



Universidade Federal
do Rio de Janeiro
Escola Politécnica

INVESTIGAÇÃO DOS CAMPOS DE VELOCIDADE, VORTICIDADE E
FUNÇÃO CORRENTE EM ESCOAMENTO LIVRE UTILIZANDO O
MÉTODO DE ELEMENTOS FINITOS

Juan de Oliveira dos Santos

Projeto de Graduação apresentado ao Curso de Engenharia Mecânica da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Gustavo Rabello dos Anjos

Daniel Barbedo Vasconcelos Santos

Rio de Janeiro

Junho de 2022



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Departamento de Engenharia Mecânica

DEM/POLI/UFRJ



INVESTIGAÇÃO DOS CAMPOS DE VELOCIDADE, VORTICIDADE E
FUNÇÃO CORRENTE EM ESCOAMENTO LIVRE UTILIZANDO O
MÉTODO DE ELEMENTOS FINITOS

Juan de Oliveira dos Santos

PROJETO FINAL SUBMETIDO AO CORPO DOCENTE DO DEPARTAMENTO DE ENGENHARIA MECÂNICA DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO MECÂNICO.

Aprovada por:

Prof. Gustavo Rabello dos Anjos, Ph.D.

Daniel Barbedo Vasconcelos Santos, M.Sc.

Prof. Roney Leon Thompson, D.Sc.

Prof. Fábio da Costa Figueiredo, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

JUNHO DE 2022

dos Santos, Juan de Oliveira

Investigação dos campos de velocidade, vorticidade e função corrente em escoamento livre utilizando o Método de Elementos Finitos/ Juan de Oliveira dos Santos. – Rio de Janeiro: UFRJ/Escola Politécnica, 2022.

XV, 113 p. 29, 7cm.

Orientadores: Gustavo Rabello dos Anjos

Daniel Barbedo Vasconcelos Santos

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Mecânica, 2022.

Referências Bibliográficas: p. 107 – 108.

1. Formulação Corrente-Vorticidade. 2. Método de Elementos Finitos. 3. Fluidodinâmica Computacional. I. dos Anjos, Gustavo Rabello *et al.*. II. Universidade Federal do Rio de Janeiro, UFRJ, Curso de Engenharia Mecânica. III. Investigação dos campos de velocidade, vorticidade e função corrente em escoamento livre utilizando o Método de Elementos Finitos.

*À minha mãe pelo dom da vida e
pelo amparo ao longo desses anos.
À meu tio Marcelo (in memo-
riam).*

Agradecimentos

Agradeço à minha família pelo suporte emocional e financeiro na minha trajetória acadêmica. À meus amigos que, por incontáveis vezes, tiveram que ouvir sobre minha paixão pela Engenharia e a descrição minuciosa de qualquer fenômeno físico que eu já tivesse estudado na faculdade. À meus mentores no meio profissional e pessoal, Rodrigo e Paulo, que me ensinaram os valores de persistência, excelência, companheirismo e que podemos sempre ser maiores que o problema que enfrentamos. À meus companheiros de classe, Bernardo e Fernando, com os quais dividi desafios, ensinamentos e aventuras que sempre resultaram em bons aprendizados e, em sua grande parte, vitórias. Por fim, à minha companheira de vida, Marina, que sempre esteve ao meu lado em toda a minha trajetória acadêmica e pessoal, me aconselhando e apoiando.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Mecânico

INVESTIGAÇÃO DOS CAMPOS DE VELOCIDADE, VORTICIDADE E
FUNÇÃO CORRENTE EM ESCOAMENTO LIVRE UTILIZANDO O
MÉTODO DE ELEMENTOS FINITOS

Juan de Oliveira dos Santos

Junho/2022

Orientadores: Gustavo Rabello dos Anjos

Daniel Barbedo Vasconcelos Santos

Programa: Engenharia Mecânica

O presente projeto de graduação tem como objetivo o desenvolvimento de um código em linguagem *Python* para a análise do escoamento livre ao redor de geometrias variadas. Para esse fim, foi utilizado o Método de Elementos Finitos para a discretização espacial e temporal das equações que regem a Formulação Corrente-Vorticidade, uma abordagem física que deriva das Equações de Navier-Stokes. Para a validação das simulações, casos bem estabelecidos na literatura foram simulados e os resultados obtidos foram comparados com curvas analíticas e dados de maior confiabilidade. A partir da validação, o código numérico foi utilizado para simular outros onze casos, cada qual com sua especificidade, e fazer uma análise comparativa de seus resultados. Os resultados obtidos foram satisfatórios e o modelo numérico se mostrou condizente com a natureza física esperada, de forma que o objetivo do trabalho foi alcançado.

Palavras-chave: Formulação Corrente-Vorticidade, Método de Elementos Finitos, Fluidodinâmica Computacional.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Mechanical Engineer

INVESTIGATION OF THE VELOCITY, VORTICITY AND
STREAMFUNCTION FIELDS OF A FREE FLOW USING THE FINITE
ELEMENTS METHOD

Juan de Oliveira dos Santos

June/2022

Advisors: Gustavo Rabello dos Anjos

Daniel Barbedo Vasconcelos Santos

Department: Mechanical Engineering

The present graduation project aimed at the development of a numerical code in *Python* language to analyze the two-dimensional free flow around a variety of geometries. To this end, Finite Elements Method was used to spacially and temporally discretize the equations that govern the Streamfunction-Vorticity Formulation, a physical approach that derives from the Navier-Stokes equations. In order to validate the simulations, well-stablished cases were simulated and the results were compared to the analytical curves and trustworthy data. From the validation, eleven simulations were carried out, each one with its own specification, and the final results were compared. The results obtained in this work were satisfactory and the numerical model showed proper result in comparison with the expected physical nature, so that the goal of this work was achieved

Keywords: Streamfunction-Vorticity Formulation, Finite Elements Method, Computational Fluid Dynamics.

Sumário

Lista de Figuras	x
Lista de Tabelas	xv
1 Introdução	1
2 Revisão Bibliográfica	5
2.1 Equações de governo	5
2.1.1 Equação da Continuidade	5
2.1.2 Equação de conservação da quantidade de movimento	7
2.2 Função corrente	10
2.3 Formulação corrente-vorticidade	11
2.4 Método de Elementos Finitos	13
2.4.1 Método dos resíduos ponderados	13
2.4.2 Método de Galerkin	16
2.4.3 Forma matricial	19
2.4.4 Malha computacional	21
2.4.5 Discretização temporal	22
2.4.6 Adimensionalização	23
3 Metodologia numérica	27
3.1 Geração de malha	27
3.2 Código numérico	28
3.2.1 Importação de bibliotecas e definição de parâmetros de simulação	29
3.2.2 Importação da malha	30

3.2.3	Criação das matrizes globais	31
3.2.4	Atribuição das condições iniciais e de contorno	32
3.2.5	<i>Loop</i> temporal	35
3.2.6	Exportação para pós-processamento	39
3.3	Pós-processamento	40
4	Validação do código numérico	42
4.1	Caso Poiseuille	42
4.2	Caso Lid-driven	44
5	Resultados	47
5.1	Escoamento ao redor de corpo quadrado	49
5.2	Escoamento ao redor de corpo triangular	54
5.2.1	Caso A: Triângulo equilátero contra o escoamento	55
5.2.2	Caso B: Triângulo isósceles contra o escoamento	59
5.2.3	Caso C: Triângulo equilátero alinhado ao escoamento	64
5.2.4	Caso D: Triângulo isósceles alinhado ao escoamento	68
5.3	Escoamento ao redor do aerofólio NACA 0012	72
5.3.1	Caso A: NACA 0012 com angulo de ataque 0°	73
5.3.2	Caso B: NACA 0012 com angulo de ataque 8°	77
5.3.3	Caso C: NACA 0012 com angulo de ataque 20°	82
5.4	Escoamento ao redor do aerofólio NACA 4415	87
5.4.1	Caso A: NACA 4415 com angulo de ataque 0°	87
5.4.2	Caso B: NACA 4415 com angulo de ataque 8°	92
5.4.3	Caso C: NACA 4415 com angulo de ataque 20°	97
5.5	Discussão dos resultados	102
6	Conclusões	105
	Referências Bibliográficas	107
A	Código Fonte	109

Lista de Figuras

1.1	Exemplo de simulação CFD para duas diferentes geometrias (SU2) . . .	2
2.1	Representação de duas possíveis funções de forma (ANJOS, 2021) . . .	17
2.2	Representação do elemento triangular (LEWIS, 2004)	21
3.1	Exemplo de malha gerada pelo <i>software Gmsh</i>	28
3.2	Fluxograma do algoritmo computacional	29
3.3	Matrizes IEN, X e Y de uma malha computacional (ANJOS, 2021) . .	31
3.4	Fluxograma do processo iterativo no tempo	35
3.5	Interface gráfica do <i>ParaView</i> exibindo o campo de velocidade de uma simulação	41
4.1	Escoamento sob placas planas (PONTES, 2009)	42
4.2	Resultados da simulação do caso <i>Poiseuille</i>	43
4.3	Resultados da simulação do caso <i>Poiseuille</i>	44
4.4	Escoamento em cavidade (CARLOS MARCHI, 2009)	44
4.5	Resultados da simulação do caso <i>Lid-driven</i>	45
4.6	Resultados de velocidade em X da simulação do caso <i>Lid-driven</i> . . .	46
4.7	Resultados de velocidade em Y da simulação do caso <i>Lid-driven</i> . . .	46
5.1	Condições iniciais e de contorno considerando uma figura central genérica.	48
5.2	Domínio e condições de contorno do caso	49
5.3	Malha computacional do caso	49
5.4	Detalhe da malha computacional do caso na região de interesse	50
5.5	Evolução temporal dos campos de velocidade u e v	51
5.6	Destaque da região onde o corte foi executado	51

5.7	Evolução temporal da função corrente ψ	52
5.8	Evolução temporal do campo de vorticidade ω	52
5.9	Evolução temporal das linhas de corrente	52
5.10	Velocidade ilustrando esteiras de von Kármán para Reynolds elevado	53
5.11	Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado	54
5.12	Domínio e condições de contorno do caso	55
5.13	Malha computacional do caso	55
5.14	Detalhe da malha computacional do caso na região de interesse	56
5.15	Evolução temporal dos campos de velocidade u e v	56
5.16	Destaque da região onde o corte foi executado	57
5.17	Evolução temporal da função corrente ψ	57
5.18	Evolução temporal do campo de vorticidade ω	57
5.19	Evolução temporal das linhas de corrente	58
5.20	Velocidade ilustrando esteiras de von Kármán para Reynolds elevado	59
5.21	Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado	59
5.22	Malha computacional do caso	60
5.23	Detalhe da malha computacional do caso na região de interesse	60
5.24	Evolução temporal dos campos de velocidade u e v	61
5.25	Destaque da região onde o corte foi executado	61
5.26	Evolução temporal da função corrente ψ	62
5.27	Evolução temporal do campo de vorticidade ω_z	62
5.28	Evolução temporal das linhas de corrente	62
5.29	Velocidade ilustrando esteiras de von Kármán para Reynolds elevado	63
5.30	Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado	63
5.31	Malha computacional do caso	64
5.32	Detalhe da malha computacional do caso na região de interesse	64
5.33	Evolução temporal dos campos de velocidade u e v	65
5.34	Destaque da região onde o corte foi executado	65
5.35	Evolução temporal da função corrente ψ	66
5.36	Evolução temporal do campo de vorticidade ω_z	66
5.37	Evolução temporal das linhas de corrente	67
5.38	Velocidade ilustrando esteiras de von Kármán para Reynolds elevado	67

5.39	Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado	68
5.40	Malha computacional do caso	68
5.41	Detalhe da malha computacional do caso na região de interesse	69
5.42	Evolução temporal dos campos de velocidade u e v	69
5.43	Destaque da região onde o corte foi executado	70
5.44	Evolução temporal da função corrente ψ	70
5.45	Evolução temporal do campo de vorticidade ω_z	71
5.46	Evolução temporal das linhas de corrente	71
5.47	Velocidade ilustrando esteiras de von Kármán para Reynolds elevado	72
5.48	Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado	72
5.49	Domínio e condições de contorno do caso	73
5.50	Malha computacional do caso	73
5.51	Detalhe da malha computacional do caso na região de interesse	74
5.52	Evolução temporal dos campos de velocidade u e v	74
5.53	Destaque da região onde o corte foi executado	75
5.54	Evolução temporal da função corrente ψ	75
5.55	Evolução temporal do campo de vorticidade ω	75
5.56	Evolução temporal das linhas de corrente	76
5.57	Velocidade do escoamento para Reynolds elevado	77
5.58	Vorticidade do escoamento para Reynolds elevado	77
5.59	Malha computacional do caso	78
5.60	Detalhe da malha computacional do caso na região de interesse	78
5.61	Evolução temporal dos campos de velocidade u e v	79
5.62	Destaque da região onde o corte foi executado	79
5.63	Evolução temporal da função corrente ψ	80
5.64	Evolução temporal do campo de vorticidade ω	80
5.65	Evolução temporal das linhas de corrente	80
5.66	Velocidade do escoamento para Reynolds elevado	81
5.67	Vorticidade do escoamento para Reynolds elevado	82
5.68	Malha computacional do caso	82
5.69	Detalhe da malha computacional do caso na região de interesse	83
5.70	Evolução temporal dos campos de velocidade u e v	84

5.71	Destaque da região onde o corte foi executado	84
5.72	Evolução temporal da função corrente ψ	85
5.73	Evolução temporal do campo de vorticidade ω	85
5.74	Evolução temporal das linhas de corrente	85
5.75	Velocidade do escoamento para Reynolds elevado	86
5.76	Vorticidade do escoamento para Reynolds elevado	86
5.77	Domínio e condições de contorno do caso	87
5.78	Malha computacional do caso	88
5.79	Detalhe da malha computacional do caso na região de interesse	88
5.80	Evolução temporal dos campos de velocidade u e v	89
5.81	Destaque da região onde o corte foi executado	89
5.82	Evolução temporal da função corrente ψ	90
5.83	Evolução temporal do campo de vorticidade ω	90
5.84	Evolução temporal das linhas de corrente	91
5.85	Velocidade do escoamento para Reynolds elevado	91
5.86	Vorticidade do escoamento para Reynolds elevado	92
5.87	Malha computacional do caso	92
5.88	Detalhe da malha computacional do caso na região de interesse	93
5.89	Evolução temporal dos campos de velocidade u e v	93
5.90	Destaque da região onde o corte foi executado	94
5.91	Evolução temporal da função corrente ψ	94
5.92	Evolução temporal do campo de vorticidade ω	94
5.93	Evolução temporal das linhas de corrente	95
5.94	Velocidade do escoamento para Reynolds elevado	96
5.95	Vorticidade do escoamento para Reynolds elevado	96
5.96	Malha computacional do caso	97
5.97	Detalhe da malha computacional do caso na região de interesse	97
5.98	Evolução temporal dos campos de velocidade u e v	98
5.99	Destaque da região onde o corte foi executado	98
5.100	Evolução temporal da função corrente ψ	99
5.101	Evolução temporal do campo de vorticidade ω	99
5.102	Evolução temporal das linhas de corrente	100

5.103	Velocidade do escoamento para Reynolds elevado	101
5.104	Vorticidade do escoamento para Reynolds elevado	101

Lista de Tabelas

5.1	Parâmetros da simulação	48
5.2	Avaliação das variáveis para comparação quantitativa	53
5.3	Avaliação das variáveis para comparação quantitativa	58
5.4	Avaliação das variáveis para comparação quantitativa	63
5.5	Avaliação das variáveis para comparação quantitativa	67
5.6	Avaliação das variáveis para comparação quantitativa	71
5.7	Avaliação das variáveis para comparação quantitativa	76
5.8	Avaliação das variáveis para comparação quantitativa	81
5.9	Avaliação das variáveis para comparação quantitativa	86
5.10	Avaliação das variáveis para comparação quantitativa	91
5.11	Avaliação das variáveis para comparação quantitativa	95
5.12	Avaliação das variáveis para comparação quantitativa	100
5.13	Avaliação comparativa das variáveis em cada caso	102
5.14	Avaliação comparativa com destaque em laranja	102
5.15	Avaliação comparativa com destaque em ciano	103
5.16	Avaliação comparativa com destaque em azul	104

Capítulo 1

Introdução

A dinâmica dos fluidos é uma importante área de pesquisa na comunidade científica e, em especial, na engenharia. Isso se deve a dois principais fatores: o primeiro, devido ao fato que fenômenos dominados pela fluidodinâmica estão largamente presentes no cotidiano de inúmeras indústrias tecnológicas, como no automobilismo, aviação, meteorologia, química, térmica, entre outros; o segundo, devido ao fato de que o equacionamento dos fenômenos responsáveis por reger o movimento dos fluidos é de grande complexidade e sua principal equação, a equação de Navier-Stokes, permanece sem solução até os dias atuais.

A principal alternativa utilizada na predição desses fenômenos, largamente utilizada na atualidade, é o desenvolvimento de protótipos em escala reduzida que simulem fisicamente o comportamento do sistema real. Muito embora essa alternativa apresente resultados satisfatórios e coerentes com as necessidades da indústria tecnológica, sua aplicação prática demanda a construção minuciosa de modelos e inúmeras simulações, o que em geral representa um alto investimento financeiro.

A partir do avanço no poder computacional, que sofreu abrupta ascensão desde o início do século XXI, iniciou-se o desenvolvimento de técnicas digitais avançadas que, a partir de modelos físicos e matemáticos desenvolvidos para obter uma solução aproximada das equações que regem a dinâmica dos fluidos, atuam na simulação numérica de domínios físicos através de algoritmos computacionais. A esse conjunto de técnicas, foi atribuído o nome Fluidodinâmica Computacional (ou CFD, *Computational Fluid Dynamics*).

Dentro da gama de possibilidades apresentadas pelas técnicas CFD, dois métodos

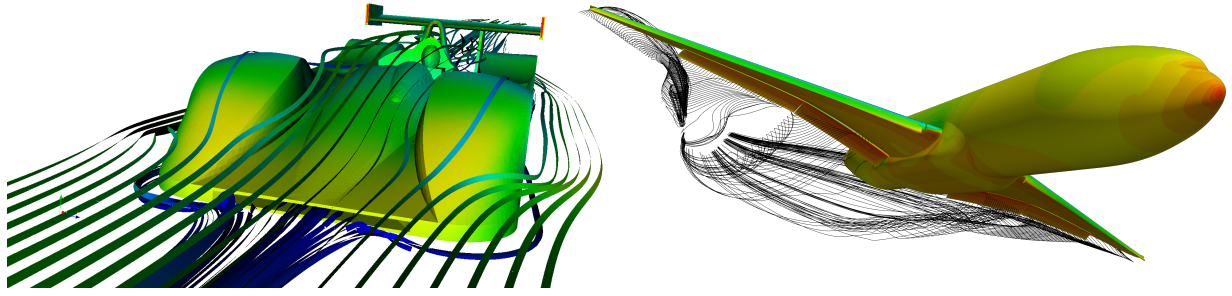


Figura 1.1: Exemplo de simulação CFD para duas diferentes geometrias (SU2)

destacam-se devido a sua grande difusão no meio acadêmico e ampla utilização por empresas de engenharia: o Método dos Volumes Finitos (FVM, *Finite Volume Method*) e o Método de Elementos Finitos (FEM, *Finite Element Method*). O presente trabalho tem como objetivo o desenvolvimento de um algoritmo numérico utilizando o FEM capaz de simular um escoamento bidimensional de um fluido monofásico, incompressível e newtoniano, analisando sua interação com diversas geometrias em um meio contínuo. Para tal fim, o modelo matemático utilizado para aproximar a equação de Navier-Stokes será a Formulação Corrente-Vorticidade, que permite o desacoplamento entre velocidade e pressão, diminuindo consideravelmente o esforço computacional necessário para a simulação. A discretização espacial foi realizada utilizando o método dos resíduos ponderados através do Método de Galerkin, enquanto a discretização temporal utiliza um esquema de diferenças finitas progressivo de primeira ordem.

A validação do código foi realizada através da simulação de casos clássicos de engenharia que tem seu comportamento bem definido na literatura. Os resultados numéricos obtidos em simulação são comparados com curvas analíticas ou com simulações numéricas de alta confiabilidade no meio científico. Os casos de validação escolhidos foram o Escoamento entre Placas Paralelas (Escoamento de Poiseuille), o Escoamento em Cavidade com Tampa Deslizante (ou Escoamento Lid-driven).

A malha computacional é uma representação discreta do domínio real a ser simulado. A construção das malhas utilizadas no presente trabalho contará com o *software Gmsh*, ferramenta que dispõe de diversas funcionalidades para o controle na criação dos elementos.

A implementação do código utilizado na criação do algoritmo numérico contou com a linguagem de programação *Python*. O *Python* conta com uma ampla gama de bibliotecas, como o *Numpy* e o *Meshio*, que possibilitam a manipulação de operações envolvendo álgebra linear e a leitura da malha computacional, imprescindíveis para a obtenção dos resultados.

O pós-processamento, procedimento utilizado para analisar os dados provenientes da simulação computacional, contará com a o *software ParaView*, que dispõe de diversas ferramentas para a visualização gráfica das variáveis, além de permitir a confecção de gráficos e geração de imagens.

O estudo foi executado em 11 (onze) geometrias distintas, que tiveram como objetivo observar o comportamento do escoamento em diferentes corpos e ângulos de ataque. São elas:

- Corpo quadrado
- Corpo triangular equilátero contra escoamento
- Corpo triangular equilátero alinhado ao escoamento
- Corpo triangular isósceles contra escoamento
- Corpo triangular isósceles alinhado ao escoamento
- Corpo de aerofólio NACA 0012 com ângulo de ataque 0°
- Corpo de aerofólio NACA 0012 com ângulo de ataque 8°
- Corpo de aerofólio NACA 0012 com ângulo de ataque 20°
- Corpo de aerofólio NACA 4415 com ângulo de ataque 0°
- Corpo de aerofólio NACA 4415 com ângulo de ataque 8°
- Corpo de aerofólio NACA 4415 com ângulo de ataque 20°

O presente trabalho foi organizado em 6 capítulos. O capítulo 2 apresenta uma revisão bibliográfica, explorando de maneira minuciosa as equações de governo do problema, o equacionamento da Formulação Corrente-Vorticidade e a aplicação do Método de Elementos Finitos.

O capítulo 3 conta com a metodologia numérica utilizada, descrevendo o procedimento de geração de malha, o desenvolvimento linha a linha do código numérico e o pós-processamento.

O capítulo 4 introduz a validação do código numérico, englobando todos os casos de validação, as condições e parâmetros de sua simulação e o processamento dos resultados, comparando com resultados já estabelecidos na literatura.

O capítulo 5 é composto pelos resultados da simulação das 11 geometrias citadas, apresentando imagens, gráficos e tabelas dos dados obtidos.

O capítulo 6 conta com a conclusão do trabalho, agregando reflexões sobre os resultados obtidos e apresentando sugestões para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

2.1 Equações de governo

Essa seção visa apresentar as principais equações diferenciais utilizadas na modelagem matemática de um escoamento. A derivação dessas equações respeita a hipótese do contínuo, que estabelece que o fluido em questão é homogêneo em qualquer parcela infinitesimal de seu volume. Dessa forma, torna-se possível modelar o escoamento a partir dos princípios gerais de conservação:

- Conservação da massa
- Conservação da quantidade de movimento

2.1.1 Equação da Continuidade

A equação da continuidade provém do princípio de conservação da massa, que versa sobre a variação da massa em determinado local do espaço. Determinando o volume em análise como volume de controle (VC) e sua fronteira como superfície de controle (SC), pode-se dizer que (PONTES, 2009):

$$\left\{ \begin{array}{l} \text{Taxa de acúmulo de massa} \\ \text{dentro do volume de controle} \end{array} \right\} + \left\{ \begin{array}{l} \text{Fluxo líquido de massa} \\ \text{que atravessa a superfície de controle} \end{array} \right\} = 0$$

Segundo (YUNUS A. ÇENGEL, 2017), em termos matemáticos, a primeira parcela dessa soma pode ser reescrita como a integral da variação de massa em todo o volume de controle:

$$\int_{VC} \frac{\partial}{\partial t} (dm) \quad (2.1)$$

Considerando uma integral no volume, a taxa de variação da massa pode ser substituída utilizando a densidade ρ :

$$\int_{VC} \frac{\partial}{\partial t} (dm) = \int_{VC} \frac{\partial}{\partial t} (\rho dV) = \int_{VC} \frac{\partial \rho}{\partial t} dV \quad (2.2)$$

Para obter o fluxo de massa atravessando a superfície de controle, aplica-se o produto do vetor velocidade (\mathbf{U}) com o vetor normal à superfície S (\mathbf{n}):

$$\oint_{SC} \rho \mathbf{U} \cdot \mathbf{n} dS \quad (2.3)$$

Na qual $\mathbf{U} = (u, v, w)$ é o vetor velocidade. Unindo ambas as formulações, chegamos à Eq. (2.4):

$$\int_{VC} \frac{\partial \rho}{\partial t} dV + \oint_{SC} \rho \mathbf{U} \cdot \mathbf{n} dS = 0 \quad (2.4)$$

Segundo o Teorema de Gauss (PONTES, 2009), sendo \mathbf{G} um vetor genérico, tem-se:

$$\oint_S \mathbf{G} \cdot \mathbf{n} dS = \int_V \nabla \cdot \mathbf{G} dV \quad (2.5)$$

Aplicando o teorema de Gauss no segundo termo da Eq. (2.4) e unindo as integrais, resulta em:

$$\int_{VC} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) \right] dV = 0 \quad (2.6)$$

Sendo dV um volume arbitrário, a equação acima é verdadeira para qualquer volume, portanto:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (2.7)$$

A Eq. (2.7) é conhecida como equação da continuidade. Tomando como hipótese a incompressibilidade do fluido, a derivada temporal da densidade ρ é igualada a zero e essa equação pode ser simplificada para a Eq. (2.8):

$$\nabla \cdot \mathbf{U} = 0 \quad (2.8)$$

2.1.2 Equação de conservação da quantidade de movimento

De maneira semelhante à adotada na seção anterior, o princípio da conservação da quantidade de movimento pode ser equacionado de acordo com sua definição, postulada pela segunda lei de Newton e que afirma que a variação da quantidade de movimento é igual a soma de todas as forças que atuam externamente ao sistema. Considerando o mesmo volume e superfície de controle (VC) analisado anteriormente e seguindo o raciocínio apresentado em (PONTES, 2009), pode-se dizer que:

$$\left\{ \begin{array}{l} \text{Taxa de acúmulo de quantidade} \\ \text{de movimento dentro do volume} \\ \text{de controle} \end{array} \right\} + \left\{ \begin{array}{l} \text{Fluxo líquido de quantidade} \\ \text{de movimento que atravessa} \\ \text{a superfície de controle} \end{array} \right\} =$$

$$\left\{ \begin{array}{l} \text{Resultante das forças aplicadas} \\ \text{à superfície de controle} \end{array} \right\} + \left\{ \text{Resultante das forças de volume} \right\}$$

Em termos matemáticos, os termos da equação podem ser reescritos da seguinte forma:

$$\left\{ \begin{array}{l} \text{Taxa de acúmulo de quantidade} \\ \text{de movimento dentro do volume} \\ \text{de controle} \end{array} \right\} = \int_{VC} \frac{\partial}{\partial t} (\rho \mathbf{U}) dV$$

$$\left\{ \begin{array}{l} \text{Fluxo líquido de quantidade} \\ \text{de movimento que atravessa} \\ \text{a superfície de controle} \end{array} \right\} = \oint_{SC} (\rho \mathbf{U}) \mathbf{U} \cdot \mathbf{n} dS$$

$$\left\{ \begin{array}{l} \text{Resultante das forças aplicadas} \\ \text{à superfície de controle} \end{array} \right\} = \oint_{SC} \boldsymbol{\sigma} \cdot \mathbf{n} dS$$

$$\left\{ \text{Resultante das forças de volume} \right\} = \int_{VC} \rho \mathbf{g} dV$$

Os símbolos $\boldsymbol{\sigma}$ e \mathbf{g} representam, respectivamente, o tensor de tensões e o vetor de aceleração da gravidade. Vale comentar que apenas a gravidade foi considerada como força volumétrica. Unindo todos os termos em uma só equação, chega-se na Eq. (2.9):

$$\int_{VC} \frac{\partial}{\partial t} (\rho \mathbf{U}) dV + \oint_{SC} (\rho \mathbf{U}) \mathbf{U} \cdot \mathbf{n} dS = \oint_{SC} \boldsymbol{\sigma} \cdot \mathbf{n} dS + \int_{VC} \rho \mathbf{g} dV \quad (2.9)$$

