

Глава 1

Лабораторные работы

1.2. Лабораторная работа №2: Алгоритм PISO. Обтекание цилиндра.

В данной работе рассматривается решение нестационарной задачи обтекания цилиндра с помощью решателя PISO с использованием потенциального решения в качестве начальных условий для ускорения сходимости. Производится анализ получившейся вихревой картины при помощи пост-обработки и визуализации в ParaView.

Задача обтекания цилиндра

Задача обтекания цилиндра потоком идеальной несжимаемой жидкости является классической иллюстративной задачей гидродинамики. Рассматривается течение, перпендикулярное образующей, а сам цилиндр подразумевается бесконечной длины, тем самым задача становится двумерной. Такое течение является потенциальным и соответствующая система уравнений имеет аналитическое решение [6].

Вводя систему координат с началом в центре основания цилиндра и обозначая его радиус как R , потенциал скоростей может

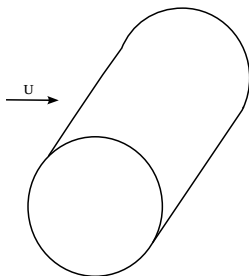


Рис. 1.1: Обтекание цилиндра

быть записан как:

$$\phi(x, y) = Ux \left(1 + \frac{R^2}{x^2 + y^2} \right). \quad (1.1)$$

Компоненты скорости \mathbf{U} могут быть найдены как частные производные по координатам, а поле давлений при помощи интеграла Бернулли. Линии тока возможно найти как постоянные значения функции тока:

$$\psi(x, y) = Uy \left(1 - \frac{R^2}{x^2 + y^2} \right) = \text{const}. \quad (1.2)$$

Исходя из данных соотношений нетрудно построить всю картину обтекания.

Построение расчетной сетки

При построении сетки в данной лабораторной работе необходимо учитывать криволинейные края блоков. Для этого разделим расчетную область на четырехугольные блоки, границы которых описываются прямыми или дугами окружности (рис. 1.2).

Теперь необходимо задать координаты вершин блоков. Чтобы облегчить определение точек, находящихся на окружности, воспользуемся языком макровыражений. Задавая ключевое слово, мы можем использовать его значение далее в конфигурационном файле, используя знак доллара (\$), например:

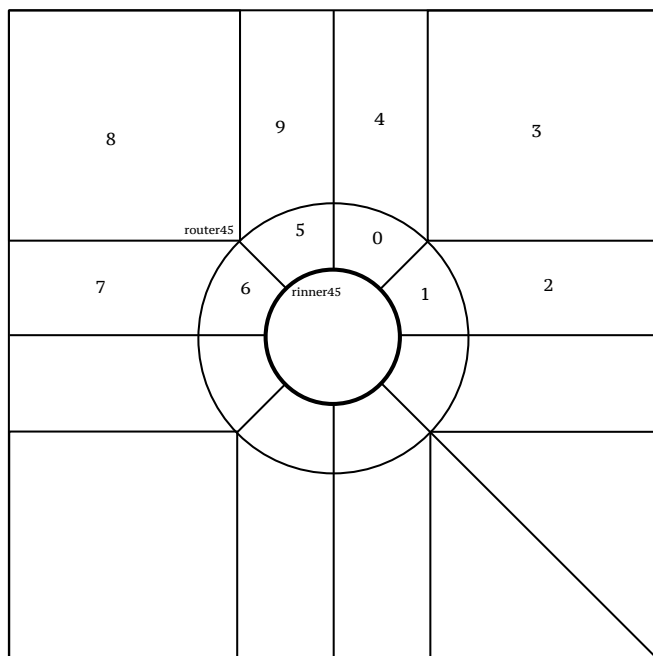


Рис. 1.2: Разбиение расчетной области на блоки **исправить картинку**

```

1  rinner 0.5; // внутренний радиус
2  zmin   -0.5;
3  xright 4;
4
5  vertices
6  (
7      ($rinner 0 $zmin)
8      ($xright 0 $zmin)

```

Это позволяет при изменении значения какого-то либо выражения исправлять лишь значение при ключевом слове. Например, чтобы изменить положение правой границы расчетной области достаточно поменять значение *xright*. Более того, в рамках конфигурационного файла возможно даже проводить несложные вычисления, используя директиву *eval*:

```
1 router 1; // внешний радиус
2 router45 #eval "$router * sqrt(0.5)";
```

Тем самым вычисляется значение, соответствующее длине проекции половины диагонали, вписанного в круг квадрата, на ортогональную ось. Это избавляет от необходимости предварительных расчетов и ручного пересчета при изменении каких-либо входных данных.

