 ребра, содержащая 40% ячеек увеличивающихся в 10 раз;
- 2) средняя 1/3 ребра, содержащая 20% ячеек постоянного размера.
- 3) конечная 1/3 ребра, содержащая 40% ячеек уменьшающихся в 10 раз.

В направлении оси  $z$  размер ячеек должен сохраняться неизменным. В результате список блоков с описанием блока с локальным измельчением расчётной сетки будет выглядеть следующим образом:

```

blocks
(
  hex (0 1 2 3 4 5 6 7) (50 50 50)
  simpleGrading (
    (
      (33 40 10)
      (33 20 1)
      (33 40 0.1)
    )
    (
      (33 40 10)
      (33 20 1)
      (33 40 0.1)
    )
  )
  1
)
);

```

А построенная простейшая расчётная сетка с локальным измельчением выглядит так, как показано на рис. 3.6.

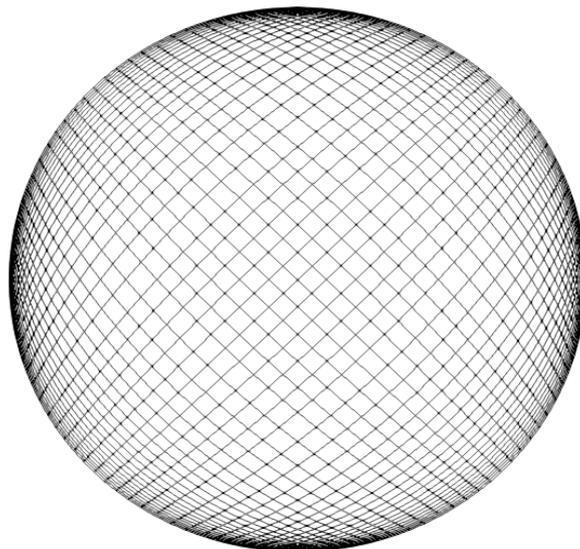


Рис. 3.6. Простейшая расчётная сетка с локальным измельчением ячеек

Как можно заметить, теперь в пристеночной области размер ячеек уменьшается, а в области оси канала остаётся неизменным и имеет максимальный размер. Однако проблема локального искажения формы ячеек вблизи вершин блока так и осталась нерешённой.

**Комплексная расчётная сетка.** Для уменьшения локальных искажений расчётной сетки вблизи стенки создадим её из нескольких

блоков (рис. 3.7), один из которых (V) будет служить для описания осевой области канала, а четыре других (I...IV) – пристеночных.

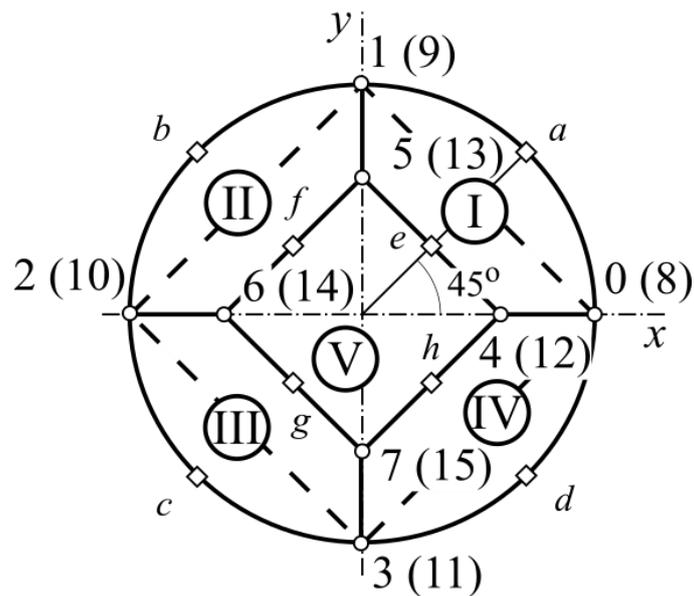


Рис. 3.7. Схема блока для построения комплексное расчётной сетки

Список вершин будет выглядеть следующим образом:

```
vertices
(
  ( 7.5  0  0)
  ( 0  7.5  0)
  (-7.5  0  0)
  ( 0 -7.5  0)
  ( 3.75 0  0)
  ( 0  3.75 0)
  (-3.75 0  0)
  ( 0 -3.75 0)
  ( 7.5  0  140)
  ( 0  7.5  140)
  (-7.5  0  140)
  ( 0 -7.5  140)
  ( 3.75 0  140)
  ( 0  3.75 140)
  (-3.75 0  140)
  ( 0 -3.75 140)
);
```

а описание блоков:

```
blocks
(
  hex (0 1 5 4 8 9 13 12) (25 25 50)
  simpleGrading (1 10 1)
```

```

hex (1 2 6 5 9 10 14 13) (25 25 50)
    simpleGrading (1 10 1)
hex (2 3 7 6 10 11 15 14) (25 25 50)
    simpleGrading (1 10 1)
hex (3 0 4 7 11 8 12 15) (25 25 50)
    simpleGrading (1 10 1)
hex (4 5 6 7 12 13 14 15) (25 25 50)
    simpleGrading (1 1 1)
);

```

Блоки 0...3, соответствующие блокам I...IV на рис. 3.7, заданы с измельчением ячеек расчётной в направлении оси  $u$  локальной системы координат. Т. к. базовой плоскостью выбрана наружная грань, то эта ось направлена от стенки канала к его центру, и для измельчения ячеек в пристеночной области необходимо задать коэффициент измельчения  $\varepsilon > 1$ .

Список рёбер, имеющих форму дуги окружности, в связи с изменением нумерации вершин, задаётся как:

```

edges
(
    arc 0 1 ( $a $a 0)
    arc 1 2 ($_a $a 0)
    arc 2 3 ($_a $_a 0)
    arc 3 0 ( $a $_a 0)
    arc 8 9 ( $a $a 140)
    arc 9 10 ($_a $a 140)
    arc 10 11 ($_a $_a 140)
    arc 11 8 ( $a $_a 140)
);

```

Список типов граничных условий на гранях блоков:

```

boundary
(
    inlet
    {
        type patch;
        faces
        (
            (0 1 5 4)
            (1 2 6 5)
            (2 3 7 6)
            (3 0 4 7)
        )
    }
);

```

```

        (4 5 6 7)
    );
}
cylinderWall
{
    type wall;
    faces
    (
        (0 1 9 8)
        (1 2 10 9)
        (2 3 11 10)
        (3 0 8 11)
    );
}
outlet
{
    type patch;
    faces
    (
        (8 12 13 9)
        (9 13 14 10)
        (10 14 15 11)
        (11 15 12 8)
        (12 15 14 13)
    );
}
);

```

*NB.* Блоки расчётной сетки, имеющие общие грани, объединяются автоматически, поэтому нет необходимости перечислять их в списке `mergePatchPairs`.

Расчётная сетка, построенная с использованием нескольких блоков, показана на рис. 3.8.

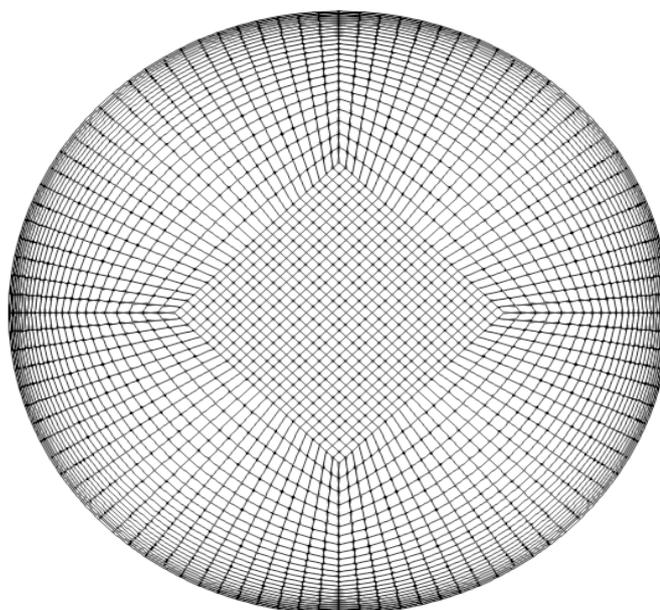


Рис. 3.8. Комплексная расчётная сетка

Такая расчетная сетка удовлетворяет нашим требованиям и даёт возможность проводить расчёты с достаточной точностью [4].

**Модификация расчётной сетки.** Расчётная сетка (рис. 3.8), построенная в предыдущем разделе, вполне пригодна для решения практических задач, однако в области перехода от центрального блока к периферийным наблюдается локальное искажение ячеек в радиальном направлении, что, потенциально, может сказаться на результатах расчёта. Попытаемся исправить положение, изменив форму поперечных рёбер внутреннего блока с отрезка прямой на дугу окружности. Для этого вычислим координаты точек  $e$ ,  $f$ ,  $g$  и  $h$  по (3.1) и получим значения

```
f 2.65165042944955321650;
_f -2.65165042944955321650;
```

Список описаний формы рёбер дополнится следующими элементами:

```
arc 4 5 ( $f $f 0)
arc 5 6 ($_f $f 0)
arc 6 7 ($_f $_f 0)
arc 7 4 ( $f $_f 0)
arc 12 13 ( $f $f 140)
arc 13 14 ($_f $f 140)
arc 14 15 ($_f $_f 140)
arc 15 12 ( $f $_f 140)
```

Построенная в результате расчётная сетка будет иметь вид, показанный на рис. 3.9.

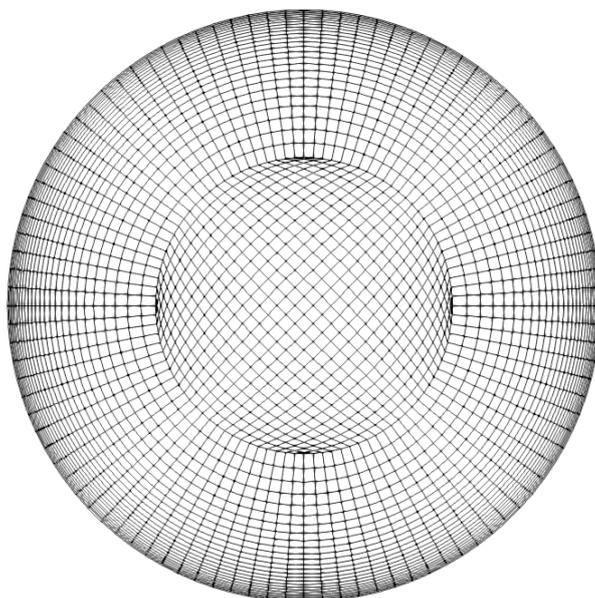


Рис. 3.9. Комплексная расчетная сетка с внутренней границей между блоками в виде окружности

Искажения размеров и формы ячеек периферийных блоков удалось успешно избежать, однако для центрального блока возникли искажения, характерные для простейшей расчетной сетки на рис. 3.4. Одновременно уменьшить искажения ячеек периферийных и центрального блока можно задав больший радиус кривизны дуг 4–5, 5–6, 6–7, 7–4, 12–13, 13–14, 14–15 и 15–12. Для этого необходимо модифицировать координаты промежуточной точки дуги  $e$ ,  $f$ ,  $g$  и  $h$  таким образом, чтобы она оказалась ближе к началу координат. Для этого умножим вычисленное ранее значение координаты  $f$  и  ${}_f$  на  $7/8$  [6]:

```
f 2.32019412576835906444;
_f -2.32019412576835906444;
```

Внутри файлов словарей *OpenFOAM* можно выполнять вычисления и подставлять их результаты в переменные с помощью оператора `#calc "...";`. Благодаря этому фрагмент с вычислением координат можно записать следующим образом:

```
a #calc "7.5*cos(degToRad(45))";
_a #calc "-1*$a";
_f #calc "$a/2*7/8";
_f #calc "-1*$f";
```

*NB.* Использование вычислений в словарях *OpenFOAM* возможно только при установленном и настроенном компиляторе *C/C++*, например, *GCC*.

*NB.* Вычисления в словарях *OpenFOAM* могут существенно замедлить процесс генерации расчётной сетки.