De maneira análoga à apresentada na seção anterior, pode-se aplicar o Teorema de Gauss:

$$\int_{VC} \frac{\partial}{\partial t} (\rho \mathbf{U}) dV + \int_{VC} \nabla \cdot (\rho \mathbf{U} \mathbf{U}) dV = \int_{VC} \nabla \cdot \boldsymbol{\sigma} dV + \int_{VC} \rho \mathbf{g} dV \quad (2.10)$$

Organizando as variáveis, chega-se a:

$$\int_{VC} \left[\frac{\partial}{\partial t} (\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{g} \right] dV = 0 \quad (2.11)$$

Visto que dV é um volume de controle arbitrário, a equação é válida para qualquer volume de controle, de forma que:

$$\frac{\partial}{\partial t} (\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{g} = 0 \quad (2.12)$$

É possível desenvolver os termos positivos dessa equação:

$$\frac{\partial}{\partial t} (\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) \longrightarrow \rho \left[\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} \right] + \mathbf{U} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) \right] \quad (2.13)$$

Como já foi visto, o último termo dessa expressão é a equação da continuidade, tendo o valor numérico de zero. Dessa forma, a Eq. (2.12) pode ser reescrita:

$$\rho \left[\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} \right] = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} \quad (2.14)$$

É interessante decompor o tensor de tensões como uma soma de tensores, um devido à pressão p e outro devido à viscosidade $\boldsymbol{\tau}$:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} \quad (2.15)$$

Com \mathbf{I} sendo a matriz identidade. O tensor $\boldsymbol{\tau}$ é chamado de tensor de tensões viscosas e, considerando um fluido homogêneo, isotrópico e newtoniano, pode ser modelado dessa maneira:

$$\boldsymbol{\tau} = \mu[\nabla\mathbf{U} + (\nabla\mathbf{U})^T] \quad (2.16)$$

Na qual μ é a viscosidade dinâmica do fluido. Unindo as Eqs. (2.15) e (2.16) à Eq. (2.14):

$$\rho \left[\frac{\partial\mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla\mathbf{U} \right] = \nabla \cdot [-p\mathbf{I} + \mu(\nabla\mathbf{U} + (\nabla\mathbf{U})^T)] + \rho\mathbf{g} \quad (2.17)$$

Considerando que a viscosidade dinâmica μ é uma grandeza escalar que independe da posição:

$$\rho \left[\frac{\partial\mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla\mathbf{U} \right] = -\nabla p + \mu(\nabla \cdot \nabla\mathbf{U} + \nabla \cdot (\nabla\mathbf{U})^T) + \rho\mathbf{g} \quad (2.18)$$

Aplicando propriedades vetoriais:

$$\rho \left[\frac{\partial\mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla\mathbf{U} \right] = -\nabla p + \mu(\nabla^2\mathbf{U} + \nabla(\nabla \cdot \mathbf{U})^T) + \rho\mathbf{g} \quad (2.19)$$

Finalmente, aplica-se a equação da continuidade incompressível à Eq. (2.19):

$$\frac{\partial\mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla\mathbf{U} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{U} + \mathbf{g} \quad (2.20)$$

Na qual $\nu = \frac{\mu}{\rho}$ é conhecida como a viscosidade cinemática do fluido. Essa equação, que tem como hipótese um fluido newtoniano, é conhecida como a equação de Navier-Stokes, apresentada aqui para um fluido incompressível, homogêneo e isentrópico. Essa equação pode ser também reescrita utilizando o conceito de derivada material $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{U} \cdot \nabla$:

$$\frac{D\mathbf{U}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{U} + \mathbf{g} \quad (2.21)$$

2.2 Função corrente

Assim como é apresentado em (YUNUS A. ÇENGEL, 2017), considerando o caso de um escoamento incompressível e bidimensional no plano xy , pode-se reescrever a equação da continuidade:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2.22)$$

Entretanto, com o objetivo de diminuir o número de variáveis envolvidas, é possível representar as duas componentes da velocidade u e v utilizando uma única variável ψ :

$$u = \frac{\partial \psi}{\partial y} \quad v = -\frac{\partial \psi}{\partial x} \quad (2.23)$$

Essa nova variável é chamada de função corrente. Note que sua utilização respeita a Eq. (2.22):

$$\frac{\partial}{\partial x} \left(\frac{\partial \psi}{\partial y} \right) + \frac{\partial}{\partial y} \left(-\frac{\partial \psi}{\partial x} \right) = 0 \longrightarrow \frac{\partial^2 \psi}{\partial x \partial y} - \frac{\partial^2 \psi}{\partial y \partial x} = 0 \quad (2.24)$$

Ela recebe esse nome pois as curvas do escoamento onde ψ é constante são equivalentes às linhas de corrente do escoamento, ou seja, são curvas tangente ao vetor velocidade local em todos os pontos. Esse fato pode ser facilmente constatado através da definição matemática de linhas de corrente:

Ao longo de uma linha de corrente, $\frac{dy}{dx} = \frac{v}{u} \rightarrow udy - vdx = 0$ (2.25)

Aplicando a definição de função corrente à Eq. (2.25):

$$\frac{\partial\psi}{\partial y}dy + \frac{\partial\psi}{\partial x}dx = 0 \quad (2.26)$$

Sabe-se que para qualquer função suave ψ de duas variáveis x e y , a variação total de ψ entre dois pontos a uma distancia infinitesimal é:

$$d\psi = \frac{\partial\psi}{\partial y}dy + \frac{\partial\psi}{\partial x}dx \quad (2.27)$$

Unindo ambas as equações, chega-se a:

$$d\psi = 0 \quad (2.28)$$

Provando que ao longo de uma linha de corrente, a função corrente ψ é constante.

2.3 Formulação corrente-vorticidade

De acordo com (PANTON, 2013), vorticidade é uma grandeza física empregada para quantificar a rotação das partículas de um fluido em movimento. Matematicamente, define-se a vorticidade $\boldsymbol{\omega}$ como:

$$\boldsymbol{\omega} = \nabla \times \mathbf{U} \quad (2.29)$$

Considerando um escoamento bidimensional, as componentes ω_x e ω_y são nulas, restando apenas a componente em z :

$$\omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \quad (2.30)$$

Aplicando o operador rotacional em toda à equação de Navier-Stokes:

$$\nabla \times \left[\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} \right] = \nabla \times \left[-\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{U} + \mathbf{g} \right] \quad (2.31)$$

Para o caso bidimensional, considerando um vetor genérico $\mathbf{v} = (v_x, v_y)$, $\nabla \times \mathbf{v} = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}$. Aplicando essa operação em cada um dos termos da Eq. (2.31):

$$\begin{aligned} \frac{\partial}{\partial y} \left[\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right] - \frac{\partial}{\partial x} \left[\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right] = \\ \frac{\partial}{\partial y} \left[-\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \nabla^2 u + \mathbf{g} \right] - \frac{\partial}{\partial x} \left[-\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \nabla^2 v + \mathbf{g} \right] \end{aligned} \quad (2.32)$$

Reorganizando os termos a montante do sinal de igualdade:

$$\begin{aligned} \frac{\partial}{\partial t} \left[\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right] + u \frac{\partial}{\partial x} \left[\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right] + v \frac{\partial}{\partial y} \left[\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right] = \\ \frac{\partial \omega_z}{\partial t} + u \frac{\partial \omega_z}{\partial x} + v \frac{\partial \omega_z}{\partial y} = \frac{D\omega_z}{Dt} \end{aligned} \quad (2.33)$$

É interessante observar que vetor aceleração gravitacional \mathbf{g} e a viscosidade ν são constantes do problema, ao passo que a pressão p está sob a ação do operador gradiente. Assim como é indicado em (PANTON, 2013), pode-se aplicar as propriedades que o rotacional de um gradiente e o rotacional de uma constante são zero, reduzindo os termos a jusante do sinal de igualdade:

$$\begin{aligned} \nabla \times \left[-\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{U} + \mathbf{g} \right] = \nabla \times [\nu \nabla^2 \mathbf{U}] = \nu \left[\frac{\partial(\nabla^2 v)}{\partial x} - \frac{\partial(\nabla^2 u)}{\partial y} \right] \\ = \nu \left[\frac{\partial^2}{\partial x^2} \left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) + \frac{\partial^2}{\partial y^2} \left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right) \right] = \nu \left[\frac{\partial^2 \omega_z}{\partial x^2} + \frac{\partial^2 \omega_z}{\partial y^2} \right] = \nu \nabla^2 \omega_z \end{aligned} \quad (2.34)$$

Dessa forma, unindo a Eq. (2.33) à Eq. (2.34), chega-se à equação de transporte de vorticidade:

$$\frac{D\omega_z}{Dt} = \nu \nabla^2 \omega_z \quad (2.35)$$

Dentre as vantagens dessa formulação, a principal é o desacoplamento entre pressão e velocidade, permitindo maior facilidade na resolução numérica do escoamento, como será apresentado na seção seguinte.

Como foi visto na seção anterior, a função corrente também relaciona as componentes da velocidade. Dessa maneira, é possível aplicar a Eq. (2.23) à Eq. (2.30):

$$\omega_z = -\frac{\partial}{\partial x} \frac{\partial \psi}{\partial x} - \frac{\partial}{\partial y} \frac{\partial \psi}{\partial y} \longrightarrow \omega_z = -\nabla^2 \psi \quad (2.36)$$

A utilização das Eq.s (2.35) e (2.36) permitirá obter o perfil de velocidades do escoamento e, portanto, sua resolução será o objeto de estudo desse trabalho.

2.4 Método de Elementos Finitos

O Método de Elementos Finitos (MEF) é um procedimento numérico para a solução de equações diferenciais em determinado domínio a partir da subdivisão desse domínio em pequenas partições, chamadas de elementos. De maneira resumida, o método consiste na avaliação de determinadas variáveis de interesse em cada elemento discreto do domínio. A justaposição da variação dessas variáveis em cada elemento é utilizada para descrever todo o campo da variável no domínio (ANJOS, 2021).

Essa seção tem como objetivo apresentar a aplicação do MEF para a resolução do problema previamente apresentado.

2.4.1 Método dos resíduos ponderados

Para obter a solução da formulação corrente-vorticidade, é necessário resolver as seguintes equações:

$$\begin{cases} \frac{D\omega_z}{Dt} = \nu \nabla^2 \omega_z \\ \nabla^2 \psi = -\omega_z \\ \mathbf{U} = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \end{cases}, \text{ para todo } \Omega \quad (2.37)$$

Onde Ω é o domínio. É possível reorganizar essas equações de forma a igualá-las a zero:

$$\begin{cases} \frac{D\omega_z}{Dt} - \nu \nabla^2 \omega_z = 0 \\ \nabla^2 \psi + \omega_z = 0 \\ \mathbf{U} - \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) = 0 \end{cases} \quad (2.38)$$

O sistema (2.38) é conhecido como formulação forte do problema. Uma solução analítica dessas equações respeitaria rigorosamente essa igualdade. Entretanto, o MEF é um método numérico discreto, de forma que sua resolução produz um resultado aproximado. Dessa forma, assume-se que cada equação gerará um resíduo R intrínseco à aproximação:

$$\begin{cases} \frac{D\omega_z}{Dt} - \nu \nabla^2 \omega_z = R_1 \\ \nabla^2 \psi + \omega_z = R_2 \\ \mathbf{U} - \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) = R_3 \end{cases} \quad (2.39)$$

Com o objetivo de minimizar o resíduo, o método dos resíduos ponderados sugere a multiplicação do resíduo por uma função de ponderação ou função peso, de forma que a integral da ponderação no domínio é igual a zero:

$$\begin{cases} \int_{\Omega} \delta R_1 d\Omega = 0 \quad \rightarrow \int_{\Omega} \delta \left[\frac{D\omega_z}{Dt} - \nu \nabla^2 \omega_z \right] d\Omega = 0 \\ \int_{\Omega} \gamma R_2 d\Omega = 0 \quad \rightarrow \int_{\Omega} \gamma \left[\nabla^2 \psi + \omega_z \right] d\Omega = 0 \\ \int_{\Omega} \zeta R_3 d\Omega = 0 \quad \rightarrow \int_{\Omega} \zeta \left[\mathbf{U} - \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \right] d\Omega = 0 \end{cases} \quad (2.40)$$

δ , γ e ζ são as funções peso. A aplicação dessa integral força que o resíduo ponderado

total seja zero no domínio de estudo. Distribuindo as integrais do sistema (2.40) e separando a última equação:

$$\left\{ \begin{array}{l} \int_{\Omega} \delta \frac{D\omega_z}{Dt} d\Omega - \int_{\Omega} \delta \nu \nabla^2 \omega_z d\Omega = 0 \\ \int_{\Omega} \gamma \nabla^2 \psi d\Omega + \int_{\Omega} \gamma \omega_z d\Omega = 0 \\ \int_{\Omega} \zeta u d\Omega - \int_{\Omega} \zeta \frac{\partial \psi}{\partial y} d\Omega = 0 \\ \int_{\Omega} \zeta v d\Omega + \int_{\Omega} \zeta \frac{\partial \psi}{\partial x} d\Omega = 0 \end{array} \right. \quad (2.41)$$

A esse novo sistema, atribui-se o nome de formulação fraca, no qual as funções foram integradas no domínio contínuo. No sistema (2.41) existem variáveis que precisam ser deriváveis à segunda ordem. Entretanto, com o objetivo de prosseguir com o cálculo, é interessante diminuir a ordem de derivação dos termos. Isso pode ser feito aplicando o Teorema de Green nas integrais onde termos de segunda ordem aparecem:

$$\int_{\Omega} \delta \nu \nabla^2 \omega_z d\Omega = - \int_{\Omega} \nabla \delta \cdot \nu \nabla \omega_z d\Omega + \int_{\Gamma} \delta (\nu \nabla \omega_z \cdot \mathbf{n}) d\Gamma \quad (2.42)$$

$$\int_{\Omega} \gamma \nabla^2 \psi d\Omega = - \int_{\Omega} \nabla \gamma \cdot \nabla \psi d\Omega + \int_{\Gamma} \gamma (\nabla \psi \cdot \mathbf{n}) d\Gamma \quad (2.43)$$

Onde Γ é a fronteira do domínio Ω e \mathbf{n} é o vetor normal orientado para o exterior de Γ . Essa formulação é conhecida como formulação fraca final do sistema devido ao fato de que o termo de segunda ordem foi redistribuído entre as variáveis de interesse e as funções peso correspondentes.

As integrais na fronteira do domínio utilizam a avaliação das propriedades nesse local, ou seja, as condições de contorno do sistema. Para o problema em questão, todas as condições de contorno são dos tipos valor prescrito, ou Dirichlet, e fluxo prescrito nulo, ou Neumann homogêneo. Por definição, as integrais na fronteira Γ resultam em valor nulo em contornos com esses tipos de condição, resultando no seguinte sistema de equações:

$$\begin{cases} \int_{\Omega} \delta \frac{D\omega_z}{Dt} d\Omega + \int_{\Omega} \nabla \delta \cdot \nu \nabla \omega_z d\Omega = 0 \\ \int_{\Omega} \gamma \omega_z d\Omega - \int_{\Omega} \nabla \gamma \cdot \nabla \psi d\Omega = 0 \\ \int_{\Omega} \zeta u d\Omega - \int_{\Omega} \zeta \frac{\partial \psi}{\partial y} d\Omega = 0 \\ \int_{\Omega} \zeta v d\Omega + \int_{\Omega} \zeta \frac{\partial \psi}{\partial x} d\Omega = 0 \end{cases} \quad (2.44)$$

2.4.2 Método de Galerkin

Até agora, o método dos resíduos ponderados partiu do princípio que as equações da Formulação Corrente-Vorticidade serão resolvidas para determinado domínio Ω do espaço e que sua discretização acarretará em um resíduo numérico. Entretanto, nada foi dito sobre a maneira como esse domínio será discretizado. Para definir as variáveis de interesse em cada elemento e discreto do domínio, será utilizado um conjunto de funções chamado de função de forma N , que é uma função de interpolação válida para cada elemento do domínio:

$$\omega_z^e = \sum_{n=1}^{np} (\omega_z)_i N_i^e \quad (2.45)$$

$$\psi^e = \sum_{n=1}^{np} \psi_i N_i^e \quad (2.46)$$

$$u^e = \sum_{n=1}^{np} u_i N_i^e \quad (2.47)$$

$$v^e = \sum_{n=1}^{np} v_i N_i^e \quad (2.48)$$

Sendo np é o número de nós que forma cada elemento do domínio, formado por ne elementos. De forma análoga, para as funções peso:

$$\delta^e = \sum_{n=1}^{np} \delta_j N_j^e \quad (2.49)$$

$$\gamma^e = \sum_{n=1}^{np} \gamma_j N_j^e \quad (2.50)$$

$$\zeta^e = \sum_{n=1}^{np} \zeta_j N_j^e \quad (2.51)$$

Nessa formulação, ω_z^e , ψ^e , u^e e v^e são constantes a se determinar, enquanto N_i^e e N_j^e são as funções de forma, que podem ser modelados de diferentes formas. Em uma dessas modelagens, chamada de método de Galerkin, as funções de forma para as variáveis e para as funções peso são as mesmas, ou seja, $N_i^e = N_j^e$.

As funções de forma podem apresentar diferentes formulações, a depender da ordem desejada. É possível, por exemplo, utilizar funções polinomiais de diferentes graus para avaliar aquela variável dentro de determinado elemento a partir do valor constante determinado em seus nós, como apresentado na Figura 2.1, que representa essas funções do elemento linear e quadrático:

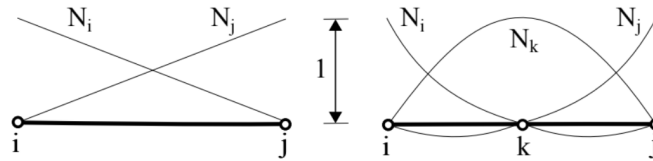


Figura 2.1: Representação de duas possíveis funções de forma (ANJOS, 2021)

A diferença de ambas está na ordem do polinômio interpolador. Por definição, as funções de forma devem receber o valor unitário no nó em questão e zero nos demais. Elementos de ordem mais baixa exigem um poder computacional menor quando comparados aos demais, sendo utilizados sempre que possível. Embora seja intuitivo concluir que elementos de ordem maior trarão resultados melhores, isso nem sempre é verdadeiro, como indica (ANJOS, 2021).

O próximo passo é aplicar essas definições no sistema (2.44):

$$\begin{cases}
\int_{\Omega^e} \sum_{i,j \in e} \delta_j N_j^e \frac{D}{Dt} [(\omega_z)_i N_i^e] d\Omega + \int_{\Omega^e} \sum_{i,j \in e} \nabla(\delta_j N_j^e) \cdot \nu \nabla [(\omega_z)_i N_i^e] d\Omega = 0 \\
\int_{\Omega^e} \sum_{i,j \in e} \gamma_j N_j^e (\omega_z)_i N_i^e d\Omega - \int_{\Omega^e} \sum_{i,j \in e} \nabla(\gamma_j N_j^e) \cdot \nabla(\psi_i N_i^e) d\Omega = 0 \\
\int_{\Omega^e} \sum_{i,j \in e} \zeta_j N_j^e u_i N_i^e d\Omega - \int_{\Omega^e} \sum_{i,j \in e} \zeta_j N_j^e \frac{\partial(\psi_i N_i^e)}{\partial y} d\Omega = 0 \\
\int_{\Omega^e} \sum_{i,j \in e} \zeta_j N_j^e v_i N_i^e d\Omega + \int_{\Omega^e} \sum_{i,j \in e} \zeta_j N_j^e \frac{\partial(\psi_i N_i^e)}{\partial x} d\Omega = 0
\end{cases} \quad (2.52)$$

É interessante notar que, agora, as integrais não estão sendo avaliadas em todo o domínio Ω , mas sim no domínio do elemento Ω^e . Também se faz necessário abrir a derivada material composta na primeira equação do sistema (2.52):

$$\begin{aligned}
\int_{\Omega^e} \sum_{i,j \in e} \delta_j N_j^e \frac{D}{Dt} [(\omega_z)_i N_i^e] d\Omega &\longrightarrow \int_{\Omega^e} \sum_{i,j \in e} \delta_j N_j^e \left[\frac{\partial(\omega_z)_i N_i^e}{\partial t} \right] d\Omega + \\
&\int_{\Omega^e} \sum_{i,j \in e} \delta_j N_j^e \mathbf{U} \cdot \nabla [(\omega_z)_i N_i^e] d\Omega
\end{aligned} \quad (2.53)$$

Por definição, as funções de forma N são constantes no tempo. Além disso, por hipótese, as variáveis locais $(\omega_z)_i$, u_i e v_i são constantes, permitindo reorganizar a primeira equação do sistema (2.52):

$$\sum_{i,j \in e} \delta_j \left[\frac{\partial(\omega_z)_i}{\partial t} \int_{\Omega^e} N_j^e N_i^e d\Omega + (\omega_z)_i \int_{\Omega^e} \mathbf{U} \cdot (N_j^e \nabla N_i^e) d\Omega + (\omega_z)_i \nu \int_{\Omega^e} \nabla N_j^e \cdot \nabla N_i^e d\Omega \right] = 0 \quad (2.54)$$

Como explicitado, a função peso δ_j se repete em todos os termos e pode ser eliminada da equação. O mesmo raciocínio aplica-se em todas as equações, de forma que é possível eliminar as funções peso γ_j e ζ_i do sistema:

$$\begin{cases}
\sum_{i,j \in e} \left[\frac{\partial(\omega_z)_i}{\partial t} \int_{\Omega^e} N_j^e N_i^e d\Omega + (\omega_z)_i \int_{\Omega^e} \mathbf{U} \cdot (N_j^e \nabla N_i^e) d\Omega + (\omega_z)_i \nu \int_{\Omega^e} \nabla N_j^e \cdot \nabla N_i^e d\Omega \right] = 0 \\
\sum_{i,j \in e} \left[(\omega_z)_i \int_{\Omega^e} N_j^e N_i^e d\Omega - \psi_i \int_{\Omega^e} \nabla N_j^e \cdot \nabla N_i^e d\Omega \right] = 0 \\
\sum_{i,j \in e} \left[u_i \int_{\Omega^e} N_j^e N_i^e d\Omega - \psi_i \int_{\Omega^e} N_j^e \frac{\partial N_i^e}{\partial y} d\Omega \right] = 0 \\
\sum_{i,j \in e} \left[v_i \int_{\Omega^e} N_j^e N_i^e d\Omega + \psi_i \int_{\Omega^e} N_j^e \frac{\partial N_i^e}{\partial x} d\Omega \right] = 0
\end{cases} \quad (2.55)$$

2.4.3 Forma matricial

As integrais apresentadas no sistema (2.55) são avaliações das funções de forma em cada elemento do domínio discretizado, ou seja, são um parâmetro dependente apenas da geometria do elemento e suas coordenadas. Dessa forma, é interessante representar essas integrais de maneira matricial, já que, uma vez definido o domínio, essas funções serão constantes do problema.

$$\begin{cases}
m^e = \int_{\Omega^e} N_i^e N_j^e d\Omega \\
g_x^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} N_j^e d\Omega \\
g_y^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} N_j^e d\Omega \\
k_{xx}^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} \frac{\partial N_i^e}{\partial x} d\Omega \\
k_{yy}^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} \frac{\partial N_i^e}{\partial y} d\Omega \\
k^e = k_{xx}^e + k_{yy}^e
\end{cases} \quad (2.56)$$

Essas matrizes são conhecidas como matrizes elementares, que são matrizes referentes a cada elemento e de dimensão a depender do tipo de elemento utilizado. Reescrevendo o sistema (2.55) utilizando a notação do sistema (2.56):

$$\left\{ \begin{array}{l} \sum_{i,j \in e} \left[\frac{\partial(\omega_z)_i}{\partial t} m^e + (\omega_z)_i (u \cdot g_x^e + v \cdot g_y^e) + (\omega_z)_i \nu k^e \right] = 0 \\ \sum_{i,j \in e} [(\omega_z)_i m^e - \psi_i k^e] = 0 \\ \sum_{i,j \in e} [u_i m^e - \psi_i g_y^e] = 0 \\ \sum_{i,j \in e} [v_i m^e + \psi_i g_x^e] = 0 \end{array} \right. \quad (2.57)$$

Cada elemento possuirá uma sistema matricial a ser resolvido. Entretanto, cada elemento possui diversos nós vizinhos compartilhados, de forma que é necessário resolver o sistema para todo o domínio. Para esse fim, é utilizada uma matriz montagem especial \mathbf{A} , que é responsável por unir as matrizes elementares em um processo conhecido como *Assembling*:

$$\left\{ \begin{array}{l} \mathbf{M} = \mathbf{A}m^e \\ \mathbf{G}_x = \mathbf{A}g_x^e \\ \mathbf{G}_y = \mathbf{A}g_y^e \\ \mathbf{K} = \mathbf{A}k^e \end{array} \right. \quad (2.58)$$

As matrizes que resultam do *Assembling* são conhecidas como matrizes globais do sistema, nomeadas como matriz de massa \mathbf{M} , matriz de rigidez \mathbf{K} e matrizes do gradiente \mathbf{G}_x e \mathbf{G}_y . Essas são as matrizes utilizadas na resolução do sistema linear para todo o domínio:

$$\left\{ \begin{array}{l} \mathbf{M} \frac{\partial \omega_z}{\partial t} + (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) \omega_z + \nu \mathbf{K} \omega_z = 0 \\ \mathbf{M} \omega_z - \mathbf{K} \psi = 0 \\ \mathbf{M} u - \mathbf{G}_y \psi = 0 \\ \mathbf{M} v + \mathbf{G}_x \psi = 0 \end{array} \right. \quad (2.59)$$

Para utilizar o sistema matricial apresentado acima e resolver do problema, basta agora definir a malha computacional utilizada para realizar a montagem das matrizes, como será apresentado na próxima seção.

2.4.4 Malha computacional

Uma malha computacional é um conjunto de elementos de geometria definida utilizados para representar um domínio físico de maneira computacional. Uma das vantagens apresentadas pelo uso do Método de Elementos Finitos é a possibilidade de utilizar elementos de diferentes formas.

Para a resolução desse problema, elementos triangulares com três nós (vértices do elemento utilizados para armazenar os valores das variáveis) serão utilizados. Isso se deve ao fato de a Formulação Corrente-Vorticidade permitir funções de forma polinomiais de primeira ordem para aproximar as variáveis de interesse no interior do elemento e pela facilidade dos elementos triangulares em se aproximar da geometria física desejada.

