
DINÂMICA DOS FLUIDOS COMPUTACIONAL

Guia prático usando o OpenFOAM

Livia Flavia Carletti Jatobá



Universidade do Estado do Rio de Janeiro, Instituto Politécnico

Licença:

Este trabalho está licenciado sob a licença Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License (CC BY-NC-SA 4.0 - Atribuição-NãoComercial-CompartilhaIgual)

Aviso Legal:

OpenFOAM® and OpenCFD® são marcas registradas por OpenCFDLimited, que produz o software OpenFOAM. Todas as marcas registradas são de seus proprietários. Este documento não foi aprovado ou endossado por OpenCFD Limited, o produtor do software OpenFOAM e detentor das marcas registradas OPENFOAM® and OpenCFD®.

CAPÍTULO 1

ESCOAMENTO EM TORNO DE UM CILINDRO

KAIO VALENTE QUINTES

Controle de versão: T01-1.0

ESTE É UM DOCUMENTO EM REVISÃO

Objetivo: Avaliar convergência numérica, através de uma variável integrada, para diferentes números de Reynolds do escoamento laminar em torno de um corpo cilíndrico.

Os arquivos deste tutorial estão disponíveis no repositório: <https://github.com/liviajatoba/cfd-openfoam/>

1.1 Definição do problema

O problema físico estudado neste tutorial consiste no escoamento de um fluido newtoniano, isotérmico e incompressível, em torno de um corpo cilíndrico. Neste tutorial, vamos utilizar os utilitários `blockMesh` e `foamLog` para as etapas de pré e pós-processamento, e o *solver* `pisoFoam` para a solução do escoamento. O `gnuplot` e `ParaView` são as ferramentas utilizadas para gerar os resultados gráficos.

A Figura 1.1 mostra a geometria do tutorial, onde R é o raio do corpo cilíndrico. O domínio computacional possui uma região anterior ao corpo sólido de comprimento $L = 20R$, e uma região posterior ao corpo sólido de comprimento $1,5L$. A altura total do domínio computacional é $2L$, e o centro do corpo cilíndrico é a origem do sistema de coordenadas. A velocidade do escoamento na entrada é conhecida e uniforme igual à $(U, 0, 0)$. Os planos superior e inferior são planos de simetria.

O escoamento em torno de um corpo cilíndrico é um problema clássico de escoamento externo em dinâmica dos fluidos. Dependendo das condições do escoamento, ou seja, do número de Reynolds,

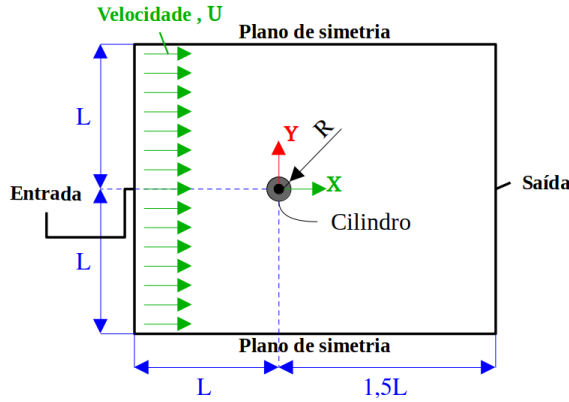


Figura 1.1 Representação esquemática do domínio computacional do escoamento em torno de um corpo cilíndrico

observa-se a separação do escoamento. A separação do escoamento é um fenômeno que ocorre quando existe um gradiente de pressão adverso, ou seja, a pressão aumenta na direção do escoamento principal. [1, 3].

A separação do escoamento pode ser analisada através da Equação de Bernoulli, que descreve a conservação de energia em um fluido e é dada por,

$$\frac{\Delta U^2}{2g} + \frac{\Delta P}{\rho g} + \Delta z = 0 \quad (1.1)$$

onde o primeiro termo é a energia cinética, o segundo o trabalho devido a pressão e o terceiro o trabalho devido a força da gravidade. Essa equação, aplicada ao escoamento em torno de um corpo cilíndrico onde $\Delta z = 0$, torna-se,

$$\frac{\Delta P}{\rho} = -\frac{\Delta U^2}{2} \quad (1.2)$$

ou seja, a variação de pressão é contrária a variação de velocidade. Assim, quando a velocidade do fluido aumenta, a pressão do escoamento diminui.

A Figura 1.2 mostra uma representação esquemática do fenômeno de separação do escoamento em torno de um corpo cilíndrico. Considere uma partícula de fluido em uma linha de corrente do escoamento, representada pela linha contínua. Nessa figura o escoamento do fluido em torno do corpo cilíndrico acontece em três regiões: frontal, superior e traseira. Na região frontal, o fluido choca-se com o cilindro em um ponto chamado de ponto de estagnação, o que leva a uma velocidade nula e um aumento de pressão, a partir deste momento o fluido desloca-se para a região superior do cilindro. A análise entre a região frontal e a região superior do corpo cilíndrico mostra que a velocidade da partícula fluida aumenta, $\nabla U > 0$, o que resulta, segundo a Eq. 1.2, em uma redução na pressão, $\nabla p < 0$.

Um comportamento oposto é observado entre a região superior do objeto e a região traseira, quer dizer, a velocidade do fluido diminui, $\nabla U < 0$, o que resulta, segundo a Eq. 1.2, em um aumento na pressão, $\nabla p > 0$. O gradiente de pressão contrário ao escoamento principal na região traseira, faz com que as partículas do fluido nesta região mudem de direção, formando uma região de escoamento reverso. Por isso, a região traseira é denominada como região de esteira, caracterizada pela formação de vórtices e baixa pressão.

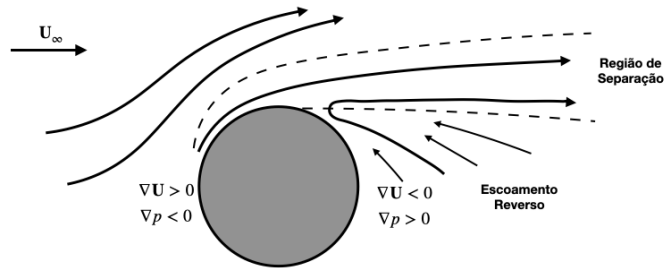


Figura 1.2 Separação do escoamento em torno de um corpo cilíndrico

Duas simulações CFD serão realizadas para a geometria 2D da Figura 1.1. Cada simulação terá um número de Reynolds, que é calculado por,

$$Re = \frac{UD}{\nu} \quad (1.3)$$

onde U é a componente x da velocidade na entrada, D o diâmetro do cilindro, ou seja $2R$, e ν a viscosidade cinemática do fluido. O número do Reynolds é o parâmetro adimensional da literatura utilizado para classificar o escoamento em laminar, transicional ou turbulento. As faixas de número de Reynolds e respectivas classificações dependem da geometria do problema. A Figura 1.3 mostra a classificação do escoamento, em função do número de Reynolds, para o escoamento em torno de um corpo cilíndrico [].

	Fluxo ameno (sem vórtices)	$Re < 5$
	Um par fixo de vórtices simétricos	$5 < Re < 40$
	Caminho de vórtices laminar	$40 < Re < 150$
	Camada limite laminar até o ponto de separação, esteira turbulenta	$150 < Re < 300$ Transição para turbulento $300 < Re < 3 \times 10^5$
	Transição da camada limite para turbulenta	$3 \times 10^5 < Re < 3 \times 10^6$
	Caminho de vórtices turbulentos, mas a esteira é mais estreita do que no caso laminar.	$Re > 3 \times 10^6$

Figura 1.3 Classificação dos regimes de escoamento em torno de um corpo cilíndrico.

Observe que este tipo de geometria produz três tipos de escoamento laminar. O primeiro, onde o Reynolds é menor que 5, as linhas de corrente da parte anterior são idênticas a parte posterior do corpo, ou seja, não ocorre a formação de vórtices atrás do objeto. A segunda, com Reynolds entre 5 e 40, observa-se a formação de dois vórtices fixos atrás do objeto. A terceira, onde o Reynolds está

entre 40 e 150, observa-se a formação de vórtices flutuantes atrás do objeto. É importante destacar que o escoamento em regime estacionário é alcançado nos dois primeiros casos. Já o terceiro, é um escoamento intrinsecamente transiente e, portanto, não atinge estado estacionário. Isto ocorre pois os vórtices que se formam atrás do objeto estão sempre mudando de posição.

O objetivo do tutorial será avaliar as diferenças entre a convergência numérica de um escoamento estacionário e um escoamento transiente. Assim, resolveremos um escoamento para Reynolds igual a 10 e 100. A velocidade na entrada é constante e igual a 1 m/s . O diâmetro do corpo cilíndrico também será constante e igual a 1 m . A viscosidade cinemática será a variável adotada para configurar os dois tipos de escoamento.

Uma variável de interesse precisa ser definida para avaliar a convergência numérica das simulações. É conveniente escolher uma variável integrada, que pode ser acompanhada ao longo do tempo. Assim, a convergência numérica para as simulações será realizada para a força, ou o coeficiente de arrasto, no corpo.

Recorde a definição da força de contato devido o escoamento de fluido,

$$\mathbf{F}_s = \int_A -P \mathbf{n} dA + \int_A \boldsymbol{\tau} \cdot \mathbf{n} dA \quad (1.4)$$

onde P é a pressão, $\boldsymbol{\tau}$ é o tensor de cisalhamento e \mathbf{n} é o vetor normal externo a superfície. A nossa variável de interesse será esta força resultante na parede do corpo cilíndrico. A componente dessa força resultante na direção do escoamento é chamada de força de arrasto (F_D) e a componente na direção normal ao escoamento é chamada de força de sustentação (F_L). A Figura 1.4 mostra a decomposição da força de contato nas forças de arrasto e sustentação.

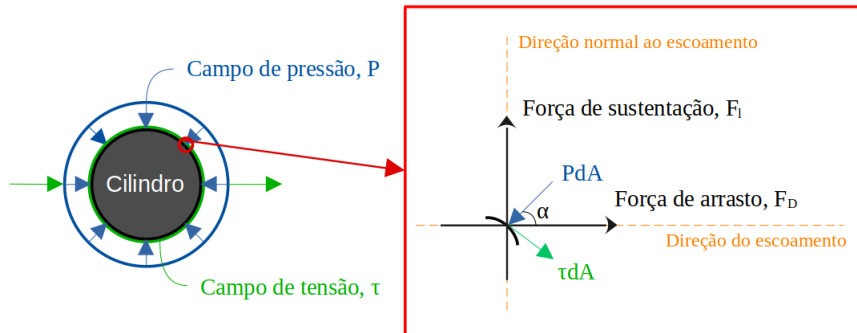


Figura 1.4 Forças de contato sobre o corpo cilíndrico e respectiva decomposição em forças de arrasto e sustentação

A literatura reporta ainda o coeficiente de arrasto (*drag Coefficient*, C_D) como uma propriedade adimensional para a análise da força de arrasto,

$$C_D = \frac{F_D}{\frac{1}{2} \rho U^2 A} \quad (1.5)$$

onde ρ é a massa específica do fluido, U é a velocidade característica do escoamento e A é a área de referencia, ou área projetada, que para o cilindro é o produto do diâmetro pelo comprimento do cilindro. A Figura 1.5 mostra os valores experimentais do coeficiente de arrasto para o escoamento em torno de um cilindro para diferentes Reynolds [2] apud [1].