После этого можно задать соответствующие блоки, где количество расчетных ячеек по разным осям также задаются в виде ключевых слов:

```
1 hex (5 4 9 10 24 23 28 29) ($nradial $nquarter $nz)
2 simpleGrading (1 1 1)
```

Для задания криволинейных краев блоков для данного случая используются дуги окружностей, которые задаются в блоке *edges* согласно синтаксису:

```
1 arc 0 5 origin (0 0 $zmin)
```

Здесь *arc* обозначает дугу окружности, далее идут номера вершин, которые эта дуга соединяет и координаты центра после слова *origin*, так как три точки полностью задают окружность.

Отметим, что как и в предыдущей лабораторной работе, так как блоки трехмерны, все вершины и дуги должны быть дублированы для другого постоянного значения на пространственной оси.

Наконец, необходимо собрать из различных кусков границу. Для прямолинейных границ (нижней и верхней) с нормалью перпендикулярной оси симметрии поставим тип *symmetryPlane*, для криволинейной границы цилиндра — *wall*, входу и выходу потока соотнесем тип *patch*, а для границы по пространственной оси *z* — тип *empty*. Условие симметрии на верхней и нижней границе соответствует случаю неограниченной по вертикали области: нормальная компонента вектора величины равна нулю, а параллельная границе зеркально отражается. Итоговый конфигурационный файл размещен в листинге 1.3.

Как и в предыдущей работе, заданная сетка создается и проверяется вводом команд *blockMesh* и *checkMesh*, вывод которых тоже полезно сохранить, перенаправив в файл:

```
1 blockMesh | tee blockMesh.log
2 checkMesh | tee checkMesh.log
```

Для ускорения работы решателей линейных уравнений рекомендуется произвести перенумерацию ячеек сетки, чтобы понизить ширину ленты матрицы. Ширина лента матрицы A определяется как наименьшее целое число k , такое, что: $A(i, j) = 0$ для $|i - j| > k$. Это означает, что все ненулевые элементы находятся не далее чем в k позициях от диагонали.

Для этого необходимо применить команду:

```
1 renumberMesh -overwrite
```

Здесь ключ *overwrite* означает перезапись файла.

Граничные условия

На примере конфигурационного файла для скорости (листинг 1.4) можно видеть, что граничные условия соответствуют геометрии расчетной сетки: там где была задана симметрия, стоят соответствующие граничные условия. На выходе потока задано условие нулевой нормальной производной, а на входе условие горизонтального потока с постоянной скоростью. На цилиндре задано условие прилипания — равенство всех компонент скорости нулю.

На выходе потока ставится условие постоянного нулевого избыточного давления, что соответствует полностью развитому течению, а на входе и самом цилиндре — условие нулевой нормальной производной давления. Нулевое давление здесь допустимо, поскольку в уравнениях изотермического течения на вычисления оказывает влияние лишь изменение давления, нежели его значение. Так как рассматриваемый решатель предназначен для несжимаемого потока, для упрощения расчетов давление делится на постоянную плотность и поэтому имеет размерность $\text{м}^2/\text{с}^2$.

Решение методом потенциала

Для нахождения потенциального решения в OpenFOAM возможно воспользоваться решателем *potentialFoam*. Его можно использовать для того чтобы найти решение, а потом воспользо-

ваться другим более продвинутым решателем, используя полученные результаты как начальные данные для ускорения сходимости. Также данный решатель можно использовать для дополнительной проверки качества сетки и её чувствительности к градиентам: для хорошей сетки число итераций алгоритма должно уменьшаться со временем.

Решатель решает следующую систему в дискретизованном виде:

$$\nabla \cdot \mathbf{U} = 0, \quad (1.3)$$

$$\nabla^2 p = 0. \quad (1.4)$$

Для выбора и настройки данного решателя достаточно изменить параметр *application* в *controlDict* и задать подходящие *fvSolution*, *fvSchemes* (листинги 1.5, 1.6 и 1.7). Отметим, что здесь для решения системы линейных уравнений для потенциала скоростей и давления используется многосеточный метод (GAMG) с релаксацией методом неполного разложения Холецкого (DIC). В словаре *potentialFlow* определяется параметр решателя, количество циклов коррекции неортогональности (*nNonOrthogonalCorrectors*). В случае неортогональности сетки, которая обычно сопутствует криволинейной геометрии, возникает погрешность при подсчете скалярного произведения градиента скорости на нормаль к грани в дискретизированном диффузионном члене и необходима поправка на неортогональную часть нормали [4].