As funções de forma utilizadas para esse tipo de elemento são amplamente descritas pela literatura, como em (LEWIS, 2004), e apresentadas nas Eqs. (2.60) e (2.61):

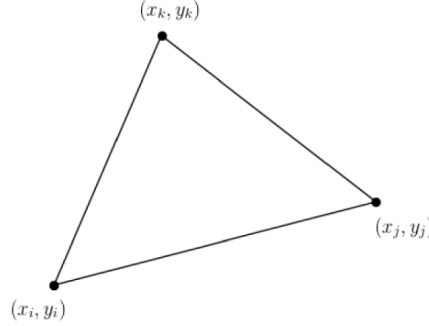


Figura 2.2: Representação do elemento triangular (LEWIS, 2004)

$$\mathbf{N}(x, y) = (N_i, N_j, N_k) \longrightarrow N_{i,j,k} = \frac{1}{2A}(a_{i,j,k} + b_{i,j,k}x + c_{i,j,k}y) \quad (2.60)$$

$$\begin{aligned} a_i &= x_j y_k - x_k y_j; & b_i &= y_j - y_k; & c_i &= x_k - x_j; \\ a_j &= x_k y_i - x_i y_k; & b_j &= y_k - y_i; & c_j &= x_i - x_k; \\ a_k &= x_i y_j - x_j y_i; & b_k &= y_i - y_j; & c_k &= x_j - x_i; \end{aligned} \quad (2.61)$$

Na qual A é a área do elemento, determinada por:

$$A = \frac{1}{2} \det \begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{bmatrix} \quad (2.62)$$

Considerando esse tipo de elemento, as matrizes elementares apresentadas no sistema (2.56) são:

$$m^e = \frac{A}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (2.63)$$

$$g_x^e = \frac{1}{6} \begin{bmatrix} b_i & b_j & b_k \\ b_i & b_j & b_k \\ b_i & b_j & b_k \end{bmatrix} \quad (2.64)$$

$$g_y^e = \frac{1}{6} \begin{bmatrix} c_i & c_j & c_k \\ c_i & c_j & c_k \\ c_i & c_j & c_k \end{bmatrix} \quad (2.65)$$

$$k^e = \frac{1}{4A} \begin{bmatrix} b_i^2 + c_i^2 & b_i b_j + c_i c_j & b_i b_k + c_i c_k \\ b_i b_j + c_i c_j & b_j^2 + c_j^2 & b_k b_j + c_k c_j \\ b_i b_k + c_i c_k & b_k b_j + c_k c_j & b_k^2 + c_k^2 \end{bmatrix} \quad (2.66)$$

2.4.5 Discretização temporal

O último passo da formulação consiste em descrever como a derivada temporal será modelada. A discretização desse termo partirá de uma aproximação utilizando um esquema de diferenças finitas, onde o termo é discretizado através de um esquema progressivo de primeira ordem (ANJOS, 2021). Tomando como ponto de partida a equação para o transporte de vorticidade:

$$\mathbf{M} \frac{\partial \omega_z}{\partial t} + (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) \omega_z + \nu \mathbf{K} \omega_z = 0 \quad (2.67)$$

Basta substituir o termo de derivada temporal por sua aproximação:

$$\mathbf{M} \left[\frac{\omega_z^{n+1} - \omega_z^n}{\Delta t} \right] + (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) \omega_z + \nu \mathbf{K} \omega_z = 0 \quad (2.68)$$

Enquanto ω_z^n representa a vorticidade avaliada no tempo atual, ω_z^{n+1} indica a vorticidade do passo de tempo seguinte. Uma vez que agora a vorticidade é discreta no tempo, é natural que seja necessário definir a qual passo de tempo essa variável pertence. Dessa necessidade, surgem duas maneiras distintas de abordar o problema. A primeira, chamada de abordagem implícita, considera que o termo de convecção utiliza os valores do novo passo de tempo e precisa ser resolvido pelo sistema linear:

$$\left(\frac{\mathbf{M}}{\Delta t} + (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) + \nu \mathbf{K} \right) \omega_z^{n+1} = \frac{\mathbf{M}}{\Delta t} \omega_z^n + c.c. \quad (2.69)$$

Na qual *c.c.* representa as condições de contorno a serem aplicadas. Já no esquema explícito, o termo convectivo utiliza os valores do passo de tempo anterior, entrando dessa forma no lado direito do sistema linear:

$$\left(\frac{\mathbf{M}}{\Delta t} + \nu \mathbf{K} \right) \omega_z^{n+1} = \left(\frac{\mathbf{M}}{\Delta t} - (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) \right) \omega_z^n + c.c. \quad (2.70)$$

2.4.6 Adimensionalização

Os fenômenos da natureza, naturalmente, dependem de diversos fatores distintos, tornando a comparação entre fenômenos uma prática não trivial. Com o intuito de permitir uma comparação mais fiel, o teorema II de Buckingham sugere que o processo de adimensionalização das equações que regem os fenômenos físicos permite relacionar eventos, desde que haja a semelhança completa. Essa semelhança é definida pela igualdade do valor numérico de constantes resultantes do processo de adimensionalização.

Em outras palavras, se dois eventos distintos, mas regidos pela mesma lei, possuírem adimensionais iguais, então o comportamento dos dois eventos também será igual. Dessa forma, o processo de adimensionalização permite relacionar diretamente os resultados da simulação com o de fenômenos de mesma natureza (FOX, 2010).

Tomando como ponto de partida o sistema (2.59):

$$\begin{cases} M \frac{\partial \omega_z}{\partial t} + (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) \omega_z + \nu \mathbf{K} \omega_z = 0 \\ M \omega_z - \mathbf{K} \psi = 0 \\ M u - \mathbf{G}_y \psi = 0 \\ M v + \mathbf{G}_x \psi = 0 \end{cases}$$

Para a adimensionalização das equações, é necessário adimensionalizar as variáveis. Para isso, serão utilizados alguns parâmetros como a velocidade característica U_0 e o comprimento característico L :

$$x^* = \frac{x}{L} \quad y^* = \frac{y}{L} \quad u^* = \frac{u}{U_0} \quad v^* = \frac{v}{U_0} \quad t^* = \frac{U_0}{L} t$$

O sobrescrito $*$ é utilizado para identificar a grandeza adimensional. Para adimensionalizar as matrizes massa, rigidez e gradiente, basta partir de sua definição, apresentada no sistema (2.56), e aplicar o modo adimensional das variáveis:

$$\mathbf{M}^* = \mathbf{M} \quad (2.71)$$

$$\frac{\partial}{\partial x} = \frac{1}{L} \frac{\partial}{\partial x^*} \longrightarrow \mathbf{G}_x^* = L \mathbf{G}_x \quad (2.72)$$

$$\frac{\partial}{\partial y} = \frac{1}{L} \frac{\partial}{\partial y^*} \longrightarrow \mathbf{G}_y^* = L \mathbf{G}_y \quad (2.73)$$

$$\frac{\partial^2}{\partial x^2} = \frac{1}{L^2} \frac{\partial^2}{\partial (x^*)^2} \quad ; \quad \frac{\partial^2}{\partial y^2} = \frac{1}{L^2} \frac{\partial^2}{\partial (y^*)^2} \longrightarrow \mathbf{K}^* = L^2 \mathbf{K} \quad (2.74)$$

O primeiro passo é adimensionalizar as variáveis do problema. Partindo da definição de vorticidade e substituindo as grandezas por seu respectivo adimensional:

$$\mathbf{M}\omega = \mathbf{G}_x v - \mathbf{G}_y u = \frac{U_0}{L} (\mathbf{G}_x^* v^* - \mathbf{G}_y^* u^*) \longrightarrow \omega^* = \frac{L}{U_0} \omega \quad (2.75)$$

O mesmo pode ser feito para a definição da função corrente. Aqui será utilizada a expressão utilizando a componente horizontal u da velocidade, mas note que o resultado para a outra componente seguirá rigorosamente o mesmo procedimento:

$$\mathbf{M}u = \mathbf{G}_y \psi \longrightarrow (\mathbf{M}^* u^*) U_0 = (\mathbf{G}_y^* \psi^*) \frac{1}{L} \longrightarrow \psi^* = \frac{\psi}{U_0 L} \quad (2.76)$$

Uma vez que todas as variáveis do problema possuem adimensionais conhecidos, é possível aplicar todas as variáveis adimensionais nas equações, resultando no sistema 2.77:

$$\left\{ \begin{array}{l} \mathbf{M}^* \frac{\partial \omega^*}{\partial t^*} + (u^* \cdot \mathbf{G}_x^* + v^* \cdot \mathbf{G}_y^*) \omega^* + \frac{\nu}{U_0 L} \mathbf{K}^* \omega^* = 0 \\ \mathbf{M}^* \omega^* - \mathbf{K}^* \psi^* = 0 \\ \mathbf{M}^* u^* - \mathbf{G}_y^* \psi^* = 0 \\ \mathbf{M}^* v^* + \mathbf{G}_x^* \psi^* = 0 \\ \mathbf{M}^* \omega^* = \mathbf{G}_x^* v^* - \mathbf{G}_y^* u^* \end{array} \right. \quad (2.77)$$

É interessante observar que houve a aparição de um termo adimensional na equação do transporte de vorticidade. Esse termo, muito conhecido na área da dinâmica dos fluidos, chama-se número de Reynolds e é utilizado para medir a relação entre as forças inerciais e as forças viscosas atuante em um escoamento:

$$\text{Re} = \frac{U_0 L}{\nu} \quad (2.78)$$

Finalmente, é possível definir a exata forma das equações a serem solucionadas no problema. O sobrescrito $*$ foi deixado de lado para uma exibição mais limpa

das equações, entretanto todas as variáveis daqui em diante contarão com o caráter adimensional.

$$\left\{ \begin{array}{l} M \frac{\partial \omega_z}{\partial t} + (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) \omega_z + \frac{1}{\text{Re}} \mathbf{K} \omega_z = 0 \\ M \omega_z - \mathbf{K} \psi = 0 \\ M u - \mathbf{G}_y \psi = 0 \\ M v + \mathbf{G}_x \psi = 0 \\ M \omega = \mathbf{G}_x v - \mathbf{G}_y u \end{array} \right. \quad (2.79)$$

Capítulo 3

Metodologia numérica

3.1 Geração de malha

Após definida a geometria de interesse para simulação, o primeiro passo é gerar a malha computacional. Para tal fim, diversos *softwares* se apresentam como opções, cada qual com suas peculiaridades e características. Para a confecção desse trabalho, foi utilizado o *software* de código aberto *Gmsh* (CHRISTOPHE GEUZAINÉ, 2017) em sua versão 3.0.6, disponível gratuitamente. Esse programa, desenvolvida Christophe Geuzaine e Jean-François Remacle e lançado sob a GNU General Public License, é repartido em módulos, sendo os dois principais:

- *Geometry* - módulo responsável por construir a figura geométrica e o domínio computacional fechado de desejo. Dispõe como principais ferramentas a criação de pontos, linhas, curvas, além de permitir a definição de faces e volumes que posteriormente serão discretizados no espaço. Além disso, nesse módulo também são definidos os *Physical groups*, grupos responsáveis por definir a quais entidades serão atribuídas as futuras condições de contorno, a serem definidas no código numérico.
- *Mesh* - módulo responsável por construir a malha computacional a partir da geometria definida. Aqui, o usuário pode controlar o tipo de elemento desejado, o tamanho mínimo e máximo dos elementos, zonas de maior refino, os diferentes algoritmos de construção, entre outros.

Todas as malhas utilizadas nesse trabalho foram confeccionadas utilizando ele-

mentos triangulares de primeira ordem e o algoritmo de triangulação *Frontal*. Após gerada a malha computacional, é necessário exportá-la no formato adequado para o tratamento via Método de Elementos Finitos, que nesse caso foi o formato `.msh`. Um exemplo de malha computacional gerada pelo *Gmsh* em sua interface gráfica pode ser observado na Figura 3.1:

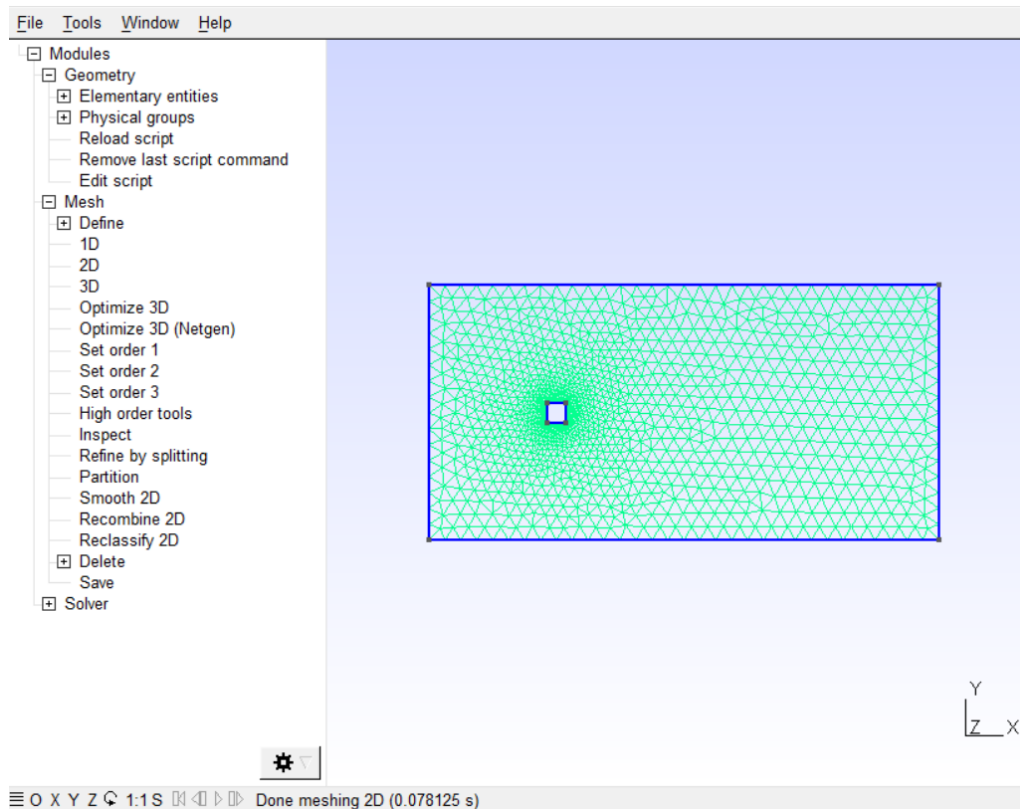


Figura 3.1: Exemplo de malha gerada pelo *software Gmsh*

3.2 Código numérico

Após gerada a malha, é necessário importá-la para o processamento em um código numérico, responsável por aplicar todo o procedimento descrito no capítulo 2 de forma computacional. Para esse fim, o código desenvolvido segue um algoritmo específico, descrito pelo fluxograma ilustrado na Figura 3.2. Para o desenvolvimento do código numérico, a linguagem de computação escolhida foi o *Python* em sua versão 3.8 devido a sua ampla aplicação no meio acadêmico.

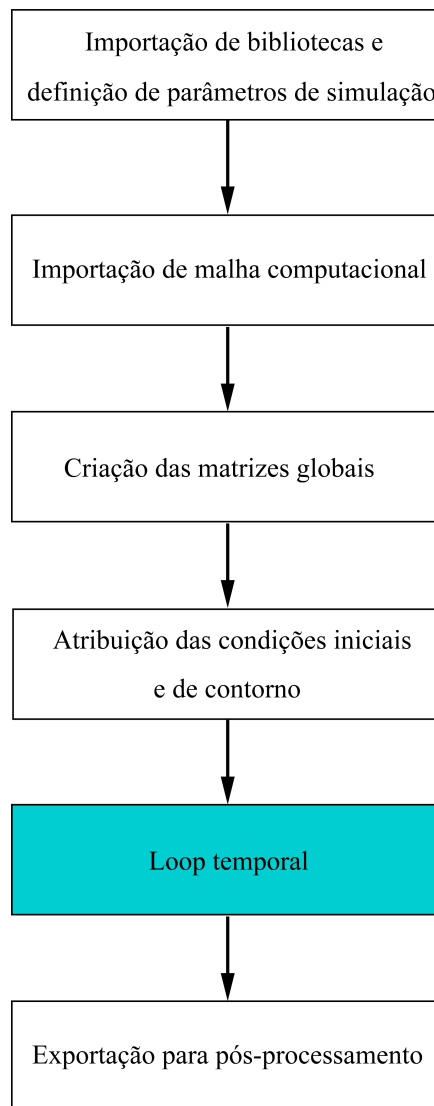


Figura 3.2: Fluxograma do algoritmo computacional

3.2.1 Importação de bibliotecas e definição de parâmetros de simulação

O primeiro passo do desenvolvimento do código é a importação das bibliotecas necessárias. Para esse trabalho, as bibliotecas importadas foram:

- *Numpy*: utilizada para os cálculos matriciais envolvendo álgebra linear
- *Meshio*: utilizada para a importação, exportação e tratamento da malha computacional

```
# importando bibliotecas
import numpy as np
import meshio
```

Logo após, define-se os parâmetros gerais de cada simulação, como a viscosidade cinemática ν , o passo de tempo Δt e o número de iterações passos :

```
# Definição das constantes do problema
nu = 0.01 # o termo "nu" foi utilizado para variar o número de Reynolds,
dt = 0.01 # uma vez que os demais termos característicos são unitários,
passos = 600 # resultando em nu = 1/Re
```

3.2.2 Importação da malha

Como foi comentado na subseção anterior, a biblioteca utilizada para o tratamento da malha é o `meshio`. Ela tem a principal função de receber um arquivo no formato `.msh` de entrada e convertê-lo em listas no *Python*.

```
msh = meshio.read('malha.msh') #importa malha computacional

X = msh.points[:,0] #lista com coordenada X dos pontos
Y = msh.points[:,1] #lista com coordenada Y dos pontos
IEN = msh.cells["triangle"] # matriz IEN
ne = IEN.shape[0] # define número de elementos

IENbound = msh.cells["line"] # lista de vértices nas faces do contorno
IENboundTypeElem = list(msh.cell_data['line']['gmshtype']) # pega índice
# de cada cc do physical group
# pra cada par de vértices (face) do IENbound
boundNames = list(msh.field_data.keys()) # pega o nome das cc's
# definidas no Gmsh
IENboundElem = [boundNames[elem] for elem in IENboundTypeElem] # relaciona elem
# ao índice da cc

# criando lista das faces (par de vértices) das células do contorno
cc = np.unique(IENbound.reshape(IENbound.size))

# atribuindo cada cc ao nome do contorno, definido no physical group do Gmsh
ccName = []
for i in range(len(X)):
    ccName.append('')
for elem in range(0, len(IENbound)):
    ccName[IENbound[elem][0]] = boundNames[IENboundElem[elem]]
    ccName[IENbound[elem][1]] = boundNames[IENboundElem[elem]]
```

A importação da malha via `meshio`, além de permitir grande acessibilidade aos dados utilizando listas compreensíveis pelo `Python`, possibilita a obtenção de parâmetros de malha importantes com grande facilidade, como o número de elementos, suas coordenadas e a matriz IEN, também chamada de matriz de conectividade, que relaciona os elementos com seus vértices. A matriz conectividade é de grande importância para a correta definição das condições de contorno e na interpolação entre células vizinhas. Um exemplo de matriz IEN é apresentado na Figura 3.3.

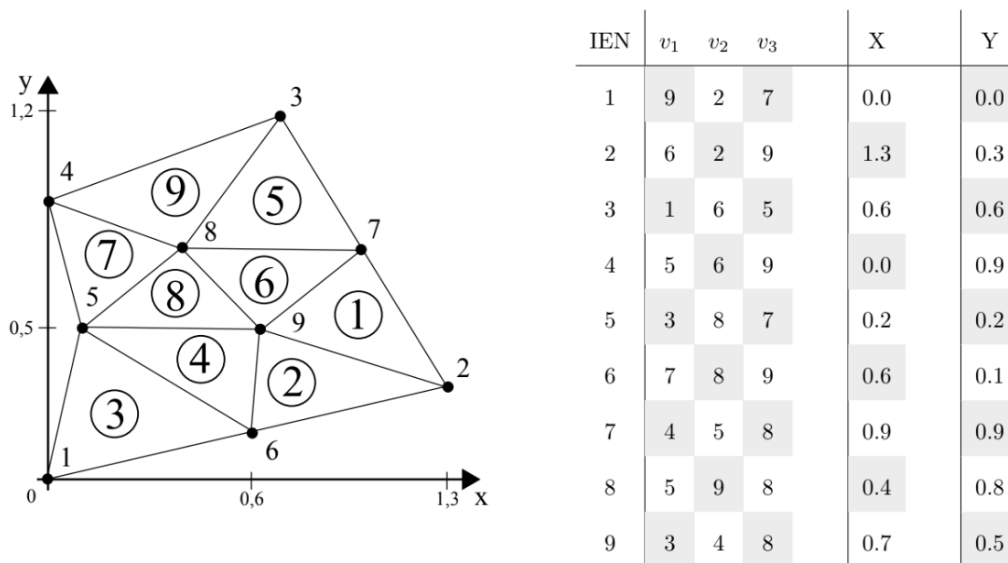


Figura 3.3: Matrizes IEN, X e Y de uma malha computacional (ANJOS, 2021)

3.2.3 Criação das matrizes globais

Nesse trecho do código, são criadas as matrizes globais do sistema vide o desenvolvimento apresentado nas Eqs. (2.63) a (2.66). Para mais informações sobre esse procedimento, visitar Apêndice A.

```

for e in range(0,ne): # loop em cada elemento
    (. . .)
    for i in range(0,3): # loop em cada vértice do elemento
        ii = IEN[e,i]
        for j in range(0,3):
            jj = IEN[e,j]

            K[ii,jj] = K[ii,jj] + kele[i,j] #
            M[ii,jj] = M[ii,jj] + mele[i,j] # Procedimento de

```

```
Gx[ii,jj] = Gx[ii,jj] + gxele[i,j] # Assembling
Gy[ii,jj] = Gy[ii,jj] + gyele[i,j] #
```

O procedimento de montagem das matrizes elementares e o *Assembling* seguem rigorosamente o procedimento apresentado na seção 2.4.4 e presente em (LEWIS, 2004). De maneira resumida, a construção dessas matrizes ocorre dentro do *loop* em cada elemento e consiste em:

1. Ler as coordenadas (x, y) de cada elemento
2. Determinar os parâmetros a, b, c e a área do elemento
3. Montar a matriz elementar

3.2.4 Atribuição das condições iniciais e de contorno

Agora que as matrizes globais foram construídas para toda a malha e os vértices para a aplicação das condições de contorno já foram identificados, é necessário programar sua atribuição. Como já foi comentado, a solução do problema em questão envolve duas condições de contorno (Dirichlet e Neumann homogêneo), cada qual com uma abordagem específica.

- Condição de Dirichlet

A condição de Dirichlet, ou condição de valor prescrito, impõe um valor específico para a variável de interesse nos vértices onde é aplicada. Dessa forma, o valor não deve ser alterado enquanto a simulação acontece, ou seja, o produto entre a linha da matriz global cujo índice é um vértice com condição de contorno Dirichlet e o valor a ser calculado deve resultar no valor pré-estabelecido. Considerando um sistema matricial genérico $\mathbf{Ax} = \mathbf{b}$:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (3.1)$$

A condição de contorno de Dirichlet é realizada através da imposição explícita do valor pré-estabelecido após a montagem do sistema linear resultante. Considerando um sistema matricial genérico $\mathbf{Ax} = \mathbf{b}$, substituindo a linha i do sistema linear pela equação do contorno $x_i = b_i$.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{ni} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix} \quad (3.2)$$

Dessa forma, o algoritmo para a aplicação dessa condição de contorno é:

1. Zerar linha i
2. Inserir 1 na diagonal i, i
3. Inserir b_i no vetor solução, do lado direito

Para exemplificar, a aplicação das condições de contorno para ψ são feitas da seguinte forma:

```
for i in cc:
    A[i,:] = 0.0 # zerando toda a linha
    A[i,i] = 1.0 # impondo 1 na diagonal
    b[i] = psicc[i] # impondo o valor definido no vetor solução
```

- Condição de Neumann homogêneo

Para o caso da condição de Neumann, ou de fluxo prescrito, seria necessário resolver uma integral na superfície de contorno Γ proveniente do processo de discretização espacial via Teorema de Green, apresentado na Eq. (2.42), e somar o resultado dessa integral no vetor solução b .

Entretanto, como foi descrito no capítulo 2, a condição utilizada na resolução do problema é a de Neumann homogêneo, ou de fluxo prescrito nulo. Isso faz com que o

resultado dessa integral na superfície seja necessariamente nulo. Como consequência, a atribuição dessa condição faz com que nenhuma alteração seja necessária na resolução do sistema linear $\mathbf{Ax} = \mathbf{b}$, bastando prosseguir com o procedimento padrão. Mais detalhes sobre as condições de contorno podem ser encontradas em (ANJOS, 2022).

- Condição para a vorticidade ω

A condição inicial e de contorno para ω se diferencia das demais por não ser definida de maneira direta (utilizando condições de valor ou fluxo prescrito), mas a partir do cálculo de outras variáveis. Dessa forma, após definidas as demais condições iniciais e de contorno, a vorticidade é computada seguindo sua definição, apresentada na Eq. (2.30):

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

Que, após o procedimento de discretização apresentado no capítulo 2, se torna:

$$\mathbf{M}\omega = \mathbf{G}_x v - \mathbf{G}_y u \quad (3.3)$$

Pelo fato de depender dos valores de velocidade, que são variáveis no tempo, essa condição terá que ser aplicada a cada *loop* temporal, como será discutido adiante.

- Implementação no código

No primeiro passo de tempo, todas as variáveis são inicializadas com o valor nulo e não há resolução de equação linear para as variáveis u , v e ψ . Por essa razão, apenas as condições no vetor solução precisam ser impostas. Como exemplo, para a entrada do domínio:

```
u = np.zeros( (npoints,1),dtype='float') #
v = np.zeros( (npoints,1),dtype='float') # Variáveis inicializadas
omega = np.zeros( (npoints,1),dtype='float') # com valor nulo
psi = np.zeros( (npoints,1),dtype='float') #

psicc = np.zeros( (npoints,1),dtype='float') # Vetor auxiliar para guardar
# informações de cc para psi
```

```

for i in cc:
    if ccName[i] == 'inflow': # cc para a entrada do domínio
        u[i] = 1.0           # Dirichlet para u
        v[i] = 0.0           # Dirichlet para v
        psicc[i] = Y[i]      # Dirichlet para psi

```

Já para ω , a condição inicial é calculada a partir da resolução da Eq. 3.3:

```

# Calculo da condicao inicial de omega (M \omega = Gx * v - Gy * u)
b = np.dot(Gx,v) - np.dot(Gy,u)
omega = np.linalg.solve(M,b)

```

3.2.5 Loop temporal

O *loop* temporal é um processo iterativo no tempo, no qual as equações do problema são resolvidas. Ao final de cada iteração, as variáveis calculadas são utilizadas como condição inicial para a iteração seguinte e esse procedimento se repete até o número final de iterações, definido pelo usuário. O processo iterativo pode ser ilustrado pelo fluxograma presente na Figura 3.4.

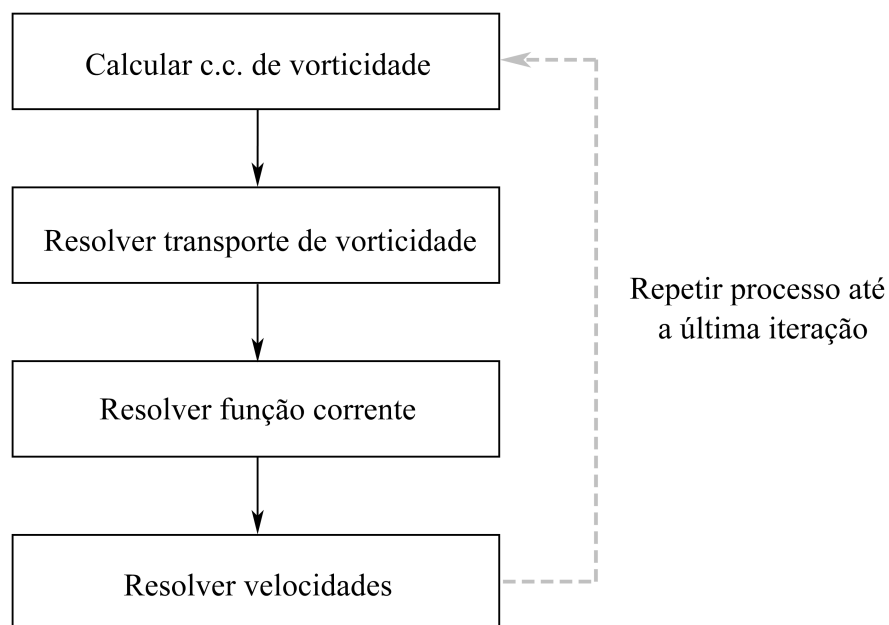


Figura 3.4: Fluxograma do processo iterativo no tempo

- Cálculo da c.c. de vorticidade

Uma vez iniciado o *loop*, o primeiro passo é o cálculo da condição de contorno da vorticidade a partir das velocidades obtidas no passo de tempo anterior. Para isso, é resolvida a Eq. (3.3) com uma pequena adaptação - os resultados serão armazenados em um vetor específico para a condição de contorno da vorticidade, ω_{cc} :

$$\mathbf{M}\omega_{cc} = \mathbf{G}_x v - \mathbf{G}_y u \quad (3.4)$$

```
# Início do loop iterativo no tempo
for t in range(passos):
    print(t) # Controle da iteração atual

    # Calculo da condicao de contorno de omega (M omegacc = Gx * v - Gy * u)
    b = np.dot(Gx,v) - np.dot(Gy,u) # Resolucao do sistema linear Ax = b,
    omegacc = np.linalg.solve(M,b) # onde A = M, x = omegacc e b = (Gx v - Gy u)
```

O vetor contendo as condições de contorno de vorticidade será usado no próximo passo, ou seja, na resolução do transporte de vorticidade.

- Resolução do transporte de vorticidade

O próximo passo é resolver o transporte de vorticidade utilizando a equação discreta no tempo (Eq. (2.68)) apresentada na seção 2.4.5. Entretanto, como foi abordado nessa seção, existem duas possíveis maneiras de abordar o problema: o método implícito e o método explícito, apresentados nas Eqs. (2.69) e (2.70), respectivamente.

$$\text{Método Implícito} \longrightarrow \left(\frac{\mathbf{M}}{\Delta t} + (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) + \nu \mathbf{K} \right) \omega^{n+1} = \frac{\mathbf{M}}{\Delta t} \omega^n + c.c.$$

$$\text{Método Explícito} \longrightarrow \left(\frac{\mathbf{M}}{\Delta t} + \nu \mathbf{K} \right) \omega^{n+1} = \left(\frac{\mathbf{M}}{\Delta t} - (u \cdot \mathbf{G}_x + v \cdot \mathbf{G}_y) \right) \omega^n + c.c.$$

Nas quais ω^n representa o valor de vorticidade calculado no passo de tempo anterior e ω^{n+1} , o valor a ser calculado no passo de tempo atual.