*NB.* Если в словаре [*system\blockMeshDict*] указать радиус канала и его длину, исходя из которых препроцессор будет вычислять положение узлов и координаты промежуточных точек, то в результате получим параметрически заданную расчётную сетку.

Результат построения расчётной сетки показан на рис. 3.10.

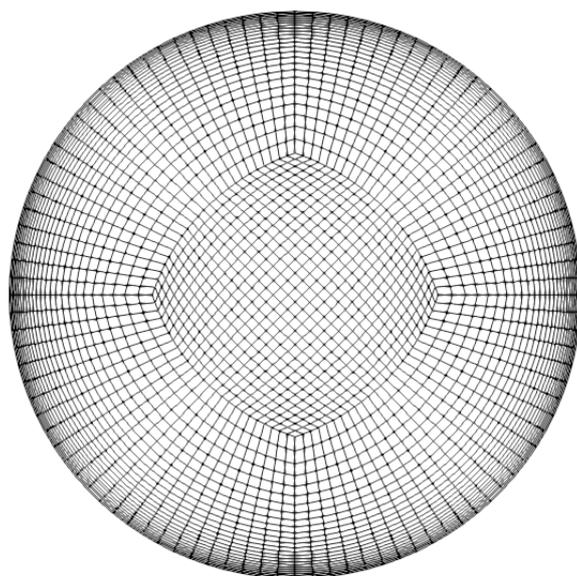


Рис. 3.10. Модифицированная комплексная расчетная сетка

Полученная расчётная сетка удовлетворяет предъявленным к ней требованиям и обладает достаточно малыми искажениями размеров и форм ячеек.

**«Параметрическая» расчётная сетка.** Взяв за основу расчётную сетку, построенную в предыдущем разделе (рис. 3.10), можно модифицировать словарь [*system\blockMeshDict*] таким образом, что геометрические параметры блоков будут изменяться в зависимости от некоторого, наперёд заданного, коэффициента  $\beta$ . Такую сетку можно

считать «параметрической», т. к. форма и размеры составляющих её ячеек будут зависеть от величины  $\beta$ .

Для этого присвоим переменной  $r_1$  значение, равное радиусу цилиндрического канала, в свою очередь равное  $x_0 = 7,5$  мм, а переменной  $r_2 - x_4 = \beta x_0$ . Чтобы препроцессор *blockMesh* выполнил необходимые вычисления, перед списком вершин блоков нужно добавить строки

```
r1 7.5;
beta #calc "2./3.";
_r1 #calc "-1*$r1";
_r2 #calc "$r1*$beta";
_r2 #calc "-1*$r2";
```

а сам список вершин примет вид:

```
vertices
(
    ($r1 0 0)
    (0 $r1 0)
    ($_r1 0 0)
    (0 $_r1 0)
    ($r2 0 0)
    (0 $r2 0)
    ($_r2 0 0)
    (0 $_r2 0)
    ($r1 0 140)
    (0 $r1 140)
    ($_r1 0 140)
    (0 $_r1 140)
    ($r2 0 140)
    (0 $r2 140)
    ($_r2 0 140)
    (0 $_r2 140)
);
```

Также изменятся строки, в которых вычисляются координаты точек  $a, b, c, d - a$  и  $\_a$ , и  $e, f, g, h - f$  и  $\_f$ :

```
a #calc "$r1*cos(degToRad(45))";
_a #calc "-1*$a";
f #calc "$r2*cos(degToRad(45))*7/8";
_f #calc "-1*$f";
```

Других изменений в словаре [*system\blockMeshDict*] нет.

Полученная в результате расчётная сетка показана на рис. 3.11.

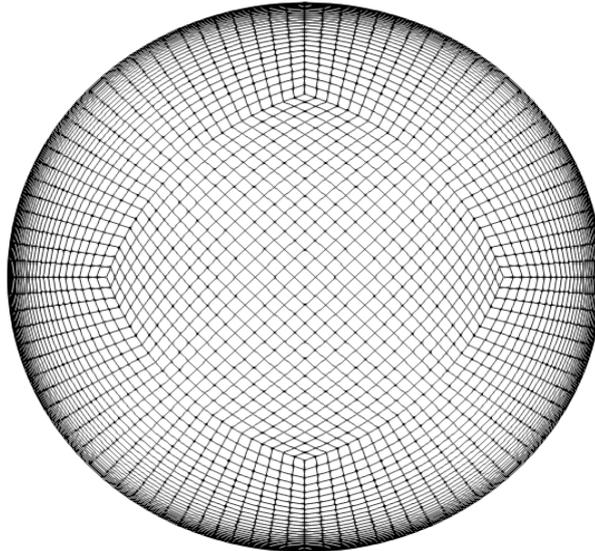


Рис. 3.11. Параметрическая расчётная сетка при  $\beta = 2/3$

Анализ построенной расчётной сетки (рис. 3.11) показывает, что она отвечает всем предъявленным требованиям, и, кроме того, обеспечивает «плавное» изменение размеров и формы расчётных ячеек, без существенных геометрических искажений.

### 3.3. Условия однозначности

**Давление.** Т. к. используется кинематическое давление  $p^* = \frac{p}{\rho}$ , то в словаре  $[0\backslash p]$  задаётся размерность  $\text{м}^2/\text{с}^2$ . Учитывая, что в начальный момент времени для всей области течения выбрано относительное кинематическое давление, равное нулю, начальные условия будут выглядеть как

```
internalField uniform 0;
```

Граничные условия по давлению на входе (`inlet`) и на стенке (`cylinderWall`) устанавливаются как `zeroGradient` (условие «бесконечности»), что означает, что за соответствующей границей давление будет равно давлению в расчётных ячейках. На выходной (`outlet`) границе устанавливается фиксированное значение кинематического давления, равное 0. В результате словарь граничных условий будет выглядеть следующим образом:

```
boundaryField
{
    inlet
    {
        type zeroGradient;
    }
}
```

```

outlet
{
    type fixedValue;
    value uniform 0;
}
cylinderWall
{
    type zeroGradient;
}
}

```

**Скорость.** Размерность скорости м/с. Т. к. в качестве начального приближения задаётся однородное поле скоростей, соответствующая запись в словаре [0\U] будет выглядеть как

```
internalField uniform (0 0 0.1);
```

В качестве граничных условий на входе в расчетную область (inlet) зададим фиксированное значение<sup>29</sup> (fixedValue). На выходе (outlet) – бесконечности<sup>30</sup> (zeroGradient). На стенке (cylinderWall) – прилипания (noSlip). Т. о. словарь граничных условий будет выглядеть как:

```

boundaryField
{
    inlet
    {
        type fixedValue;
        value uniform (0 0 0.1);
    }
    cylinderWall
    {
        type noSlip;
    }
    outlet
    {
        type zeroGradient;
    }
}

```

### 3.4. Свойства жидкости

Так как изучается ламинарное течение воздуха, являющегося Ньютоновской жидкостью, то в словаре [constant\transportProperties] необходимо задать свойство

<sup>29</sup> первого рода, Дирихле.

<sup>30</sup> второго рода, Неймана

```
transportModel Newtonian;
```

Кинематическая вязкость воздуха при 20°C составляет  $\nu = 1,5 \cdot 10^{-5} \text{ м}^2/\text{с}$ . Поэтому в словарь [*constant\transportProperties*] нужно внести следующую строку

```
nu [ 0 2 -1 0 0 0 0 ] 1.5e-5;
```

### 3.5. Турбулентность

Свойства, относящие к модели турбулентности, хранятся в словарях [*constant\turbulenceProperties*] и [*constant\RASProperties*].

Объектом исследования является ламинарное течение жидкости. Поэтому в словарь [*constant\turbulenceProperties*] достаточно поместить всего одну запись:

```
simulationType laminar;
```

В словаре [*constant\RASProperties*] также необходимо указать, что турбулентность моделируется (*RASModel*) с помощью ламинарной модели (*laminar*). Соответственно, для течения турбулентность (*turbulence*) нужно выключить (*off*). Также отсутствует необходимость вывода на печать (*printCoeffs*) коэффициентов модели турбулентности в начале расчёта (устанавливается значение *off*). Словарь, в результате, выглядит как

```
RASModel laminar;  
turbulence off;  
printCoeffs off;
```

### 3.6. Управление процессом расчёта

Параметры, используемые для управления процессом расчёта, указываются в словаре [*constant\ControlDict*]. В этом словаре указывается приложение для расчёта параметров течения:

```
application simpleFoam;
```

начальное:

```
startFrom latestTime;  
startTime 0;
```

и конечное

```
stopAt endTime;  
endTime 1000;
```

время модели, шаг дискретизации по времени<sup>31</sup>

```
deltaT 1;
```

<sup>31</sup> т. к. рассчитывается стационарное поле скоростей, в качестве шага по времени можно задать любое разумное значение.

и параметры записи результатов расчёта на диск

```
writeControl runTime;  
writeInterval 1000;  
purgeWrite 0;  
writeFormat ascii;  
writePrecision 6;  
writeCompression off;  
timeFormat general;  
timePrecision 6;  
runTimeModifiable false;
```

Для записи результатов выбрано значение, соответствующее последнему шагу по времени, что позволяет избежать сохранения промежуточных результатов, не имеющих практической ценности.

Формат записываемых данных не отличается от проекта из гл.2.

### 3.7. Параметры численной схемы

Параметры дискретизации приведены в словаре `[system\fvSchemes]`.

Так как исследуется стационарное течение, то производная по времени  $\frac{\partial}{\partial t}$  равна нулю, а в словаре параметров численной схемы необходимо сделать следующую запись:

```
ddtSchemes  
{  
    default steadyState;  
}
```

В качестве схемы для дискретизации диффузионных (градиентных,  $\nabla$ ) членов выберем трёхточечную симметричную схему второго порядка точности:

```
gradSchemes  
{  
    default Gauss linear;  
}
```

По умолчанию схему дискретизации адвективных членов (дивергенции,  $\nabla \cdot$ ) задавать не будем. Для дискретизации конвективных членов вида  $\nabla \cdot (\phi \mathbf{U})$  применим трёхточечную симметричную схему с ограничением (bounded). Это связано с тем, что полная производная зависимой переменной  $e$ , равная

$$\frac{De}{Dt} = \frac{\partial e}{\partial t} + \mathbf{U} \cdot \nabla e = \frac{\partial e}{\partial t} + \nabla \cdot (\mathbf{U}e) - (\nabla \cdot \mathbf{U})e,$$

включает в свой состав член с одним из сомножителей ( $\nabla \cdot \mathbf{U}$ ), для несжимаемого течения равным нулю в силу того, что уравнение неразрывности имеет вид  $\nabla \cdot \mathbf{U} = 0$ . Однако в процессе численного решения, во время выполнения промежуточных итераций, поле скоростей может не удовлетворять уравнению неразрывности<sup>32</sup> т. о., что  $\nabla \cdot \mathbf{U} \neq 0$ . В этом случае для ускорения сходимости и уменьшения погрешности промежуточных вычислений необходимо использовать специальный вариант ограничения<sup>33</sup> зависимой переменной, связанный с дискретизацией «нулевого» члена.

В качестве схемы дискретизации адвективного источникового члена, имеющего диффузионную природу, такого как  $\nabla \cdot \nu(\nabla \mathbf{U})^T$ , применим трёхточечную симметричную схему.

В результате словарь для схем дискретизации адвективных членов будет выглядеть как:

```
divSchemes
{
    default none;
    div(phi,U) bounded Gauss linear;
    div((nuEff*dev2(T(grad(U)))) Gauss linear;
}
```

Для дискретизации лапласианов  $\nabla^2$  используем трёхточечную симметричную схему с корректировкой неортогональности<sup>34</sup> расчётной сетки:

```
laplacianSchemes
{
    default Gauss linear corrected;
}
```

В качестве метода интерполяции полей зависимых переменных выберем линейную интерполяцию:

```
interpolationSchemes
{
    default linear;
}
```

<sup>32</sup> это касается только итеративной процедуры решения СЛАУ. В начале шага по времени, и в конце его, нарушения уравнения неразрывности наблюдаться не будет.

<sup>33</sup> *boundness*

<sup>34</sup> для дискретизации без такой корректировки, неортогональность расчётной сетки может составлять до 5°, с корректировкой – до 70°. На расчётных сетках с неортогональностью более 80° получить сходящееся решение очень трудно.