Запуск решения и анализ результатов

Для запуска решения воспользуемся командой:

```
1 potentialFoam -writep | tee potentialFoam.log
```

Как можно заметить, после выполнения команд не появляется новых временных каталогов, так как *potentialFoam* по своей сути стационарный решатель, свои результаты он перезаписывает в нулевое время. Аргумент команды *writep* обозначает перезапись не только начальной скорости, но и давления. Текстовый вывод команды выглядит следующим образом:

```
1 Calculating potential flow
2 GAMG: Solving for Phi, Initial residual = 5.77125e-05,
```

```

3 Final residual = 6.48403e-07, No Iterations 1
4 GAMG: Solving for Phi, Initial residual = 1.79131e-06,
5 Final residual = 9.37665e-08, No Iterations 1
6 Continuity error = 3.73305e-06
7 Interpolated velocity error = 1.15029e-05
8
9 Calculating approximate pressure field
10 GAMG: Solving for p, Initial residual = 7.3855e-05,
11 Final residual = 5.07864e-07, No Iterations 2
12 GAMG: Solving for p, Initial residual = 1.13124e-05,
13 Final residual = 6.81457e-07, No Iterations 1
14 ExecutionTime = 0.02 s ClockTime = 0 s

```

Здесь можно видеть количество итераций для изначального приближения и работу цикла коррекции неортогональности при вычислении полей давления и скорости.

Однако, потенциальное течение не особенно интересно для реальных задач, особенно если необходимо учитывать вязкость и нестационарные эффекты, такие как образование вихрей. Используя результаты потенциального решения как отправную точку, можно воспользоваться нестационарным решателем.

Описание алгоритма PISO

Одним из исторически первых универсальных и эффективных методов решения нестационарных проблем является алгоритм PISO (Pressure Implicit with Splitting of Operators) [1, 3]. Алгоритм предназначен для решения нестационарных уравнений гидродинамики для вязкой несжимаемой жидкости:

$$\nabla \cdot \mathbf{U} = 0, \quad (1.5)$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) = -\nabla p + \mu \Delta \mathbf{U}. \quad (1.6)$$

Алгоритм построен по схеме предиктор-корректор, его общая схема такова:

Шаг 1: На первом шаге происходит конечно-объемная дискретизация уравнений движения (1.6), обозначим получившуюся систему следующим образом:

$$\mathbf{M}\mathbf{U} = -\nabla p. \quad (1.7)$$

Данное уравнение возможно решить относительно скорости \mathbf{U} , используя давление, полученное на предыдущем шаге (для инициализации используются начальные и граничные значения). Это представляет собой *шаг предиктора*.

Шаг 2: Обозначим как \mathbf{A}_p диагональную часть матрицы \mathbf{M} и вычислим следующий оператор \mathbf{H} :

$$\mathbf{H} = \mathbf{A}_p^{-1} \mathbf{U} - \mathbf{M} \mathbf{U}. \quad (1.8)$$

Шаг 3: Теперь, подставляя скорость, полученную на шаге предиктора, найдем давление из следующего уравнения:

$$\nabla \cdot (\mathbf{A}_p^{-1} \nabla p) = \nabla \cdot (\mathbf{A}_p^{-1} \mathbf{H}). \quad (1.9)$$

Данный шаг повторяется в рамках цикла коррекции неортогональности, так как при дискретизации диффузионных членов используется поправка в столбец свободных членов [ссылка?](#).

Шаг 4: Полученное на предыдущем шаге давление используем для коррекции скорости:

$$\mathbf{U} = \mathbf{A}_p^{-1} \mathbf{H} - \mathbf{A}_p^{-1} \nabla p. \quad (1.10)$$

Так как такая скорость может больше не удовлетворять уравнению (1.7), будем повторять шаги 2-4 до сходимости. Это составляет *цикл коррекции*.

1.3. Параметры расчетной схемы

Для того, чтобы применить данный алгоритм в OpenFOAM, необходимо поменять название приложения в файле *controlDict* и отредактировать или заменить файлы *fvSchemes* 1.8 и *fvSolutions* 1.9 В блоке *solvers* файла *fvSolutions* описываются параметры решателей систем линейных уравнений для нахождения соответствующих переменных:

- *solver* — используемый алгоритм решения,
- *smoother* — метод сглаживания матрицы системы уравнений,
- *tolerance* и *relTol* — абсолютная и относительная разница между итерациями алгоритма.

В блоке *PISO* описаны параметры алгоритма:

- *nCorrectors* — число циклов коррекции,
- *nNonOrthogonalCorrectors* — число циклов коррекции неортогональности,
- *pRefCell*, *pRefValue* — фиксированное значение давления в одной ячейке, используются, если отсутствуют числовые граничные условия на давление.