De maneira resumida, a alteração entre os dois métodos permite ao usuário escolher se deseja que a velocidade seja tratada de maneira implícita, ou seja, aplicada na matriz A na resolução do sistema linear, ou de maneira explícita, ou seja, aplicada no vetor solução b na resolução do sistema linear. Ambas as maneiras estão corretas e a escolha entre uma ou outra é de escolha do usuário.

Para a solução do sistema linear, é necessário seguir os seguintes passos:

1. Construir matriz A
2. Construir vetor solução b
3. Aplicar condições de contorno para a variável
4. Resolver sistema linear

Como a construção da matriz A e do vetor b diferem quanto ao método utilizado, ambos apresentam diferentes implementações no código. Pelo método implícito:

```
# Resolvendo o transporte da vorticidade
A_1 = (1.0/dt) * M + nu*K

# Modo IMPLÍCITO: vgo entrando na matriz A
uI = np.diag(u[:,0]) # Transforma o vetor u em uma matriz diagonal
vI = np.diag(v[:,0]) # Transforma o vetor v em uma matriz diagonal
vgoI = np.dot(uI,Gx) + np.dot(vI,Gy) # Constrói matriz vgo = Gx u + Gy v

A_1 += vgoI # Adiciona a matriz vgo à matriz A

# Vetor solução b para o sist. linear do transporte da vorticidade
b_1 = (1.0/dt) * np.dot(M,omega)

# Imposição das cc's para omega
for i in cc:
    A_1[i,:] = 0.0 # zerando toda a linha
    A_1[i,i] = 1.0 # impondo 1 na diagonal
    b_1[i] = omegacc[i] # cc definida no início do loop com base
                    # nas velocidades do passo anterior

# Resolve transporte de vorticidade, atualizando o valor de omega
omega = np.linalg.solve(A_1,b_1)
```

Já pelo método explícito:

```
# Resolvendo o transporte da vorticidade
A_1 = (1.0/dt) * M + nu*K

# Modo EXPLÍCITO: vgoI entrando no vetor solução b
uI = np.diag(u[:,0]) # Transforma o vetor u em uma matriz diagonal
vI = np.diag(v[:,0]) # Transforma o vetor v em uma matriz diagonal
vgoI = np.dot(uI,Gx) + np.dot(vI,Gy) # Constrói matriz vgo = Gx u + Gy v

# Vetor solução b para o sist. linear do transporte da vorticidade
b_1 = (1.0/dt) * np.dot(M,omega) - np.dot(vgoI,omega)

# Imposição das cc's para omega
for i in cc:
    A_1[i,:] = 0.0 # zerando toda a linha
    A_1[i,i] = 1.0 # impondo 1 na diagonal
    b_1[i] = omegacc[i] # cc definida no início do loop com base
                        # nas velocidades do passo anterior

# Resolve transporte de vorticidade, atualizando o valor de omega
omega = np.linalg.solve(A_1,b_1)
```

- Resolução da função corrente

Uma vez atualizado o vetor vorticidade, é possível prosseguir com o cálculo da função corrente seguindo o sistema (2.59):

$$\mathbf{M}\omega = \mathbf{K}\psi \quad (3.5)$$

De maneira semelhante à aplicada na vorticidade, as matrizes serão construídas, as condições de contorno serão aplicadas e o sistema linear será resolvido.

```
# Resolvendo função corrente K*psi = M*omega, inicializando vetor solução b
b_2 = np.dot(M,omega)

# Inicializando matriz A
A_2 = K

# Impondo ccs de psi
for i in cc:
    A_2[i,:] = 0.0 # zerando toda a linha
```

```

A_2[i,i] = 1.0 # impondo 1 na diagonal
b_2[i] = psicc[i]

psi = np.linalg.solve(A_2,b_2)

```

- Resolução das velocidades

Por fim, as velocidades na direção x e y podem ser obtidas através da atualização de ψ seguindo as respectivas equações apresentadas no sistema (2.59):

$$M u = G_y \psi \quad (3.6)$$

$$M v = -G_x \psi \quad (3.7)$$

```

# Resolvendo velocidades M*u = Gy*psi e M*v = - Gx*psi
b_3 = np.dot(Gy,psi)
b_4 = np.dot(-Gx,psi)

u = np.linalg.solve(M,b_3)
v = np.linalg.solve(M,b_4)

for i in cc:
    if ccName[i] == 'inflow':
        u[i] = 1.0
        v[i] = 0.0
    if ccName[i] == 'farfield':
        v[i] = 0.0
    if ccName[i] == 'noSlip':
        u[i] = 0.0
        v[i] = 0.0

```

Em posse dos novos valores de velocidade e após a aplicação das condições de contorno, já foram gerados todos os dados necessários para uma nova iteração no tempo.

3.2.6 Exportação para pós-processamento

Na maioria dos casos, é interessante observar como a simulação avança a cada iteração, tanto para o controle dos parâmetros numéricos como para observação

do comportamento físico daquela simulação (por exemplo, o desenvolvimento de um escoamento até atingir o estado estacionário). Para esse fim, é interessante exportar os dados gerados a cada iteração.

No presente trabalho, a análise dos dados foi feita utilizando o *software ParaView*, que será apresentado na próxima seção. Para que os dados gerados pela simulação sejam compreensíveis pelo programa, a biblioteca `meshio` dispõe de uma função específica capaz de exportar os dados no formato `.vtk`, compatível com o *ParaView*.

```
# Exportando dados para pós-processamento
if t in range(0,passos):
    point_data={'u':u}
    data_vy={'v':v}
    data_psi={'psi':psi}
    data_omega={'omega':omega}
    point_data.update(data_v)
    point_data.update(data_psi)
    point_data.update(data_omega)
    meshio.write_points_cells('Iter'+str(t)+'.vtk',msh.points,
                              msh.cells,point_data=point_data,)
```

Esse passo marca o fim do processo iterativo no tempo e, ao fim de todas as iterações, do código numérico.

3.3 Pós-processamento

Como foi comentado na seção anterior, os dados gerados no presente trabalho serão analisados e o pós-processados através do *software* de código aberto *ParaView* (DEVELOPERS, 2020) em sua versão 5.8, disponível gratuitamente. Esse programa, desenvolvido pela empresa Kitware Inc., é uma ferramenta criada para a análise e visualização de dados utilizando parâmetros quantitativos e qualitativos.

Como algumas de suas principais funcionalidades, pode-se citar:

- Investigar localmente e globalmente o valor numérico de campos escalares e vetoriais e sua evolução no tempo
- Manipular campos existentes através de procedimentos matemáticos para a

geração de novos campos (é possível, por exemplo, aplicar o operador gradiente a um campo gerado em simulação)

- Realizar cortes bidimensionais e tridimensionais para a inspeção de regiões específicas do domínio computacional
- Gerar gráficos da evolução espacial e temporal das variáveis de interesse
- Proporcionar a visualização direta de fenômenos de interesse em simulações de fluidodinâmica computacional, como linhas de corrente e vetores velocidade

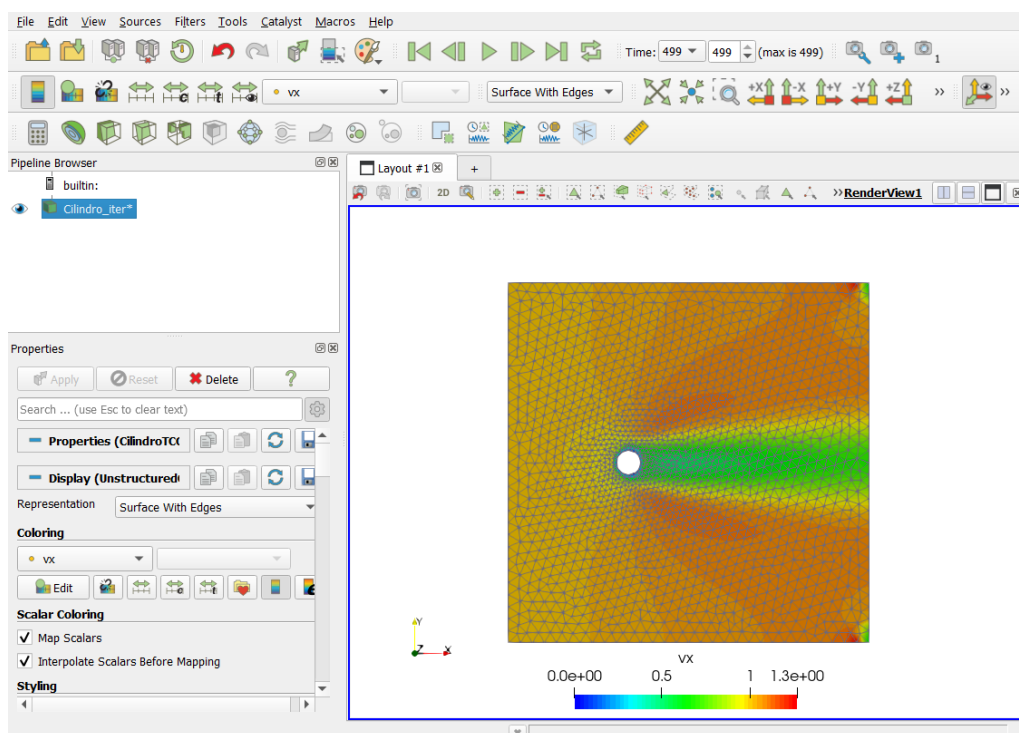


Figura 3.5: Interface gráfica do *ParaView* exibindo o campo de velocidade de uma simulação

Todas as imagens, tanto do domínio computacional quanto dos resultados, foram geradas utilizando o *software ParaView*. Para mais informações sobre as ferramentas presentes na aplicação, basta acessar a página do fabricante.

Capítulo 4

Validação do código numérico

Esse capítulo tem por objetivo a validação do código numérico apresentado, utilizando como método de validação a simulação de casos conhecidos no mundo da engenharia e a comparação dos resultados obtidos com as soluções presentes na literatura. Para esse fim, o código desenvolvido será utilizado para simular dois casos clássicos: o caso *Lid-driven cavity flow* e o caso *Poiseuille*.

4.1 Caso Poiseuille

O escoamento de *Poiseuille* é um caso clássico na engenharia, onde escoam-se um fluido newtoniano, monofásico e incompressível entre duas placas planas e paralelas. Como hipótese, assume-se um escoamento permanente e totalmente desenvolvido, apresentando um gradiente de pressão $\partial p/\partial x$ constante. Ao adotar essas condições, obtém-se como resultado o escoamento apresentado na Figura 4.1.

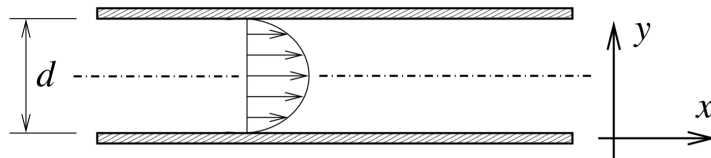


Figura 4.1: Escoamento sob placas planas (PONTES, 2009)

Para validar a simulação, os resultados numéricos obtidos são comparados à solução analítica adimensional do caso (Eq.(4.1)), obtida através da solução das equações de Navier-Stokes para o caso bidimensional e a aplicação das condições de contorno (DA CUNHA, 2020):

$$\begin{cases} u = 6(y - y^2) \\ v = 0 \end{cases} \quad (4.1)$$

As velocidades em x e em y são representadas, respectivamente, por u e v , enquanto d é a distância entre as placas, estabelecida como $d = 1$.

O caso simulado conta com uma velocidade de entrada uniforme de $u = 1$ e uma velocidade nula para os demais contornos. A função corrente ψ foi definida como 1 para parede superior e 0 para a inferior. Para a simulação numérica desse caso, foi utilizada uma malha computacional de 1158 elementos e um número de Reynolds de $Re = 1$.

Os resultados das variáveis de interesse obtidos na simulação pode ser encontrado na Figura 4.2.

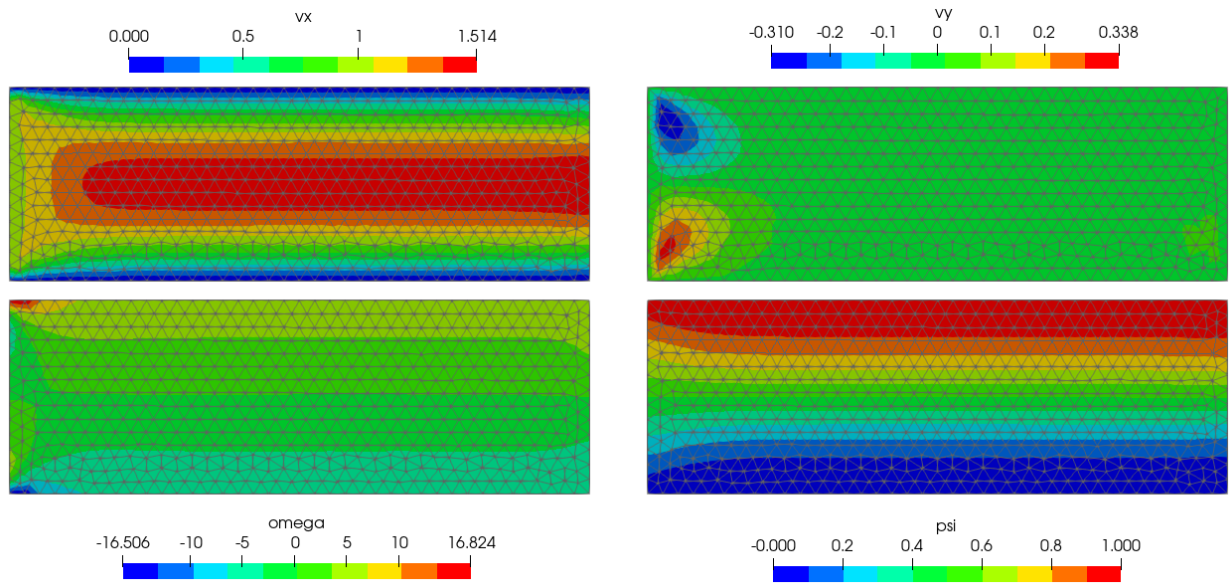


Figura 4.2: Resultados da simulação do caso *Poiseuille*

Com o objetivo de validar quantitativamente a simulação, foram extraídos valores de velocidade horizontal de um corte transversal do caso, tornando possível uma comparação gráfica entre a curva analítica e os resultados computacionais. Tais resultados podem ser encontrados na Figura 4.3.

Os resultados da simulação apresentam boa concordância com a curva obtida analiticamente.

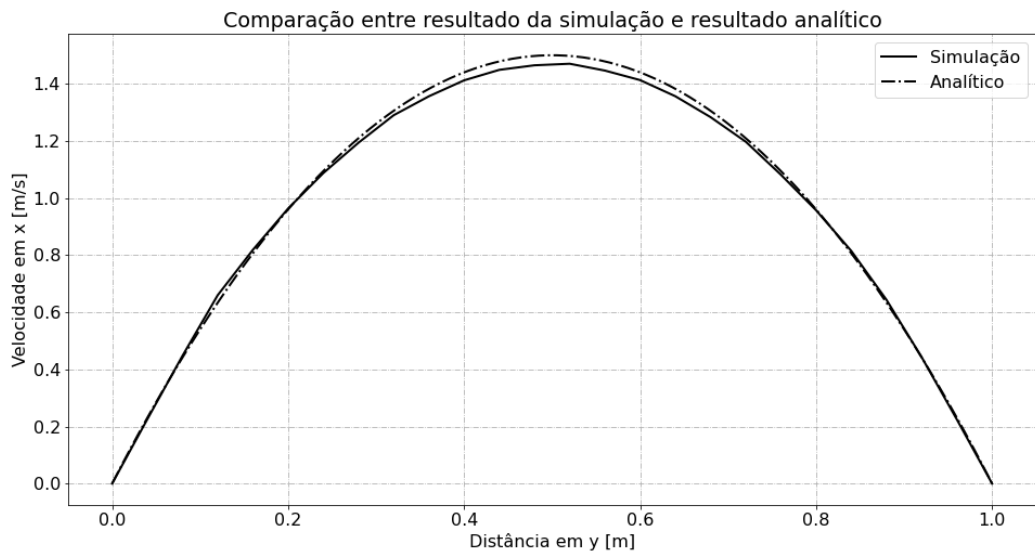


Figura 4.3: Resultados da simulação do caso *Poiseuille*

4.2 Caso Lid-driven

O caso *Lid-driven*, também conhecido como Escoamento em Cavidade, consiste no escoamento de um fluido newtoniano, monofásico e incompressível dentro de uma cavidade quadrada, causado pelo deslocamento da superfície superior. Esse deslocamento faz com que o fluido adquira um movimento circular sob a ação de forças viscosas, conforme apresentado na Figura 4.4.

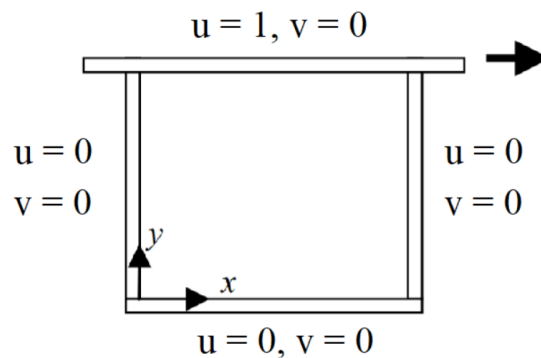


Figura 4.4: Escoamento em cavidade (CARLOS MARCHI, 2009)

O caso simulado conta com uma velocidade de $u = 1$ para a parede superior e velocidade nula para os demais contornos. A função corrente ψ foi definida como 0 para todas os contornos. Para a simulação numérica desse caso, foi utilizada uma malha computacional de 544 elementos e um número de Reynolds de $Re = 10$,

tornando possível uma posterior comparação com os resultados encontrados por (CARLOS MARCHI, 2009).

Os resultados das variáveis de interesse obtidos na simulação podem ser encontrados na Figura 4.5.

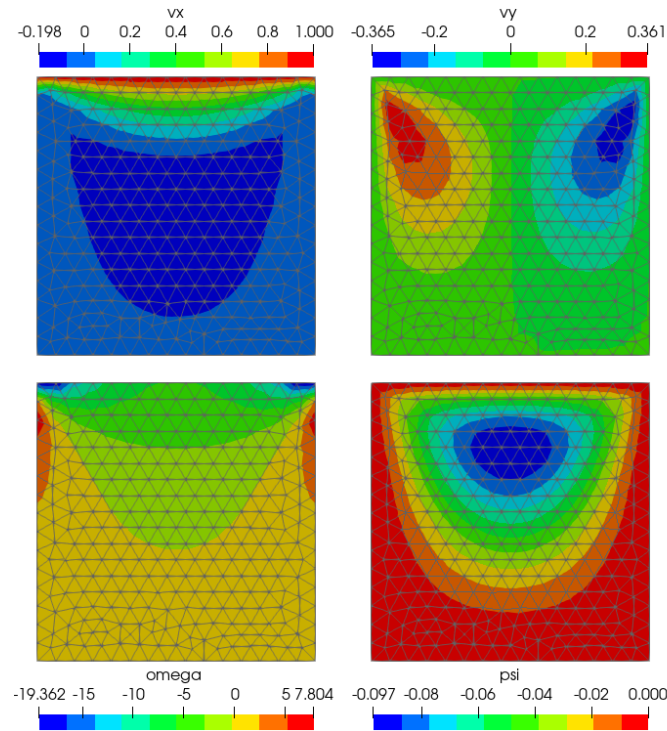


Figura 4.5: Resultados da simulação do caso *Lid-driven*

Com o objetivo de validar quantitativamente a simulação, foram extraídos valores de velocidade horizontal e vertical do fluido em cortes na região central do domínio computacional ($x = 0.5$ e $y = 0.5$). Dessa forma, é possível comparar graficamente esses dados com os resultados obtidos por (CARLOS MARCHI, 2009), estudo no qual o caso *Lid-driven* foi simulado utilizando o Método de Volumes Finitos com aproximação numérica de segunda ordem em uma malha computacional fina (1024 x 1024 elementos). Tais resultados podem ser encontrados nas Figuras 4.6 e 4.7. Os resultados da simulação apresentam boa concordância com os apresentados em (CARLOS MARCHI, 2009).

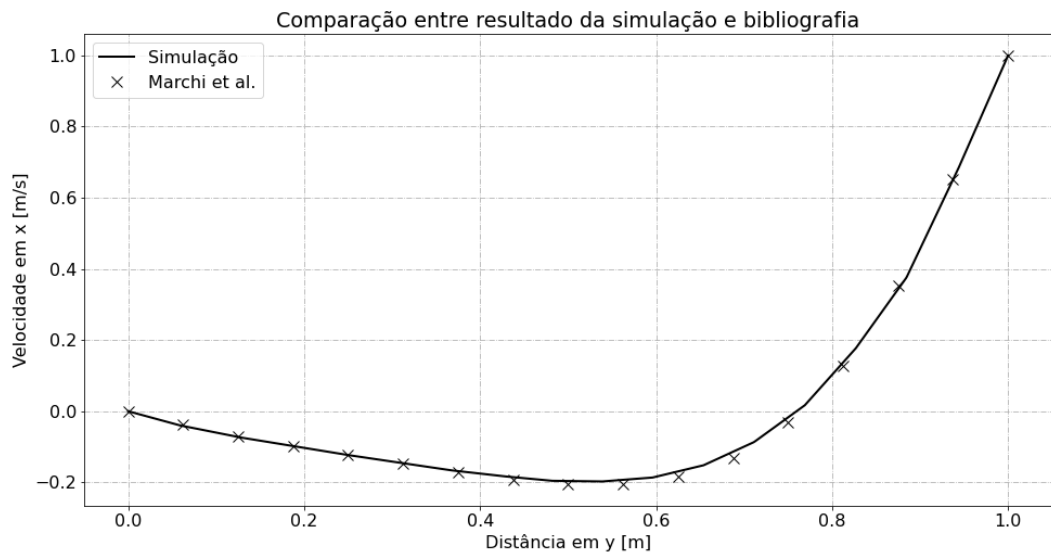


Figura 4.6: Resultados de velocidade em X da simulação do caso *Lid-driven*



Figura 4.7: Resultados de velocidade em Y da simulação do caso *Lid-driven*

Capítulo 5

Resultados

Nesse capítulo serão apresentados os resultados obtidos para a simulação de cada geometria. No total, serão simuladas cinco diferentes geometrias, cada qual com diferentes ângulos de ataque, totalizando onze simulações. Os resultados de cada simulação serão apresentados a partir dos seguintes tópicos:

- especificação da malha computacional e condições de contorno
- resultados das variáveis de interesse da simulação
- pós-processamento

Todas as simulações apresentam condições iniciais e de contorno semelhantes, apresentadas na Figura 5.1. De maneira resumida, as seguintes condições serão aplicadas:

- Condição de entrada (*inflow*)
 - Condição de Dirichlet para velocidade horizontal: $u = 1.0$
 - Condição de Dirichlet para velocidade vertical: $v = 0.0$
 - Condição de Dirichlet para função corrente: $\psi = y$
- Condição de saída (*outflow*)
 - Condição de Neumann homogêneo para função corrente: $\nabla\psi \cdot \mathbf{n} = 0.0$
- Condição de parede afastada (*farfield*)

- Condição de Neumann homogêneo para velocidade horizontal: $\nabla u \cdot \mathbf{n} = 0.0$
- Condição de Dirichlet para velocidade vertical: $v = 0.0$
- Condição de Dirichlet para função corrente: $\psi = y$
- Condição de não-escorregamento para a geometria central (*noSlip*)
 - Condição de Dirichlet para velocidade horizontal: $u = 0.0$
 - Condição de Dirichlet para velocidade vertical: $v = 0.0$
 - Condição de Dirichlet para função corrente: $\psi = y_c$, na qual y_c representa o valor da altura central do domínio

A vorticidade é calculada a partir dos demais campos, como foi apresentado no capítulo 2.

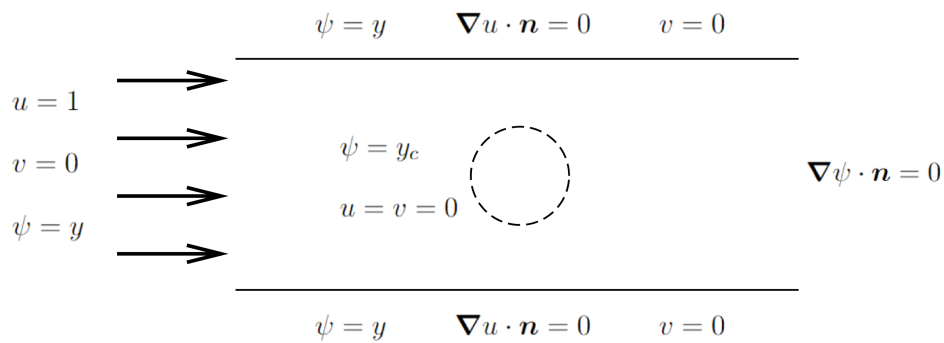


Figura 5.1: Condições iniciais e de contorno considerando uma figura central genérica.

Para a confecção das malhas computacionais, foi utilizado o *software* de código aberto *Gmsh*. Já para o pós-processamento, foi utilizado o *software* de código aberto *ParaView*. Em todas as simulações, os seguintes parâmetros foram utilizados:

Passo de tempo Δt	0.01 [s]
Número de Reynolds (Re)	100
Número de Iterações	600

Tabela 5.1: Parâmetros da simulação

5.1 Escoamento ao redor de corpo quadrado

Para a simulação desse caso, foi discretizado um domínio retangular com um quadrado na parte anterior, permitindo uma melhor observação do comportamento do fluido na parte posterior. O quadrado foi posicionado de maneira a encontrar o escoamento perpendicularmente. O domínio e suas faces de contorno são representadas na Figura 5.2.

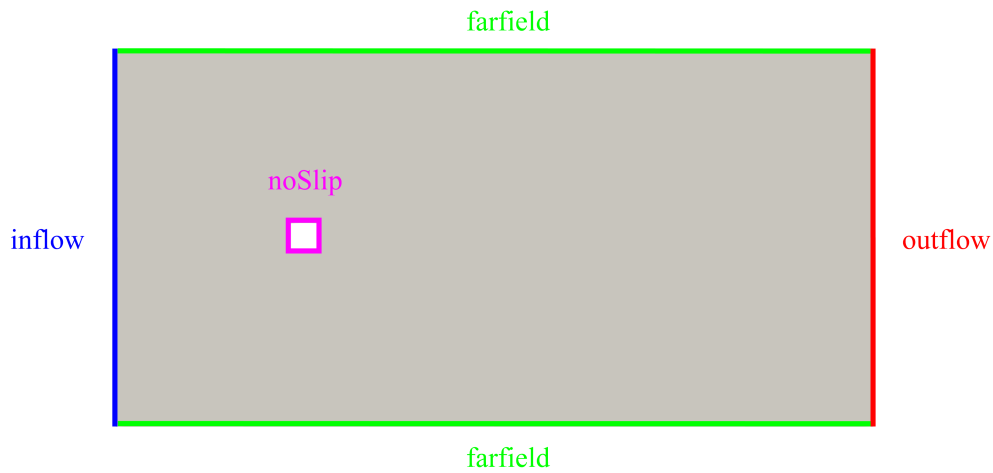


Figura 5.2: Domínio e condições de contorno do caso

Tomando esse domínio como referência, foi construída uma malha computacional com 6520 elementos triangulares, apresentada na Figura 5.3.

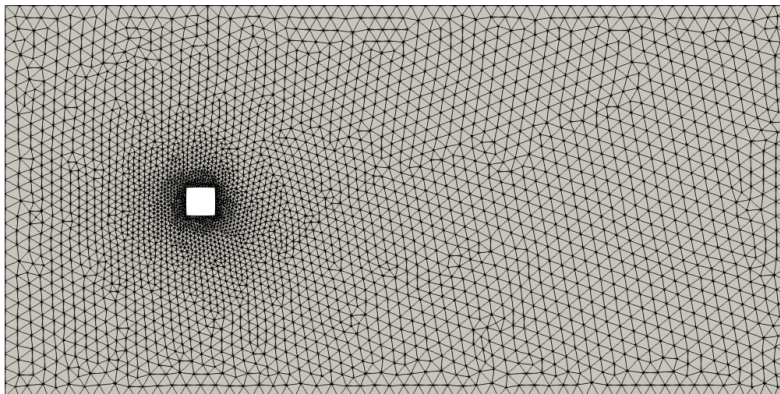


Figura 5.3: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão

no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.4.