Для вычисления нормальной составляющей вектора градиента на грани расчётной ячейки выбираем схему с корректировкой неортогональности расчётной сетки:

```
snGradSchemes
{
    default corrected;
}
```

В заключение перечислим потоки величин через грани ячеек, которые рассчитываются приложением (в нашем случае *simpleFoam*) в процессе моделирования, аналогично заданным на т. н. «шахматной» сетке [8]. В нашем случае СДУЧП включает в себя  $\nabla p^*$ , вычисление которого через потоки на гранях ячейки приведёт к увеличению устойчивости численной схемы и гладкости полученного решения:

```
fluxRequired
{
    default no;
    p;
}
```

### 3.8. Параметры модуля решения

Параметры модуля решения<sup>35</sup> находятся в словаре [*system\fvSolutions*].

В словаре

```
solvers
{
    ...
}
```

хранятся описания параметров модулей решения СЛАУ дискретных аналогов соответствующих зависимых переменных.

Для поля давления выберем солвер GAMG – *generalised geometric-algebraic multi-grid*, в основу которого положен принцип вычисления начального приближения на огрублённой расчётной сетке. Алгоритм GAMG отличается достаточно высокой скоростью решения СЛАУ дискретных аналогов. Для его корректной работы необходимо задать число ячеек (по умолчанию 10) для огрубления расчётной сетки `nCellsInCoarsestLevel`. Другие параметры связаны с распространением грубого решения по расчётной сетке – `nPreSweeps`, максималь-

---

<sup>35</sup> солвера, калька с англ. *solver* – «решатель»

ного числа итераций – `maxIter`, выбора алгоритма огрубления расчётной сетки – `smoother`. В итоге, параметры солвера для поля давления задаются в виде словаря:

```
p
{
    solver GAMG;
    tolerance 1e-07;
    relTol 0.01;
    smoother GaussSeidel;
    nPreSweeps 1;
    nCellsInCoarsestLevel 500;
    maxIter 100;
}
```

Для расчёта поля скоростей можно использовать солвер *PBiCG – preconditioned biconjugate gradient* для асимметричных матриц. Для него необходима некоторая предварительная обработка матрицы коэффициентов **A** – `preconditioner`, которая осуществляется с помощью алгоритма `DILU`<sup>36</sup>. Словарь параметров модуля решения СЛАУ для поля скоростей будет выглядеть как:

```
U
{
    solver PBiCG;
    preconditioner DILU;
    tolerance 1e-8;
    relTol 0;
}
```

В *simpleFoam* используется алгоритм *SIMPLE – semi-implicit method for pressure-linked equation* (полуявный метод для уравнений, связанных давлением), позволяющий итеративным путём вычислить поля давления и скорости, удовлетворяющие уравнениям сохранения (неразрывности и Навье-Стокса). Для этого алгоритма необходимо задать число циклов корректировки неортогональности расчётной сетки `nNonOrthogonalCorrectors` равное нулю, т. к. решается стационарная задача, и разность между значениями двух последовательных итераций<sup>37</sup>, в качестве условия сходимости – `residualControl`:

```
SIMPLE
{
    nNonOrthogonalCorrectors 0;
    residualControl
```

<sup>36</sup> *diagonal incomplete-LU*, применяется для асимметричных матриц.

<sup>37</sup> для полей давления и скорости

```
{
    p 1e-5;
    U 1e-5;
}
```

Для улучшения устойчивости и ускорения сходимости решения применяется метод нижней релаксации. Его коэффициенты<sup>38</sup> задаются в следующем словаре:

```
relaxationFactors
{
    fields
    {
        p 0.5;
    }
    equations
    {
        U 0.7;
    }
}
```

### 3.9. Генерация расчетной сетки

Перед началом моделирования необходимо сгенерировать расчётную сетку. Для этого нужно в командной строке набрать *blockMesh* и нажать клавишу *<Enter>*.

### 3.10. Расчёт

Для того, чтобы запустить процесс расчёта, необходимо в командной строке набрать *simpleFoam* и нажать *<Enter>*. Работа программы может занять достаточно продолжительное время, от пяти минут и более.

### 3.11. Обработка результатов

В графическом постпроцессоре *paraFoam* для построения поля скоростей или давлений лучше всего выбрать некоторое сечение *Slice* нажав на пиктограмму . Построение скалярного поля модулей вектора скорости в среднем сечении расчетной области вдоль оси канала должно привести к результату, подобному показанному на рис. 3.12.

---

<sup>38</sup> если не задавать соответствующие коэффициенты, то метод нижней релаксации использоваться не будет.

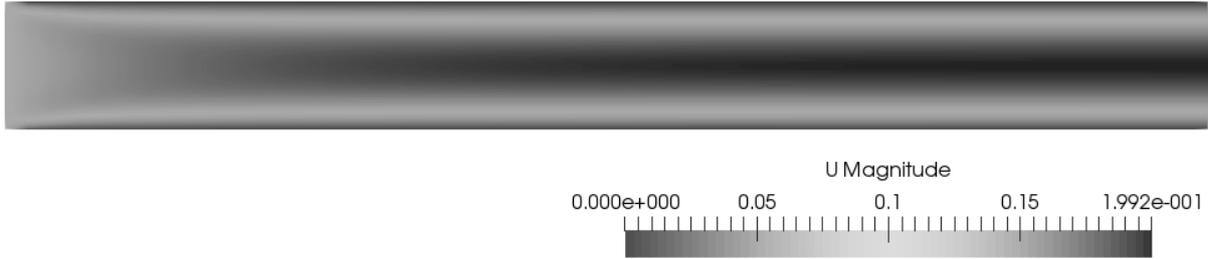


Рис. 3.12. Поле модуля вектора скорости

Кроме прочего, определённый интерес может представлять изменение того или иного параметра в пространстве или во времени. Для того чтобы извлечь необходимую информацию из результатов расчётов необходимо создать словарь `[system\sampleDict]`, в котором определить схему интерполяции

```
interpolationScheme cellPoint;
```

например, по ячейкам. Формат данных

```
setFormat raw;
```

скалярное или векторное, как в нашем случае, поле значений зависимой переменной

```
fields ( U );
```

и набор линий вдоль которых нужно определить значения зависимых переменных

```
type sets;
```

Сами линии вводятся в виде словаря `sets (...);`, содержащего информацию о сечениях для построения соответствующих характеристик. В нашем случае, нас интересует эпюра скоростей на выходе из канала:

```
outlet
{
    type uniform;
    axis xyz;
    start (0 0 0.14);
    end (0.0075 0 0.14);
    nPoints 20;
}
```

состоящая из значений в двадцати равноотстоящих (`uniform`) точках, лежащих на радиусе окружности в направлении оси  $x$  от центра к периферии в выходном сечении канала.

Вторая интересующая нас область лежит вдоль оси канала и также состоит из двадцати равноотстоящих точек:

```
axis
{
    type uniform;
    axis xyz;
    start (0 0 0);
    end (0 0 0.14);
    nPoints 20;
}
```

После создания словаря нужно в командной строке набрать *postProcess -func sampleDict* и нажать *<Enter>*.

В результате, будет сформирована папка [*postProcessing*] в которой, для каждого шага по времени, будут размещены файлы с названиями, составленными из названия сечения и названия зависимой переменной. Например, для нашего случая<sup>39</sup>, это будут файлы [*postProcessing\sampleDict\449\axis\_U*] для кривой описывающей изменение скорости вдоль оси канала, и [*postProcessing\sampleDict\449\outlet\_U*] для эпюры скоростей потока на выходе из канала.

Ниже приведен фрагмент файла [*postProcessing\sampleDict\449\outlet\_U*].

```
0.0003 0 0.14 -8.92719e-007 1.07045e-011 0.198644
0.0007 0 0.14 -1.73248e-006 1.18773e-011 0.197005
0.0011 0 0.14 -2.47142e-006 1.28184e-011 0.194275
0.0015 0 0.14 -3.07102e-006 1.62266e-011 0.190458
0.0019 0 0.14 -3.50553e-006 1.99932e-011 0.185562
0.0023 0 0.14 -3.75295e-006 2.1702e-011 0.179538
0.0027 0 0.14 -3.81064e-006 2.54946e-011 0.172362
0.0031 0 0.14 -3.71614e-006 3.59588e-011 0.164193
0.0035 0 0.14 -3.48930e-006 5.43502e-011 0.155005
0.0039 0 0.14 -3.14684e-006 3.54845e-011 0.144341
0.0043 0 0.14 -2.69764e-006 1.91838e-011 0.133302
0.0047 0 0.14 -2.19792e-006 9.75997e-012 0.120758
0.0051 0 0.14 -1.68926e-006 1.33612e-012 0.107016
0.0055 0 0.14 -1.20433e-006 -4.57318e-012 0.092145
0.0059 0 0.14 -7.78809e-007 -6.85903e-012 0.076077
```

---

<sup>39</sup> имя папки с номером шага по времени приведено в качестве примера, реальное имя будет отличаться в зависимости от ряда факторов

0.0063	0	0.14	-4.35327e-007	-7.81811e-012	0.058842
0.0067	0	0.14	-1.90373e-007	-9.21257e-012	0.040417
0.0071	0	0.14	-4.71736e-008	-6.23598e-012	0.020813
0.0075	0	0.14	-7.55159e-011	-1.38339e-012	0.000871

Файл состоит из строк, каждая из которых описывает значение зависимой переменной в точке пространства с координатами  $x$ ,  $y$  и  $z$  приведёнными в первых трёх столбцах, соответственно. Если бы мы исследовали скалярное поле, например давления ( $p$ ), то четвёртый столбец, содержащий значение зависимой переменной, был бы последним. Но, т. к. нас интересует эпюра скоростей ( $U$ ), а скорость – величина векторная, то в последних трёх столбцах приведены компоненты вектора скорости  $U_x$ ,  $U_y$  и  $U_z$  вдоль осей координат  $x$ ,  $y$  и  $z$ , соответственно. Найти модуль вектора скорости можно воспользовавшись известным выражением

$$U = \sqrt{U_x^2 + U_y^2 + U_z^2}.$$

Для течения в прямом канале круглого сечения компоненты вектора скорости  $U_x$  и  $U_y$  настолько малы<sup>40</sup>, что их можно принять равными нулю. Тогда значение в последнем, шестом, столбце и можно считать величиной модуля вектора скорости.

Полученное решение интересно сравнить с аналитическим [5, стр. 361].

Построение эпюры относительной скорости в радиальном направлении и кривой изменения скорости вдоль оси канала можно осуществить по выражению, полученному в результате аналитического решения уравнений ламинарного движения вязкой несжимаемой жидкости

$$\frac{u(x, r)}{U} = 2 \left( 1 - \frac{r^2}{a^2} \right) - 4 \sum_{m=1}^{\infty} \frac{1}{\gamma_m^2} \left[ 1 - \frac{J_0 \left( r \frac{\gamma_m}{a} \right)}{J_0(\gamma_m)} \right] e^{\frac{\gamma_m^2 v}{U a^2} x},$$

---

<sup>40</sup> их возникновение связано с влиянием ошибок округления на результаты итерационного процесса решения.

где  $a$  – радиус канала, м;  $J_0$  – функция Бесселя нулевого порядка;  $\gamma_1 = 5,136$ ,  $\gamma_2 = 8,417$ ,  $\gamma_3 = 11,620$ ,  $\gamma_4 = 14,796$ ,  $\gamma_5 = 17,960$ .

### Задание

1. Создайте проект *«laminarTube»* и проведите расчёт ламинарного стационарного течения в круглом трубопроводе с модифицированной комплексной расчётной сеткой (рис. 3.10). Проанализируйте результаты расчёта. Сравните полученные результаты с аналитическими. Сделайте выводы.

2. Для проекта п. 1 измените граничные условия, например `cylinderWall` с условия прилипания на условия непротекания (`slip`). Проанализируйте полученные результаты. Сравните их с результатами полученными в п. 1. Сделайте выводы.

3. Для проекта п. 1 измените начальные условия, например задайте начальное поле скоростей равное нулю<sup>41</sup>. Проведите расчёт. Сравните время, необходимое работы программы и результаты расчёта с полученными в п. 1. Сделайте выводы.