Файл *fvSchemes* описывает схемы дискретизации для различных членов уравнения:

- *ddtSchemes*: дискретизация производной по времени $\partial/\partial t$; *default Euler* — используется явная разностная схема первого порядка;
- *gradSchemes*: дискретизация оператора градиента ∇ , значение *default Gauss linear* соответствует линейной интерполяции из центра ячейки на ребро с интегрированием по Гауссу;
- *divSchemes*: дискретизация оператора дивергенции $\nabla \cdot$, для поля скорости *div(phi,U)*:
 - *default none* — обозначает отсутствие схемы по умолчанию для всех членов,
 - *div(phi,U) Gauss limitedLinearV 1* — для дивергенции скорости используется схема с линейной интерполяцией и коэффициентом ограничителя 1. Буква V означает, что ограничитель для векторной величины рассчитывается в направлении наиболее сильно изменяющихся градиентов;
- *laplacianSchemes*: дискретизация лапласиана *default Gauss linear corrected* — линейная интерполяция с интегрированием по Гауссу с коррекцией неортогональности;
- *interpolationSchemes*: *default linear* — схема линейной интерполяции применяется по умолчанию для различных полей;
- *snGradSchemes*: дискретизация градиента по нормали: *default corrected* — схема расчета градиента скалярного поля вдоль нормали к поверхности с коррекцией неортогональности.

Расчет при различных числах Рейнольдса

Экспериментальные данные свидетельствуют, что качественное поведение потока за цилиндром зависит от числа Рейнольдса [2, 5]):

$$Re = \frac{\rho U d}{\mu} = \frac{U d}{\nu}, \quad (1.11)$$

где d — характерный размер (диаметр цилиндра), μ — динамическая вязкость, ν — кинематическая вязкость. Последняя как раз является тем параметром, который необходимо менять в файле *transportProperties* (листинг 1.10) для получения необходимых чисел Рейнольдса.

Для решения задачи теперь достаточно запустить расчет командой:

```
1  pisoFOAM
```

Это приведет к следующему текстовому выводу:

```
1  Time = 0.001
2  Courant Number mean: 0.0533538 max: 0.130023
3  smoothSolver: Solving for Ux, Initial residual = 0.000498407,
4  Final residual = 6.6746e-07, No Iterations 2
5  smoothSolver: Solving for Uy, Initial residual = 0.000914777,
6  Final residual = 1.18646e-06, No Iterations 2
7  GAMG: Solving for p, Initial residual = 0.323646,
8  Final residual = 0.0293731, No Iterations 5
9  GAMG: Solving for p, Initial residual = 0.007652,
10 Final residual = 0.0003832, No Iterations 5
11 GAMG: Solving for p, Initial residual = 0.00133851,
12 Final residual = 9.29803e-05, No Iterations 6
13 time step continuity errors : sum local = 3.50201e-09,
14 global = -2.50197e-10, cumulative = -2.50197e-10
15 GAMG: Solving for p, Initial residual = 0.00119567,
16 Final residual = 9.22534e-05, No Iterations 4
17 GAMG: Solving for p, Initial residual = 0.000220632,
18 Final residual = 1.80378e-05, No Iterations 4
19 GAMG: Solving for p, Initial residual = 7.04996e-05,
20 Final residual = 8.58288e-07, No Iterations 8
21 time step continuity errors : sum local = 3.33395e-11,
22 global = -2.13806e-12, cumulative = -2.52335e-10
23 ExecutionTime = 0.17 s  ClockTime = 0 s
```

Здесь можно видеть времена симуляции, выполнения и процессорное время (*Time*, *ExecutionTime*, *ClockTime*), среднее и максимальное числа Куранта (*Courant Number*), число итераций (*No Iterations*, начальную и конечную невязку (*Initial, Final residual*) решателей систем линейных уравнений для каждой компоненты скорости и давления, а также невязки уравнения неразрывности (*continuity error*): глобальную (*global*), локальную (*local*) и накопленный итог (*cumulative*). Невязки решателей полезны при диагностике и настройке параметров в *fvSolution*, невязки уравнения неразрывности, исходя из физичности решения, должны быть близки к нулю.

Вычисление необходимо производить до возникновения повторяющегося паттерна вихря за цилиндром.

Для анализа образовавшейся вихревой дорожки можно вычислить завихренность $\nabla \times \mathbf{U}$, используя команду пост-обработки:

```
1 postProcess -func vorticity
```

Результат удобно визуализировать с использованием *Paraview*, для этого в рабочей директории необходимо ввести команду:

```
1 paraFoam
```

В появившемся окне графического интерфейса возможно наглядно увидеть все рассчитанные в результате моделирования поля физических характеристик, а также совершать преобразование над ними, смотреть значения в отдельных точках, строить графики и диаграммы. Для начала работы необходимо нажать кнопку *Apply* (Принять) и выбрать в верхней панели поле завихренности (*vorticity*) и подходящий временной шаг. Более подробно об использовании *ParaView* можно узнать в приложении ??.

Задание для лабораторной работы

- Задать начальные условия согласно варианту, восстановить граничные условия.
- Провести численное моделирование решателем *potentialFOAM*, попробовать различные значения параметра циклов коррекции (0, 2) и сравнить результаты.
- Сравнить полученные результаты с аналитическим решением.