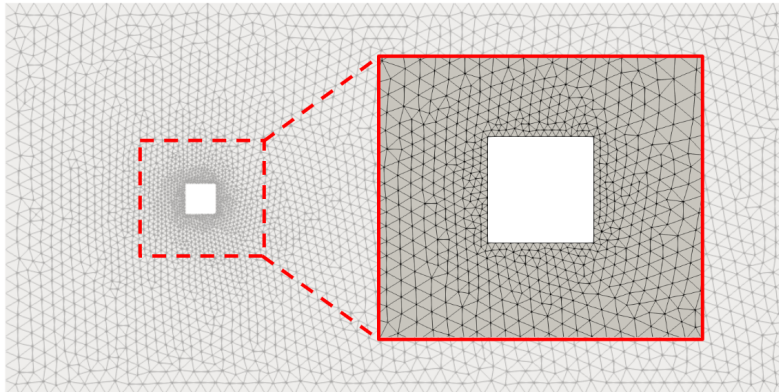


Figura 5.4: Detalhe da malha computacional do caso na região de interesse

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.5.

É interessante notar como a velocidade, principalmente na direção horizontal, varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.6. Os resultados das grandezas podem ser observados nas Figuras 5.7 e 5.8.

Os resultados de ψ , em conformidade com os da velocidade e da vorticidade, sugerem a aparição de uma zona após a geometria onde pontos de circulação seriam formados. Para avaliar esse comportamento, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.9.

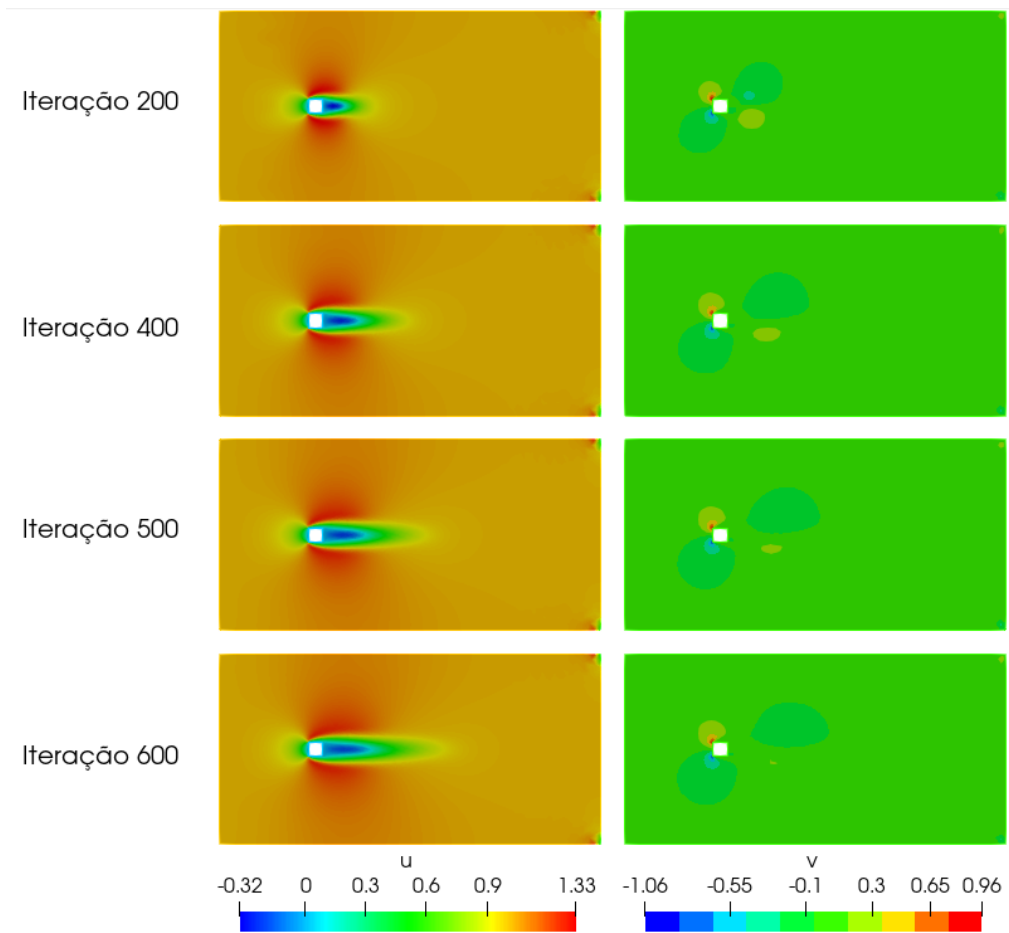


Figura 5.5: Evolução temporal dos campos de velocidade u e v

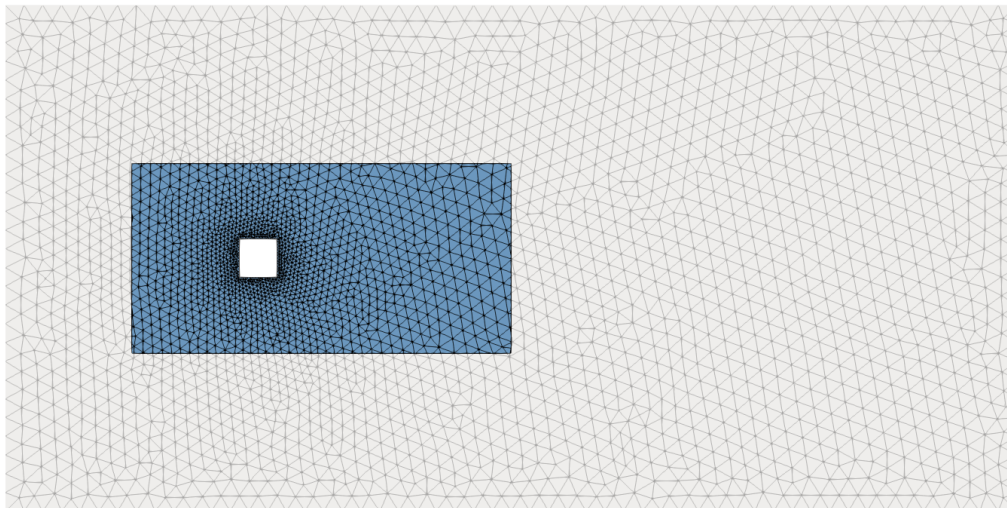


Figura 5.6: Destaque da região onde o corte foi executado

Esse resultado demonstra a aparição de vórtices na região posterior da geometria simulada, o que é um resultado esperado e amplamente reportado na literatura específica (PANTON, 2013).

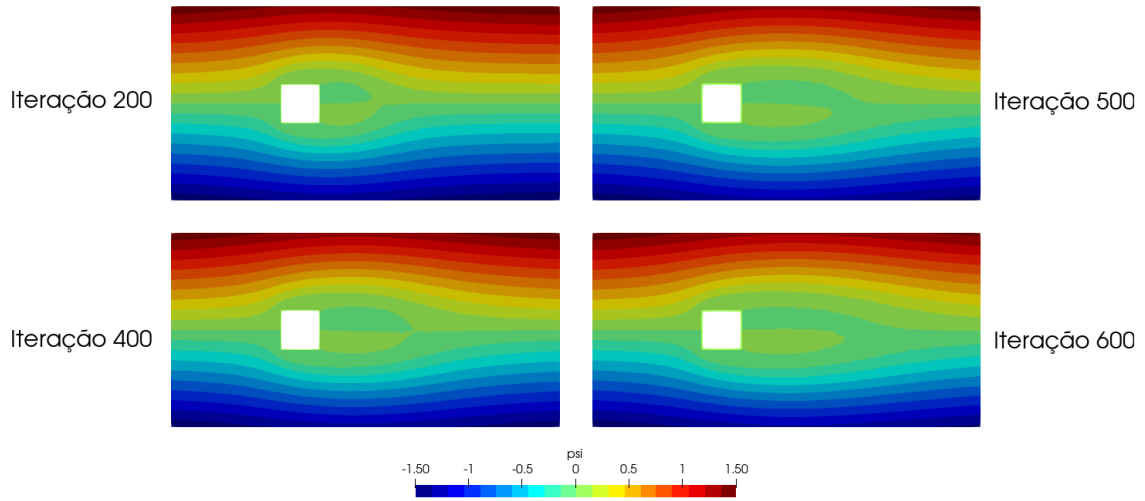


Figura 5.7: Evolução temporal da função corrente ψ

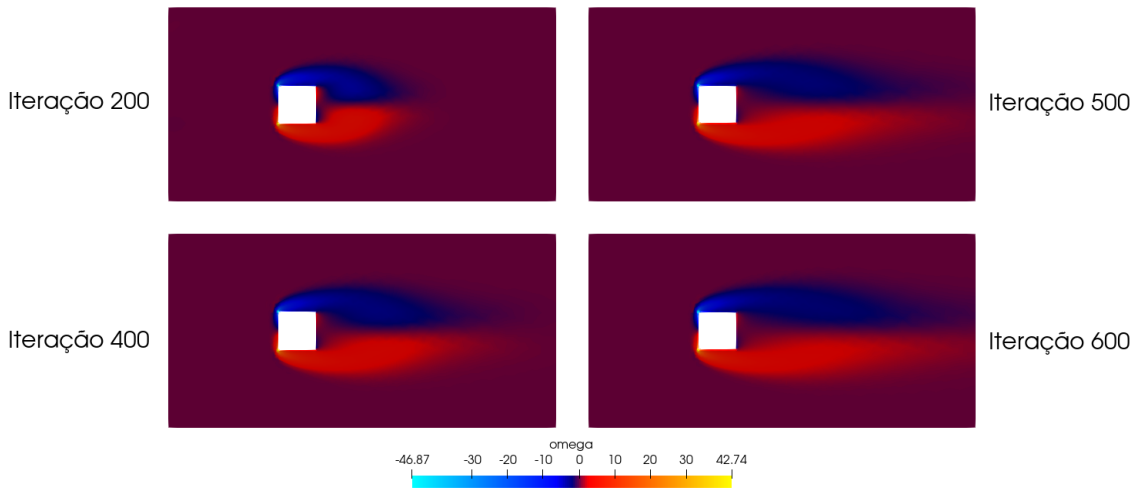


Figura 5.8: Evolução temporal do campo de vorticidade ω

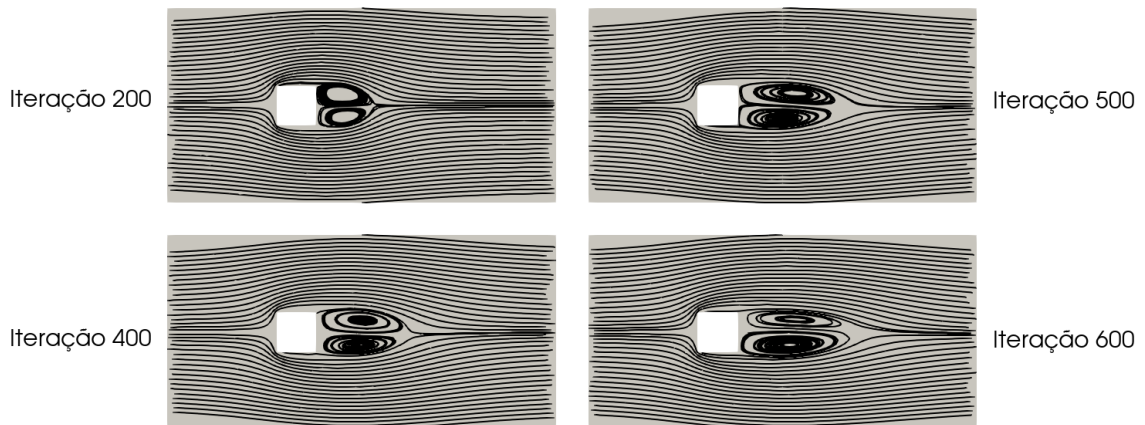


Figura 5.9: Evolução temporal das linhas de corrente

Para fins de comparação entre as simulações, alguns resultados numéricos das variáveis foram extraídos. Além dos dados de u , v e ω , o *software ParaView* permite o cálculo do gradiente dos campos, o que é interessante para identificar, por exemplo, variações de velocidade. Tais dados podem ser encontrados na Tabela 5.2.

Variável	Min.	Máx.
u	-0.32	1.33
v	-1.06	0.95
$ \nabla U $	0.0	39.13
ω	-49.72	45.09

Tabela 5.2: Avaliação das variáveis para comparação quantitativa

De acordo com a literatura, o aumento do número de Reynolds proporciona o descolamento e aparecimento de esteiras de vórtices, conhecido como esteiras de von Kármán. Com o intuito de verificar a capacidade do código de capturar esse comportamento, uma nova simulação com elevado número de Reynolds foi feita ($Re = 300$). Os resultados obtidos podem ser vistos nas Figuras 5.10 e 5.11.

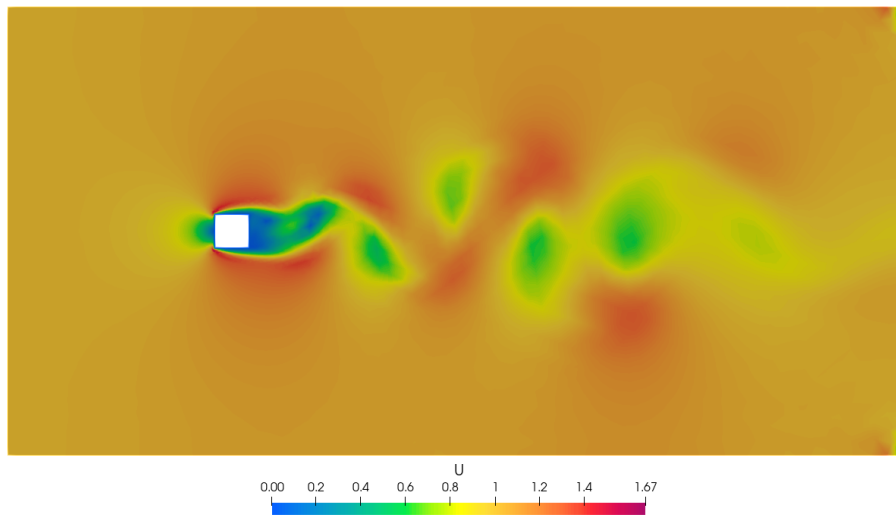


Figura 5.10: Velocidade ilustrando esteiras de von Kármán para Reynolds elevado

De fato, para um maior número de Reynolds, há o aparecimento das esteiras de von Kármán. Esse resultado reforça a confiabilidade da formulação corrente-

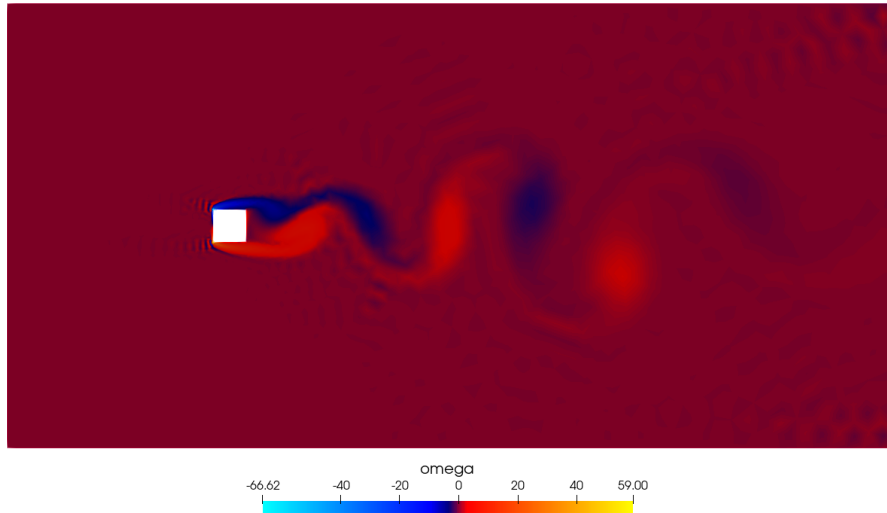


Figura 5.11: Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado

vorticidade em representar matematicamente e fisicamente os fenômenos naturais regidos pelas equações de Navier-Stokes, além de validar o potencial do código numérico desenvolvido.

5.2 escoamento ao redor de corpo triangular

Para a simulação desses casos, foi discretizado um domínio retangular com um triângulo na parte anterior, permitindo uma melhor observação do comportamento do fluido na parte posterior. O triângulo foi posicionado de maneira a encontrar o escoamento perpendicularmente. Esse conjunto de casos conta com a simulação de quatro domínios diferentes:

- Caso A: Triângulo equilátero contra o escoamento
- Caso B: Triângulo isósceles contra o escoamento
- Caso C: Triângulo equilátero de encontro ao escoamento
- Caso D: Triângulo isósceles de encontro ao escoamento

Tomando o caso A como referência, o domínio e suas faces de contorno são representadas na Figura 5.12.

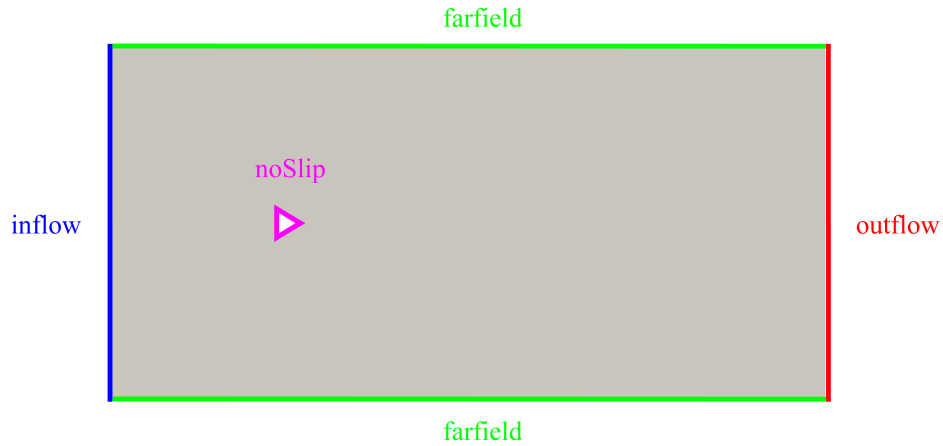


Figura 5.12: Domínio e condições de contorno do caso

5.2.1 Caso A: Triângulo equilátero contra o escoamento

Tomando o domínio da Figura 5.12 como referência, foi construída uma malha computacional com 5968 elementos triangulares, apresentada na Figura 5.13.

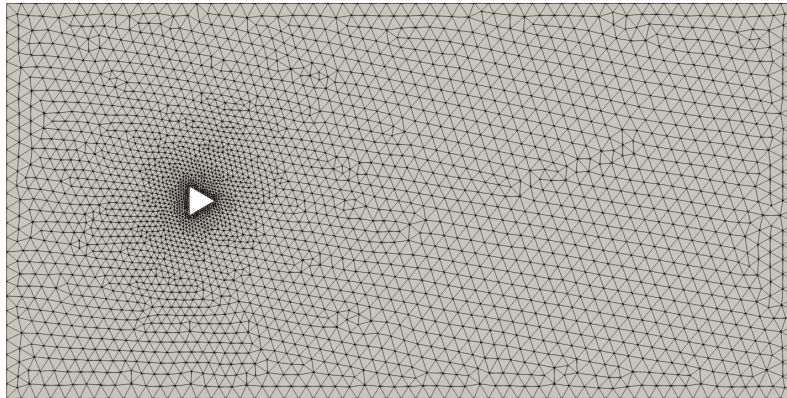


Figura 5.13: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.14.

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.15.

É interessante notar como a velocidade, principalmente na direção horizontal,

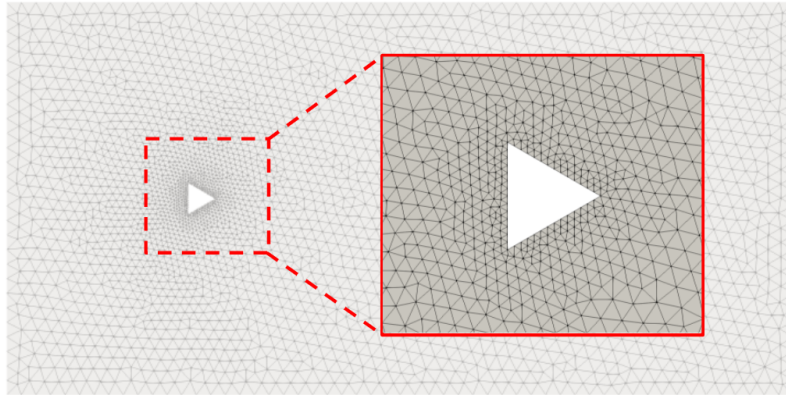


Figura 5.14: Detalhe da malha computacional do caso na região de interesse

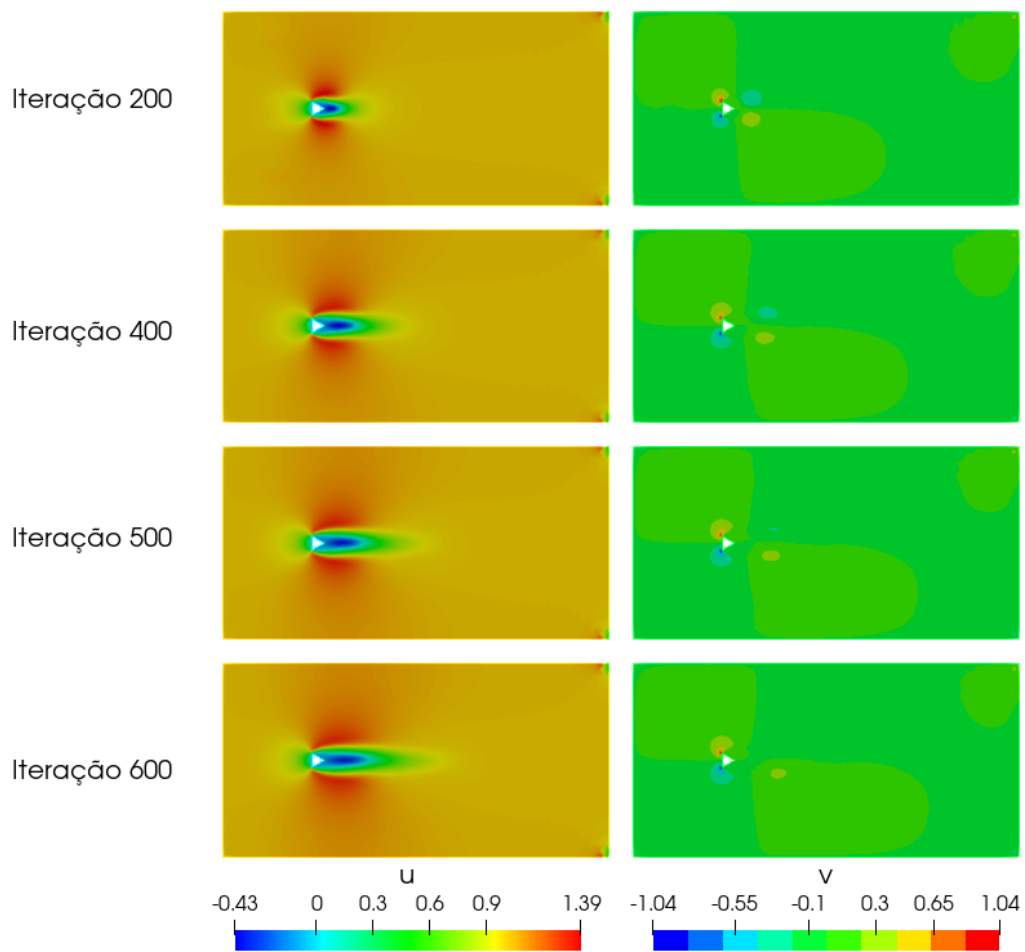


Figura 5.15: Evolução temporal dos campos de velocidade u e v

varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.16. Os resultados das grandezas podem ser observados nas Figuras 5.17 e 5.18.

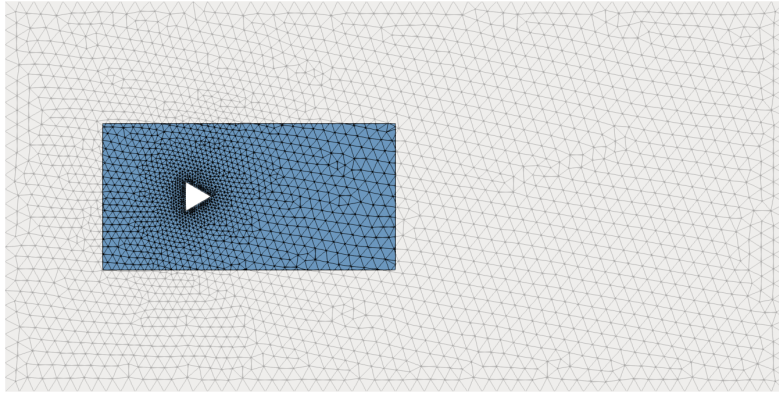


Figura 5.16: Destaque da região onde o corte foi executado

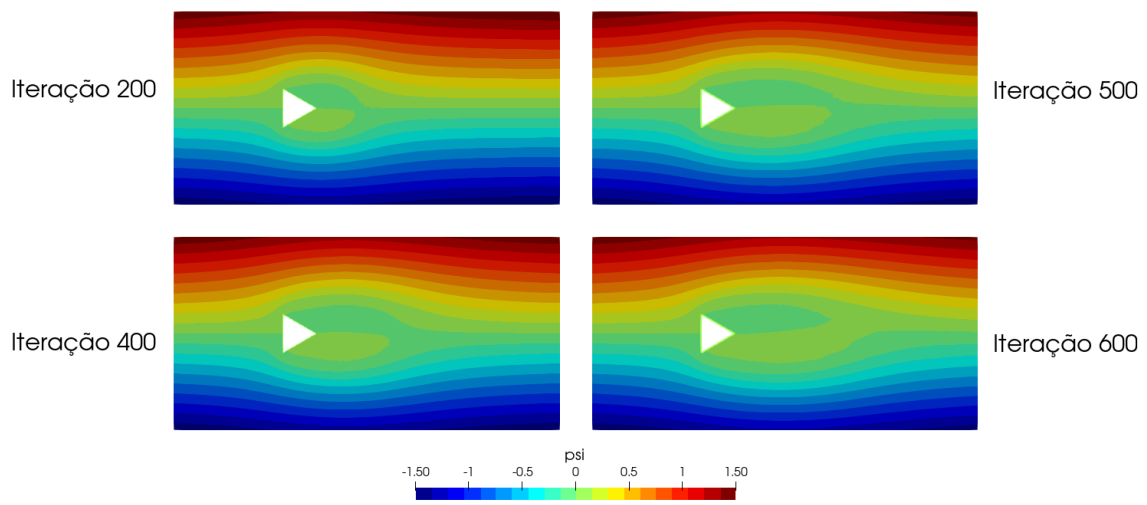


Figura 5.17: Evolução temporal da função corrente ψ

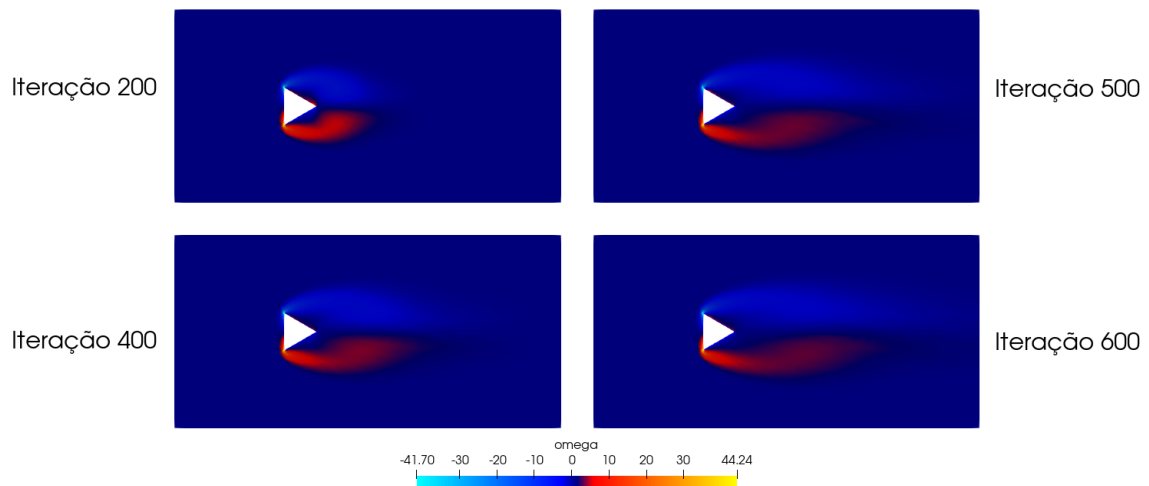


Figura 5.18: Evolução temporal do campo de vorticidade ω

Os resultados de ψ , em conformidade com os da velocidade e da vorticidade, sugerem a aparição de uma zona após a geometria onde pontos de circulação seriam formados. Para avaliar esse comportamento, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.19.