4. Попробуйте провести расчёты с простой расчётной сеткой с локальным измельчением ячеек и с комплексной расчётной сеткой. Сравните результаты расчётов между собой и с результатами, полученными в п. 1. Сделайте выводы.

5. Используя словарь *blockMeshDict* для параметрической расчётной сетки (рис. 3.11) проведите ряд расчётов с параметром  $\beta$ , изменяющим свою величину от 1/2 до 4/5. Проанализируйте полученные результаты. Оцените, как изменяется сходимость решения и его погрешность.

### Вопросы для самоконтроля

1. Для чего по оси канала должны быть расположены объёмы расчётных ячеек, а не их рёбра?

---

<sup>41</sup> `value uniform (0 0 0);` вместо `value uniform (0 0 0.1);`

2. Почему необходимо выполнять измельчение расчётной сетки вблизи стенок?

3. Каким образом можно задать криволинейное ребро блока расчётной сетки в виде дуги окружности?

4. Как можно сократить количество вводимой информации и снизить вероятность опечатки в словарях *OpenFOAM*?

5. Почему простейшая расчётная сетка (рис. 3.4), даже с локальным измельчением ячеек (рис. 3.6), либо совсем не должна применяться для моделирования течений, либо должна применяться с изрядной долей осторожности?

6. Каким образом осуществляется изменение размеров ячеек вдоль одной из осей локальной системы координат блока?

7. Как осуществить локальное измельчение ячеек вдоль оси локальной системы координат блока?

8. Как предотвратить запись промежуточных результатов расчёта?

9. Каким образом можно извлечь из результатов расчётов информацию об изменении зависимой переменной вдоль произвольного отрезка прямой внутри расчётной области?

10. Какова структура файла, описывающего изменение векторной величины вдоль отрезка прямой внутри расчётной области?

#### 4. ТЕПЛООБМЕН ПРИ ТУРБУЛЕНТНОМ ТЕЧЕНИИ ГАЗА

Исследование процесса теплообмена при течении жидкости в каналах является одной из практически важных задач не только в теплоэнергетике, но и в самых различных областях науки и техники. Для определения среднего коэффициента теплоотдачи от стенки канала к жидкости при турбулентном режиме течения может использоваться следующее уравнение подобия:

$$Nu = 0,021 Re^{0,8} Pr_{ж}^{0,43} \left( \frac{Pr_{ж}}{Pr_{ст}} \right)^{0,25}, \quad (4.1)$$

где  $Nu = \frac{\alpha l}{\lambda}$  – число Нуссельта;  $Re$  – число Рейнольдса;  $Pr_{ж}$  – число Прандтля, вычисленное для температуры жидкости на входе в канал;  $Pr_{ст}$  – число Прандтля, вычисленное для температуры жидкости, равной температуре стенки канала;  $\alpha$  – коэффициент теплоотдачи,  $m^2/c$ ;  $\lambda$  – коэффициент теплопроводности,  $Вт/(м \cdot К)$ ;  $l$  – характерный размер, равный диаметру канала круглого сечения, или гидравлическому диаметру – некруглого,  $m$ .

Уравнение (4.1) применимо в пределах  $Re = 1 \cdot 10^4 \dots 5 \cdot 10^6$  и  $Pr_{ж} = 0,6 \dots 2500$ .

Для воздуха выражение (4.1) может быть упрощено:

$$Nu = 0,018 Re^{0,8}. \quad (4.2)$$

В случае канала, относительная длина которого  $l/d < 50$ , вычисленное значение коэффициента теплоотдачи нужно умножить на величину поправочного коэффициента  $\varepsilon_l$  из табл. 4.1.

Таблица 4.1

Поправочный коэффициент $\varepsilon_l$					
$l/d$	$Re = 1 \cdot 10^4$	$2 \cdot 10^4$	$5 \cdot 10^4$	$1 \cdot 10^5$	$1 \cdot 10^6$
1	1,65	1,51	1,34	1,28	1,14
2	1,50	1,40	1,27	1,22	1,11
5	1,34	1,27	1,18	1,15	1,08
10	1,23	1,18	1,13	1,10	1,05
15	1,17	1,13	1,10	1,08	1,04
20	1,13	1,10	1,08	1,06	1,03
30	1,07	1,05	1,04	1,03	1,02
40	1,03	1,02	1,02	1,02	1,01
50	1,00	1,00	1,00	1,00	1,00

Для течения в изогнутых каналах (змеевиках) наблюдается интенсификация теплообмена в связи с перемешиванием жидкости благодаря вторичным вихрям, образующимся в поперечном сечении. Для учёта этого явления необходимо умножить полученное значение коэффициента теплоотдачи на поправочный коэффициент  $\epsilon_{зм}$ :

$$\epsilon_{зм} = 1 + 3,6 \frac{d}{D}, \quad (4.3)$$

где  $d$  – диаметр канала, м;  $D$  – диаметр спирали змеевика, м.

#### 4.1. Постановка задачи

Изучим теплообмен при вынужденной конвекции при течении вязкой сжимаемой жидкости в колене змеевика радиусом  $R = 25$  мм цилиндрического сечения радиусом  $r = 7,5$  мм (рис. 4.1).

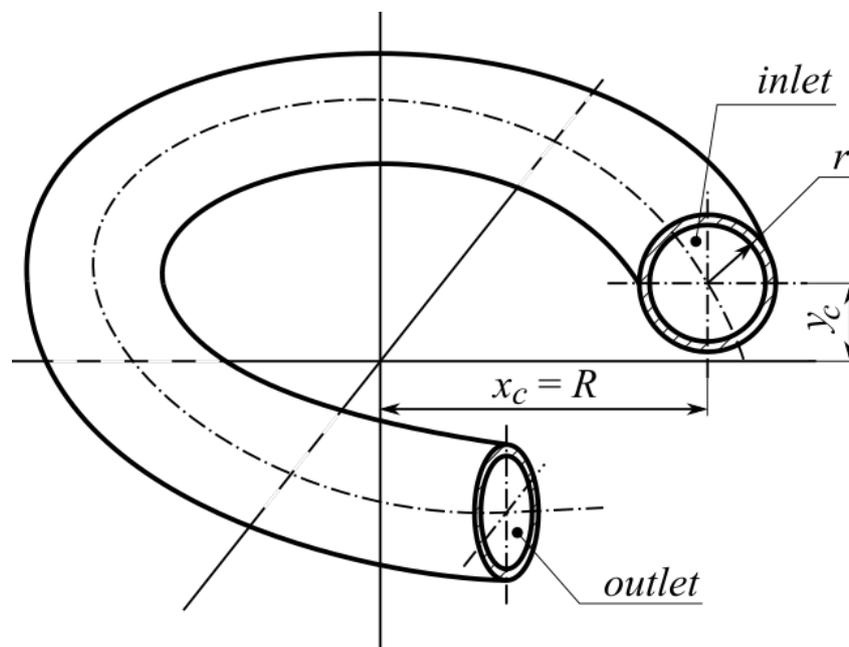


Рис. 4.1. Расчётная схема

Длина канала составит  $l = 2\pi R = 2 \cdot \pi \cdot 25 \text{ мм} \approx 157 \text{ мм}$ , а относительная длина  $l/(2r) = 157 \text{ мм}/15 \text{ мм} \approx 10,5$ . Скорость невозмущённого потока на входе в канал примем равной 50 м/с, тогда число Рейнольдса будет равно  $Re = 5 \cdot 10^4$ . На стенке канала применяется условие прилипания. В выходном сечении – условие бесконечности.

В качестве начальных условий применяются однородные поля скорости – равное 0 м/с, и давления – равного 100 кПа.

В качестве жидкости будет выступать воздух, обладающий<sup>42</sup> следующими свойствами: кинематическая вязкость  $\nu = 1,51 \cdot 10^{-5} \text{ м}^2/\text{с}$ ; коэффициент теплопроводности  $\lambda = 0,0259 \text{ Вт}/(\text{м} \cdot \text{К})$ ; плотность  $\rho = 1,204 \text{ кг}/\text{м}^3$ ; удельная массовая изобарная теплоёмкость  $c_p = 1,005 \text{ кДж}/(\text{кг} \cdot \text{К})$ ; число Прандтля  $Pr = 0,713$ .

Т. к. рассматриваемая задача имеет стационарный<sup>43</sup> характер – в уравнениях, описывающих движение вязкой несжимаемой жидкости, нестационарные члены, зависящие от времени  $t$ , такие как производные по времени  $\frac{\partial}{\partial t}$ , приравниваются нулю.

Уравнение неразрывности приобретает вид:

$$\nabla \cdot (\rho \mathbf{U}) = 0.$$

В качестве уравнения движения используется уравнение Навье-Стокса для стационарного турбулентного движения вязкой сжимаемой жидкости:

$$\nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \rho \mathbf{g} + \nabla \cdot [2\mu_{eff} D(\mathbf{U})] - \nabla \left[ \frac{2}{3} \mu_{eff} (\nabla \cdot \mathbf{U}) \right]$$

где  $p$  – статическое давление, Па;  $\mathbf{g}$  – вектор объемного ускорения, как правило, ускорения свободного падения, вызванного гравитацией,  $\text{м}/\text{с}^2$ ;  $\mu_{eff} = \mu + \mu_t$  – эффективная динамическая вязкость;  $\mu$  – вязкость;  $\mu_t$  – турбулентная вязкость;  $D(\mathbf{U}) = \frac{1}{2} [\nabla \mathbf{U} + (\nabla \mathbf{U})^T]$  – тензор скорости деформаций;  $-\nabla p + \rho \mathbf{g} = -\nabla p_{rgh} - (\mathbf{g} \cdot \mathbf{r}) \nabla \rho$ ;  $p_{rgh} = p - \rho \mathbf{g} \cdot \mathbf{r}$ ;  $\mathbf{r}$  – радиус-вектор.

В качестве зависимой переменной для уравнения сохранения энергии можно использовать либо удельную энтальпию  $h$ , либо удельную внутреннюю энергию  $e$ , связанные выражением  $h = e + \frac{p}{\rho}$ :

– удельная энтальпия:

$$\nabla \cdot (\rho \mathbf{U} h) + \nabla \cdot (\rho \mathbf{U} K) = \nabla \cdot (\alpha_{eff} \nabla h) + \rho \mathbf{U} \cdot \mathbf{g};$$

– удельная внутренняя энергия

$$\nabla \cdot (\rho \mathbf{U} e) + \nabla \cdot (\rho \mathbf{U} K) + \nabla \cdot (p \mathbf{U}) = \nabla \cdot (\alpha_{eff} \nabla e) + \rho \mathbf{U} \cdot \mathbf{g},$$

где  $K = \frac{|\mathbf{U}|^2}{2}$  – удельная кинетическая энергия потока;  $\alpha_{eff} = \frac{\rho \nu_t}{Pr_t} + \frac{\mu}{Pr} = \frac{\rho \nu_t}{Pr_t} + \frac{k}{c_p}$  – эффективный коэффициент температуропроводности<sup>44</sup>;

<sup>42</sup> при 20°C

<sup>43</sup> изучается установившееся движение жидкости

<sup>44</sup> в отечественной литературе –  $a$

$\lambda$  – коэффициент теплопроводности;  $c_p$  – удельная массовая изобарная теплоёмкость;  $\mu$  – динамическая вязкость;  $\nu_t = C_\mu \frac{k^2}{\varepsilon}$  – турбулентная кинематическая вязкость;  $Pr$  – число Прандтля;  $Pr_t$  – турбулентное число Прандтля;  $\frac{p}{\rho}$  – кинематическое давление.