- Провести численное моделирование обтекания цилиндра решателем *pisoFOAM* для разных чисел Рейнольдса.
- Построить поля завихренности, используя *Paraview*, сравнить качественно с экспериментальными результатами (например, [2, 5]).

Вариант	Числа Рейнольдса
1	32,65,102
2	55,73,161
3	8,28,33
4	16,33,50
5	8,28,80
6	36,50,112

Таблица 1.1: Варианты для второй лабораторной работы

1.4. Листинги кода

Для лабораторной работы №2

```
1  /*----- C++ -----*\
2  | ===== |
3  |  \ \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \ / O peration | Version: v2206 |
5  |  \ \ / A nd | Website: www.openfoam.com |
6  |  \ \ / M anipulation |
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        blockMeshDict;
14 }
15 // ***** //
16
17 scale 1;
18
19 rinner 0.5;
20 router 1;
21 rinner45 #eval "$rinner * sqrt(0.5)";
22 router45 #eval "$router * sqrt(0.5)";
23 xright 4;
24 xleft -2;
```

```

25 ymax 2;
26 zmin -0.5;
27 zmax 0.5;
28 nradial 80;
29 nquarter 20;
30 nxrouter 200;
31 nyrouter 40;
32 nz 1;
33
34 vertices
35 (
36 ($rinner 0 $zmin ) // 0
37 ($router 0 $zmin ) // 1
38 ($xright 0 $zmin ) // 2
39 ($xright $router45 $zmin ) // 3
40 ($router45 $router45 $zmin ) // 4
41 ($rinner45 $rinner45 $zmin ) // 5
42 ($xright $ymax $zmin ) // 6
43 ($router45 $ymax $zmin ) // 7
44 (0 $ymax $zmin ) // 8
45 (0 $router $zmin ) // 9
46 (0 $rinner $zmin ) // 10
47 (-$rinner 0 $zmin ) // 11
48 (-$router 0 $zmin ) // 12
49 ($xleft 0 $zmin ) // 13
50 ($xleft $router45 $zmin ) // 14
51 (-$router45 $router45 $zmin ) // 15
52 (-$rinner45 $rinner45 $zmin ) // 16
53 ($xleft $ymax $zmin ) // 17
54 (-$router45 $ymax $zmin ) // 18
55
56 ( $rinner 0 $zmax ) // 19
57 ( $router 0 $zmax ) // 20
58 ( $xright 0 $zmax ) // 21
59 ( $xright $router45 $zmax ) // 22
60 ( $router45 $router45 $zmax ) // 23
61 ( $rinner45 $rinner45 $zmax ) // 24
62 ( $xright $ymax $zmax ) // 25
63 ( $router45 $ymax $zmax ) // 26
64 ( 0 $ymax $zmax ) // 27
65 ( 0 $router $zmax ) // 28
66 ( 0 $rinner $zmax ) // 29
67 ( -$rinner 0 $zmax ) // 30
68 ( -$router 0 $zmax ) // 31
69 ( $xleft 0 $zmax ) // 32
70 ( $xleft $router45 $zmax ) // 33
71 ( -$router45 $router45 $zmax ) // 34
72 ( -$rinner45 $rinner45 $zmax ) // 35
73 ( $xleft $ymax $zmax ) // 36
74 ( -$router45 $ymax $zmax ) // 37
75
76 ($xright -$router45 $zmin ) // 38
77 ($router45 -$router45 $zmin ) // 39
78 ($rinner45 -$rinner45 $zmin ) // 40
79 ($xright -$ymax $zmin ) // 41
80 ($router45 -$ymax $zmin ) // 42