APRESENTAR DISCUSSÃO DA LITERATURA QUE MOSTRA QUE O C_D NÃO É SUFICIENTE PARA FAZER VALIDAÇÃO DOS CASOS E PRECISA AINDA DA DISTÂNCIA ENTRE OS VÓRTICES - INCLUIR CITACAO - PASSAR ARTIGO PARA KAIO

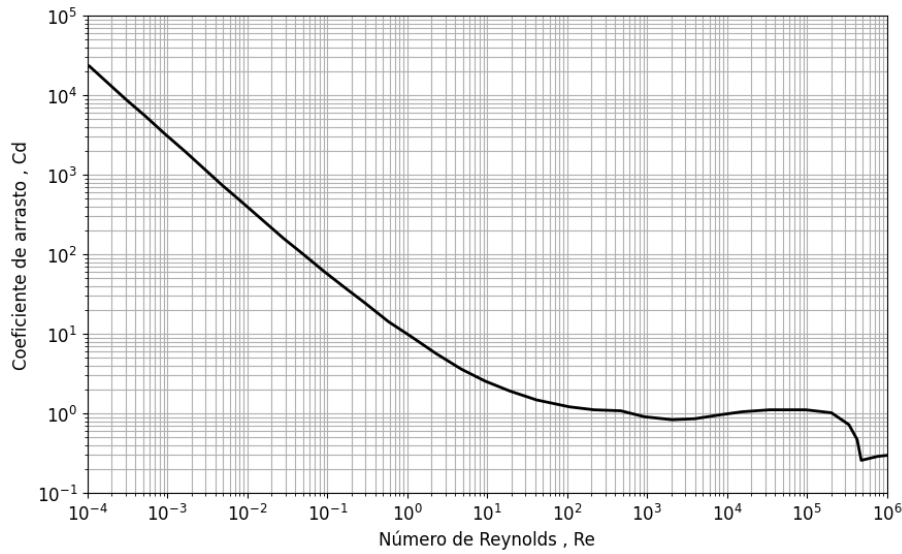


Figura 1.5 Curvas dos dados experimentais do coeficiente de arrasto para o escoamento em torno de uma esfera, disco e cilindro [2] apud [1].

A seguir são detalhadas as etapas necessárias para a configuração das simulações, ou seja, a construção da geometria, malha, configurações das propriedades do fluido e os parâmetros para a solução numérica.

1.2 Geometria e malha

A geometria do campo de escoamento em torno do corpo cilíndrico será gerada por meio do arquivo `blockMeshDict` no diretório `system`.

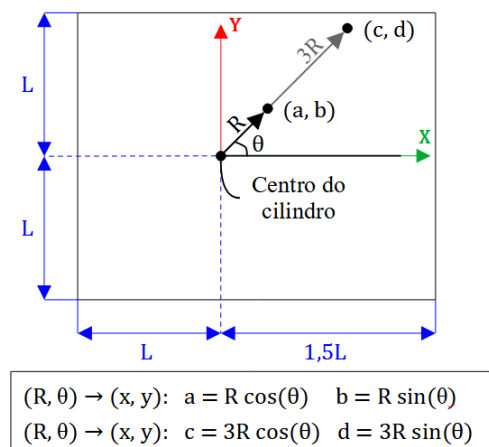


Figura 1.6 Construção da geometria em torno do corpo cilíndrico

Para construir essa geometria, o campo de escoamento em torno do corpo cilíndrico C é colocado em função do comprimento do escoamento L , do raio do cilindro R e do ângulo θ . Esquematisando isto matematicamente:

$$C(L, R, \theta) \quad (1.6)$$

Na Figura 1.6 pode ser visto essas variáveis na representação esquemática do problema físico que são utilizadas para determinar os pontos em torno do cilindro, sendo o raio R e o ângulo θ coordenadas polares. Deste modo, será feito a conversão das coordenadas polares para as coordenadas cartesianas, como mostrado no *frame* abaixo da representação esquemática para os pontos (a, b) e (c, d) , obtidos respectivamente, a uma distância R e $3R$ da origem.

A seguir, considerando o raio do cilindro $R = 1m$ e o ângulo $\theta = 45^\circ$, consequentemente são determinados os valores do raio do bloco $3R$ e, inclusive, o comprimento do campo de escoamento, no qual $L = 20R$, visto serem dependentes do raio R . Desta maneira, foram obtidos os pontos para a construção da geometria em torno do cilindro mostrada na Figura 1.7.

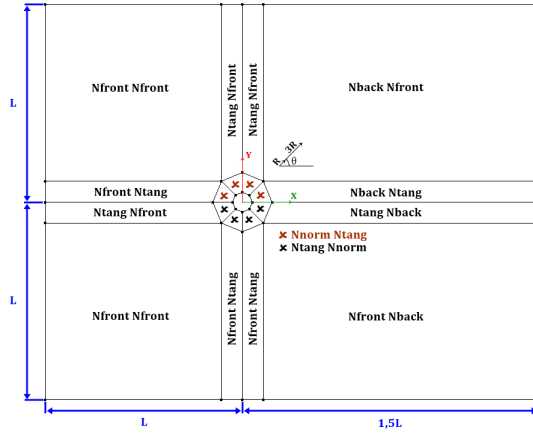


Figura 1.7 Geometria em torno do corpo cilíndrico

Nesta figura, percebe-se também, quatro variáveis (N_{front} , N_{back} , N_{norm} , N_{tang}) nas regiões da geometria, as quais determinam o número de divisões que serão realizadas em cada eixo para a construção da malha. As especificações dos valores de todas as variáveis necessárias para a construção da geometria e, posteriormente, da malha, são apresentadas no código 1.1 do arquivo `blockMeshDict` com os cálculos requeridos segundo o procedimento delineado até aqui representado na Figura 1.6.

Código 1.1 Definição dos valores e cálculos no `BlockMeshDict`

```
// Conversão de unidades
convertToMeters 1;

// D [m] do corpo cilíndrico
Diam 2;

// malha
Ntang 10;
Nnorm 20;
Nfront 30;
Nback 50;
Nz 1;

// dimensão de z
```



```

maxZ 0.5;
minZ #calc "-1.0*$maxZ";

// R [m] do cilindro
radius      #calc "0.5*$Diam";
radiusNeg   #calc "-1.0*$radius";
Angle       45.0;
radAngle    #calc "degToRad($Angle)";

a           #calc "$radius*cos($radAngle)"; // Eixo x
aNeg        #calc "-1.0*$a";               // Eixo -x
b           #calc "$radius*sin($radAngle)"; // Eixo y
bNeg        #calc "-1.0*$b";               // Eixo -y

// R [m] do bloco
radiusBlock #calc "3.0*$radius";
radiusBlockNeg #calc "-1.0*$radiusBlock";

c           #calc "$radiusBlock*cos($radAngle)"; // Eixo x
cNeg        #calc "-1.0*$c";                   // Eixo -x
d           #calc "$radiusBlock*sin($radAngle)"; // Eixo y
dNeg        #calc "-1.0*$d";                   // Eixo -y

// dominio [m]
L           #calc "20.0*$radius"; // Eixo y
LNeg        #calc "-1.0*$L";      // Eixo -y
Lback       #calc "1.5*$L";       // Eixo x
Lfront      #calc "-1.0*$L";      // Eixo -x

```

Enquanto, no código 1.2 do mesmo arquivo, são apresentadas as coordenadas, ou melhor, os vértices que formam a geometria, os quais são determinados pelas variáveis declaradas e calculadas no código 1.1. Importante ressaltar, os pontos determinados para a construção da geometria são nomeados numericamente em forma de comentário, o que é fundamental para a construção dos blocos que formam a geometria no OPENFOAM, como será visto mais adiante.

Código 1.2 Definição dos vértices da geometria no BlockMeshDict

```

vertices
(
    //back up
    ($radius      0      $minZ) //0
    ($radiusBlock 0      $minZ) //1
    ($Lback       0      $minZ) //2
    ($Lback       $d      $minZ) //3
    ($c           $d      $minZ) //4
    ($a           $b      $minZ) //5
    ($Lback       $L      $minZ) //6
    ($c           $L      $minZ) //7
    (0            $L      $minZ) //8
    (0            $radiusBlock $minZ) //9
    (0            $radius  $minZ) //10
    ($radiusNeg   0      $minZ) //11
    ($radiusBlockNeg 0      $minZ) //12
    ($Lfront      0      $minZ) //13
    ($Lfront      $d      $minZ) //14
    ($cNeg        $d      $minZ) //15
    ($aNeg        $b      $minZ) //16
    ($Lfront      $L      $minZ) //17
    ($cNeg        $L      $minZ) //18

    //front up

```

8 ESCOAMENTO EM TORNO DE UM CILINDRO

```

($radius      0      $maxZ) //19
($radiusBlock 0      $maxZ) //20
($lback       0      $maxZ) //21
($lback       $d      $maxZ) //22
($c           $d      $maxZ) //23
($a           $b      $maxZ) //24
($lback       $L      $maxZ) //25
($c           $L      $maxZ) //26
(0            $L      $maxZ) //27
(0            $radiusBlock $maxZ) //28
(0            $radius  $maxZ) //29
($radiusNeg   0      $maxZ) //30
($radiusBlockNeg 0      $maxZ) //31
($lfront      0      $maxZ) //32
($lfront      $d      $maxZ) //33
($cNeg        $d      $maxZ) //34
($aNeg        $b      $maxZ) //35
($lfront      $L      $maxZ) //36
($cNeg        $L      $maxZ) //37

//back down
($lback       $dNeg    $minZ) //38
($c           $dNeg    $minZ) //39
($a           $bNeg    $minZ) //40
($lback       $LNeg    $minZ) //41
($c           $LNeg    $minZ) //42
(0            $LNeg    $minZ) //43
(0            $radiusBlockNeg $minZ) //44
(0            $radiusNeg $minZ) //45
($lfront      $dNeg    $minZ) //46
($cNeg        $dNeg    $minZ) //47
($aNeg        $bNeg    $minZ) //48
($lfront      $LNeg    $minZ) //49
($cNeg        $LNeg    $minZ) //50

//front down
($lback       $dNeg    $maxZ) //51
($c           $dNeg    $maxZ) //52
($a           $bNeg    $maxZ) //53
($lback       $LNeg    $maxZ) //54
($c           $LNeg    $maxZ) //55
(0            $LNeg    $maxZ) //56
(0            $radiusBlockNeg $maxZ) //57
(0            $radiusNeg $maxZ) //58
($lfront      $dNeg    $maxZ) //59
($cNeg        $dNeg    $maxZ) //60
($aNeg        $bNeg    $maxZ) //61
($lfront      $LNeg    $maxZ) //62
($cNeg        $LNeg    $maxZ) //63

);

```

Por outro lado, estes pontos foram enumerados sobre quatro planos *back up*, *front up*, *back down* e *front down*, os quais estão representados graficamente na Figura 1.8, onde são identificados segundo a sua posição e cor, sendo portanto, o *back up* na cor preta, o *front up* na cor vermelha, *back down* na cor azul e o *front down* na cor verde. Os valores das coordenadas dos vértices que constroem esse gráfico são dados pela Tabela 1.1 considerando-os com três dígitos significativos caso sejam irracionais. Essas coordenadas são as representadas no código 1.2 pelas variáveis declaradas e calculadas no código 1.1.