Т. к. режим течения жидкости – турбулентный, для корректного расчёта поля течения необходимо дополнить систему уравнений математической моделью турбулентности. Используем хорошо известную, пусть и не лишённую определённых недостатков,  $k$ - $\varepsilon$  модель турбулентности<sup>45</sup>:

$$\frac{D}{Dt}(\rho k) = \nabla \cdot (\rho D_k \nabla k) + G_k - \frac{2}{3} \rho (\nabla \cdot \mathbf{U}) k - \rho \varepsilon + S_k;$$

$$\frac{D}{Dt}(\rho \varepsilon) = \nabla \cdot (\rho D_\varepsilon \nabla \varepsilon) + \frac{C_1 G_k \varepsilon}{k} - \left( \frac{2}{3} C_1 + C_3 \right) \rho (\nabla \cdot \mathbf{U}) k - C_2 \rho \frac{\varepsilon^2}{k} + S_\varepsilon$$

где  $C_\mu$ ,  $C_1$ ,  $C_2$ ,  $C_3$ ,  $\sigma_k$ ,  $\sigma_\varepsilon$  – коэффициенты<sup>46</sup>  $k$ - $\varepsilon$  модели турбулентности, табл. 4.2;

Таблица 4.2

Коэффициенты  $k$ - $\varepsilon$  модели турбулентности

$C_\mu$	$C_1$	$C_2$	$C_3$	$\sigma_k$	$\sigma_\varepsilon$
0,09	1,44	1,92	-0,33	1	1,3

Для моделирования конвективного теплообмена при установившемся турбулентном движении вязкой сжимаемой жидкости можно использовать солвер *buoyantSimpleFoam*<sup>47</sup>, в котором реализована математическая модель на основе описанных ДУЧП.

## 4.2. Расчётная сетка

Расчётную сетку будем строить на основе словаря [*system\blockMeshDict*] для параметрической расчётной сетки (рис.

<sup>45</sup> [www.openfoam.com/documetation/cpp-guide/html/guide-turbulence-ras-k-epsilon.html](http://www.openfoam.com/documetation/cpp-guide/html/guide-turbulence-ras-k-epsilon.html)

<sup>46</sup> величина коэффициента  $C_3$  ( $C_{3,RDT}$ ) взята из работы *S. H. El Tahry. k-epsilon equation for compressible reciprocating engine flows. 7(4):345-353, 1983*

<sup>47</sup> неустановившегося – *buoyantPimpleFoam*, несжимаемой жидкости – *buoyant-BoussinesqSimpleFoam* или *buoyantBoussinesqPimpleFoam*

3.11) при значении  $\beta = 2/3$ . Координаты<sup>48</sup> узлов и промежуточных точек начальной плоскости (при  $z = 0$ ) приведены в табл. 4.3.

Таблица 4.3

Координаты узлов и промежуточных точек, мм							
узел	$x$	$y$	$z$	точка	$x$	$y$	$z$
0	7,5	0,0	0,0	$a$	5,3033	5,3033	0,0000
1	0,0	7,5	0,0	$b$	-5,3033	5,3033	0,0000
2	-7,5	0,0	0,0	$c$	-5,3033	-5,3033	0,0000
3	0,0	-7,5	0,0	$d$	5,3033	-5,3033	0,0000
4	5,0	0,0	0,0	$e$	3,0936	0,0936	0,0000
5	0,0	5,0	0,0	$f$	-3,0936	3,0936	0,0000
6	-5,0	0,0	0,0	$g$	-3,0936	-3,0936	0,0000
7	0,0	-5,0	0,0	$h$	3,0936	-3,0936	0,0000

Так как *blockMesh* не может корректно обработать сильно деформированные блоки расчётной сетки, то область течения придётся разбить на несколько<sup>49</sup> частей. Соответственно, расчётная сетка будет генерироваться не из пяти, а из пятнадцати блоков. Для этого определим четыре плоскости, назовём их базовыми, на которых зададим расположение узлов блока расчётной сетки. Для прямого канала таких плоскостей нужно всего две. Каждая базовая плоскость будет повёрнута относительно другой на угол  $120^\circ$  с осью вращения параллельной оси  $u$  и проходящей через точку с координатами  $x_c = 25$  мм и  $z_c = 0$  мм.

Преобразования координат, или аффинные преобразования, могут быть заданы с помощью матриц перехода, таких как матрицы переноса и поворота<sup>50</sup>. Матрица переноса (перемещения) в трёхмерном пространстве имеет вид

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (4.4)$$

Матрица поворота в трёхмерном пространстве относительно оси  $u$  равна

<sup>48</sup> координаты в таблице приведены с точностью, недостаточной для построения расчётной сетки пригодной для практических расчётов

<sup>49</sup> минимум три

<sup>50</sup> Ещё одним видом матрицы перехода является матрица масштабирования, однако её использование в рассматриваемом случае нецелесообразно.

$$\mathbf{R} = \begin{pmatrix} \cos\gamma & 0 & \sin\gamma & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\gamma & 0 & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (4.5)$$

где  $\gamma$  – угол поворота базовой плоскости, °. Параллельный перенос системы координат осуществляется с помощью матрицы переноса (4.4) в которой значения  $\Delta x$ ,  $\Delta y$  и  $\Delta z$  равняются смещению начала старой системы координат относительно начала новой<sup>51</sup>. Для последовательных преобразований координат матрицы перехода можно перемножать

$$\mathbf{A} = \mathbf{C}_1 \times \mathbf{R} \times \mathbf{Y} \times \mathbf{C}_2, \quad (4.6)$$

где  $\mathbf{C}_1$  – матрица прямого переноса системы координат;  $\mathbf{R}$  – матрица поворота относительно новой системы координат;  $\mathbf{Y}$  – матрица переноса в новой системе координат;  $\mathbf{C}_2$  – матрица обратного переноса системы координат. В итоге, векторное произведение такой, результирующей, матрицы перехода на радиус-вектор точки

$$\mathbf{x}' = \mathbf{A} \times \mathbf{x} \quad (4.7)$$

позволит определить координаты узлов и промежуточных точек на базовых плоскостях.

Матрицы перехода в выражении (4.6), для рассматриваемого случая, записываются<sup>52</sup> в виде:

Прямой перенос системы координат:

$$\mathbf{C}_1 = \begin{pmatrix} 1 & 0 & 0 & -25 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Обратный перенос системы координат:

$$\mathbf{C}_2 = \begin{pmatrix} 1 & 0 & 0 & 25 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Поворот относительно оси  $y$  вычисляется с помощью матрицы поворота (4.5).

Смещение по оси  $y$ :

<sup>51</sup> Так, если осуществляется перенос системы координат из точки (0,0,0) в точку (1,0,0), то  $\Delta x = -1$ ,  $\Delta y = 0$  и  $\Delta z = 0$ .

<sup>52</sup> угловые размеры в °, линейные – в мм.

$$Y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -10,75 \frac{\gamma}{180^\circ} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

*NB.* Координаты узлов и промежуточных точек необходимо вводить в словарь `[system\blockMeshDict]` с максимальной точностью. В противном случае возможны искажения расчётной сетки, способные привести к недостоверности результатов расчёта.

Словарь `vertices`, полученный в результате вычислений по (4.6) выглядит следующим образом<sup>53</sup>:

```
vertices
(
  (7.5 0.0 0.0)
  (0.0 7.5 0.0)
  (-7.5 0.0 0.0)
  (0.0 -7.5 0.0)
  (5.0 0.0 0.0)
  (0.0 5.0 0.0)
  (-5.0 0.0 0.0)
  (0.0 -5.0 0.0)
  (33.75 -7.1666666666666666 15.155444566227677)
  (37.5 0.33333333333333339 21.65063509461097)
  (41.25 -7.1666666666666666 28.14582562299426)
  (37.5 -14.6666666666666666 21.65063509461097)
  (35.0 -7.1666666666666666 17.320508075688775)
  (37.5 -2.1666666666666666 21.65063509461097)
  (40.0 -7.1666666666666666 25.98076211353316)
  (37.5 -12.1666666666666666 21.65063509461097)
  (33.75 -14.333333333333332 -15.155444566227674)
  (37.5 -6.8333333333333332 -21.650635094610962)
  (41.25 -14.333333333333332 -28.145825622994252)
  (37.5 -21.833333333333332 -21.650635094610962)
  (35.0 -14.333333333333332 -17.32050807568877)
  (37.5 -9.3333333333333332 -21.650635094610962)
  (40.0 -14.333333333333332 -25.980762113533153)
  (37.5 -19.333333333333332 -21.650635094610962)
  (7.5 -21.5 -4.2862637970157365e-15)
  (0.0 -14.0 -6.123233995736766e-15)
  (-7.5 -21.5 -7.960204194457796e-15)
```

<sup>53</sup> некоторые координаты округлены для улучшения наглядности

```

(0.0 -29.0 -6.123233995736766e-15)
(5.0 -21.5 -4.898587196589413e-15)
(0.0 -16.5 -6.123233995736766e-15)
(-5.0 -21.5 -7.34788079488412e-15)
(0.0 -26.5 -6.123233995736766e-15)
);

```

**А построенные с использованием этих узлов блоки:**

```

blocks
(
  hex (0 1 5 4 8 9 13 12) (25 25 66) simpleGrading (1 10 1)
  hex (1 2 6 5 9 10 14 13) (25 25 66) simpleGrading (1 10 1)
  hex (2 3 7 6 10 11 15 14) (25 25 66) simpleGrading (1 10 1)
  hex (3 0 4 7 11 8 12 15) (25 25 66) simpleGrading (1 10 1)
  hex (4 5 6 7 12 13 14 15) (25 25 66) simpleGrading (1 1 1)
  hex (8 9 13 12 16 17 21 20) (25 25 66)
    simpleGrading (1 10 1)
  hex (9 10 14 13 17 18 22 21) (25 25 66)
    simpleGrading (1 10 1)
  hex (10 11 15 14 18 19 23 22) (25 25 66)
    simpleGrading (1 10 1)
  hex (11 8 12 15 19 16 20 23) (25 25 66)
    simpleGrading (1 10 1)
  hex (12 13 14 15 20 21 22 23) (25 25 66)
    simpleGrading (1 1 1)
  hex (16 17 21 20 24 25 29 28) (25 25 66)
    simpleGrading (1 10 1)
  hex (17 18 22 21 25 26 30 29) (25 25 66)
    simpleGrading (1 10 1)
  hex (18 19 23 22 26 27 31 30) (25 25 66)
    simpleGrading (1 10 1)
  hex (19 16 20 23 27 24 28 31) (25 25 66)
    simpleGrading (1 10 1)
  hex (20 21 22 23 28 29 30 31) (25 25 66)
    simpleGrading (1 1 1)
);

```

**Криволинейные рёбра описываются в словаре**

```

edges
(
  ...
)

```

К «поперечным» рёбрам в основании блока, описываемым дугами окружностей, добавляются «продольные» – в направлении оси  $z$ . Для них положение промежуточных точек вычисляется по (4.7) поворотом

узлов (табл. 4.3) на угол  $\gamma = 60^\circ, 180^\circ,$  и  $330^\circ$ . Соответствующая часть словаря `edges` будет выглядеть как<sup>54</sup>

```
arc 0 8 (16.25 -3.583333333333333 15.155444566227676)
arc 1 9 (12.5 3.916666666666667 21.650635094610966)
arc 2 10 (8.75 -3.583333333333333 28.145825622994256)
arc 3 11 (12.5 -11.083333333333332 21.650635094610966)
arc 4 12 (15.0 -3.583333333333333 17.32050807568877)
arc 5 13 (12.5 1.416666666666667 21.650635094610966)
arc 6 14 (10.0 -3.583333333333333 25.980762113533157)
arc 7 15 (12.5 -8.583333333333332 21.650635094610966)
arc 8 16 (42.5 -10.75 0.0)
arc 9 17 (50.0 -3.25 0.0)
arc 10 18 (57.5 -10.75 0.0)
arc 11 19 (50.0 -18.25 0.0)
arc 12 20 (45.0 -10.75 0.0)
arc 13 21 (50.0 -5.75 0.0)
arc 14 22 (55.0 -10.75 0.0)
arc 15 23 (50.0 -15.75 0.0)
arc 16 24 (16.25 -17.916666666666668 -15.155444566227676)
arc 17 25 (12.5 -10.416666666666668 -21.650635094610966)
arc 18 26 (8.75 -17.916666666666668 -28.145825622994256)
arc 19 27 (12.5 -25.416666666666668 -21.650635094610966)
arc 20 28 (15.0 -17.916666666666668 -17.32050807568877)
arc 21 29 (12.5 -12.916666666666668 -21.650635094610966)
arc 22 30 (10.0 -17.916666666666668 -25.980762113533157)
arc 23 31 (12.5 -22.916666666666668 -21.650635094610966)
```