```

```

81     (0 -$ymax $zmin ) // 43
82     (0 -$router $zmin ) // 44
83     (0 -$rinner $zmin ) // 45
84     ($xleft -$router45 $zmin ) // 46
85     (-$router45 -$router45 $zmin ) // 47
86     (-$rinner45 -$rinner45 $zmin ) // 48
87     ($xleft -$ymax $zmin ) // 49
88     (-$router45 -$ymax $zmin ) // 50
89
90     ( $xright -$router45 $zmax ) // 51
91     ( $router45 -$router45 $zmax ) // 52
92     ( $rinner45 -$rinner45 $zmax ) // 53
93     ( $xright -$ymax $zmax ) // 54
94     ( $router45 -$ymax $zmax ) // 55
95     ( 0 -$ymax $zmax ) // 56
96     ( 0 -$router $zmax ) // 57
97     ( 0 -$rinner $zmax ) // 58
98     ( $xleft -$router45 $zmax ) // 59
99     ( -$router45 -$router45 $zmax ) // 60
100    ( -$rinner45 -$rinner45 $zmax ) // 61
101    ( $xleft -$ymax $zmax ) // 62
102    ( -$router45 -$ymax $zmax ) // 63
103    );
104
105    blocks //nradial = 80, nquarter = 20, nxouter = 200, nyouter = 40
106    (
107        hex (5 4 9 10 24 23 28 29) ($nradial $nquarter $nz) grading (1 1 1) // 0
108        hex (0 1 4 5 19 20 23 24) ($nradial $nquarter $nz) grading (1 1 1) // 1
109        hex (1 2 3 4 20 21 22 23) ($nxouter $nquarter $nz) grading (1 1 1) // 2
110        hex (4 3 6 7 23 22 25 26) ($nxouter $nyouter $nz) grading (1 1 1) // 3
111        hex (9 4 7 8 28 23 26 27) ($nquarter $nyouter $nz) grading (1 1 1) // 4
112        hex (15 16 10 9 34 35 29 28) ($nradial $nquarter $nz) grading (1 1 1) // 5
113        hex (12 11 16 15 31 30 35 34) ($nradial $nquarter $nz) grading (1 1 1) // 6
114        hex (13 12 15 14 32 31 34 33) ($nquarter $nquarter $nz) grading (1 1 1) // 7
115        hex (15 18 17 14 34 37 36 33) ($nyouter $nquarter $nz) grading (1 1 1) // 8
116        hex (9 8 18 15 28 27 37 34) ($nyouter $nquarter $nz) grading (1 1 1) // 9
117
118        hex (40 45 44 39 53 58 57 52) ($nquarter $nradial $nz) grading (1 1 1) // 10
119        hex (0 40 39 1 19 53 52 20) ($nquarter $nradial $nz) grading (1 1 1) // 11
120        hex (1 39 38 2 20 52 51 21) ($nquarter $nxouter $nz) grading (1 1 1) // 12
121        hex (39 42 41 38 52 55 54 51) ($nyouter $nxouter $nz) grading (1 1 1) // 13
122        hex (44 43 42 39 57 56 55 52) ($nyouter $nquarter $nz) grading (1 1 1) // 14
123        hex (48 47 44 45 61 60 57 58) ($nradial $nquarter $nz) grading (1 1 1) // 15
124        hex (11 12 47 48 30 31 60 61) ($nradial $nquarter $nz) grading (1 1 1) // 16
125        hex (12 13 46 47 31 32 59 60) ($nquarter $nquarter $nz) grading (1 1 1) // 17
126        hex (47 46 49 50 60 59 62 63) ($nquarter $nyouter $nz) grading (1 1 1) // 18
127    hex (44 47 50 43 57 60 63 56) ($nquarter $nyouter $nz) grading (1 1 1) // 19
128    );
129
130    edges
131    (
132        arc 0 5 origin (0 0 $zmin)
133        arc 5 10 origin (0 0 $zmin)
134        arc 1 4 origin (0 0 $zmin)
135        arc 4 9 origin (0 0 $zmin)
136        arc 19 24 origin (0 0 $zmax)

```

```

137     arc 24 29 origin (0 0 $zmax)
138     arc 20 23 origin (0 0 $zmax)
139     arc 23 28 origin (0 0 $zmax)
140
141     arc 11 16 origin (0 0 $zmin)
142     arc 16 10 origin (0 0 $zmin)
143     arc 12 15 origin (0 0 $zmin)
144     arc 15 9 origin (0 0 $zmin)
145     arc 30 35 origin (0 0 $zmax)
146     arc 35 29 origin (0 0 $zmax)
147     arc 31 34 origin (0 0 $zmax)
148     arc 34 28 origin (0 0 $zmax)
149
150     arc 0 40 origin (0 0 $zmin)
151     arc 40 45 origin (0 0 $zmin)
152     arc 1 39 origin (0 0 $zmin)
153     arc 39 44 origin (0 0 $zmin)
154     arc 19 53 origin (0 0 $zmax)
155     arc 53 58 origin (0 0 $zmax)
156     arc 20 52 origin (0 0 $zmax)
157     arc 52 57 origin (0 0 $zmax)
158     arc 11 48 origin (0 0 $zmin)
159     arc 48 45 origin (0 0 $zmin)
160     arc 12 47 origin (0 0 $zmin)
161     arc 47 44 origin (0 0 $zmin)
162     arc 30 61 origin (0 0 $zmax)
163     arc 61 58 origin (0 0 $zmax)
164     arc 31 60 origin (0 0 $zmax)
165     arc 60 57 origin (0 0 $zmax)
166 );
167
168     boundary
169     (
170     top
171     {
172     type symmetryPlane;
173     faces
174     (
175     (7 8 27 26)
176     (6 7 26 25)
177     (8 18 37 27)
178     (18 17 36 37)
179     );
180     }
181     bottom
182     {
183     type symmetryPlane;
184     faces
185     (
186     (49 50 63 62)
187     (50 43 56 63)
188     (43 42 55 56)
189     (42 41 54 55)
190     );
191     }
192     inlet