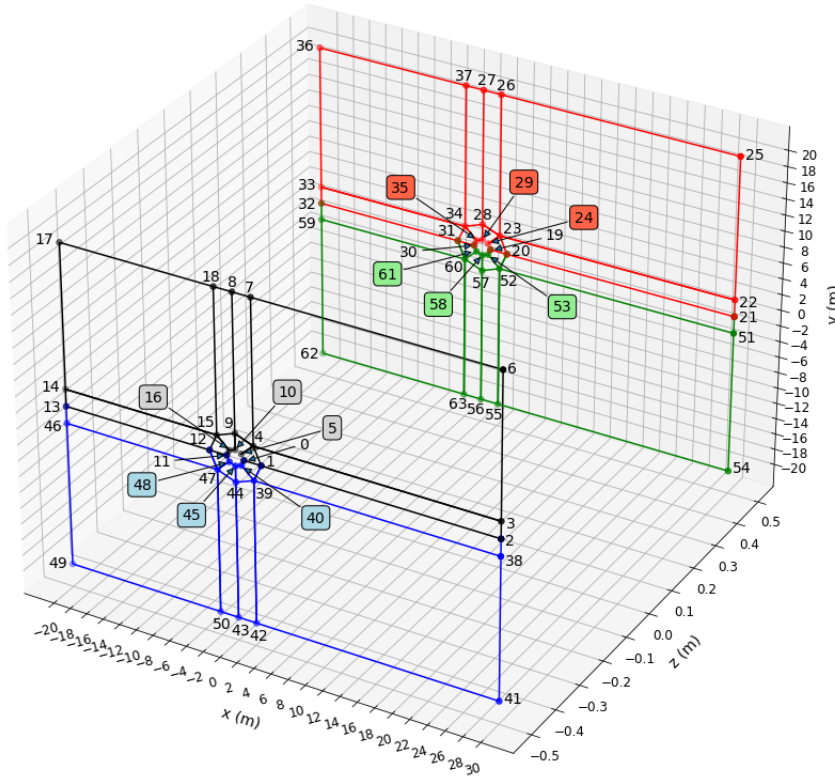


Figura 1.8 Geometria tridimensional do campo de escoamento em torno do corpo cilíndrico

Apesar dos planos da parte *up* e da parte *down* não estarem ligados, uma vez que tornaria o gráfico visivelmente poluído, essa figura mostra claramente a geometria tridimensional do campo de escoamento em torno do corpo cilíndrico, visto que fica nítido a formação da geometria tridimensional quando se liga mentalmente os planos *back up* e *front up*, e os planos *back down* e *front down*. Nas Figuras 1.9 e 1.10 são mostrados, respectivamente, esses planos, em uma visão bidimensional.

Nessas figuras, observa-se que os planos tanto na parte *up* como na parte *down* são formados por regiões as quais são indicadas por letras maiúsculas. As especificações dessas regiões são importantes para um melhor entendimento de um trecho do arquivo `blockMeshDict` dado pelo código 1.3 responsável pela construção dos blocos que formam a geometria e a malha, uma vez que os blocos são criados pela união das regiões dos planos *up* e das regiões dos planos *down*, isso fica claro ao notar nesse código, tanto na parte *up* quanto na *down*, um comentário indicando as letras das regiões que constroem cada bloco, como por exemplo, o bloco A-A na parte *up* designa o acoplamento da região A do plano *back up* com a região A do plano *front up*.

Neste momento, os códigos 1.1-1.3 são executados no Terminal 1.1 do sistema operacional LINUX fazendo o uso do comando **blockMesh** para gerar a geometria e do comando **paraFoam** para exibir o *paraView* dela, que é apresentado na Figura 1.11. Lembrando, isso deve ser feito no diretório do caso, o qual aqui é o `OpenFOAM/cylinderFlow`, e mais ainda, como se trata de gerar a geo-

metria, as variáveis da malha (N_{front} , N_{back} , N_{norm} , N_{tang}) devem ser todas colocadas com valores unitários.

Código 1.3 Construção dos blocos no BlockMeshDict

```

blocks
(

// --> up

// A-A
hex (5 4 9 10 24 23 28 29) ($Nnorm $Ntang $Nz) simpleGrading (2 1 1)

// B-B
hex (0 1 4 5 19 20 23 24) ($Nnorm $Ntang $Nz) simpleGrading (2 1 1)

// C-C
hex (1 2 3 4 20 21 22 23) ($Nback $Ntang $Nz) simpleGrading (6 1 1)

// D-D
hex (4 3 6 7 23 22 25 26) ($Nback $Nfront $Nz) simpleGrading (6 7 1)

// E-E
hex (9 4 7 8 28 23 26 27) ($Ntang $Nfront $Nz) simpleGrading (1 7 1)

// F-F
hex (15 16 10 9 34 35 29 28) ($Nnorm $Ntang $Nz) simpleGrading (0.5 1 1)

// G-G
hex (12 11 16 15 31 30 35 34) ($Nnorm $Ntang $Nz) simpleGrading (0.5 1 1)

// H-H
hex (13 12 15 14 32 31 34 33) ($Nfront $Ntang $Nz) simpleGrading (0.15 1 1)

// I-I
hex (14 15 18 17 33 34 37 36) ($Nfront $Nfront $Nz) simpleGrading (0.15 7
1)

// J-J
hex (15 9 8 18 34 28 27 37) ($Ntang $Nfront $Nz) simpleGrading (1 7 1)

// --> down

// A-A
hex (40 45 44 39 53 58 57 52) ($Ntang $Nnorm $Nz) simpleGrading (1 2 1)

// B-B
hex (0 40 39 1 19 53 52 20) ($Ntang $Nnorm $Nz) simpleGrading (1 2 1)

// C-C
hex (1 39 38 2 20 52 51 21) ($Ntang $Nback $Nz) simpleGrading (1 6 1)

// D-D
hex (39 42 41 38 52 55 54 51) ($Nfront $Nback $Nz) simpleGrading (7 6 1)

// E-E
hex (44 43 42 39 57 56 55 52) ($Nfront $Ntang $Nz) simpleGrading (7 1 1)

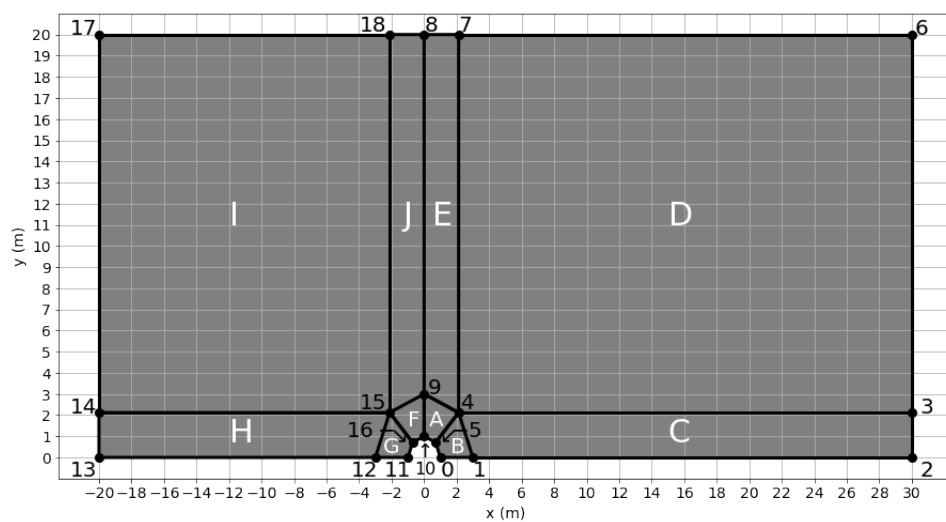
// F-F
hex (47 44 45 48 60 57 58 61) ($Ntang $Nnorm $Nz) simpleGrading (1 0.5 1)

// G-G

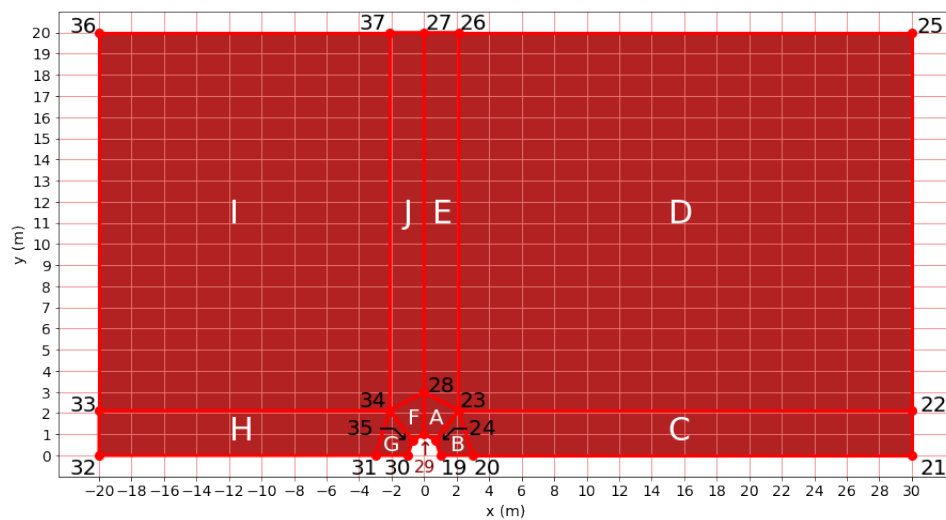
```

back up		front up	
Pontos	Coordenadas	Pontos	Coordenadas
0	(1 0 -0.5)	19	(1 0 0.5)
1	(3 0 -0.5)	20	(3 0 0.5)
2	(30 0 -0.5)	21	(30 0 0.5)
3	(30 2.12 -0.5)	22	(30 2.12 0.5)
4	(2.12 2.12 -0.5)	23	(2.12 2.12 0.5)
5	(0.707 0.707 -0.5)	24	(0.707 0.707 0.5)
6	(30 20 -0.5)	25	(30 20 0.5)
7	(2.12 20 -0.5)	26	(2.12 20 0.5)
8	(0 20 -0.5)	27	(0 20 0.5)
9	(0 3 -0.5)	28	(0 3 0.5)
10	(0 1 -0.5)	29	(0 1 0.5)
11	(-1 0 -0.5)	30	(-1 0 0.5)
12	(-3 0 -0.5)	31	(-3 0 0.5)
13	(-20 0 -0.5)	32	(-20 0 0.5)
14	(-20 2.12 -0.5)	33	(-20 2.12 0.5)
15	(-2.12 2.12 -0.5)	34	(-2.12 2.12 0.5)
16	(-0.707 0.707 -0.5)	35	(-0.707 0.707 0.5)
17	(-20 20 -0.5)	36	(-20 20 0.5)
18	(-2.12 20 -0.5)	37	(-2.12 20 0.5)
black down		front down	
Pontos	Coordenadas	Pontos	Coordenadas
38	(30 -2.12 -0.5)	51	(30 -2.12 0.5)
39	(2.12 -2.12 -0.5)	52	(2.12 -2.12 0.5)
40	(0.707 -0.707 -0.5)	53	(0.707 -0.707 0.5)
41	(30 -20 -0.5)	54	(30 -20 0.5)
42	(2.12 -20 -0.5)	55	(2.12 -20 0.5)
43	(0 -20 -0.5)	56	(0 -20 0.5)
44	(0 -3 -0.5)	57	(0 -3 0.5)
45	(0 -1 -0.5)	58	(0 -1 0.5)
46	(-20 -2.12 -0.5)	59	(-20 -2.12 0.5)
47	(-2.12 -2.12 -0.5)	60	(-2.12 -2.12 0.5)
48	(-0.707 -0.707 -0.5)	61	(-0.707 -0.707 0.5)
49	(-20 -20 -0.5)	62	(-20 -20 0.5)
50	(-2.12 -20 -0.5)	63	(-2.12 -20 0.5)