Для построения «поперечных» рёбер необходимо, с помощью (4.7), выполнить поворот и смещение промежуточных точек  $a...h$  до всех базовых плоскостей. Соответствующая часть словаря `edges` будет выглядеть следующим образом<sup>55</sup>:

```
arc 0 1 (5.303300858899107 5.303300858899107 0.0)
arc 1 2 (-5.303300858899107 5.303300858899107 0.0)
arc 2 3 (-5.303300858899107 -5.303300858899107 0.0)
arc 3 0 (5.303300858899107 -5.303300858899107 0.0)
arc 4 5 (3.0935921676911455 3.093592167691145 0.0)
arc 5 6 (-3.0935921676911455 3.093592167691145 0.0)
arc 6 7 (-3.0935921676911455 -3.093592167691145 0.0)
arc 7 4 (3.0935921676911455 -3.093592167691145 0.0)
arc 8 9
    (34.84834957055044 -1.8633658077675594 17.05784182689251)
arc 9 10
    (40.151650429449546 -1.8633658077675594 26.24342836232943)
```

<sup>54</sup> некоторые координаты округлены для улучшения читаемости

<sup>55</sup> некоторые координаты округлены для улучшения читаемости

```

arc 10 11
    (40.151650429449546 -12.469967525565773 26.24342836232943)
arc 11 8
    (34.84834957055044 -12.469967525565773 17.05784182689251)
arc 12 13
    (35.95320391615442 -4.0730744989755205 18.971505688441866)
arc 13 14
    (39.046796083845564 -4.0730744989755205 24.32976450078007)
arc 14 15
    (39.046796083845564 -10.260258834357812 24.32976450078007)
arc 15 12
    (35.95320391615442 -10.260258834357812 18.971505688441866)
arc 16 17
    (34.848349570550454 -9.030032474434226 -17.057841826893)
arc 17 18 (40.15165042944957 -9.030032474434226 -26.24342836233)
arc 18 19 (40.15165042944957 -19.63663419223244 -26.24342836233)
arc 19 16
    (34.848349570550454 -19.63663419223244 -17.057841826893)
arc 20 21
    (35.953203916154436 -11.239741165642187 -18.97150568844)
arc 21 22
    (39.046796083845585 -11.239741165642187 -24.32976450078)
arc 22 23
    (39.046796083845585 -17.426925501024478 -24.32976450078)
arc 23 20
    (35.953203916154436 -17.426925501024478 -18.97150568844)
arc 24 25 (5.303300858899107 -16.196699141100893 0.0)
arc 25 26 (-5.303300858899107 -16.196699141100893 0.0)
arc 26 27 (-5.303300858899107 -26.803300858899107 0.0)
arc 27 24 (5.303300858899107 -26.803300858899107 0.0)
arc 28 29 (3.0935921676911455 -18.406407832308854 0.0)
arc 29 30 (-3.0935921676911455 -18.406407832308854 0.0)
arc 30 31 (-3.0935921676911455 -24.593592167691146 0.0)
arc 31 28 (3.0935921676911455 -24.593592167691146 0.0)

```

**Грани блоков расчётной сетки, на которых применяются те или иные граничные условия, описаны в словаре**

```

boundary
(
    ...
)

```

Всего будем применять три типа граничных условий: для входного сечения колена змеевика (`inlet`), для выходного сечения (`outlet`) и на твёрдой стенке (`cylinderWall`). Принципы описания граней не отличаются от применяемых ранее для построения других расчётных сеток.

К входному сечению относятся грани блоков, образованные узлами 0...7 на первой базовой плоскости. Описание границы находится в словаре `inlet`:

```
inlet
{
    type patch;
    faces
    (
        (0 1 5 4)
        (1 2 6 5)
        (2 3 7 6)
        (3 0 4 7)
        (4 5 6 7)
    );
}
```

Стенки канала образованы «*продольными*» внешними гранями, такими как, например, (0 1 9 8) и т. п. Описание границы находится в словаре `cylinderWall`:

```
cylinderWall
{
    type wall;
    faces
    (
        (0 1 9 8)
        (1 2 10 9)
        (2 3 11 10)
        (3 0 8 11)
        (8 9 17 16)
        (9 10 18 17)
        (10 11 19 18)
        (11 8 16 19)
        (16 17 25 24)
        (17 18 26 25)
        (18 19 27 26)
        (19 16 24 27)
    );
}
```

Выходное сечение канала образовано гранями, лежащими на последней базовой плоскости (узлы 24...31). Описание границы находится в словаре `outlet`:

```
outlet
{
    type patch;
    faces
    (
        (24 25 29 28)
        (25 26 30 29)
        (26 27 31 30)
        (27 24 28 31)
        (28 29 30 31)
    );
}
```

*NB.* В случае «ручного» ввода данных есть возможность внести произвольные искажения в расчётную сетку из-за опечаток, ошибок округления и т. п. В целях уменьшения вероятности подобных ошибок можно создавать скрипты для формирования исходных данных на языках программирования, подобных *Python* (см. приложение).

Расчётная сетка, построенная с использованием данных из словаря [*system\blockMeshDict*] показана на рис. 4.2.

Анализ построенной расчётной сетки показывает, что искажения формы и размеров ячеек на границах блоков минимальны, однако присутствует несущественное отклонение формы канала от окружности в поперечном сечении. Такое искажение, связанное с погрешностью округления промежуточных вычислений, достаточно невелико и не должно оказать существенного влияния на результаты расчёта.

Для автоматической генерации словаря [*system\blockMeshDict*] можно воспользоваться скриптом на языке *Python* – *blockMeshDictGen.py* (см. приложение). Для его работы необходим *Python* версии 3.5 и выше с установленным пакетом *NumPy*. Запуск скрипта осуществляется командой `python blockMeshDictGen.py > ./system/blockMeshDict`, после чего словарь с данными для построения расчётной сетки сформируется в папке [*system*].

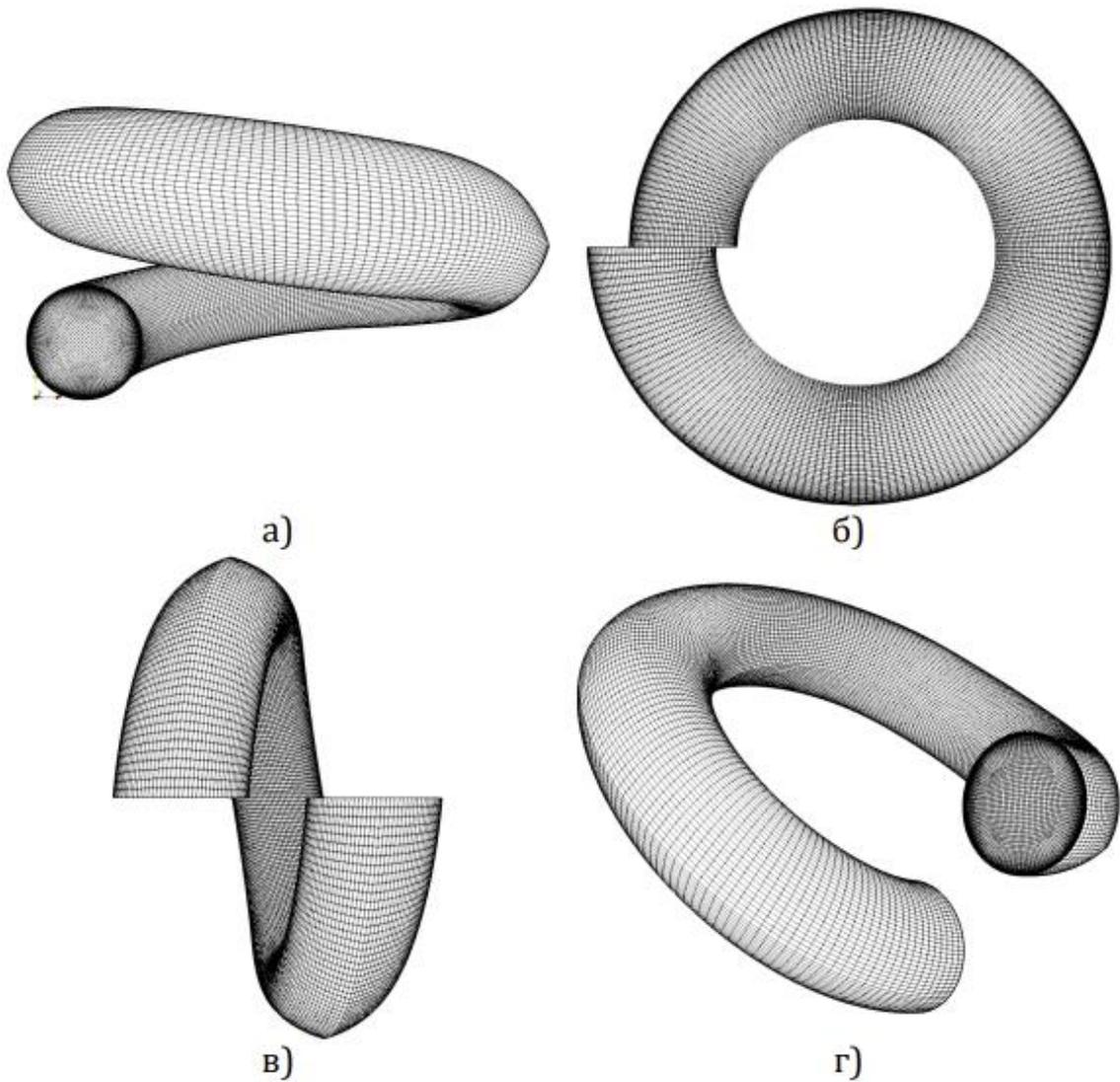


Рис. 4.2. Расчётная сетка: а) вид в направлении оси  $z$ , выходное сечение; б) вид в направлении оси  $y$ ; в) вид в направлении оси  $x$ , входное сечение справа, выходное – слева; г) общий вид, входное сечение

### 4.3. Условия однозначности

**Параметры турбулентности.** Интенсивность турбулентности для течения в канале [9]:

$$I = 0,16Re^{-\frac{1}{8}}.$$

Масштаб турбулентности вычисляется [11] по формуле

$$l = C_\mu \frac{k^{\frac{3}{2}}}{\varepsilon}, \text{ или } l = C_\mu^{\frac{3}{4}} \frac{k^{\frac{3}{2}}}{\varepsilon},$$

где  $C_\mu = 0,09$  для  $k$ - $\varepsilon$  модели турбулентности.

В случае течения в канале можно использовать выражение<sup>56</sup>

$$l = 0,038d_h,$$

где  $d_h = \frac{4F}{S}$  – гидравлический диаметр;  $F$  – площадь поперечного сечения канала;  $S$  – периметр поперечного сечения канала.

Теперь начальные и граничные условия для  $k$ – $\varepsilon$  модели турбулентности можно рассчитать по следующим выражениям:

турбулентная кинематическая вязкость

$$\nu_t = \sqrt{\frac{3}{2}} U \cdot l \cdot l$$

кинетическая энергия турбулентности

$$k = \frac{3}{2} (U \cdot l)^2$$

диссипация кинетической энергии турбулентности

$$\varepsilon = C_\mu \frac{k^{\frac{3}{2}}}{l} \text{ или } \varepsilon = C_\mu^{\frac{3}{4}} \frac{k^{\frac{3}{2}}}{l}$$

скорость диссипации турбулентной кинетической энергии

$$\omega = \frac{\sqrt{k}}{l} \text{ или } \omega = C_\mu^{-\frac{1}{4}} \frac{\sqrt{k}}{l}$$

**Кинетическая энергия турбулентности.** Начальные и граничные условия для кинетической энергии турбулентности  $k$  находятся в словаре [0\k]. В качестве начального приближения выбирается равномерное поле  $k = 0,001$ .

```
internalField uniform 0.001;
```

В списке boundaryField задаются граничные условия:

– на входе, как величина, вычисляемая исходя из интенсивности кинетической энергии турбулентности:

```
inlet
{
    type turbulentIntensityKineticEnergyInlet;
    intensity 0.041376;
    value uniform 6.4211;
}
```

– на стенке, в виде пристеночной функции:

---

<sup>56</sup> иногда, при использовании понятия длины перемешивания (например, для каналов некруглого сечения), вместо постоянного коэффициента 0,038 применяют коэффициент 0,07

```
cylinderWall
{
    type kqRWallFunction;
    value uniform 0.001;
}
```

– в выходном сечении, в виде условия бесконечности:

```
outlet
{
    type zeroGradient;
}
```

**Диссипация кинетической энергии турбулентности.** Начальные и граничные условия для диссипации кинетической энергии турбулентности  $\epsilon$  содержатся в словаре [ $0\backslash\epsilon$ ]. В качестве начального приближения задано равномерное поле  $\epsilon = 1$ .