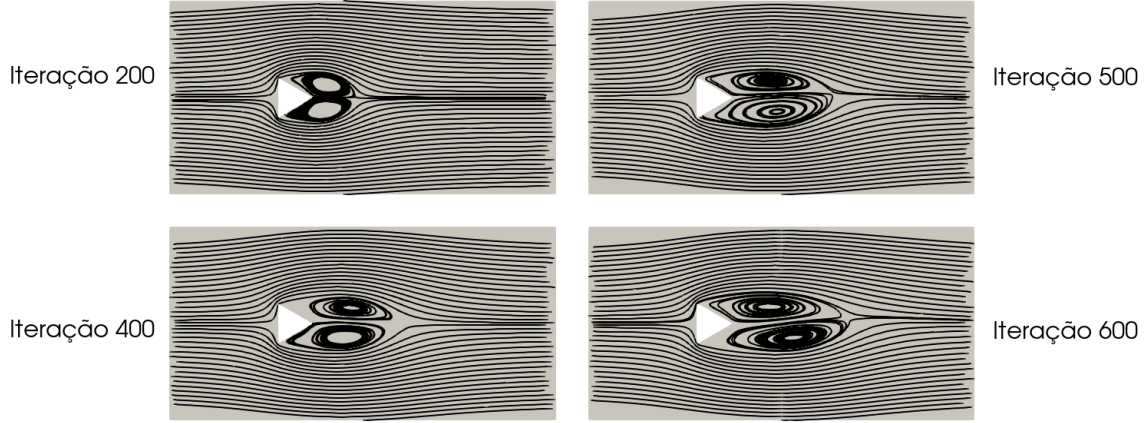


Figura 5.19: Evolução temporal das linhas de corrente

Esse resultado demonstra a aparição de vórtices na região posterior da geometria simulada, o que é um resultado esperado e amplamente reportado na literatura específica (PANTON, 2013). Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.3.

Variável	Min.	Máx.
u	-0.38	1.31
v	-0.97	0.96
$ \nabla U $	0.0	46.70
ω	-41.70	44.24

Tabela 5.3: Avaliação das variáveis para comparação quantitativa

De maneira análoga a comentada no caso anterior, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.20 e 5.21.

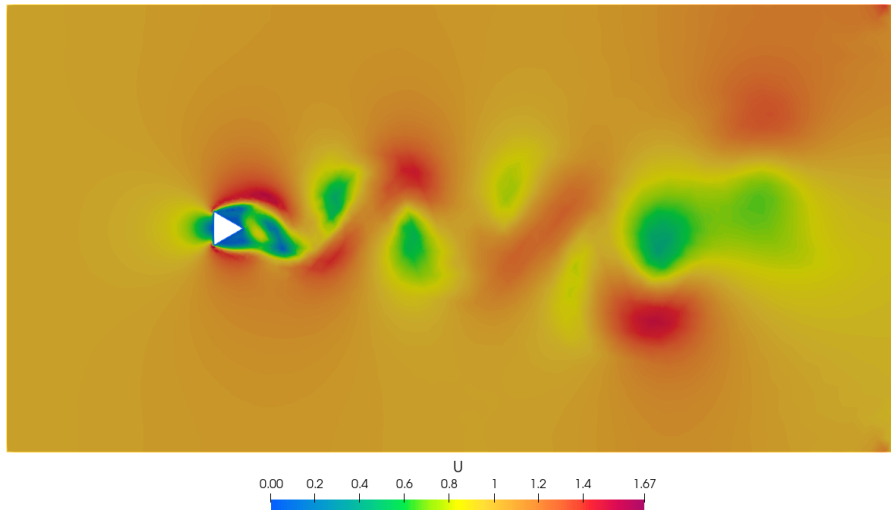


Figura 5.20: Velocidade ilustrando esteiras de von Kármán para Reynolds elevado

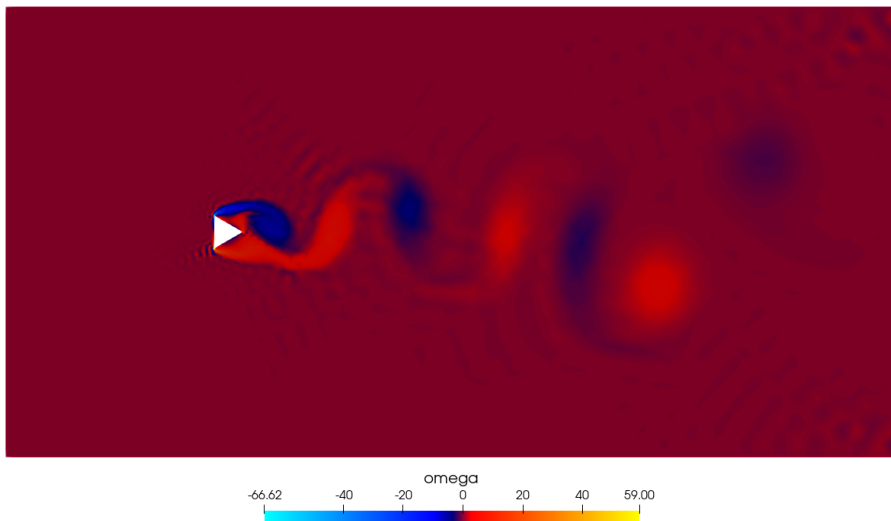


Figura 5.21: Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado

5.2.2 Caso B: Triângulo isósceles contra o escoamento

Tomando o domínio da Figura 5.12 como referência, foi construída uma malha computacional com 6608 elementos triangulares, apresentada na Figura 5.22.

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.23.

Após o término da simulação, os resultados obtidos foram pós-processados e

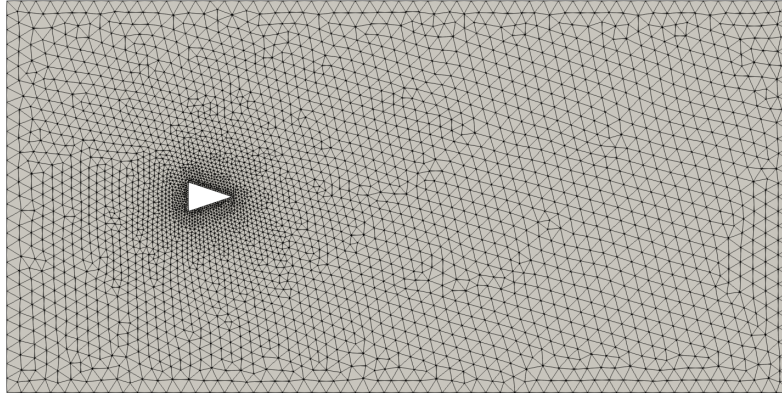


Figura 5.22: Malha computacional do caso

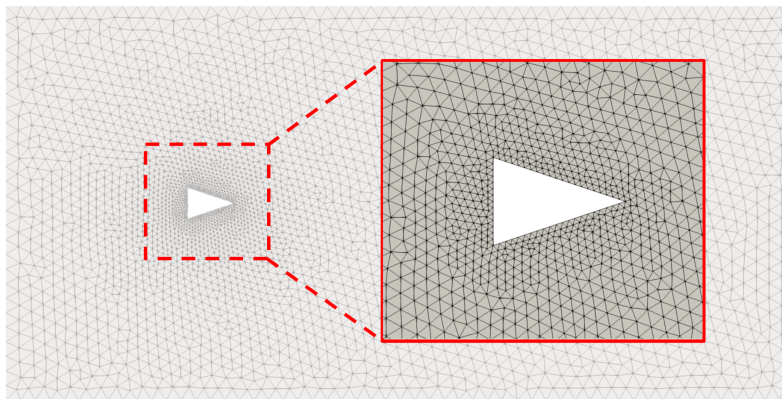


Figura 5.23: Detalhe da malha computacional do caso na região de interesse

avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.24.

É interessante notar como a velocidade, principalmente na direção horizontal, varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.25. Os resultados das grandezas podem ser observados nas Figuras 5.26 e 5.27.

Os resultados de ψ , em conformidade com os da velocidade e da vorticidade, sugerem a aparição de uma zona após a geometria onde pontos de circulação seriam formados. Para avaliar esse comportamento, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.28.

Esse resultado demonstra a aparição de vórtices na região posterior da geometria simulada, o que é um resultado esperado e amplamente reportado na literatura

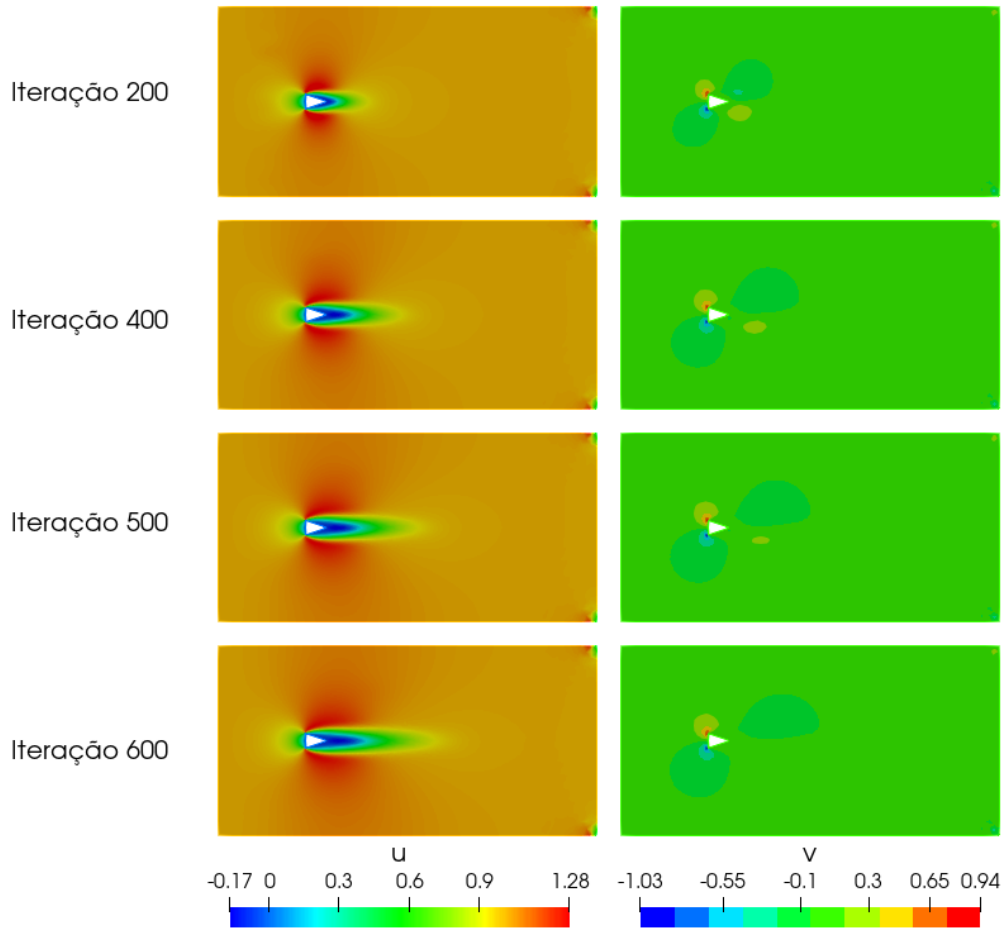


Figura 5.24: Evolução temporal dos campos de velocidade u e v

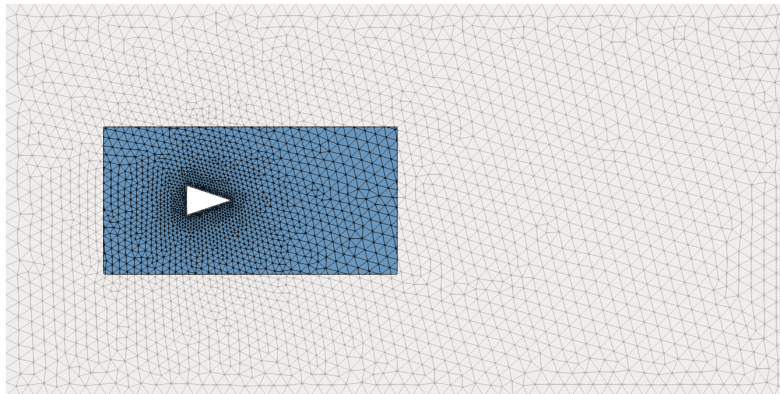


Figura 5.25: Destaque da região onde o corte foi executado

específica (PANTON, 2013). Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.4.

De maneira análoga a comentada nos casos anteriores, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as

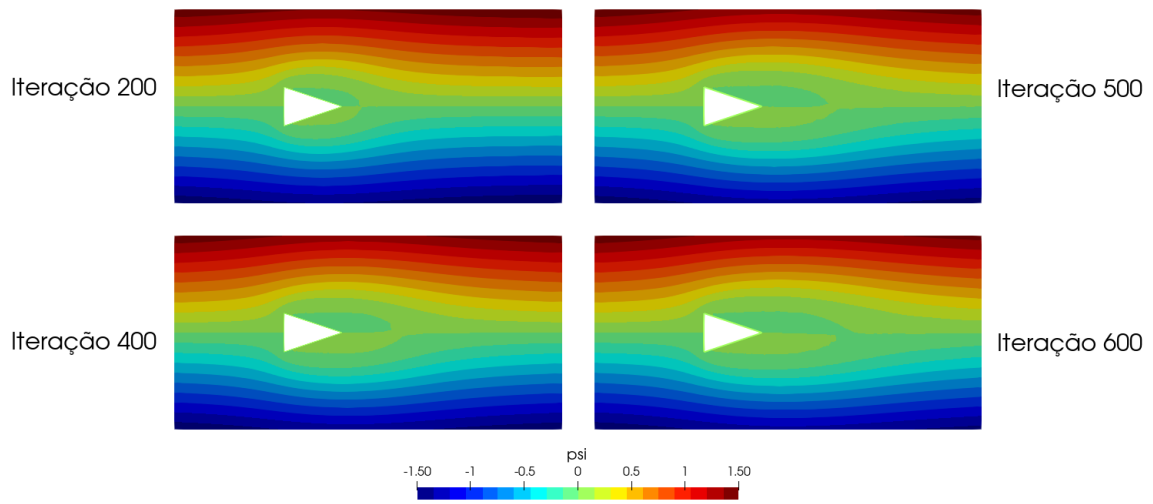


Figura 5.26: Evolução temporal da função corrente ψ

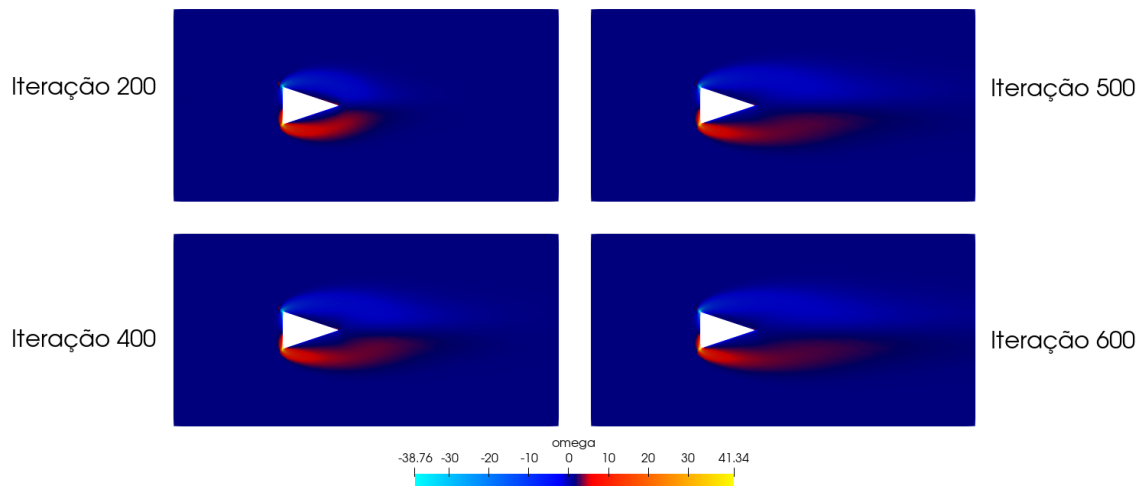


Figura 5.27: Evolução temporal do campo de vorticidade ω_z

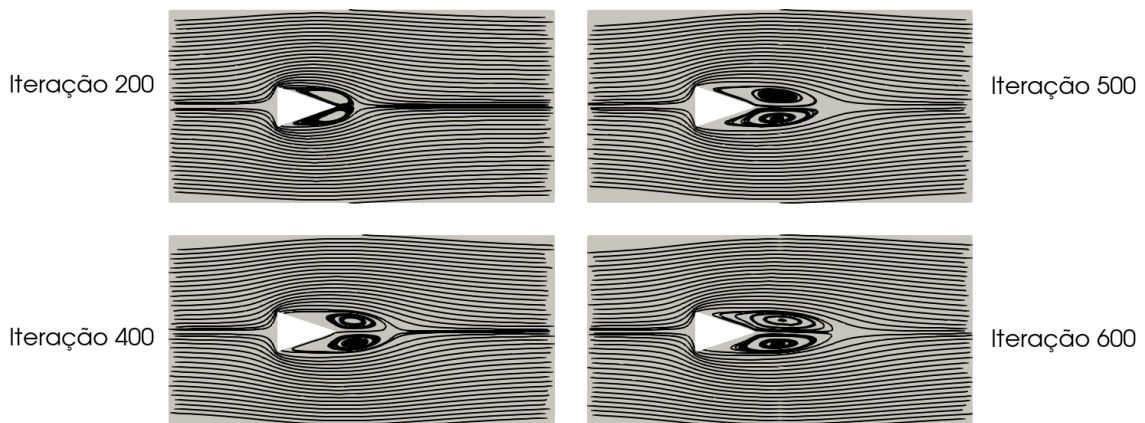


Figura 5.28: Evolução temporal das linhas de corrente

Variável	Min.	Máx.
u	-0.17	1.28
v	-1.03	0.94
$ \nabla U $	0.0	32.70
ω	-38.76	41.34

Tabela 5.4: Avaliação das variáveis para comparação quantitativa

esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.29 e 5.30.

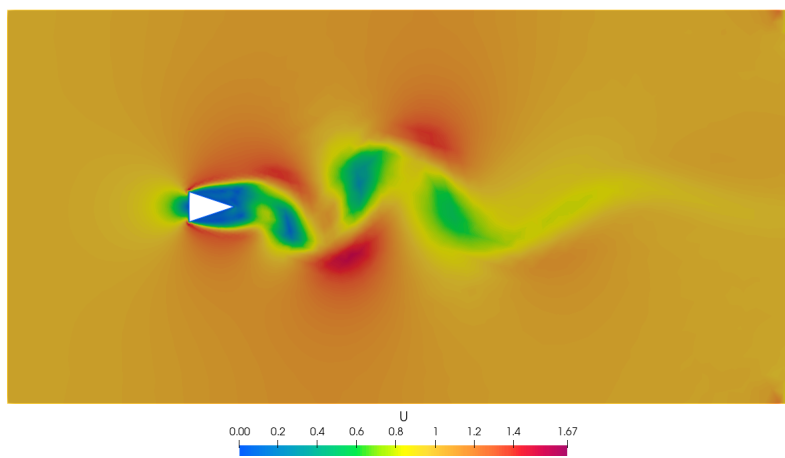


Figura 5.29: Velocidade ilustrando esteiras de von Kármán para Reynolds elevado

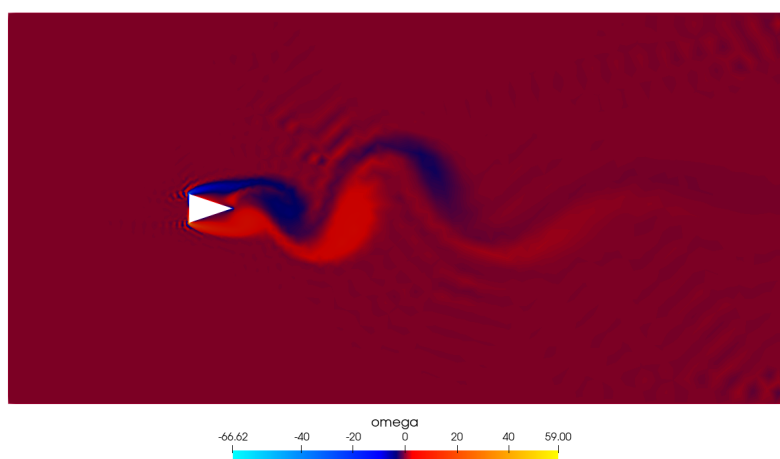


Figura 5.30: Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado

5.2.3 Caso C: Triângulo equilátero alinhado ao escoamento

Tomando o domínio da Figura 5.12 como referência, foi construída uma malha computacional com 6156 elementos triangulares, apresentada na Figura 5.31.

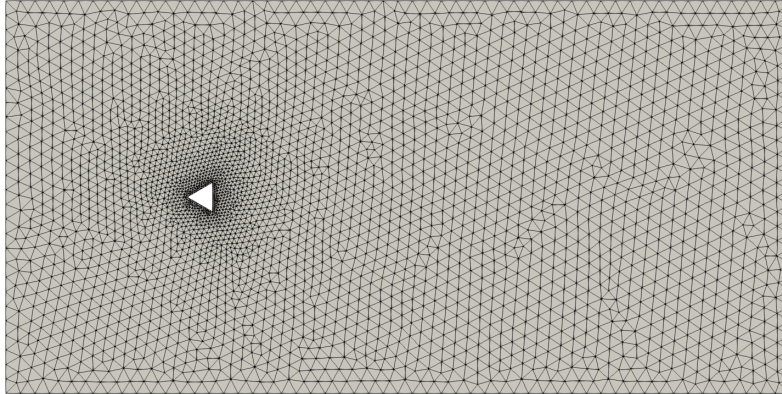


Figura 5.31: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.32.

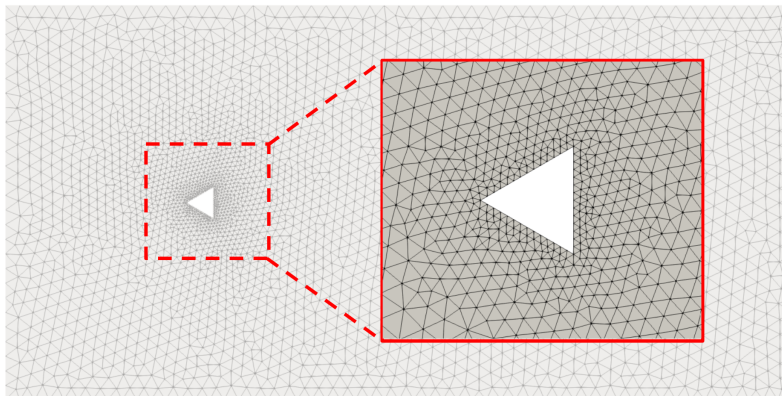


Figura 5.32: Detalhe da malha computacional do caso na região de interesse

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.33.

É interessante notar como a velocidade, principalmente na direção horizontal, varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é

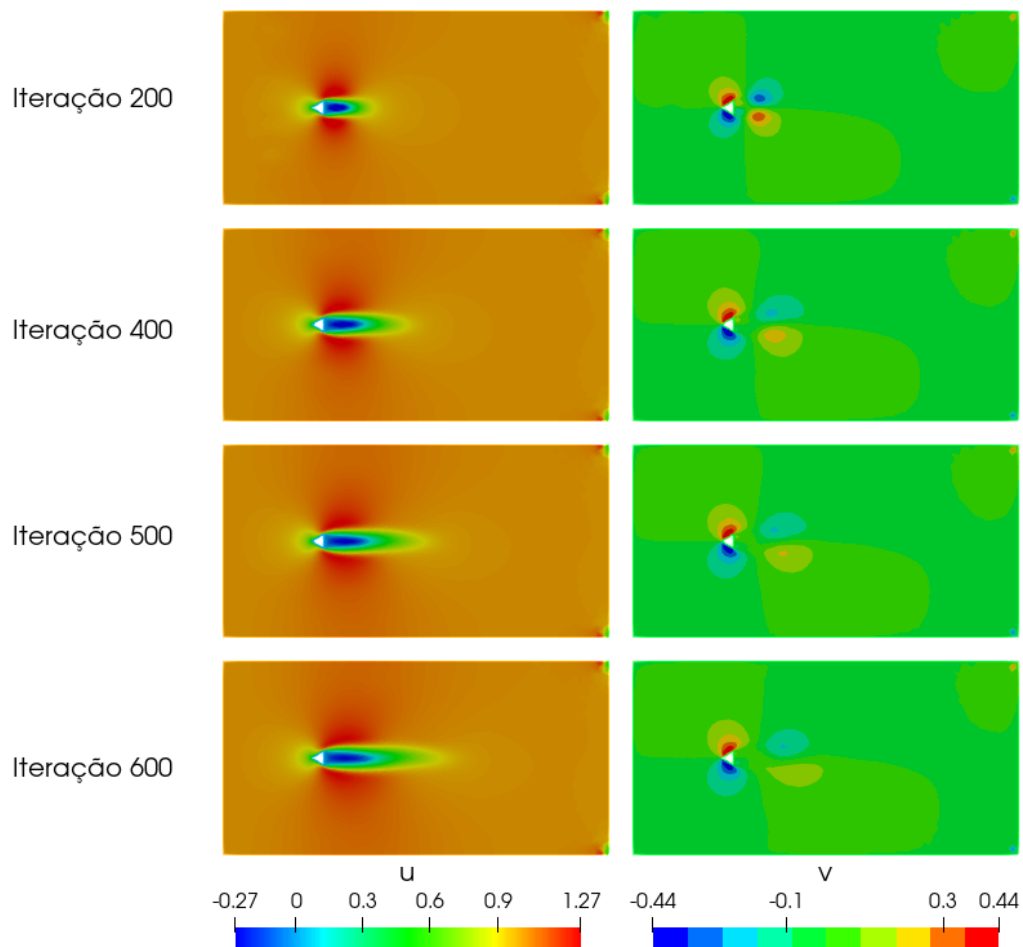


Figura 5.33: Evolução temporal dos campos de velocidade u e v

típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.34. Os resultados das grandezas podem ser observados nas Figuras 5.35 e 5.36.