Tabela 1.1 Valores dos vértices nos planos up e down

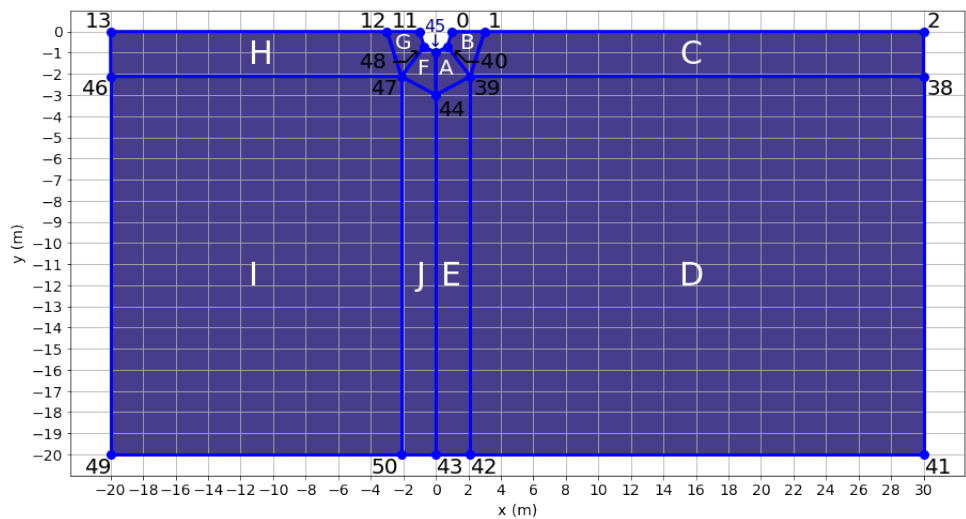


(a) Plano *back up*

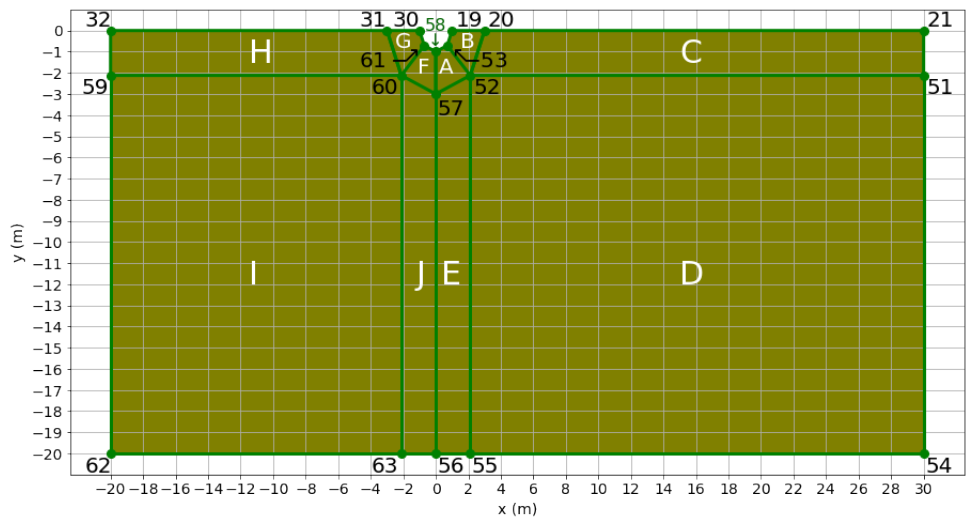


(b) Plano *front up*

Figura 1.9 Planos *up*



(a) Plano back down



(b) Plano front down

Figura 1.10 Planos down

14 ESCOAMENTO EM TORNO DE UM CILINDRO

```
hex (12 47 48 11 31 60 61 30) ($Ntang $Nnorm $Nz) simpleGrading (1 0.5 1)

// H-H
hex (13 46 47 12 32 59 60 31) ($Ntang $Nfront $Nz) simpleGrading (1 0.15 1)

// I-I
hex (46 49 50 47 59 62 63 60) ($Nfront $Nfront $Nz) simpleGrading (7 0.15
1)

// J-J
hex (47 50 43 44 60 63 56 57) ($Nfront $Ntang $Nz) simpleGrading (7 1 1)

);
```

Terminal 1.1 Construção da geometria e da malha

```
>> OpenFOAM/cylinderFlow blockMesh
>> OpenFOAM/cylinderFlow paraFoam
```

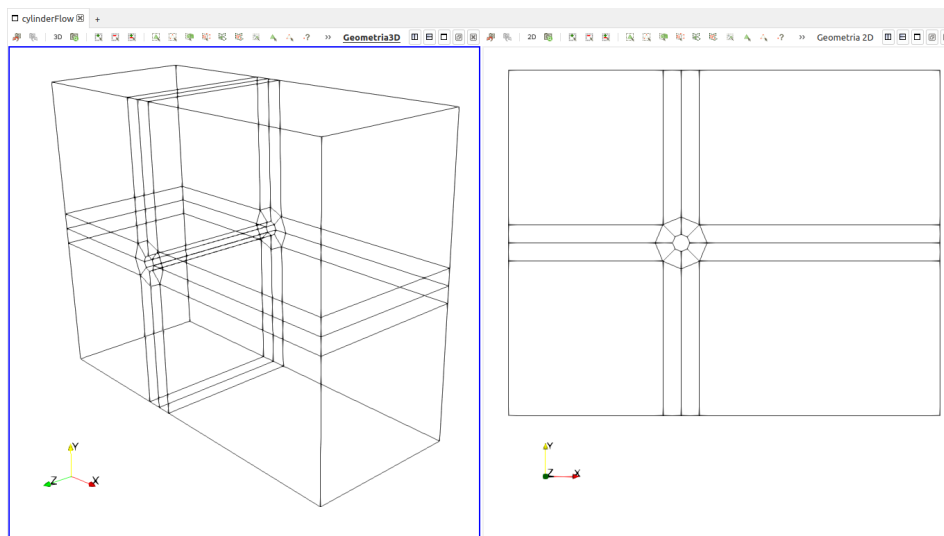


Figura 1.11 Geometria em torno do corpo cilíndrico no OPENFOAM

No entanto, para a construção da malha é usado o mesmo Terminal 1.1, mas agora, com as divisões da malha dada pelos valores declarados no código 1.1, além de um acréscimo do código 1.4 nesse código, para a declaração de novas variáveis e, f, g, h que serão novas coordenadas (e, f) e (g, h) a serem usadas no código 1.5 para transformar as arestas, que formam o raio do cilindro e do bloco, em arcos.

Na figura 1.12, é mostrado um *paraView* da configuração de duas malhas do campo de escoamento próximo ao cilindro. A malha-1 (cor azul) foi construída sem a transformação das arestas em arcos, enquanto a malha-2 (cor preta) foi construída com essa transformação. Nota-se claramente que a malha-2 é mais ajustada ao cilindro e ao bloco do que a malha-1, uma vez que as arestas aderem mais a geometria do cilindro no formato curvo em comparação ao formato linear. Deste modo, na Figura 1.13 é mostrado um *paraView* bidimensional da malha em torno do corpo cilíndrico.

Código 1.4 Definição de valores e cálculos no BlockMeshDict para a construção de arcos

```

/ Arcos
halfAngle      #calc "($Angle)/2";
radHalfAngle   #calc "degToRad($halfAngle)";

// Cilindro
e              #calc "$radius*cos($radHalfAngle)"; // Eixo x
eNeg           #calc "-1.0*$e";                  // Eixo -x
f              #calc "$radius*sin($radHalfAngle)"; // Eixo y
fNeg           #calc "-1.0*$f";                  // Eixo -y

// Bloco
g              #calc "$radiusBlock*cos($radHalfAngle)"; // Eixo x
gNeg           #calc "-1.0*$g";                  // Eixo -x
h              #calc "$radiusBlock*sin($radHalfAngle)"; // Eixo y
hNeg           #calc "-1.0*$h";                  // Eixo -y

```

Código 1.5 Definição dos arcos no BlockMeshDict

```

edges
(
    //up
    arc 0 5 ($e $f $minZ)
    arc 5 10 ($f $e $minZ)
    arc 19 24 ($e $f $maxZ)
    arc 24 29 ($f $e $maxZ)
    arc 11 16 ($eNeg $f $minZ)
    arc 16 10 ($fNeg $e $minZ)
    arc 30 35 ($eNeg $f $maxZ)
    arc 35 29 ($fNeg $e $maxZ)

    arc 1 4 ($g $h $minZ)
    arc 4 9 ($h $g $minZ)
    arc 20 23 ($g $h $maxZ)
    arc 23 28 ($h $g $maxZ)
    arc 12 15 ($gNeg $h $minZ)
    arc 15 9 ($hNeg $g $minZ)
    arc 31 34 ($gNeg $h $maxZ)
    arc 34 28 ($hNeg $g $maxZ)

    //down
    arc 0 40 ($e $fNeg $minZ)
    arc 40 45 ($f $eNeg $minZ)
    arc 19 53 ($e $fNeg $maxZ)
    arc 53 58 ($f $eNeg $maxZ)
    arc 11 48 ($eNeg $fNeg $minZ)
    arc 48 45 ($fNeg $eNeg $minZ)
    arc 30 61 ($eNeg $fNeg $maxZ)
    arc 61 58 ($fNeg $eNeg $maxZ)

    arc 1 39 ($g $hNeg $minZ)
    arc 39 44 ($h $gNeg $minZ)
    arc 20 52 ($g $hNeg $maxZ)
    arc 52 57 ($h $gNeg $maxZ)
    arc 12 47 ($gNeg $hNeg $minZ)
    arc 47 44 ($hNeg $gNeg $minZ)