В списке `boundaryField` задаются граничные условия:

– на входе, как величина, вычисляемая исходя из масштаба турбулентности:

```
inlet
{
    type turbulentMixingLengthFrequencyInlet;
    mixingLength 0.00057;
    value uniform 2569.1047;
}
```

– на стенке, в виде пристеночной функции:

```
cylinderWall
{
    type epsilonWallFunction;
    value uniform 2569.1047;
}
```

– в выходном сечении, в виде условия бесконечности:

```
outlet
{
    type zeroGradient;
}
```

**Скорость.** Граничные условия и начальное поле скорости заданы в словаре [ $0\backslash U$ ]. В качестве начального приближения используется нулевое поле скорости. На входной границе задано равномерное поле скорости, в котором касательные<sup>57</sup> составляющие вектора скорости равны 0, а нормальная<sup>58</sup> – равна 50 м/с.

---

<sup>57</sup> В направлении осей  $x$  и  $y$ .

<sup>58</sup> В направлении оси  $z$ .

В качестве граничных условий на стенке заданы условия прилипания, а в выходном сечении – бесконечности.

**Давление.** Начальное приближение поля давления и граничные условия по давлению задаются в словарях  $[0\backslash p]$  и  $[0\backslash p\_rgh]$ . Во втором словаре задаётся поле давления за вычетом гидростатической силы  $p_{rgh} = p - \rho gh$ , что позволяет учесть наличие архимедовых сил, приводящих к возникновению естественной конвекции. Размерность давления:  $\text{Н/м}^2 = \text{кг} \cdot \text{м}^{-1} \cdot \text{с}^{-2}$ . Т. к. рассматривается конвективный теплообмен при вынужденном турбулентном течении жидкости, а также учитывая малый перепад высот входного и выходного сечения колена змеевика, естественной конвекцией в процессе теплообмена можно пренебречь. Соответственно, содержимое двух этих словарей будет идентичным. В качестве граничных условий на входе и на твёрдой стенке заданы условия бесконечности, а на выходе – постоянное давление, равное 100 кПа.

**Температура.** Начальное приближение и граничные условия для поля температуры хранятся в словаре  $[0\backslash T]$ . Единицей измерения температуры является К. В качестве начального приближения используется равномерное поле температуры  $T_0 = 293 \text{ К} = 20 \text{ }^\circ\text{C}$  Температура газа, поступающего в колено змеевика задаётся постоянной, и также равной 293 К:

```
inlet
{
    type fixedValue;
    value uniform 293;
}
```

а температура стенки равна 373 К:

```
cylinderWall
{
    type fixedValue;
    value uniform 373;
}
```

На выходе из колена змеевика задано условие бесконечности:

```
outlet
{
    type zeroGradient;
}
```

**Турбулентная вязкость.** Начальные и граничные условия для поля кинематической турбулентной вязкости  $\nu_t$  находятся в словаре

[0\mathit{nut}]. В качестве начального приближения задается нулевое поле  $v_t$ , на входной и выходной границах – вычисляемые значения, с начальным приближением, равным нулю:

```
inlet
{
    type calculated;
    value uniform 0;
}
outlet
{
    type calculated;
    value uniform 0;
}
```

На твердой стенке граничное условие задается в виде пристеночной функции, с начальным приближением, равным нулю:

```
cylinderWall
{
    type nutkWallFunction;
    value uniform 0;
}
```

**Коэффициент температуропроводности.** Начальное приближение и граничные условия для турбулентного коэффициента температуропроводности записываются в словаре [0\mathit{alphat}]. В качестве начального приближения задано нулевое поле  $\alpha_t$ . Во входном и выходном сечениях – вычисляемое значение, с начальным приближением, равным нулю:

```
"inlet|outlet"
{
    type calculated;
    value uniform 0;
}
```

На твердой стенке задано граничное условие специального вида:

```
cylinderWall
{
    type compressible::alphatWallFunction;
    mut mut
    Prt 0.95;
    value uniform 0;
}
```

с начальным приближением, равным нулю.

Значение турбулентного коэффициента температуропроводности вычисляется по формуле [12]

$$\alpha_t = \frac{\mu_t}{Pr_t},$$

где  $\mu_t$  – турбулентная вязкость<sup>59</sup>, в словаре граничных условий задаётся в виде вычисляемого значения  $\mu_t$ ;  $Pr_t$  – турбулентное число Прандтля, выбирается исходя из свойств жидкости, по умолчанию  $Pr_t = 0,85$ .

**Свойства жидкости.** Свойства жидкости, равно как и дополнительные величины, входящие в расчётные уравнения, задаются в словарях, находящихся в папке [*constant*].

В файле [*constant\thermophysicalProperties*] описаны физические свойства жидкости [7]. В нём, в словаре

```
thermoType
{
    ...
}
```

описываются:<sup>60</sup> тип рабочего тела `type` `heRhoThermo` – сжимаемое, базируется на изменении плотности, с моделированием теплопередачи; модель переноса `transport` `const` – подразумевающая неизменность значений динамической вязкости  $\mu$  и числа Прандтля  $Pr$ ; термодинамическую модель `thermo` `hConst` – предполагающую постоянные значения изобарной теплоемкости  $c_p$  и теплоты плавления<sup>61</sup>  $H_f$ ; уравнение состояния `equationOfState` `perfectGas` – в виде уравнения Клапейрона в форме  $\rho = \frac{p}{RT}$ ; выбор зависимой переменной в уравнении энергии `energy` `sensibleEnthalpy` – относительная энтальпия, не учитывающая энтальпию образования компонентов смеси; свойства компонентов смеси `specie` `specie` - словарь или ссылка на подсловарь.

Затем, в словаре

```
mixture
{
    ...
}
```

задаются конкретные величины, описывающие свойства жидкости.

В подсловаре `specie` вводится число молей `nMoles` и молекулярный вес `molWeight` газа или компонента смеси. В подсловаре `thermodynamics` – изобарная теплоёмкость  $c_p$  и теплота плавления  $h_f$ .

<sup>59</sup> кинематическая.

<sup>60</sup> Каждая строка оканчивается точкой с запятой – «;»

<sup>61</sup> *heat of fusion*

В подсловаре `transport` – динамическая вязкость `mu` и число Прандтля `Pr`.

В словаре [`constant\turbulenceProperties`] хранится информация об используемой в процессе расчёта модели турбулентности.

Для моделирования турбулентного течения, в этот словарь необходимо включить строку

```
simulationType RAS;
```

которая указывает, что будет использована модель турбулентности на основе осреднения напряжений Рейнольдса<sup>62</sup>. Затем, в словаре `RAS`, производится выбор модели турбулентности `RASModel kEpsilon` и указывается на необходимость её использования в процессе расчёта – `turbulence on`, а также включается вывод на экран коэффициентов используемой модели турбулентности в начале расчёта – `printCoeffs on`.

В словаре [`constant\g`] вводится информация об ускорении свободного падения, под действием которого наблюдается свободная конвекция. Т. к. в рассматриваемой задаче влиянием силы тяжести можно пренебречь, в этом словаре вводится величина вектора ускорения свободного падения, каждый из компонентов которого равен нулю.

#### 4.4. Параметры численной схемы

Используемые схемы аппроксимации производных и операторов сохраняются в словаре [`system\fvSchemes`]. Для вычисления дивергенции используется схема разностей против потока – `upwind`, обеспечивающая высокую эффективность вычислений и устойчивость решения.

#### 4.5. Параметры модуля решения

Параметры модуля решения систем линейных алгебраических уравнений в словаре [`system\fvSolution`] дополнены описанием солверов и их параметров для поля давления – `p_rgh`, энтальпии – `h`, и кинетической энергии турбулентности и скорости её диссипации – `"(k|epsilon)"`.

В словаре

```
SIMPLE
{
    ...
}
```

---

<sup>62</sup> *RAS – Renolds-averaged simulation.*

заданы величины, необходимые для работы алгоритма *SIMPLE* в процессе совместного решения уравнений движения и уравнения для поля давления.

Т. к. в процессе решения используется схема с нижней/верхней релаксацией, в словаре

```
relaxationFactors
{
    ...
}
```

задаются величины коэффициента релаксации для полей переменных *fields* и уравнений *equations*.

## 4.6. Обработка результатов

**Расчётный коэффициент теплоотдачи.** Расчётное значение коэффициента теплоотдачи определяется исходя из уравнения Ньютона–Рихмана:

$$q = \alpha \Delta T = \alpha (T_{\text{ст}} - T_{\text{ж}}).$$

**Определение плотности теплового потока.** Тепловой поток<sup>63</sup>  $W$  от стенки к движущейся жидкости можно определить, выполнив команду *buoyantSimpleFoam -postProcess -func wallHeatFlux*. Для того, чтобы рассчитать плотность теплового потока<sup>64</sup>  $q$ , необходимо полученную величину разделить на площадь поверхности теплообмена. В случае цилиндрической поверхности получим:

$$q = \frac{W}{F} = \frac{W}{\pi \cdot d \cdot l}$$

**Определение коэффициента теплоотдачи.** Для определения расчётного значения коэффициента теплоотдачи необходимо полученную величину плотности теплового потока, вычисленную по (4.8), разделить на температурный напор

---

<sup>63</sup> Вт = кг<sup>1</sup> · м<sup>2</sup> · с<sup>-3</sup>

<sup>64</sup> Дж/(м<sup>2</sup> · с) = Вт/м<sup>2</sup> = кг<sup>1</sup> · с<sup>-3</sup>

$$\alpha_p = \frac{q}{\Delta T}$$

**Теоретическое значение коэффициента теплоотдачи.** Т. к. жидкостью, участвующей в процессе конвективного теплообмена, является воздух, для определения числа Нуссельта используем уравнение подобия (4.2):

$$Nu = 0.018Re^{0.8} = 0.018 \cdot (5 \cdot 10^4)^{0.8} = 103.$$

Учитывая, что относительная длина канала  $< 50$ , а сам канал имеет изогнутую форму, полученное значение  $Nu$  необходимо умножить на поправочные коэффициенты, величина которых приведена в табл. 4.1 и вычисляется по уравнению (4.3):

$$Nu_{зм} = \varepsilon_l \varepsilon_{зм} Nu = 1,13 \cdot 2,08 \cdot 103 = 242.$$

Тогда, теоретическое значение коэффициента теплоотдачи определится как

$$\alpha_T = \frac{Nu_{зм} \lambda}{l} = \frac{242 \cdot 0,0259 \text{ Вт}/(\text{м} \cdot \text{К})}{0,015 \text{ м}} = 418 \text{ Вт}/(\text{м}^2 \cdot \text{К}).$$

### Задание

1. Создайте проект «*turbulentCoil*» и проведите расчёт конвективного теплообмена при вынужденном турбулентном течении воздуха в змеевике при  $Re = 50000$  и  $\Delta T = 10^\circ\text{C}$ ,  $50^\circ\text{C}$ ,  $100^\circ\text{C}$ . Вычислите расчётное значение коэффициента теплоотдачи. Сравните расчётное значение коэффициента теплоотдачи с теоретическим.

2. Для проекта п. 1 проанализируйте характер течения на удалении от входного сечения.

3. Проведите ряд расчётов для проекта из п. 1 при значениях  $Re = 10000$ ,  $20000$ ,  $30000$  и  $40000$ , и  $\Delta T = 10^\circ\text{C}$ ,  $50^\circ\text{C}$ ,  $100^\circ\text{C}$ . Сравните полученные результаты с теоретическими для тех же условий.