```

```

193 {
194     type patch;
195     faces
196     (
197         (14 13 32 33)
198         (17 14 33 36)
199         (46 13 32 59)
200         (46 49 62 59)
201     );
202 }
203 outlet
204 {
205     type patch;
206     faces
207     (
208         (2 3 22 21)
209         (3 6 25 22)
210         (38 51 21 2)
211         (41 54 51 38)
212     );
213 }
214 cylinder
215 {
216     type wall;
217     faces
218     (
219         (10 5 24 29)
220         (5 0 19 24)
221         (16 10 29 35)
222         (11 16 35 30)
223         (48 11 30 61)
224         (45 48 61 58)
225         (40 45 58 53)
226         (0 40 53 19)
227     );
228 }
229 frontAndBack
230 {
231     type empty;
232     faces
233     (
234         (5 10 9 4)
235         (24 23 28 29)
236         (0 5 4 1)
237         (19 20 23 24)
238         (1 4 3 2)
239         (20 21 22 23)
240         (4 7 6 3)
241         (23 22 25 26)
242         (4 9 8 7)
243         (28 23 26 27)
244         (16 15 9 10)
245         (35 29 28 34)
246         (12 15 16 11)
247         (31 30 35 34)
248         (13 14 15 12)

```



```

249         (32 31 34 33)
250         (14 17 18 15)
251         (33 34 37 36)
252         (15 18 8 9)
253         (34 28 27 37)
254         (45 40 39 44)
255         (58 57 52 53)
256         (40 0 1 39)
257         (53 52 20 19)
258         (39 1 2 38)
259         (52 51 21 20)
260         (39 38 41 42)
261         (52 55 54 51)
262         (44 39 42 43)
263         (57 56 55 52)
264         (47 48 45 44)
265         (60 57 58 61)
266         (12 11 48 47)
267         (31 60 61 30)
268         (13 12 47 46)
269         (32 59 60 31)
270         (49 46 47 50)
271         (62 63 60 59)
272         (50 47 44 43)
273         (63 56 57 60)
274     );
275 }
276 );
277 mergePatchPairs
278 (
279 );
280 // *****

```

Рис. 1.3: Файл blockMeshDict для обтекания цилиндра

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: v2206 |
5  | \ \ / A n d | Website: www.openfoam.com |
6  | \ \ / M a n i p u l a t i o n |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     object        U;
14 }
15 // *****
16
17 dimensions      [0 1 -1 0 0 0 0];

```

```

18
19 internalField    uniform (0 0 0);
20
21 boundaryField
22 {
23     top
24     {
25         type      symmetryPlane;
26     }
27
28     outlet
29     {
30         type      zeroGradient;
31     }
32
33     bottom
34     {
35         type      symmetryPlane;
36     }
37
38     inlet
39     {
40         type      uniformFixedValue;
41         uniformValue    constant (1 0 0);
42     }
43
44     cylinder
45     {
46         type      fixedValue;
47         value      uniform (0 0 0);
48     }
49
50     frontAndBack
51     {
52         type      empty;
53     }
54 }
55
56
57 // *****

```

Рис. 1.4: Файл граничных условий по скорости для обтекания цилиндра

```

1  /*----- C++ -----*\
2  | ===== |
3  | \ \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O peration | Version: v2206 |
5  | \ \ / A nd | Website: www.openfoam.com |
6  | \ \ / M anipulation |
7  /*-----*/
8  FoamFile

```

```

9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object         controlDict;
14 }
15 // * * * * *
16
17 application      pisoFoam;
18
19 startFrom        startTime;
20
21 startTime        0;
22
23 stopAt           endTime;
24
25 endTime          50;
26
27 deltaT           0.001;
28
29 writeControl      adjustable;
30
31 writeInterval     1;
32
33 purgeWrite        0;
34
35 writeFormat       ascii;
36
37 writePrecision    6;
38
39 writeCompression  off;
40
41 timeFormat        general;
42
43 timePrecision     6;
44
45 runTimeModifiable true;
46
47 // * * * * *

```

Рис. 1.5: Файл controlDict для обтекания цилиндра

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O peration | Version: v2206 |
5  | \ \ / A nd | Website: www.openfoam.com |
6  | \ \ / M anipulation |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;

```

```

11     format      ascii;
12     class       dictionary;
13     object      fvSchemes;
14 }
15 // *****
16
17 ddtSchemes
18 {
19     default      steadyState;
20 }
21
22 gradSchemes
23 {
24     default      leastSquares;
25 }
26
27 divSchemes
28 {
29     default      none;
30
31     div(phi,U)   bounded Gauss linear;
32
33     div(div(phi,U)) Gauss linear;
34 }
35
36 laplacianSchemes
37 {
38     default      Gauss linear corrected;
39 }
40
41 interpolationSchemes
42 {
43     default      linear;
44 }
45
46 snGradSchemes
47 {
48     default      corrected;
49 }
50
51
52 // *****