```

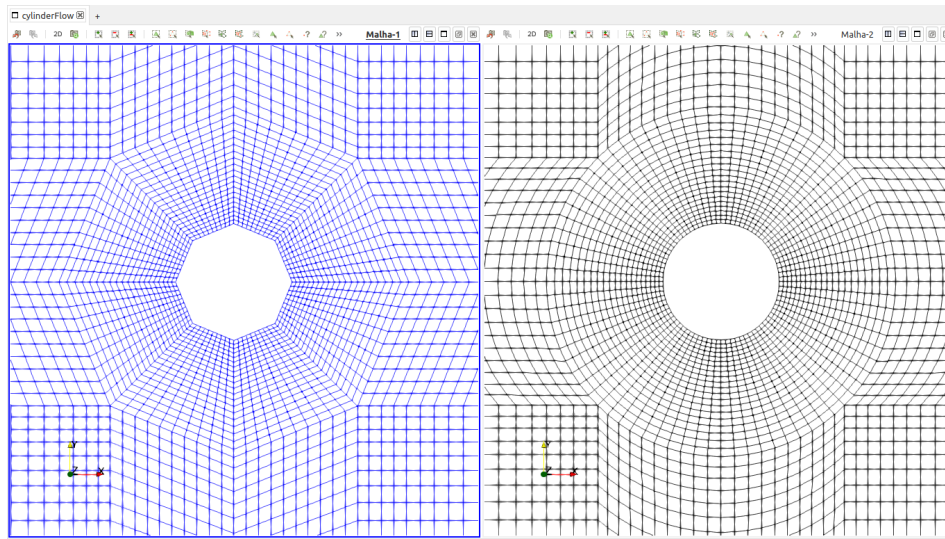


Figura 1.12 Configurações de duas malhas próximas ao cilindro no OPENFOAM

```
arc 31 60 ($gNeg $hNeg $maxZ)
arc 60 57 ($hNeg $gNeg $maxZ)
```

);

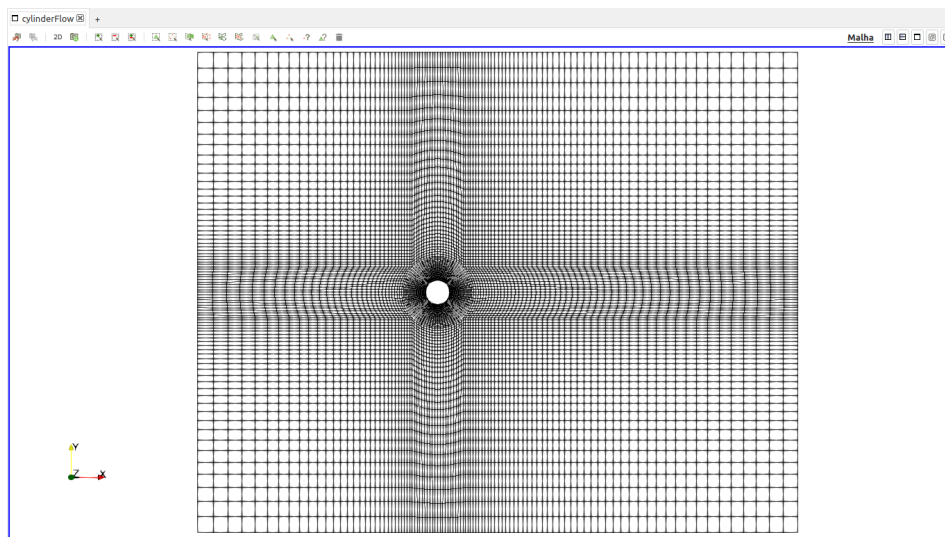


Figura 1.13 Malha em torno do corpo cilíndrico no OPENFOAM

Desta forma, com a malha construída, as fronteiras topológicas são especificadas na geometria do problema físico proposto, como esquematizado na Figura 1.14, por meio do Código 1.6 no arquivo `blockMeshDict`. Logo, porém, as fronteiras criadas são: `out`, `sym1`, `sym2`, `in`, `cylinder`, `back` e `front`. Essas fronteiras são definidas automaticamente no arquivo

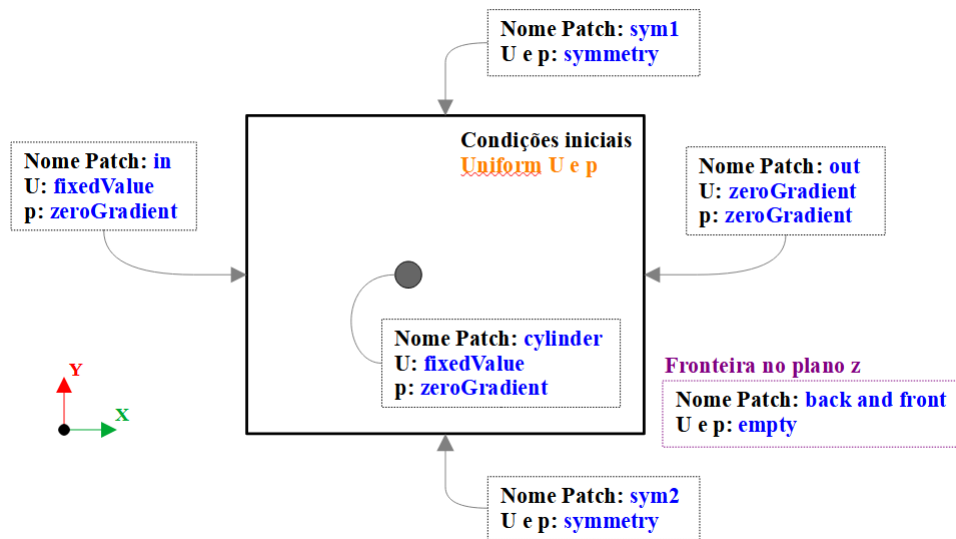


Figura 1.14 Fronteiras topológicas no OPENFOAM

boundary localizado no diretório `constant/polyMesh` quando a malha é gerada pelo arquivo `BlockMeshDict`. Ademais, é importante saber essas definições, visto que é por meio delas que são estabelecidas as condições inicial e de contorno das variáveis envolvidas no problema físico, no caso, a velocidade e a pressão.

Código 1.6 Definição das fronteiras topológicas no `blockMeshDict`

```
patches
(
    patch in
    (
        //up
        (14 13 32 33)
        (17 14 33 36)
        //down
        (13 46 59 32)
        (46 49 62 59)
    )
    patch out
    (
        //up
        (2 3 22 21)
        (3 6 25 22)
        //down
        (38 2 21 51)
        (41 38 51 54)
    )
    wall cylinder
    (
        //up
        (10 5 24 29)
        (5 0 19 24)
        (16 10 29 35)
```

```

        (11 16 35 30)
        //down
        (40 45 58 53)
        (0 40 53 19)
        (45 48 61 58)
        (48 11 30 61)
    )
    symmetryPlane sym1
    (
        (7 8 27 26)
        (6 7 26 25)
        (8 18 37 27)
        (18 17 36 37)
    )

    symmetryPlane sym2
    (
        (43 42 55 56)
        (42 41 54 55)
        (50 43 56 63)
        (49 50 63 62)
    )
    empty back
    (
        //up
        (5 10 9 4)
        (0 5 4 1)
        (1 4 3 2)
        (4 7 6 3)
        (9 8 7 4)
        (16 15 9 10)
        (11 12 15 16)
        (12 13 14 15)
        (14 17 18 15)
        (15 18 8 9)
        //down
        (45 40 39 44)
        (40 0 1 39)
        (1 2 38 39)
        (39 38 41 42)
        (44 39 42 43)
        (48 45 44 47)
        (11 48 47 12)
        (13 12 47 46)
        (46 47 50 49)
        (47 44 43 50)
    )
    empty front
    (
        //up
        (24 23 28 29)
        (20 23 24 19)
        (21 22 23 20)
        (22 25 26 23)
        (26 27 28 23)
        (28 34 35 29)
        (34 31 30 35)
        (33 32 31 34)
        (36 33 34 37)
        (37 34 28 27)
        //down

```

```

(53 58 57 52)
(19 53 52 20)
(21 20 52 51)
(51 52 55 54)
(52 57 56 55)
(58 61 60 57)
(61 30 31 60)
(31 32 59 60)
(60 59 62 63)
(57 60 63 56)
)
);

mergePatchPairs
(
);

```

Por conseguinte, no Terminal 1.2, é executado o comando **checkMesh** para realizar uma verificação da qualidade geométrica e topológica da malha, cuja uma das finalidades é observar se está tudo certo com o direcionamento dos vetores normais em relação as faces que formam a geometria.

Terminal 1.2 Verificação da malha

```
>> OpenFoam/cylinderFlow checkMesh
```

1.3 Configurações do escoamento

Contudo, com a malha construída no arquivo `blockMeshDict`, a próxima etapa de pré-processamento é a definição das propriedades do fluido, do regime de escoamento (laminar ou turbulento) e das condições iniciais e de contornos das variáveis.

As propriedades do fluido são especificadas no arquivo `transportProperties` no diretório `constant`. Nesse arquivo, é definido o modelo do fluido (Newtonian) e sua viscosidade cinemática ($\nu = \text{nu}$), como mostrado no Código 1.7.

Código 1.7 Definição das propriedades do fluido no `transportProperties`

```

transportModel  Newtonian;
nu              nu [ 0 2 -1 0 0 0 0 ] 0.025;

```

No mesmo diretório, no arquivo `turbulenceProperties`, é definido o regime de escoamento o qual pode ser laminar ou turbulento. Neste tutorial é especificado como laminar, como mostrado no Código 1.8, no entanto, caso seja necessário analisar o regime de turbulência, existem duas opções de modelos de turbulência: RAS ou LES.

A opção RAS refere-se aos modelos de turbulência baseados nas equações de Navier-Stokes de médias no tempo (*Reynolds Average Simulation*) e LES são os modelos de grande escala (*Large Eddy Simulation*). Além do mais, existem diversos modelos de turbulência do tipo RAS no OPEN-FOAM, porém, neste tutorial os seguintes serão investigados: `kEpsilon` e `kOmegaSST`. Os modelos do tipo LES não serão investigados neste tutorial. No Código 1.9, é mostrado o arquivo `turbulenceProperties` para o caso turbulento, com a especificação do modelo de turbulência: RAS, e por meio de um subdicionário chamado RAS é definido o tipo de modelo: `kEpsilon`.

Código 1.8 Definição do regime de escoamento laminar no `turbulenceProperties`

```
simulationType laminar;
```

Código 1.9 Definição do regime de escoamento turbulento no turbulenceProperties

```
simulationType RAS;

RAS
{
    RASModel      kEpsilon;

    turbulence     on;

    printcoeffs    on;
}
```

A condição de contorno e inicial dos campos de velocidade (U) e pressão (p) são especificadas em um arquivo para cada variável no diretório 0 do caso. O Código 1.10 apresenta a especificação da condição de contorno e inicial do campo de velocidade.

A primeira definição do arquivo é o vetor que representa a unidade da grandeza (m/s), em seguida, o valor da velocidade dentro do domínio é inicializado como uniforme e nulo. As condições de contorno são especificadas para cada fronteira a partir do nome escolhido na etapa de construção de malha. As sete fronteiras criadas, neste caso, foram: out, sym1, sym2, in, cylinder, back e front. Cada fronteira será especificada através de um tipo (type) e um valor (value).

O tipo fixedValue é uma condição de contorno de primeiro tipo onde o valor da variável é prescrita. Enquanto, o tipo zeroGradient atribui o valor da variável do volume de controle vizinho para a face da fronteira, ou seja, isto significa que o gradiente da pressão normal a face da fronteira é nulo. Por fim, o tipo symmetryPlane é uma condição de plano de simetria (condição de deslizamento). Por fim, o tipo empty é usado pois o problema é bidimensional.