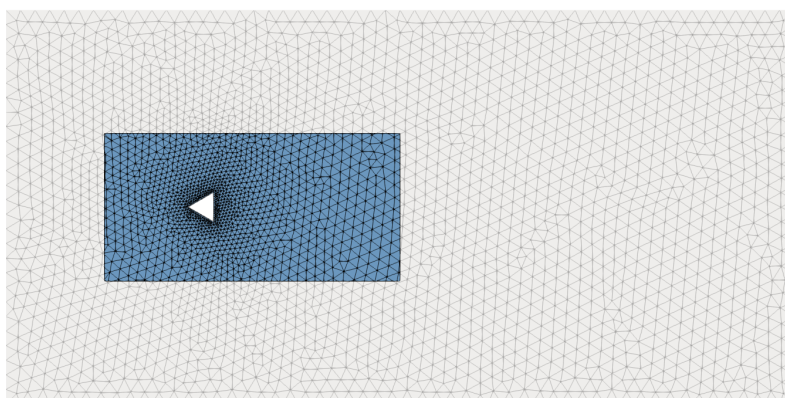


Figura 5.34: Destaque da região onde o corte foi executado

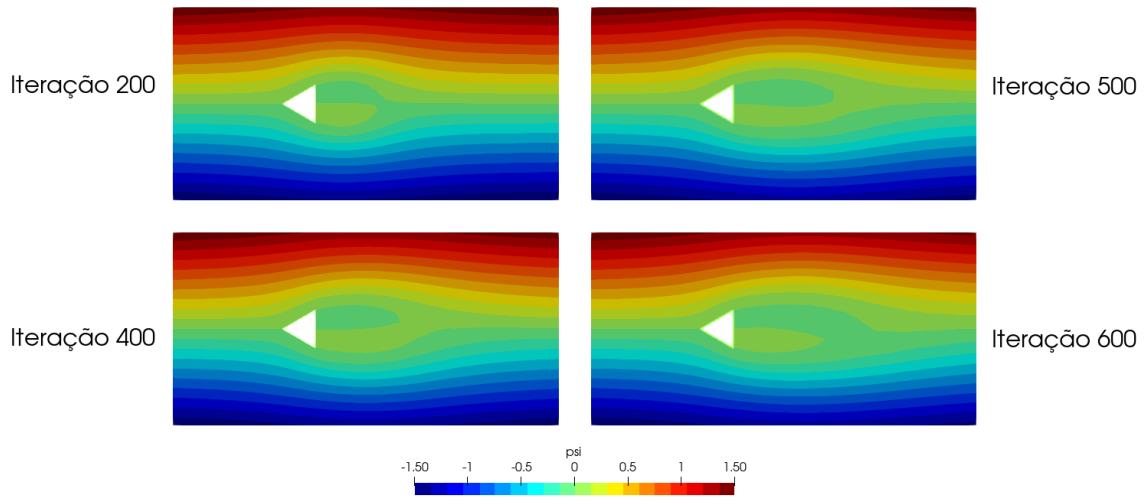


Figura 5.35: Evolução temporal da função corrente ψ

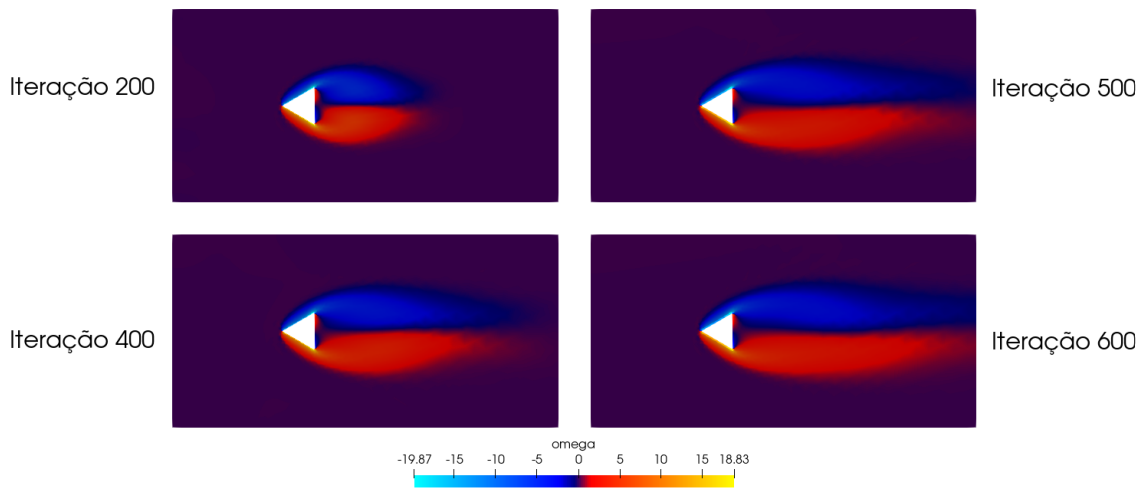


Figura 5.36: Evolução temporal do campo de vorticidade ω_z

Os resultados de ψ , em conformidade com os da velocidade e da vorticidade, sugerem a aparição de uma zona após a geometria onde pontos de circulação seriam formados. Para avaliar esse comportamento, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.37.

Esse resultado demonstra a aparição de vórtices na região posterior da geometria simulada, o que é um resultado esperado e amplamente reportado na literatura específica (PANTON, 2013). Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.5.

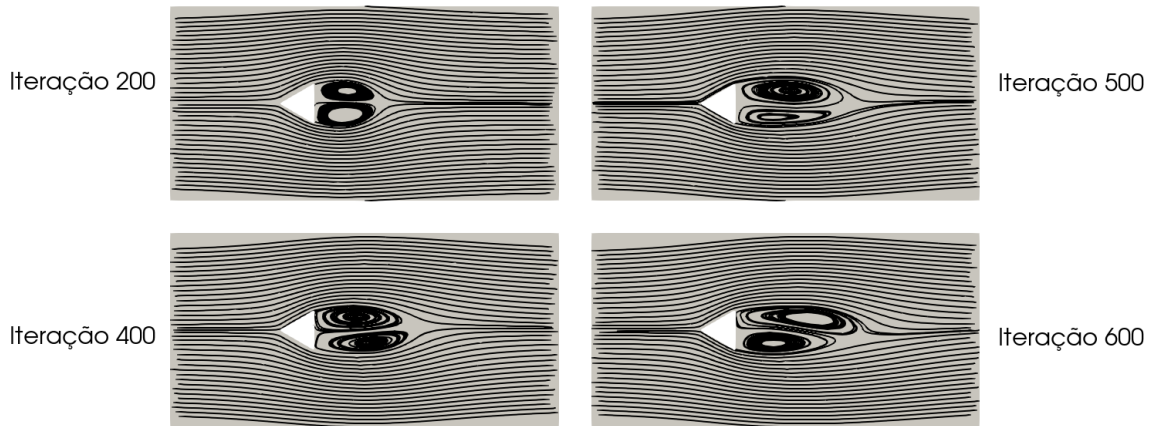


Figura 5.37: Evolução temporal das linhas de corrente

Variável	Min.	Máx.
u	-0.27	1.27
v	-0.44	0.44
$ \nabla U $	0.0	20.72
ω	-19.87	18.83

Tabela 5.5: Avaliação das variáveis para comparação quantitativa

De maneira análoga a comentada nos casos anteriores, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.38 e 5.39.

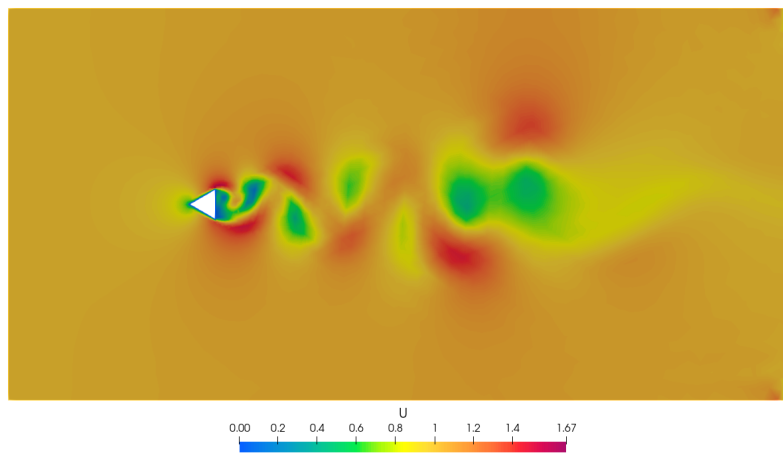


Figura 5.38: Velocidade ilustrando esteiras de von Kármán para Reynolds elevado

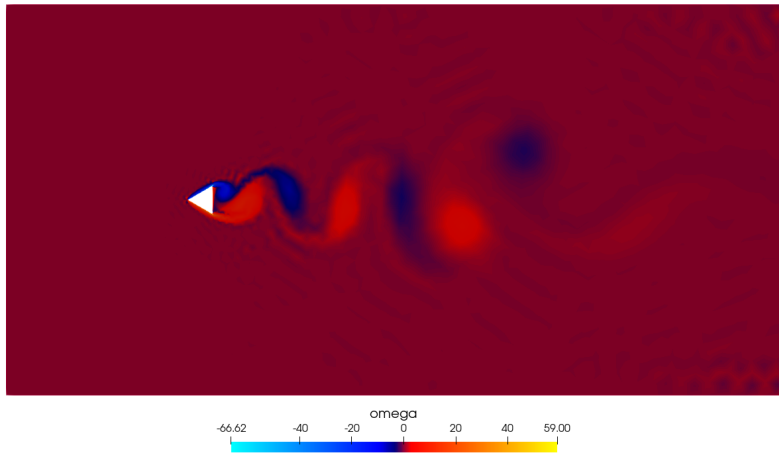


Figura 5.39: Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado

5.2.4 Caso D: Triângulo isósceles alinhado ao escoamento

Tomando o domínio da Figura 5.12 como referência, foi construída uma malha computacional com 6426 elementos triangulares, apresentada na Figura 5.40.

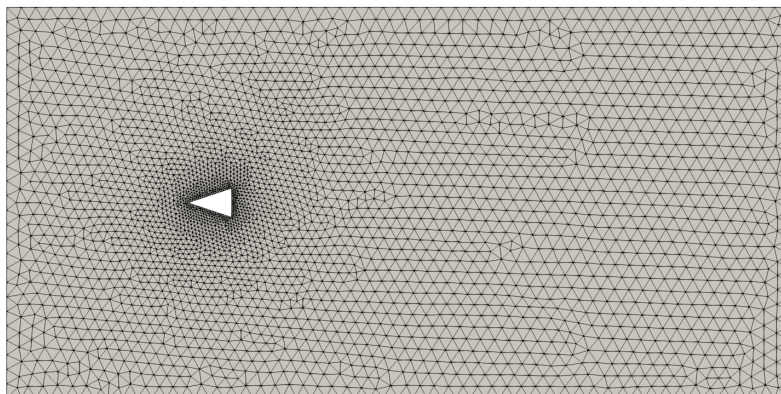


Figura 5.40: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.41.

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.42.

É interessante notar como a velocidade, principalmente na direção horizontal,

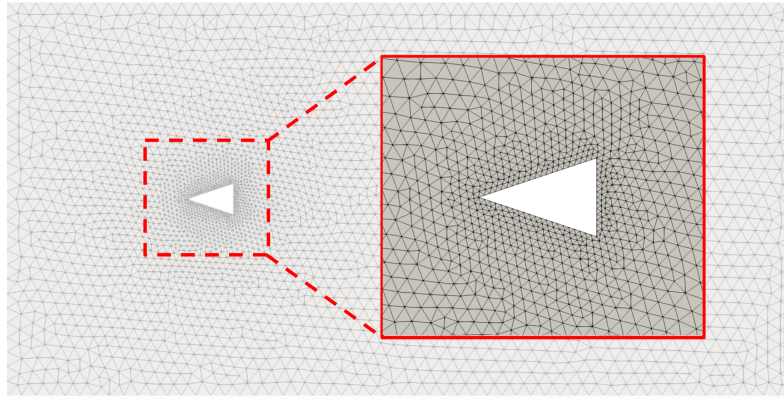


Figura 5.41: Detalhe da malha computacional do caso na região de interesse

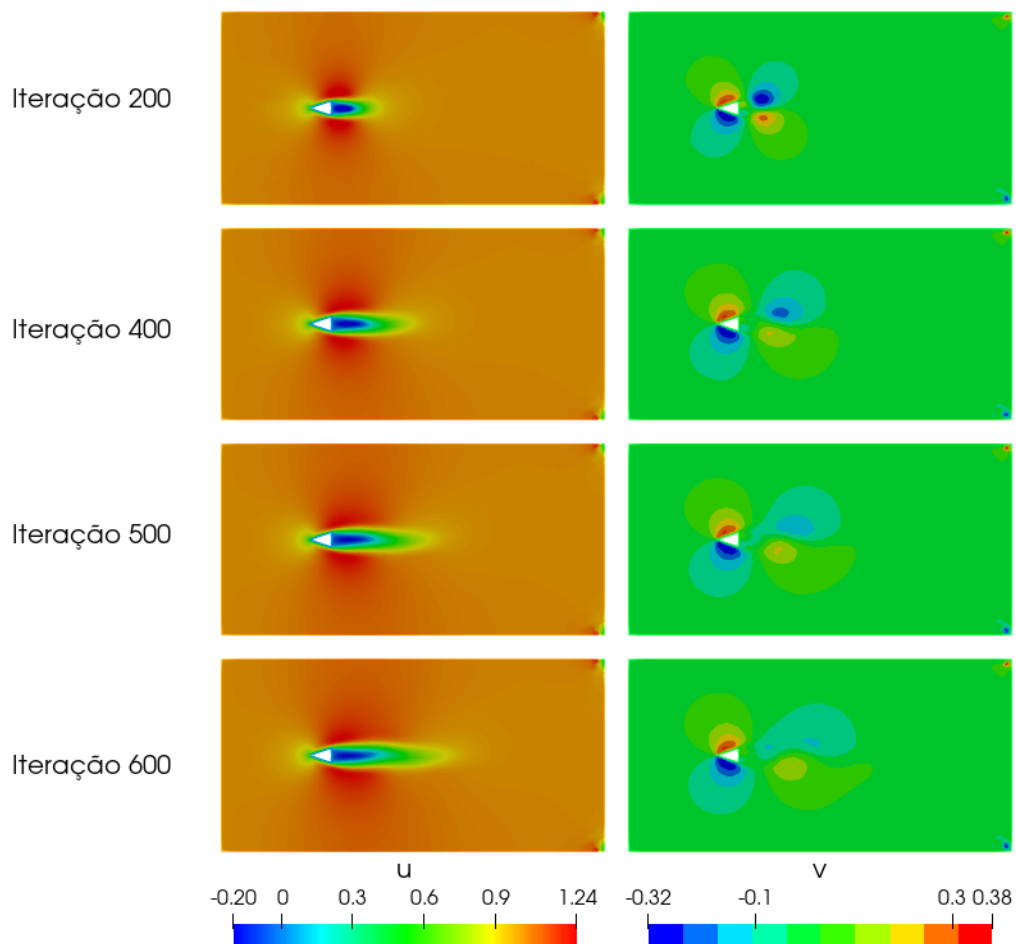


Figura 5.42: Evolução temporal dos campos de velocidade u e v

varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.43. Os resultados das grandezas podem ser observados

nas Figuras 5.44 e 5.45.

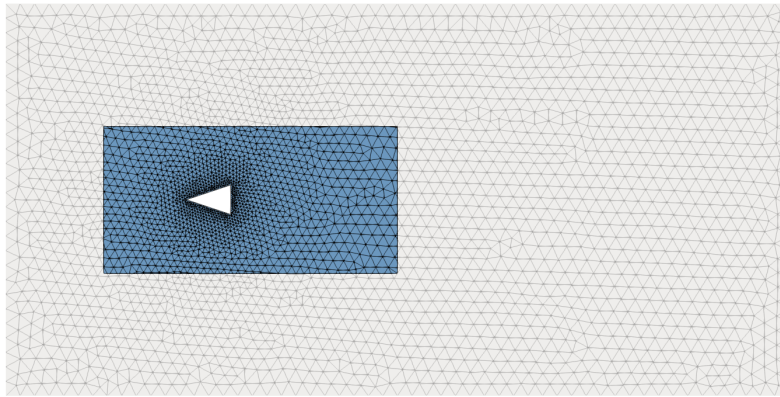


Figura 5.43: Destaque da região onde o corte foi executado

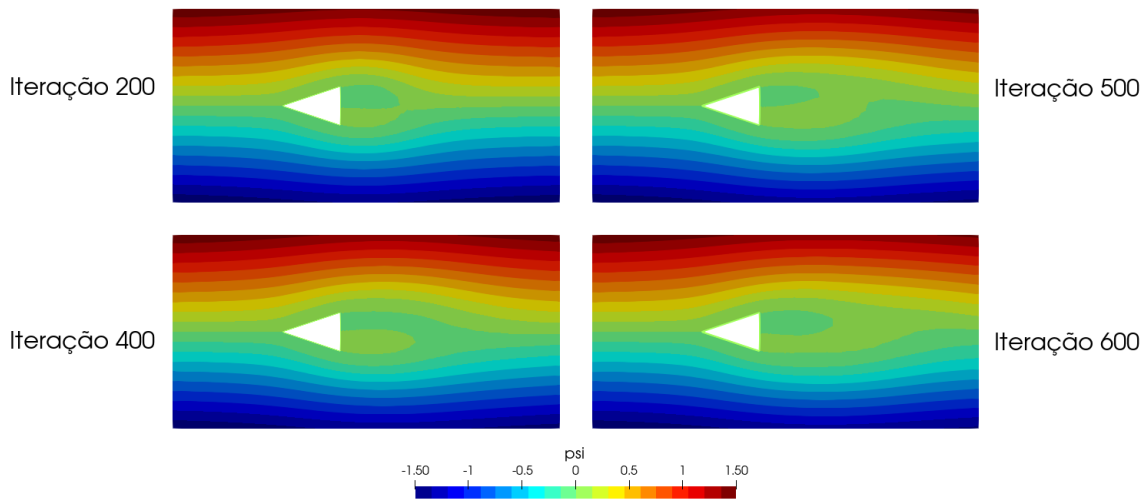


Figura 5.44: Evolução temporal da função corrente ψ

Os resultados de ψ , em conformidade com os da velocidade e da vorticidade, sugerem a aparição de uma zona após a geometria onde pontos de circulação seriam formados. Para avaliar esse comportamento, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.46.

Esse resultado demonstra a aparição de vórtices na região posterior da geometria simulada, o que é um resultado esperado e amplamente reportado na literatura específica (PANTON, 2013). Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.6.

De maneira análoga a comentada nos casos anteriores, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as

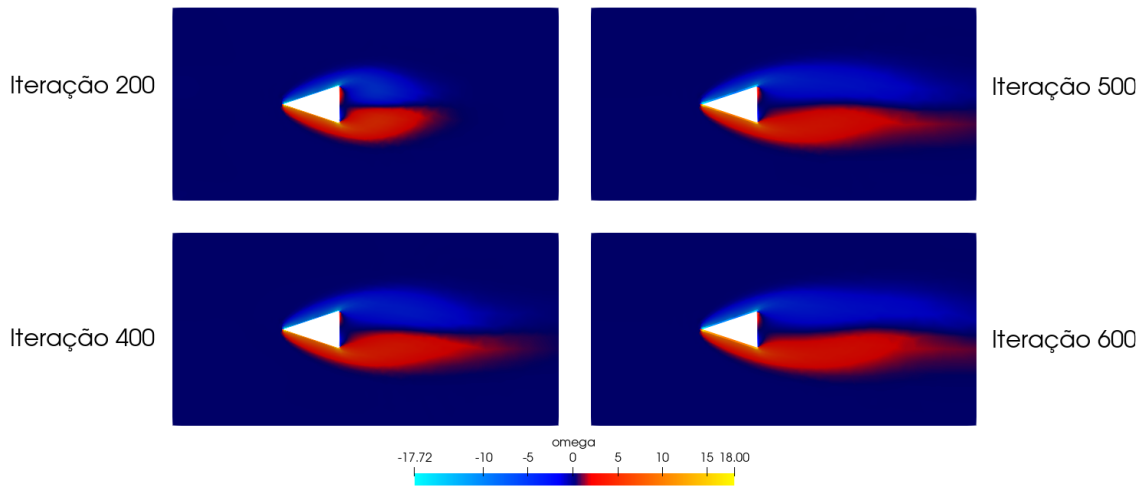


Figura 5.45: Evolução temporal do campo de vorticidade ω_z

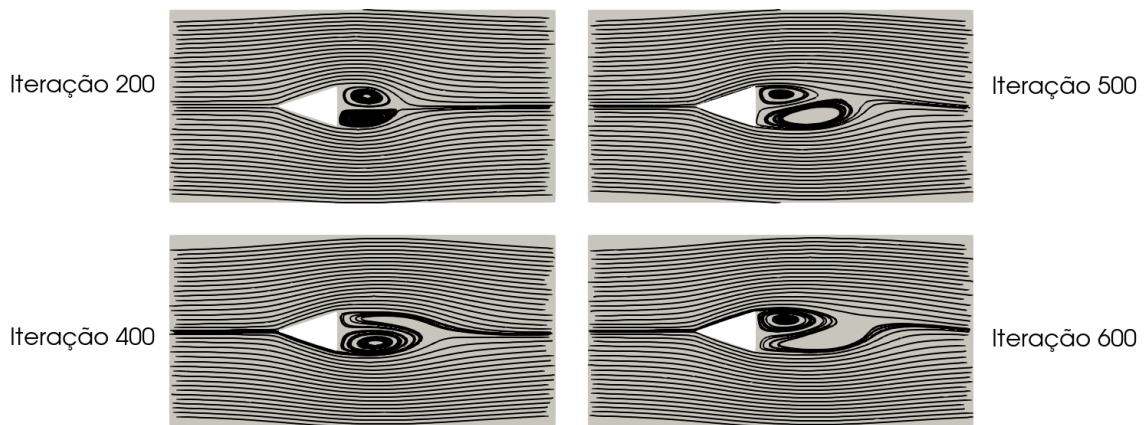


Figura 5.46: Evolução temporal das linhas de corrente

Variável	Min.	Máx.
u	-0.20	1.24
v	-0.32	0.38
$ \nabla U $	0.0	15.82
ω	-17.72	18.00

Tabela 5.6: Avaliação das variáveis para comparação quantitativa

esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.47 e 5.48.

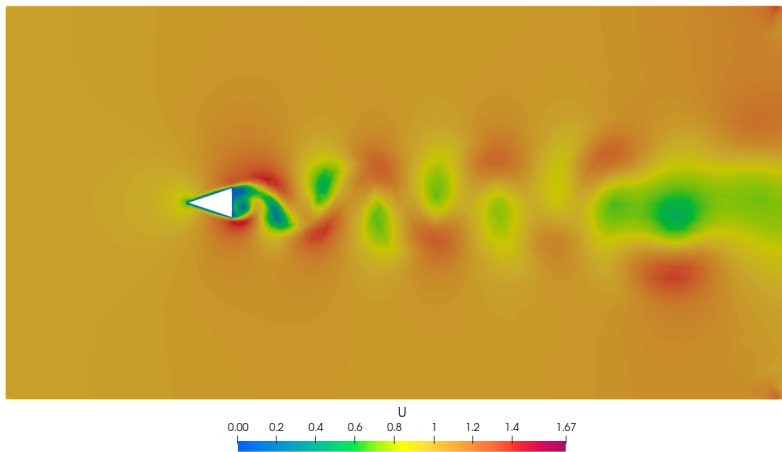


Figura 5.47: Velocidade ilustrando esteiras de von Kármán para Reynolds elevado

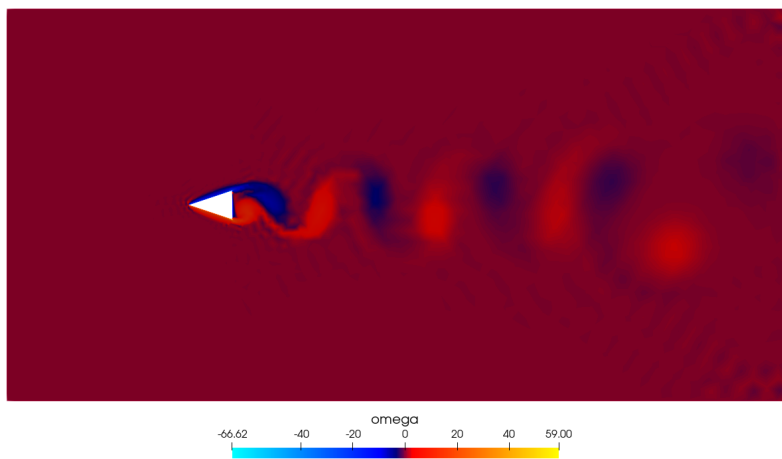


Figura 5.48: Vorticidade ilustrando esteiras de von Kármán para Reynolds elevado

5.3 Escoamento ao redor do aerofólio NACA 0012

Para a simulação desses casos, foi discretizado um domínio ao redor de um aerofólio NACA 0012, posicionado na parte anterior e permitindo uma melhor observação do comportamento do fluido na parte posterior. O aerofólio foi posicionado no centro, variando o seu ângulo de ataque para três situações distintas:

- Caso A: NACA 0012 com ângulo de ataque 0°
- Caso B: NACA 0012 com ângulo de ataque 8°
- Caso C: NACA 0012 com ângulo de ataque 20°

Tomando o caso A como referência, o domínio e suas faces de contorno são representadas na Figura 5.49.

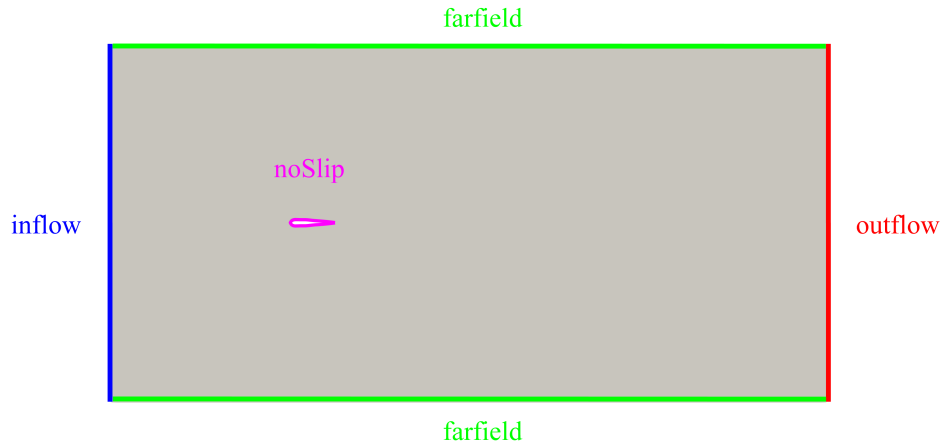


Figura 5.49: Domínio e condições de contorno do caso

5.3.1 Caso A: NACA 0012 com ângulo de ataque 0°

Tomando o domínio da Figura 5.49 como referência, foi construída uma malha computacional com 9672 elementos triangulares, apresentada na Figura 5.50.

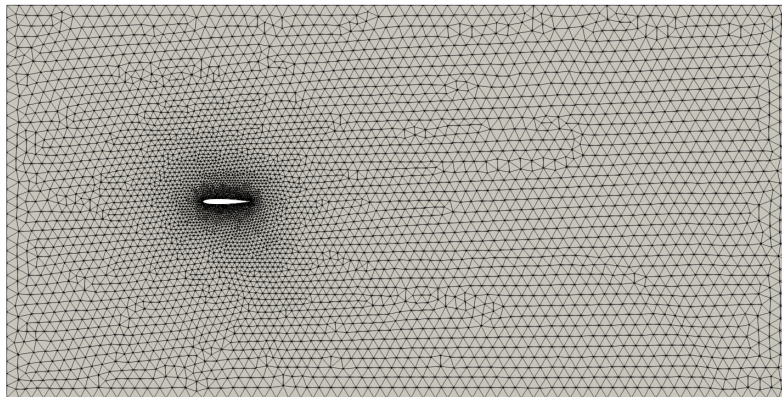


Figura 5.50: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.51.

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.52.

É interessante notar como a velocidade, principalmente na direção horizontal,

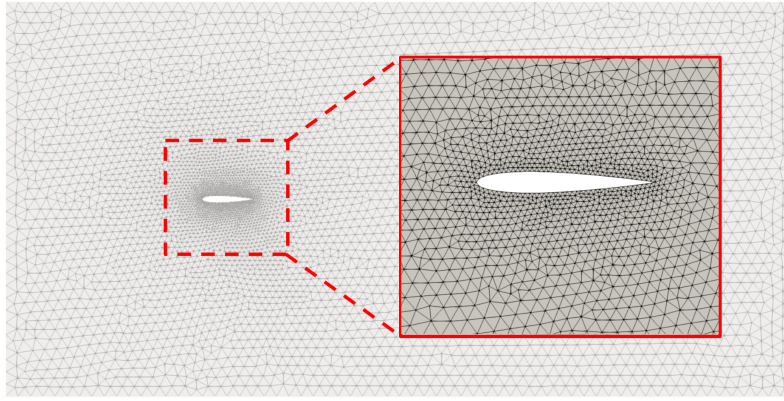


Figura 5.51: Detalhe da malha computacional do caso na região de interesse

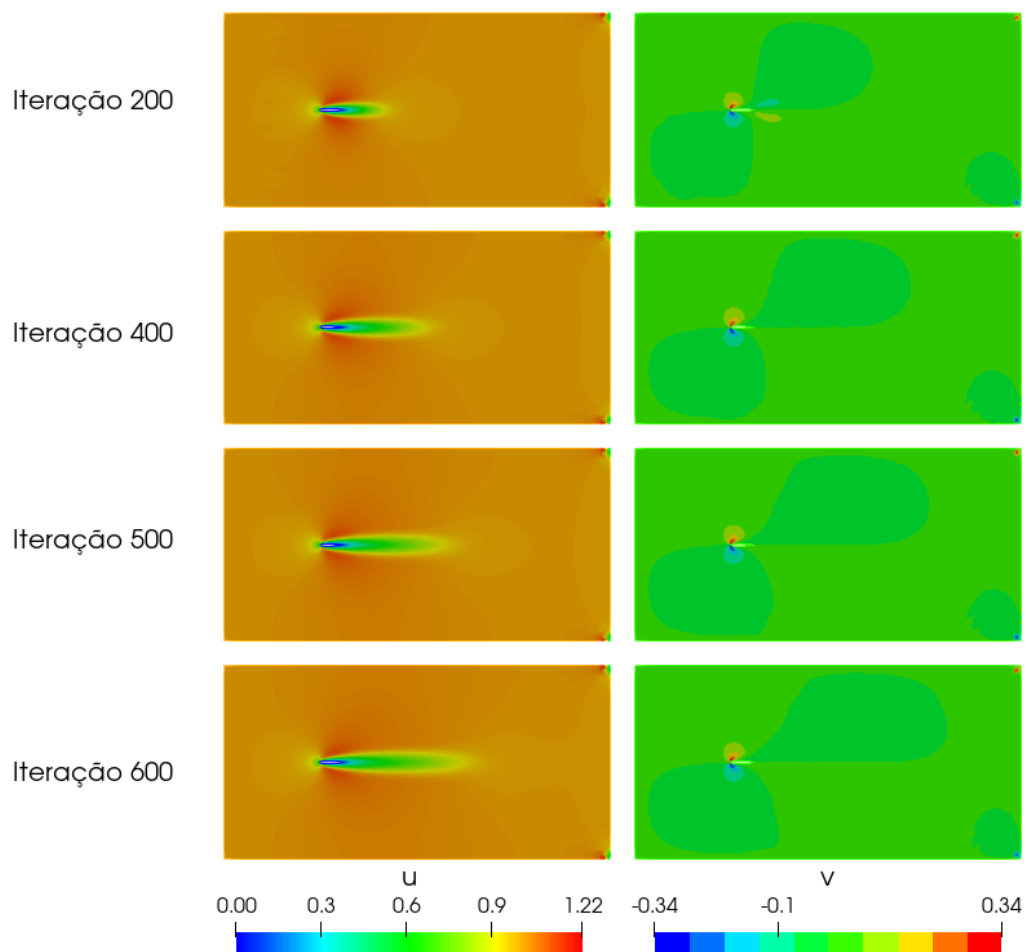


Figura 5.52: Evolução temporal dos campos de velocidade u e v

varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.53. Os resultados das grandezas podem ser observados nas Figuras 5.54 e 5.55.