```

Рис. 1.6: Файл fvSchemes для обтекания цилиндра (potentialFoam)

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O peration | Version: v2206 |
5  | \ \ / A nd | Website: www.openfoam.com |
6  | \ \ / M anipulation |
7  /*-----*/

```

```

8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object         fvSolution;
14 }
15 // ***** //
16
17 solvers
18 {
19     Phi
20     {
21         solver      GAMG;
22         smoother     DIC;
23         tolerance    1e-06;
24         relTol       0.01;
25     }
26
27     p
28     {
29         $Phi;
30     }
31 }
32
33 potentialFlow
34 {
35     nNonOrthogonalCorrectors 3;
36 }
37
38
39 // ***** //

```

Рис. 1.7: Файл fvSolution для обтекания цилиндра (potentialFoam)

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: v2206 |
5  | \ \ / A n d | Website: www.openfoam.com |
6  | \ \ / M a n i p u l a t i o n |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object         fvSchemes;
14 }
15 // ***** //
16 ddtSchemes
17 {

```

```

18     default      Euler;
19 }
20
21 gradSchemes
22 {
23     default      Gauss linear;
24 }
25
26 divSchemes
27 {
28     default      none;
29     div(phi,U)   Gauss limitedLinearV 1;
30     div(phi,k)   Gauss limitedLinear 1;
31     div(phi,epsilon) Gauss limitedLinear 1;
32     div(phi,omega) Gauss limitedLinear 1;
33     div(phi,R)   Gauss limitedLinear 1;
34     div(R)       Gauss linear;
35     div(phi,nuTilda) Gauss limitedLinear 1;
36     div((nuEff*dev2(T(grad(U)))) Gauss linear;
37 }
38
39 laplacianSchemes
40 {
41     default      Gauss linear corrected;
42 }
43
44 interpolationSchemes
45 {
46     default      linear;
47 }
48
49 snGradSchemes
50 {
51     default      corrected;
52 } // *****

```

Рис. 1.8: Файл fvSchemes для обтекания цилиндра (PISO)

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O peration | Version: v2206 |
5  | \ \ / A nd | Website: www.openfoam.com |
6  | \ \ / M anipulation |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        fvSolution;
14 }

```

```

15 // * * * * *
16
17 solvers
18 {
19     p
20     {
21         solver          GAMG;
22         tolerance        1e-06;
23         relTol           0.1;
24         smoother         GaussSeidel;
25     }
26
27     pFinal
28     {
29         $p;
30         tolerance        1e-06;
31         relTol           0;
32     }
33
34     "(U|k|epsilon|omega|R|nuTilda)"
35     {
36         solver           smoothSolver;
37         smoother          GaussSeidel;
38         tolerance         1e-05;
39         relTol            0;
40     }
41 }
42
43 PISO
44 {
45     nCorrectors          2;
46     nNonOrthogonalCorrectors 2;
47     pRefCell              0;
48     pRefValue             0;
49 }
50
51 // * * * * *

```

Рис. 1.9: Файл fvSolution для обтекания цилиндра (PISO)

```

1  /*-----*-- C++ -*-----*\
2  | ===== |
3  | \ \      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O peration   | Version: v2206 |
5  | \ \      / A nd         | Website: www.openfoam.com |
6  | \ \      / M anipulation |
7  \*-----*--
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;

```

```
13     object      transportProperties;  
14 }  
15 // * * * * *  
16  
17 transportModel  Newtonian;  
18  
19 nu              2e-4; // Re = 200  
20  
21  
22 // * * * * *
```

Рис. 1.10: Файл transportProperties для обтекания цилиндра

Список литературы

1. Ferziger J. H., Perić M. Computational Methods for Fluid Dynamics. — Springer Berlin Heidelberg, 2002.
2. Homann F. Einfluß großer Zähigkeit bei Strömung um Zylinder // Forschung auf dem Gebiete des Ingenieurwesens. — 1936. — Янв. — Т. 7, № 1. — С. 1—10.
3. Issa R. Solution of the implicitly discretised fluid flow equations by operator-splitting // Journal of Computational Physics. — 1986. — Янв. — Т. 62, № 1. — С. 40—65.
4. Jasak H. Error Analysis and Estimation for the Finite Volume Method With Applications to Fluid Flows // Direct. — 1996. — Янв. — Т. М.
5. Zdravkovich M. M. Smoke observations of the formation of a Kármán vortex street // Journal of Fluid Mechanics. — 1969. — Июль. — Т. 37, № 3. — С. 491—496.
6. Валландер С. В. Лекции по гидроаэромеханике. — 2-е изд. — Издательство Санкт-Петербургского государственного университета, 2005.