O valor da velocidade na fronteira de entrada (entrada do campo de escoamento, in) é $U_e = 1\text{ m/s}$ e no entorno do cilindro (cylinder) é $U_c = 0\text{ m/s}$, enquanto na saída (saída do campo de escoamento, out) é atribuído o valor ... o tipo zeroGradient e na parte superior e inferior ... symmetryPlane. Por fim, o problema é bidimensional assim, a fronteira front e Back é do tipo empty

Código 1.10 Arquivo de condição inicial/contorno para velocidade (U)

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

internalField    uniform (1 0 0);

boundaryField
{
    in
    {
        type      fixedValue;
        value      uniform (1 0 0);
    }
    out
    {
        type      zeroGradient;
    }
    cylinder
    {
```

```

        type          fixedValue;
        value          uniform (0 0 0);
    }
    sym1
    {
        type          symmetryPlane;
    }
    sym2
    {
        type          symmetryPlane;
    }
    back
    {
        type          empty;
    }
    front
    {
        type          empty;
    }
}

```

O Código 1.11, apresenta a especificação da condição de contorno e inicial do campo de pressão. Note que o campo de pressão especificado na solução do escoamento incompressível pelo **simpleFoam** é um campo de pressão relativa (manométrica) dividido pela massa específica do fluido, ou seja, a pressão é na verdade p/ρ . Por este motivo, a unidade do campo de pressão é m^2/s^2 . A condição de contorno adotada para a pressão nas fronteiras in, out e cylinder é do tipo zeroGradient, nas fronteiras sym1 e sym2 é do tipo symmetryPlane e por fim, nas fronteiras front e Back é do tipo empty.

Código 1.11 Arquivo de condição inicial/contorno para a pressão relativa (p)

```

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

internalField        uniform 0;

boundaryField
{
    in
    {
        type          zeroGradient;
    }
    out
    {
        type          zeroGradient;
    }
    cylinder
    {
        type          zeroGradient;
    }
    sym1
    {
        type          symmetryPlane;
    }
    sym2
    {
        type          symmetryPlane;
    }
    back
    {
        type          empty;
    }
}

```

```

    }
    front
    {
        type          empty;
    }
}

```

1.4 Configurações da solução numérica

A próxima etapa trata-se da definição dos procedimentos numéricos para a solução do modelo matemático que representa o problema físico. Essa etapa é dividida em três partes a serem especificadas, sendo: O controle da simulação, a discretização e interpolação dos termos da EDP, e a solução do sistema algébrico. Essas definições são realizadas, respectivamente, pelos arquivos `controlDict`, `fvSchemes` e `fvSolution` no diretório `system` do caso.

No arquivo `controlDict`, apresentado no Código ??, é realizada a especificação do `Solver` de acordo com as características do escoamento (estacionário, incompressível, laminar e newtoniano), que no caso, como visto, é o **simpleFoam**, além disso, é definido o tempo de simulação (0 à 600 segundos), o número de diretórios durante esse intervalo de tempo a serem gerados (de 100 em 100 segundos, 6 diretórios), o número de algarismos significativos, entre outros.

Código 1.12 `controlDict`

```

application      simpleFoam;

startFrom        startTime;

startTime        0;

stopAt           endTime;

endTime          600;

deltaT           1;

writeControl      timeStep;

writeInterval     10;

purgeWrite        0;

writeFormat       ascii;

writePrecision    6;

writeCompression off;

timeFormat        general;

timePrecision     6;

runTimeModifiable true;

```

Ademais, é definido também, as funções a serem calculadas, que no caso do escoamento em torno de um corpo cilindro serão calculadas a força de arrasto e sustentação além dos seus respectivos coeficientes adimensionais.

Código 1.13 controlDict

```

functions
{
    force
    {
        type            forces;
        libs             ("libforces.so");
        writeControl     timeStep;
        writeInterval    1;
        log              yes;
        patches          (cylinder);
        rho              rhoInf;
        rhoInf           1;
        CofR             (0 0 0);
    }

    forceCoef
    {
        type            forceCoeffs;
        libs             ("libforces.so");
        writeControl     timeStep;
        writeInterval    1;
        log              yes;
        rho              rhoInf;
        rhoInf           1;
        patches          (cylinder);
        liftDir          (0 1 0);
        dragDir          (1 0 0);
        CofR             (0 0 0);
        pitchAxis        (0 0 1);
        magUInf          1;
        lRef              1; //comp de referencia para calculo do Cm
        Aref              1; //area projetada do cilindro = D*Z
    }
}

```

das discretizações a serem realizadas nos termos temporal, difusivo, advectivo e fonte, incluindo a interpolação dos termos difusivo e advectivo dado que o OPENFOAM usa o método de volumes finitos.

Código 1.14 fvSchemes

```

ddtSchemes
{
    default            steadyState;
}

gradSchemes
{
    default            Gauss linear;
}

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

```

```

laplacianSchemes
{
    default          Gauss linear corrected;
}

interpolationSchemes
{
    default          linear;
}

snGradSchemes
{
    default          corrected;
}

wallDist
{
    method meshWave;
}

```

E também do método de solução do sistema algébrico formado pela discretização do modelo matemático

Código 1.15 fvSolution

```

solvers
{
    p
    {
        solver          GAMG;
        smoother         GaussSeidel;
        tolerance        1e-6;
        relTol           0.1;
    }

    U
    {
        solver          smoothSolver;
        smoother         symGaussSeidel;
        tolerance        1e-6;
        relTol           0.1;
    }
}

SIMPLE
{
    /*residualControl
    {
        p                1e-5;
        U                1e-5;
    }*/
    nNonOrthogonalCorrectors 0;
    pRefCell                0;
    pRefValue                0;
}

```

Código 1.16 fvSolution

```

potentialFlow
{
    nNonOrthogonalCorrectors 10;
}

relaxationFactors
{
    fields
    {
        p            0.3;
    }
    equations
    {
        U            0.7;
    }
}

```

1.5 Solução do escoamento

1.5.1 Convergência numérica

Código 1.17 gnuplot dos resíduo inicial dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)

```

reset

set encoding utf8
set terminal postscript eps enhanced color "Times-Roman" 18
set autoscale
set lmargin 13
set bmargin 4

set output "residuals.eps"

set ylabel "{/*1.5{/Italic Resíduo inicial}" offset 1,0
set xlabel "{/*1.5{/Italic t(s)} }" offset 0,0

set logscale y
set format y "%.1e"

plot "logs/Ux_0" using ($1):($2) title "{U_x}" with lines lt 1 lw 2.5, \
     "logs/Uy_0" using ($1):($2) title "{U_y}" with lines lt 2 lw 2.5, \
     "logs/p_1"  using ($1):($2) title "{p}" with lines lt 3 lw 2.5

```

Código 1.18 gnuplot do coeficiente de arrasto dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)

```

reset

set terminal postscript eps enhanced color "Times-Roman" 18
set autoscale
set lmargin 13
set bmargin 4

set output "Cd.eps"

```

```

set xlabel "{/*1.5{\\Italic t(s)}" offset 1,0
set ylabel "{/*1.5{\\Italic C_d} }" offset 0,0

set xrange [10:200]
set key left

plot "postProcessing/forceCoef/0/forceCoeffs.dat" using ($1):($3) title "{
C_d}" with lines lt 3 lw 2.5

```

Código 1.19 gnuplot da força de arrasto dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)

```

reset

set terminal postscript eps enhanced color "Times-Roman" 18
set autoscale
set lmargin 13
set bmargin 4

set output "Fd.eps"

set xlabel "{/*1.5{\\Italic t(s)}" offset 1,0
set ylabel "{/*1.5{\\Italic F_d (N)} }" offset 0,0

set xrange [10:200]
set key left

plot "postProcessing/force/0/forces.dat" using ($1):($2+$5) title "{F_x}"
with lines lt 3 lw 2.5

```

Código 1.20 gnuplot da força de sustentação dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)

```

reset

set terminal postscript eps enhanced color "Times-Roman" 18
set autoscale
set lmargin 13
set bmargin 4

set output "Fl.eps"

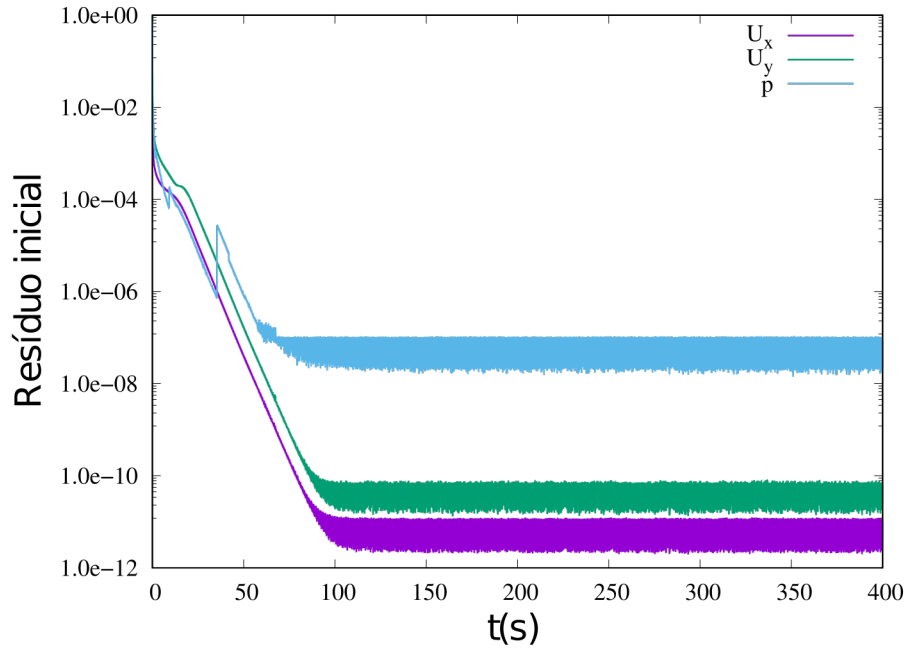
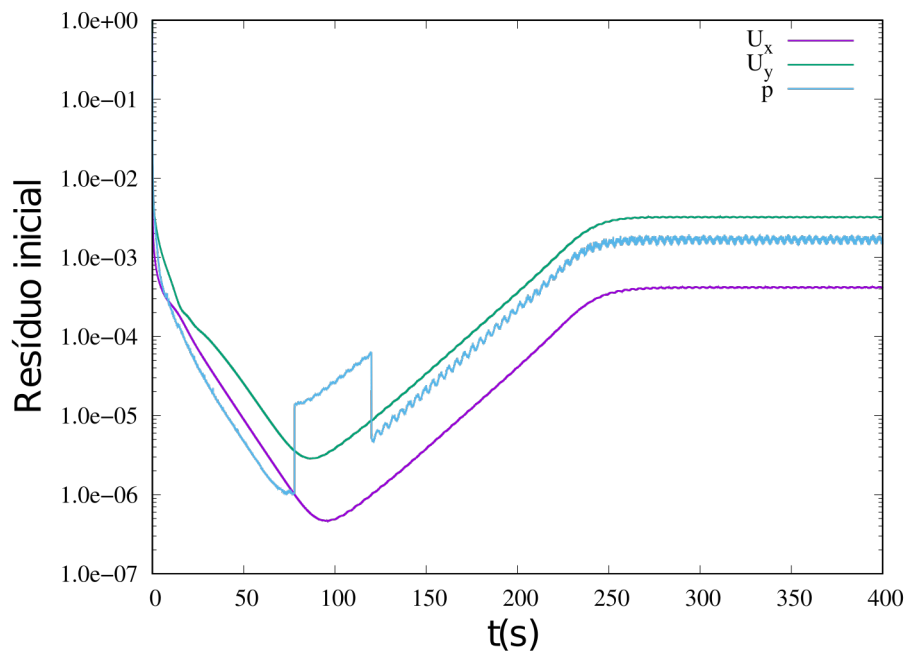
set xlabel "{/*1.5{\\Italic t(s)}" offset 1,0
set ylabel "{/*1.5{\\Italic F_l (N)} }" offset 0,0