**NB.** Помните о необходимости корректного задания начальных и граничных условий для каждого случая.

## Вопросы для самоконтроля

1. Каким образом вычисляется теоретическое значение числа Нуссельта для конвективного теплообмена при вынужденном турбулентном движении жидкости в канале?
2. Каким образом вычисляется теоретическое значение числа Нуссельта для конвективного теплообмена при вынужденном турбулентном движении воздуха в канале?
3. Как определяется поправочный коэффициент в случае конвективного теплообмена при вынужденном турбулентном движении жидкости в канале относительная длина которого  $< 50$ ?
4. Как определяется поправочный коэффициент в случае конвективного теплообмена при вынужденном турбулентном движении жидкости в изогнутом канале?
5. Каким образом, зная число Нуссельта, можно вычислить теоретическое значение коэффициента теплообмена?
6. Каким образом можно определить тепловой поток от стенки к газу по результатам расчёта в *OpenFOAM*?
7. Как вычисляется плотность теплового потока от стенки к газу по результатам расчёта в *OpenFOAM*?
8. Каким образом вычисляется расчётное значение коэффициента теплоотдачи?
9. Чем можно объяснить разницу в значениях расчётного и теоретического коэффициентов теплоотдачи?
10. Опишите характер течения в поперечном сечении изогнутого канала.

## ЗАКЛЮЧЕНИЕ

В учебном пособии был описан процесс установки программного комплекса *OpenFOAM* и даны рекомендации по его настройке. Рассмотрены задачи расчёта параметров ламинарных и турбулентных потоков сжимаемой и несжимаемой жидкости. Детально рассмотрено решение задачи конвективного теплообмена. Приведены рекомендации по построению расчётных сеток при моделировании течений в каналах, в том числе теплоэнергетических установок, рекомендации по анализу результатов расчётных исследований течений жидкостей и газов могут применяться в процессе обучения, в частности при курсовом и дипломном проектировании.

Приведённые в пособии сведения должны послужить базой для дальнейшего изучения методов математического моделирования гидрогазодинамических явлений в области энергетического машиностроения.

При подготовке исходных данных, проведении расчётных исследований и анализе их результатов необходимо использовать современные пакеты прикладных программ как общего, так и специального назначения, дополняющих своей функциональностью возможности программного комплекса *OpenFOAM*. Предполагается, что сведения относительно возможностей и особенностей работы с такими пакетами прикладных программ имеются в документации и методических указаниях по их использованию.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. OpenFOAM for Windows // CFDsupport : [сайт]. – 2023. – URL <https://www.cfdsupport.com/install-openfoam-for-windows.html> (дата обращения: 19.07.2023).
2. Tomislav Marić, Jens Höpken, Kyle Mooney. The Technology Primer. – sourceflux, 2014. – 442 p. – ISBN 978-3-00-046757-8.
3. Справочное руководство пакета программ OpenFOAM для вычислительной гидродинамики с лицензией открытого исходного кода. Руководство пользователя. Версия 1.7.1 (Перевод на русский). – openfoam.org, 2010. – 71 с.
4. Стационарное ламинарное течение в трубе // Блог Загоскин А. А. : [сайт]. – 2015. – URL <http://azagoskin.ru/?p=195> (дата обращения: 19.07.2023).
5. Слѣзкин Н. А. Динамика несжимаемой жидкости. – М. : Гос. изд-во техн.-теор. лит-ры, 1955. – 519 с.
6. Cylinder mesh using blockMesh with m4 macro // Multiphase and Complex Fluids Flow Lab at California State University, Long Beach : [сайт]. – 2016. – URL <https://www.ehsanmadadi.com/cylinder-mesh/> (дата обращения: 19.07.2023).
7. Christopher J. Greenshields. OpenFOAM User Guide version 4.0. – The OpenFOAM Foundation Ltd. – 2016. – 228 p.
8. Патанкар, С. Численные методы решения задач теплообмена и динамики жидкости / пер. с англ. под ред. В. Д. Виоленского. – М. : Энергоатомиздат, 1984. – 124 с. ил.
9. Turbulence intensity // CFD Online : [сайт]. – 2022. – URL [https://www.cfd-online.com/Wiki/Turbulence\\_intensity](https://www.cfd-online.com/Wiki/Turbulence_intensity) (дата обращения 19.07.2023).
10. Turbulence length scale // CFD Online : [сайт]. – 2012. – URL [https://www.cfd-online.com/Wiki/Turbulence\\_length\\_scale](https://www.cfd-online.com/Wiki/Turbulence_length_scale) (дата обращения: 19.07.2023).
11. Turbulence free-stream boundary conditions // CFD Online : [сайт]. – 2014. – URL [https://www.cfd-online.com/Wiki/Turbulence\\_free-stream\\_boundary\\_conditions](https://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions) (дата обращения: 19.07.2023).
12. Convective Heat Transfer Analysis // SIMSCALE : [сайт]. – 2022. – URL <https://www.simscale.com/docs/analysis-types/convective-heat-transfer-analysis/> (дата обращения: 19.07.2023).
13. Турбулентное течение в трубе. Конвективный теплообмен // Блог Загоскин А. А. : [сайт]. – 2015. – URL <http://azagoskin.ru/?p=329> (дата обращения: 19.07.2023).

## ПРИЛОЖЕНИЕ

### Скрипт генерации блочной расчетной сетки (*blockMeshDictGen.py*)

```
from math import pi, sin, cos, tan, radians, atan, asin, acos, degrees
import numpy as np

def rotate_section_by_arbitrary_angle(angle, R, center, section):
    gamma = radians(angle)
    Xc, Yc, Zc = center
    x, y, z = section
    xlc = Xc - R * cos(gamma)
    ylc = y + Yc * gamma / pi
    zlc = Zc + R * sin(gamma)
    return (xlc, ylc, zlc)

r1 = 7.5      # радиус канала
Xc = 25      # координаты
Yc = -10.75  # центра образующей
Zc = 0       # окуржности
base_plane_count = 4          # число базовых плоскостей (= число блоков +
1)
intermediate_planes_count = 1 # число промежуточных точек (всегда = 1)
cell_count = 25

block_base_nodes = [ [0, 1, 5, 4], [1, 2, 6, 5], [2, 3, 7, 6], [3, 0, 4, 7],
[4, 5, 6, 7]]
cylinderWall_nodes = [ [0, 1], [1, 2], [2, 3], [3, 0] ]
arc_connected_nodes = [ 0, 1, 2, 3, 4, 5, 6, 7 ]
arc_connected_nodes_r1 = [
    [0, 1], [1, 2], [2, 3], [3, 0], [4, 5], [5, 6], [6, 7], [7, 4] ]
R = np.array([-1, 0, 1, 0, -2./3., 0, 2./3., 0]) * r1 + Xc
R1 = np.array([
    -1, 1, 1, -1,
    -2./3.*7./8., 2./3.*7./8.,
    2./3.*7./8., -2./3.*7./8.]) * r1 * cos(radians(45)) + Xc
x = np.array([1, 0, -1, 0, 2. / 3., 0, -2. / 3., 0]) * r1
y = np.array([0, 1, 0, -1, 0, 2. / 3., 0, -2. / 3.]) * r1
z = np.array([0, 0, 0, 0, 0, 0, 0, 0])

print(
    """FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}

convertToMeters 0.001;

""")
plane_nodes_count = len(x)
base_plane_angle_increment = 360 / (base_plane_count - 1)
```

```

intermediate_plane_angle_increment = base_plane_angle_increment /\
    (intermediate_planes_count + 1)

base_plane_angles = np.array( np.arange(0, 361, base_plane_angle_increment)
)
base_planes = [
    rotate_section_by_arbitrary_angle(
        a, R, (Xc, Yc, Zc), (x, y, z)) for a in base_plane_angles ]
print('vertices\n(')
for base_plane in base_planes:
    print(
        '\n'.join(
            ['({} {} {})].format(
                base_plane[0][pt],
                base_plane[1][pt],
                base_plane[2][pt]
            ) for pt in range(plane_nodes_count)])
    )
print(');')
x_cell_count, y_cell_count, z_cell_count =\
    cell_count, cell_count, int(cell_count * 8 / (base_plane_count - 1))
print('blocks\n(')
for plane in range(base_plane_count - 1):
    for nodes in block_base_nodes:
        if nodes != block_base_nodes[-1]:
            y_edge_grading = 10
        else:
            y_edge_grading = 1
        print(
            'hex ({} {} {} {} {} {} {} {} {})) ({} {} {})) simpleGrading (1 {} 1)'.format(
                *[n + plane * plane_nodes_count for n in nodes],
                *[n + (plane + 1) * plane_nodes_count for n in nodes],
                x_cell_count, y_cell_count, z_cell_count,
                y_edge_grading
            )
        )
    )
print(');')
intermediate_plane_angles = np.array(
    np.arange(intermediate_plane_angle_increment,
        2. * degrees(pi) / (base_plane_count - 1),
        intermediate_plane_angle_increment
    )
)
intermediate_planes = []
for plane_angle in base_plane_angles[0:-1]:
    for a in intermediate_plane_angles + plane_angle:
        intermediate_planes.append(
            rotate_section_by_arbitrary_angle(a, R, (Xc, Yc, Zc), (x, y, z)))
print('edges\n(')
for plane in range(base_plane_count - 1):
    for node in arc_connected_nodes:
        print('arc {} {} ({} {} {})].format(
            node + plane * plane_nodes_count,
            node + (plane + 1) * plane_nodes_count,
            *[
                intermediate_planes[plane][coord][node]    for    coord    in
            range(3)

```

```

    ]))
ptx1 = np.array([
    1, -1, -1, 1, 2./3.*7./8., -2./3.*7./8., -2./3.*7./8., 2./3.*7./8.
    ]) * cos(radians(45)) * r1
pty1 = np.array([
    1, 1, -1, -1, 2./3.*7./8., 2./3.*7./8., -2./3.*7./8., -2./3.*7./8.
    ]) * sin(radians(45)) * r1
ptz1 = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0])
ptrs1 = [
    rotate_section_by_arbitrary_angle(
        a, R1, (Xc, Yc, Zc), (ptx1, pty1, ptz1
    ) for a in base_plane_angles]
for plane in range(base_plane_count):
    for i, nodes in enumerate(arc_connected_nodes_r1):
        print('arc {} {} ({} {} {})' .format(
            *[n + plane * plane_nodes_count for n in nodes],
            *[ptrs1[plane][coord][i] for coord in range(3)
        ]))
print(');')
inlet_outlet_nodes = block_base_nodes
print('boundary\n(')
print(' inlet\n {')
print(' type patch;\n faces\n (')
for nodes in inlet_outlet_nodes:
    print(' ({} {} {} {})' .format(*nodes))
print(' );')
print('}')
print(' cylinderWall\n {')
print(' type wall;\n faces\n (')
for plane in range(base_plane_count - 1):
    for nodes in cylinderWall_nodes:
        print(' ({} {} {} {})' .format(
            *[n + plane * plane_nodes_count for n in nodes],
            *[n + (plane + 1) * plane_nodes_count for n in reversed(nodes)]
        ))
print(' );')
print('}')
print(' outlet\n {')
print(' type patch;\n faces\n (')
for nodes in inlet_outlet_nodes:
    print(' ({} {} {} {})' .format(
        *[n + (base_plane_count - 1) * plane_nodes_count for n in nodes]))
print(' );')
print('}')
print(');')

```

*Учебное издание*

АБАЛЯЕВ Андрей Юрьевич  
ЛЮХТЕР Александр Борисович

ОСНОВЫ МОДЕЛИРОВАНИЯ ГАЗОДИНАМИЧЕСКИХ ПРОЦЕССОВ  
И КОНВЕКТИВНОГО ТЕПЛООБМЕНА В ПРОГРАММНОМ КОМПЛЕКСЕ  
*OpenFOAM*

Учебное пособие

*Издается в авторской редакции*

Подписано в печать 26.07.23.  
Формат 60×84/16. Усл. печ. л. 6,05. Тираж 50 экз.

Заказ

Издательство

Владимирского государственного университета  
имени Александра Григорьевича и Николая Григорьевича Столетовых.  
600000, г. Владимир, ул. Горького, 87.