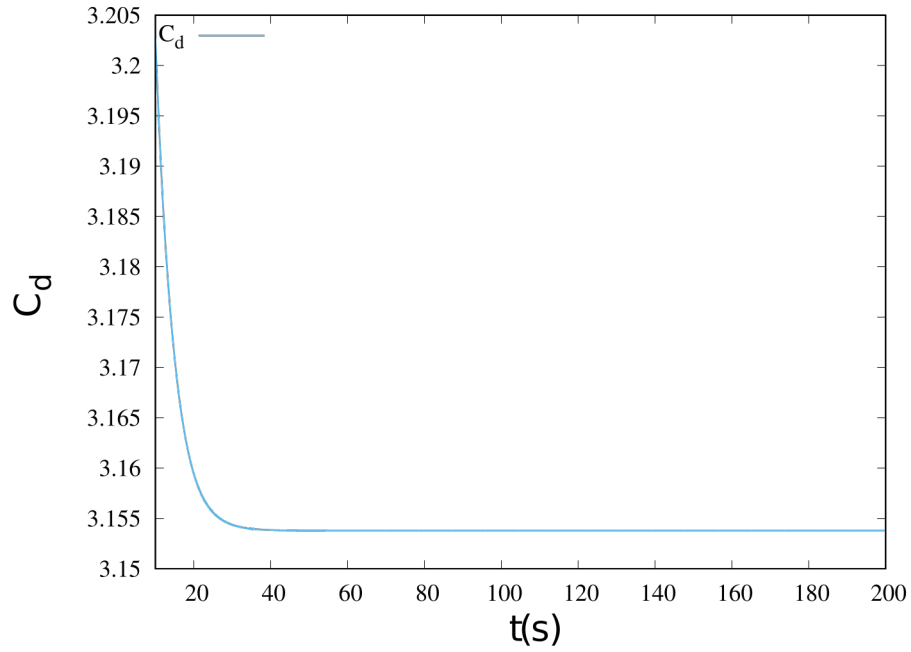
set xrange [10:200]
set key right

plot "postProcessing/force/0/forces.dat" using ($1):($3+$6) title "{F_y}"
with lines lt 2 lw 2.5

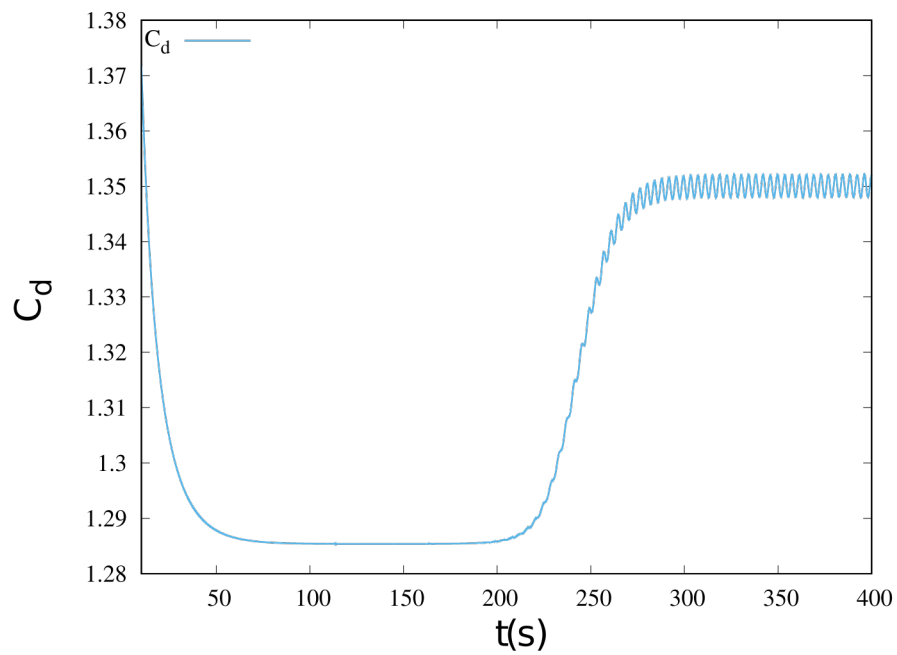
```

1.6 Análise dos resultados

(a) $Re = 10$ (b) $Re = 100$ **Figura 1.15** Resíduo inicial dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)

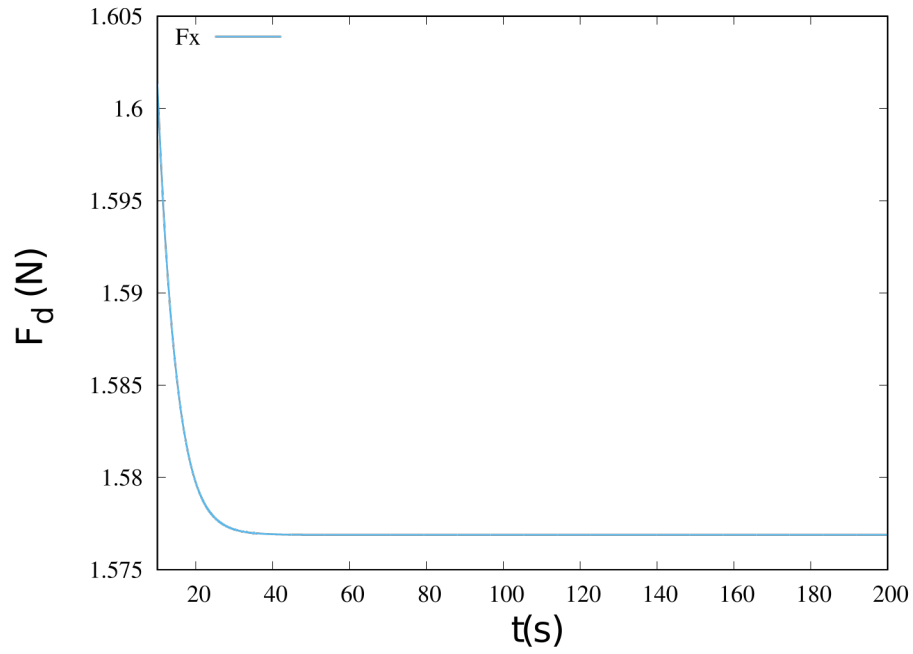
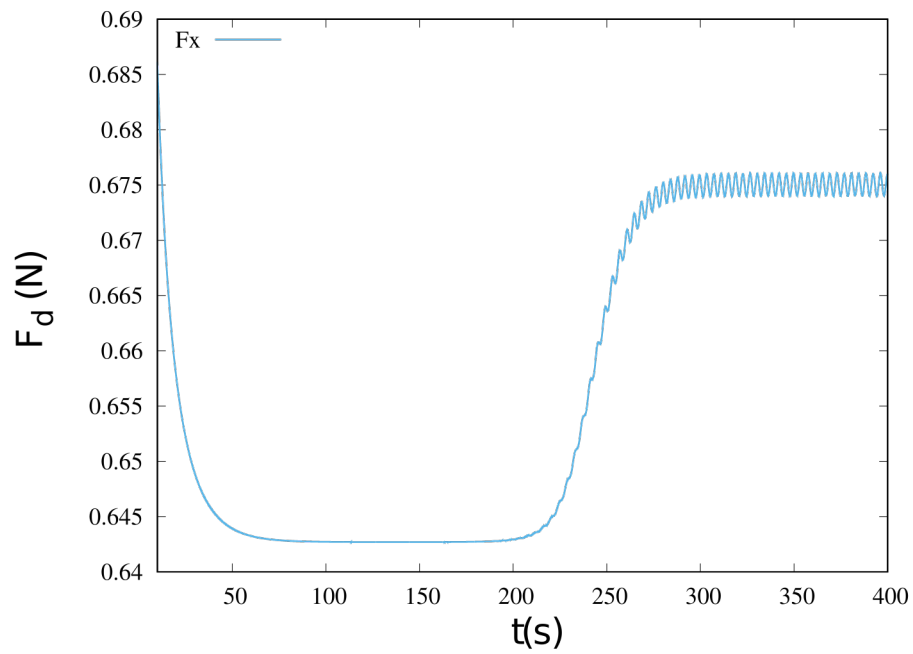


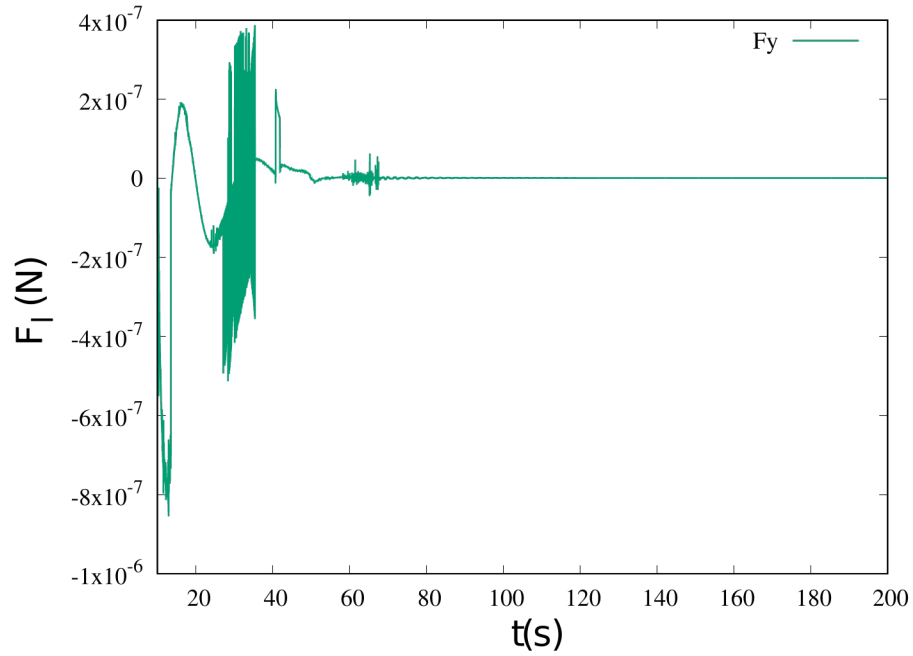
(a) $Re = 10$



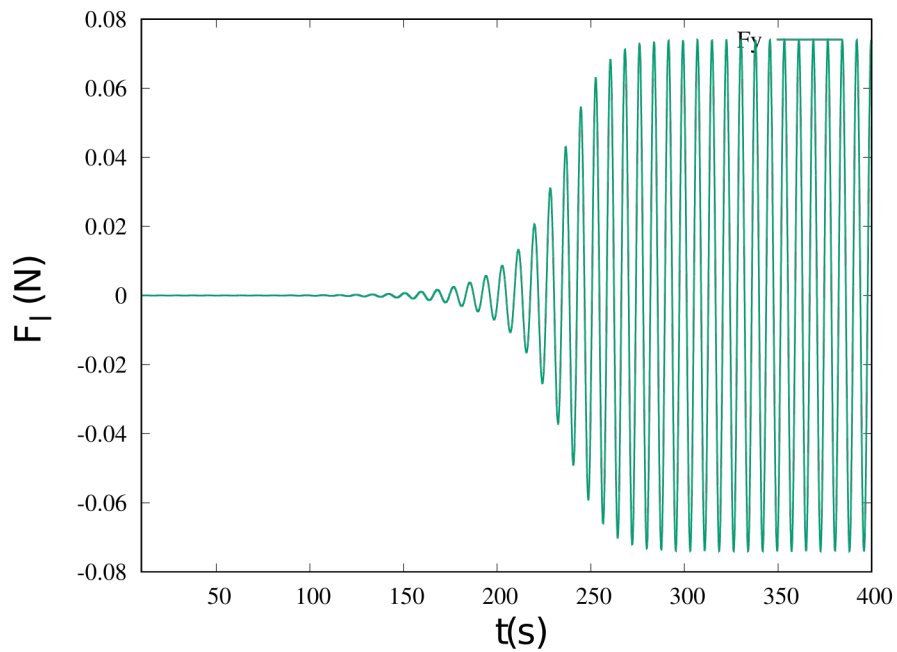
(b) $Re = 100$

Figura 1.16 Coeficiente de arrasto dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)

(a) $Re = 10$ (b) $Re = 100$ **Figura 1.17** Força de arrasto dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)



(a) $Re = 10$



(b) $Re = 100$

Figura 1.18 Força de sustentação dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)

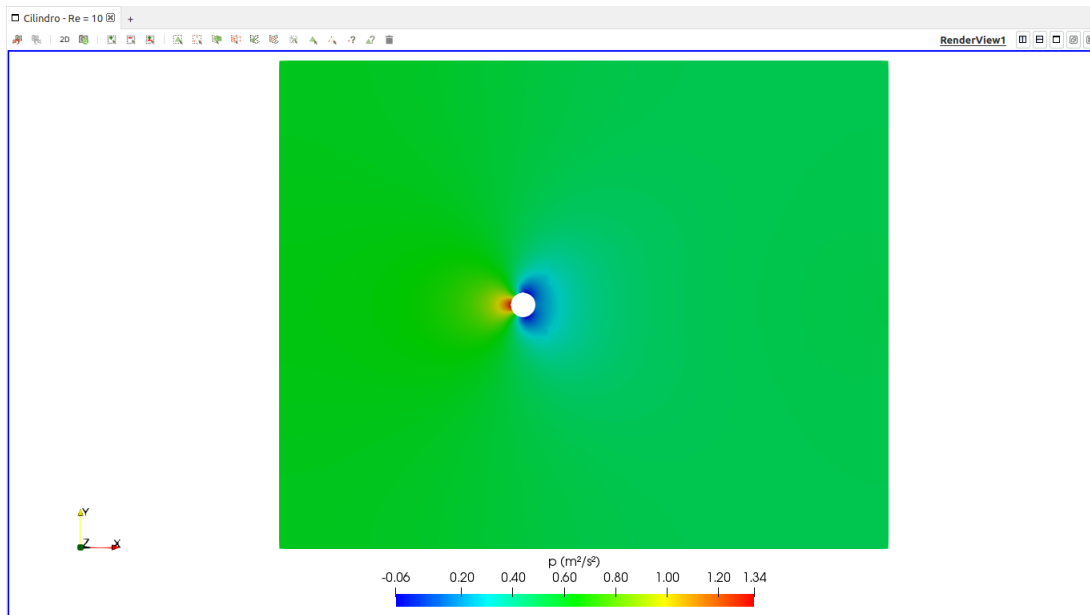
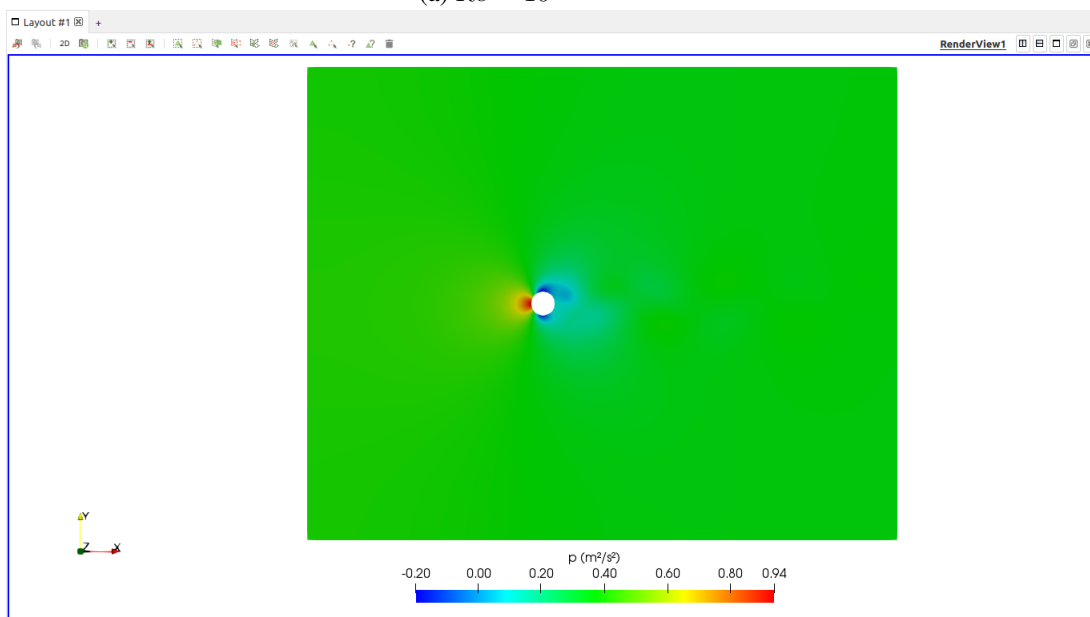
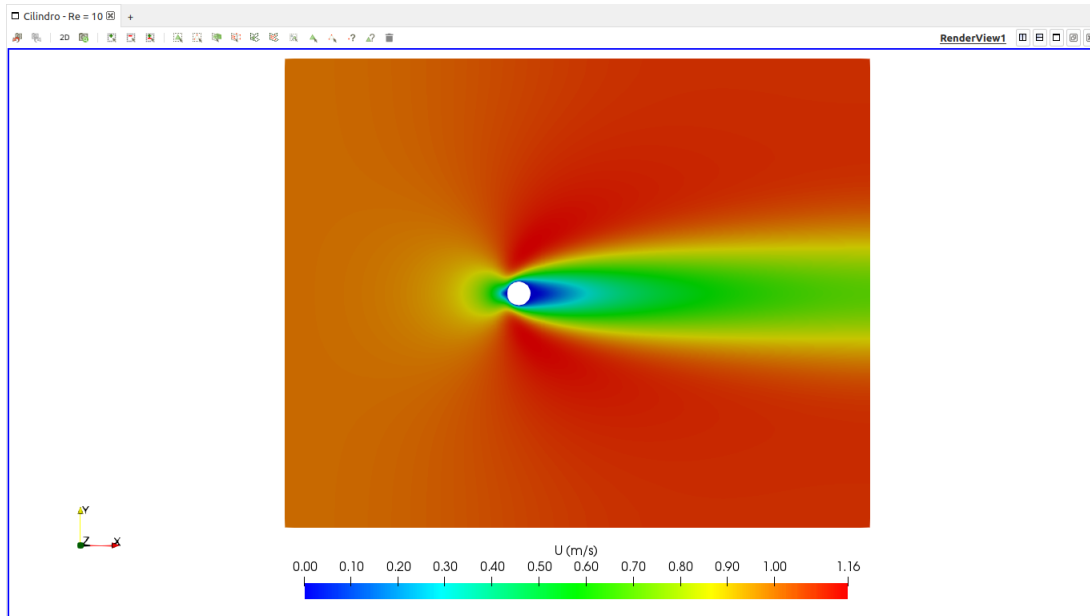
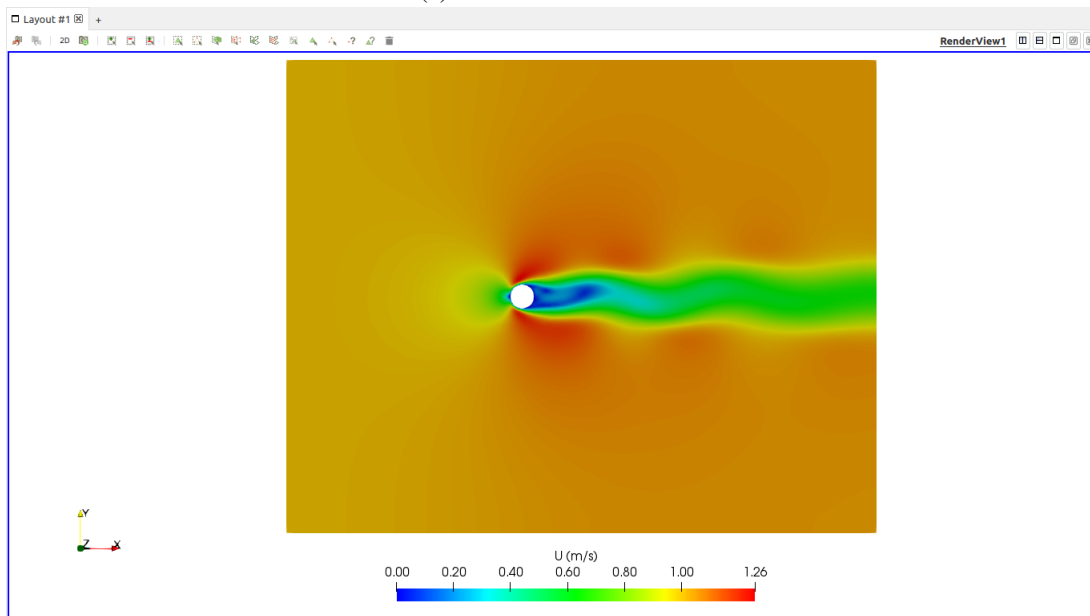
(a) $Re = 10$ (b) $Re = 100$

Figura 1.19 ParaView do campo de pressão dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)



(a) $Re = 10$



(b) $Re = 100$

Figura 1.20 ParaView do campo de velocidade dos escoamentos estacionário ($Re = 10$) e transiente ($Re = 100$)

EXERCÍCIOS

1.1 Execute um caso com as seguintes especificações:

Apresente os seguintes resultados:

- a) convergência de malha
- b) Demonstra a convergência do caso
- c) convergência de malha
- d) Linhas de corrente

1.2

Referências Bibliográficas

- [1] Brodkey, Robert S and Hershey, Harry C., Transport Phenomena: a Unified Approach, McGraw-Hill Chemical Engineering Series, 1988.
- [2] Lapple, C. E. and Shepherd, C. B., CALCULATION OF PARTICLE TRAJECTORIES, Industrial & Engineering Chemistry, Vol. 32,605–617, 1940.
- [3] Prieve, Dennis, Advanced Fluid Mechanics with Vector Field Theory, Carnegie Mellon University, 2016.
- [4] Çengel, Y. A.; Cimbala, J. M. Mecânica dos fluidos: fundamentos e aplicações. 3 ed. Porto Alegre, RS: AMGH, 2015.