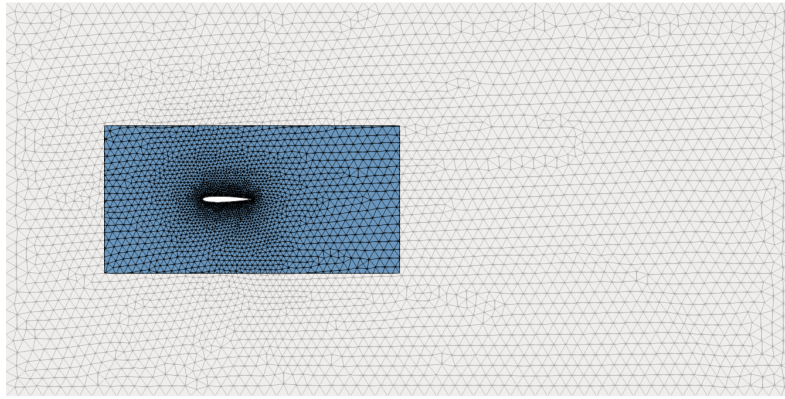


Figura 5.53: Destaque da região onde o corte foi executado

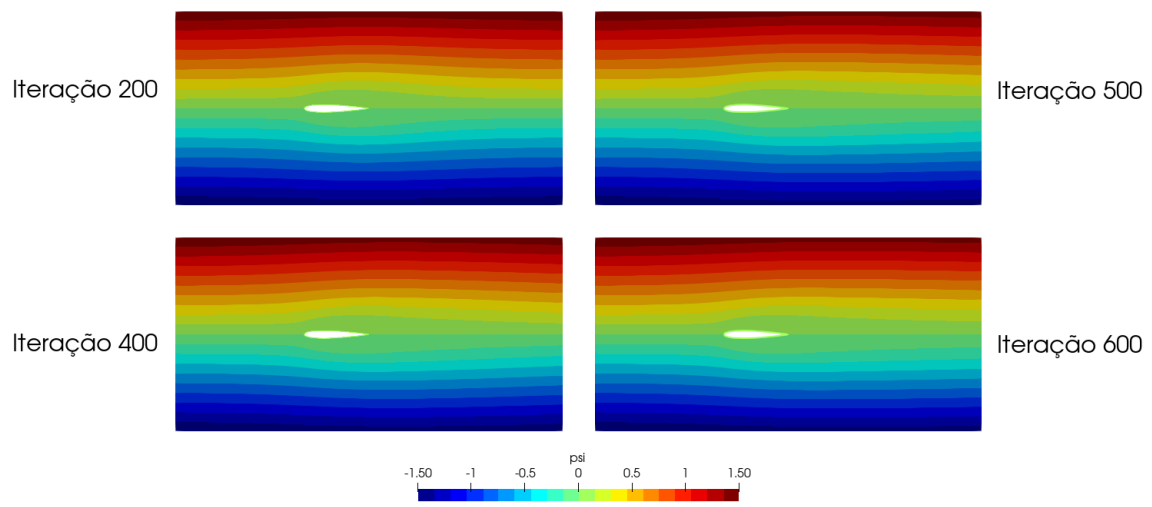


Figura 5.54: Evolução temporal da função corrente ψ

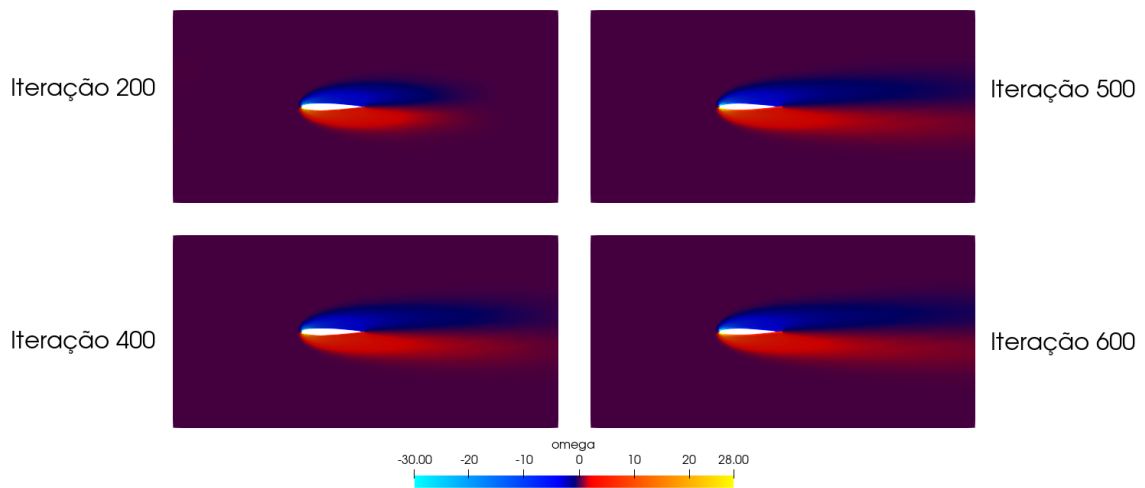


Figura 5.55: Evolução temporal do campo de vorticidade ω

Embora os resultados de ψ , em conformidade com os da velocidade e da vorticidade, não indiquem diretamente o surgimento de zonas de circulação, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.56, com o objetivo de investigar o comportamento dessas linhas.

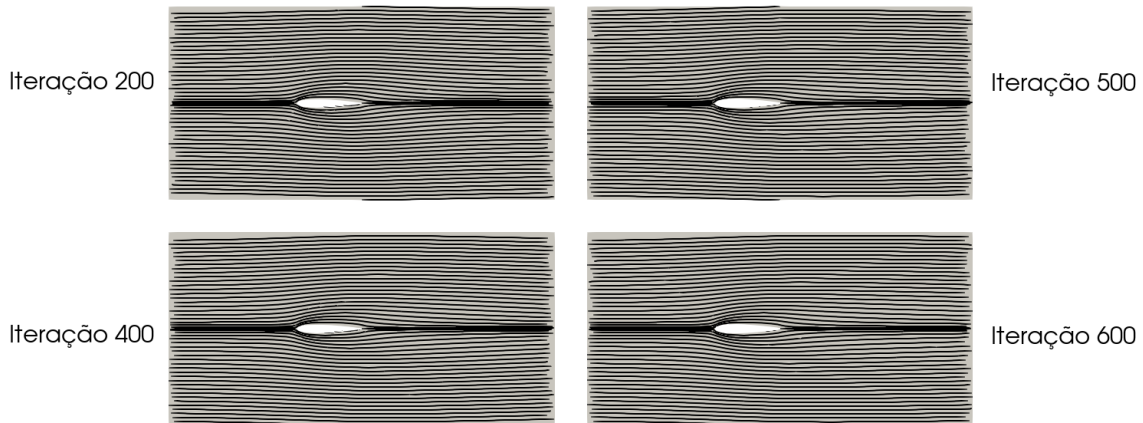


Figura 5.56: Evolução temporal das linhas de corrente

Como esperado, não houve a aparição de vórtices na região posterior desse perfil. Esse é um resultado previsto para perfis de aerofólio, tendo em vista que o aparecimento de zonas de recirculação seria uma barreira para a geração de forças de sustentação, que é o principal objetivo de um aerofólio. Sua presença, dessa forma, representaria uma falha no *design* dessas geometrias. O comportamento apontado sugere concordância com o que é observado na natureza. Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.7.

Variável	Min.	Máx.
u	0.00	1.22
v	-0.34	0.34
$ \nabla U $	0.0	26.87
ω	-33.43	28.57

Tabela 5.7: Avaliação das variáveis para comparação quantitativa

De maneira análoga a comentada nos casos anteriores, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as

esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.57 e 5.58.

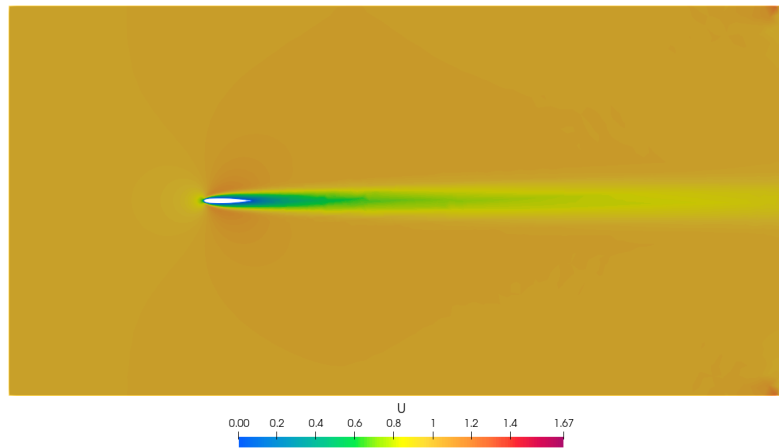


Figura 5.57: Velocidade do escoamento para Reynolds elevado

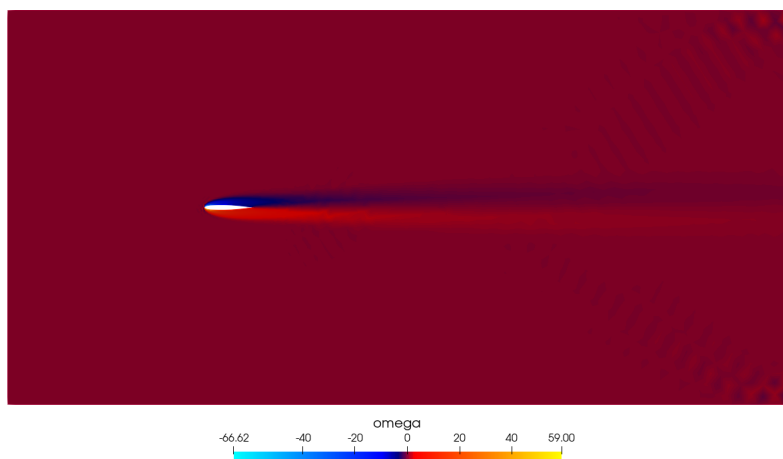


Figura 5.58: Vorticidade do escoamento para Reynolds elevado

Para o caso do NACA 0012 totalmente alinhado com o escoamento, não há a formação aparente de vórtices, como apresentado na Figura 5.57. Como consequência desse fato, nota-se um escoamento similar ao observado para $Re = 100$.

5.3.2 Caso B: NACA 0012 com angulo de ataque 8°

Tomando o domínio da Figura 5.49 como referência, foi construída uma malha computacional com 10016 elementos triangulares, apresentada na Figura 5.59.

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip*

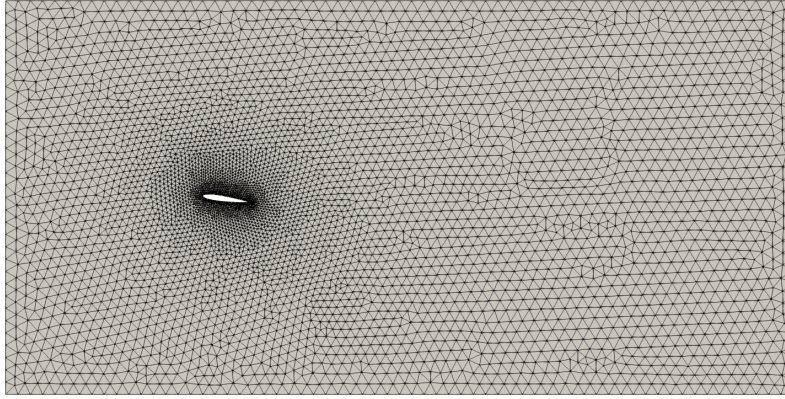


Figura 5.59: Malha computacional do caso

atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.60.

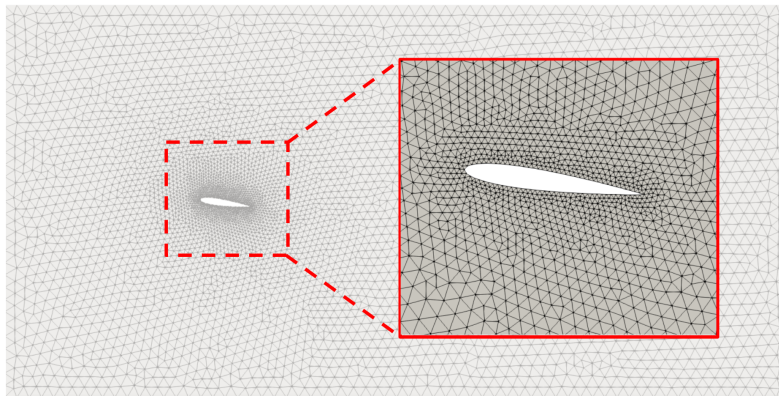


Figura 5.60: Detalhe da malha computacional do caso na região de interesse

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.61.

É interessante notar como a velocidade, principalmente na direção horizontal, varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.62. Os resultados das grandezas podem ser observados nas Figuras 5.63 e 5.64.

Embora os resultados de ψ , em conformidade com os da velocidade e da vorticidade, não indiquem diretamente o surgimento de zonas de circulação, foi gerado

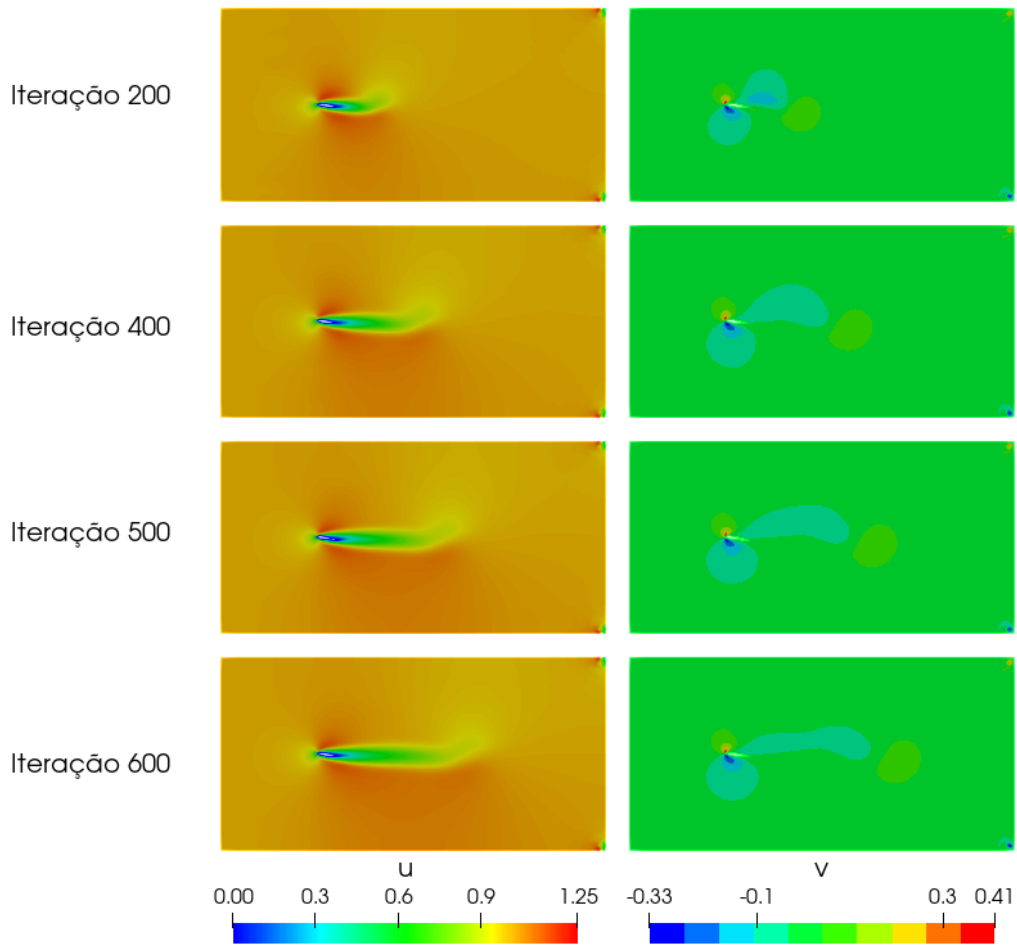


Figura 5.61: Evolução temporal dos campos de velocidade u e v

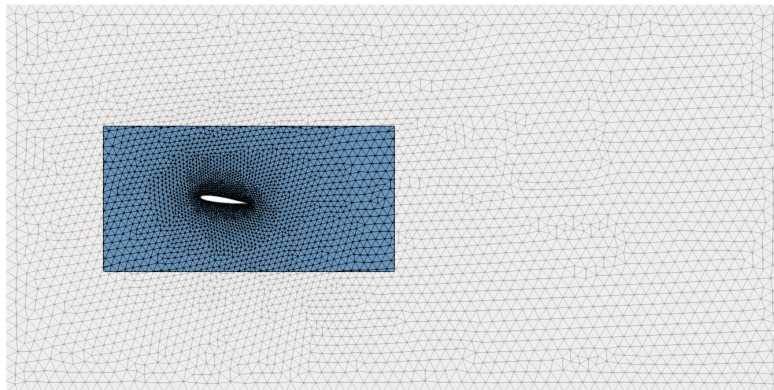


Figura 5.62: Destaque da região onde o corte foi executado

um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.65, com o objetivo de investigar o comportamento dessas linhas.

Como esperado, para uma pequena angulação, não houve a aparição de zonas de recirculação na região posterior desse perfil. De maneira análoga ao caso

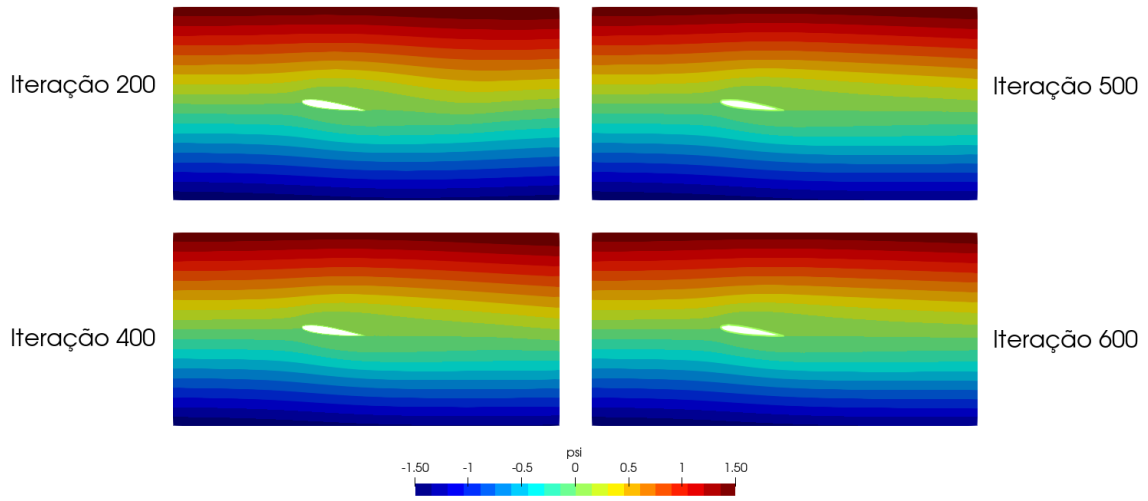


Figura 5.63: Evolução temporal da função corrente ψ

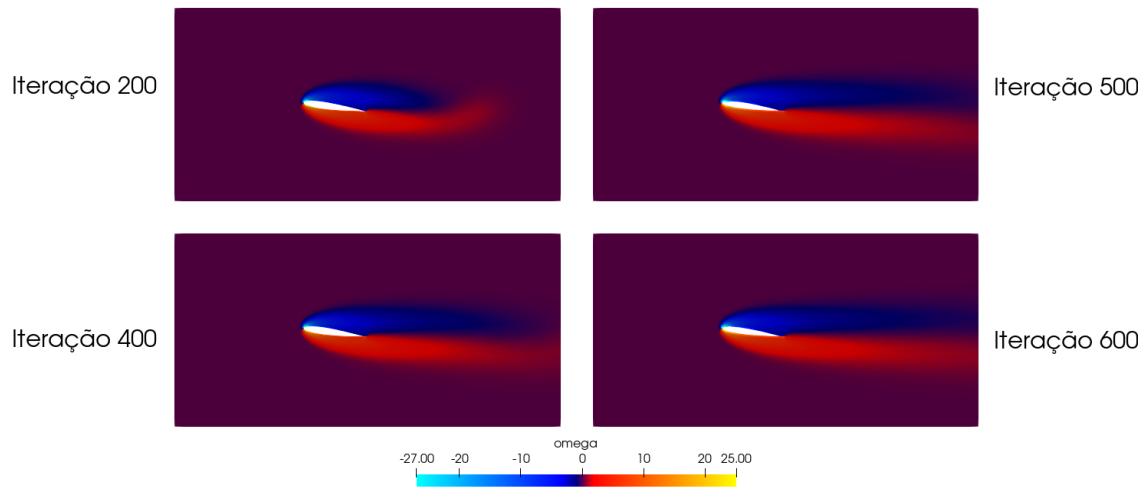


Figura 5.64: Evolução temporal do campo de vorticidade ω

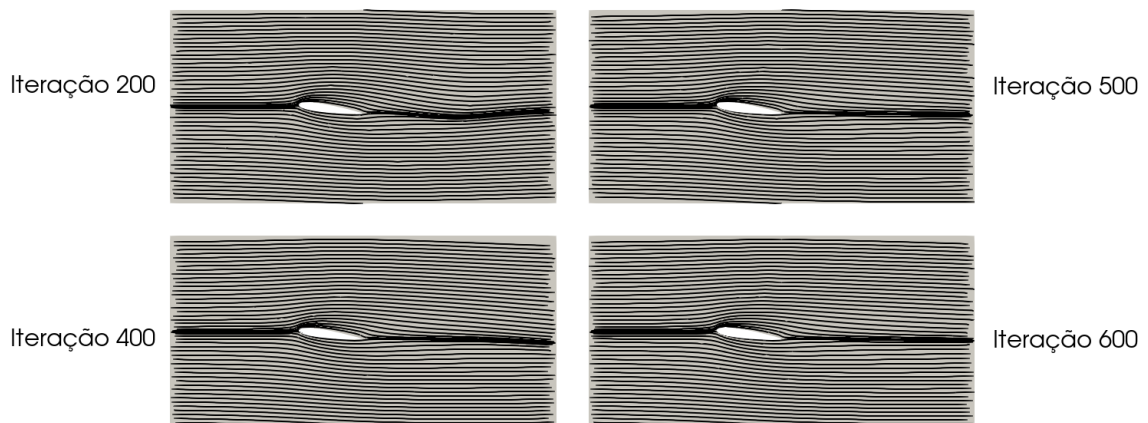


Figura 5.65: Evolução temporal das linhas de corrente

anterior, esse é um resultado previsto para perfis de aerofólio, sustentando a tese de que para pequenas angulações, o aerofólio mantém o bom desempenho, sem a aparição de zonas de recirculação. O comportamento apontado, dessa forma, sugere concordância com o que é observado na natureza. Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.8.

Variável	Min.	Máx.
u	0.00	1.25
v	-0.33	0.41
$ \nabla U $	0.0	29.74
ω	-37.72	21.58

Tabela 5.8: Avaliação das variáveis para comparação quantitativa

De maneira análoga a comentada nos casos anteriores, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.66 e 5.67.

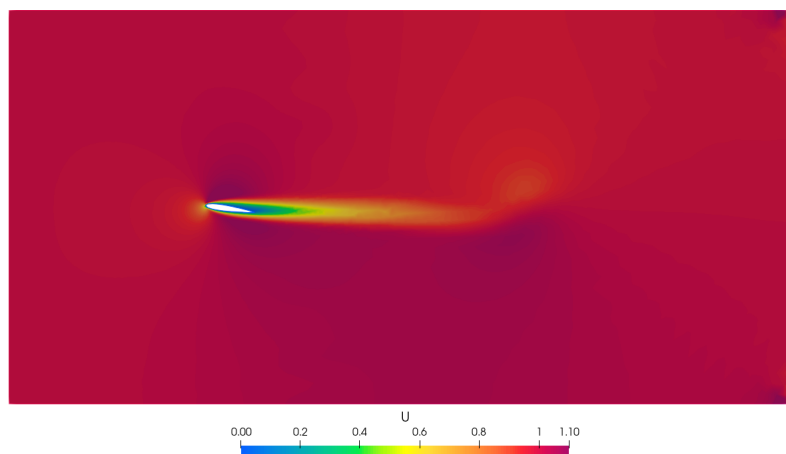


Figura 5.66: Velocidade do escoamento para Reynolds elevado

Para o caso do NACA 0012 com baixo ângulo de ataque (8 graus de inclinação), embora não haja aparecimento da esteira de von Kármán, é possível observar um aumento na velocidade e na vorticidade em uma região superior do escoamento,

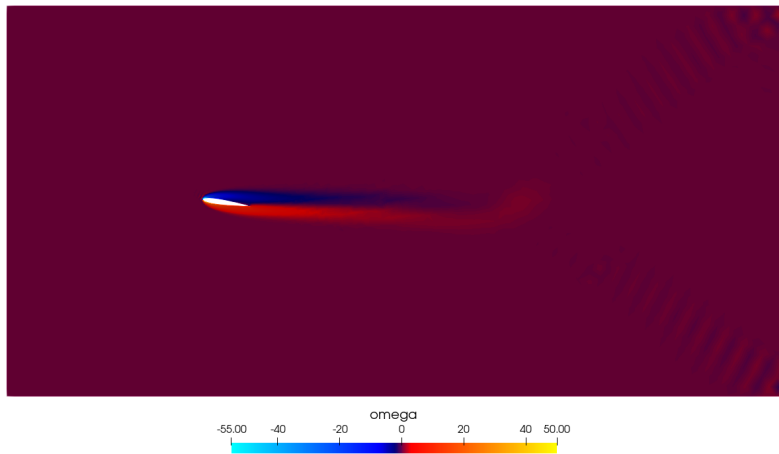


Figura 5.67: Vorticidade do escoamento para Reynolds elevado

como apresentado nas Figuras 5.66 e 5.67. Esse fato é agravado com o aumento do número de Reynolds. A explicação para isso é que a diferença de velocidades na parte superior de aerofólios, sob a condição de Kutta (na qual os fluidos devem deixar o aerofólio suavemente), pressupõe a criação de um vórtice que abandona o bordo de fuga no início do escoamento. Esse mecanismo é responsável pela geração das forças de sustentação e pode ser amplamente observado na literatura (BOTTASSO, 2019) e em aeronaves, o que valida o comportamento observado.

5.3.3 Caso C: NACA 0012 com ângulo de ataque 20°

Tomando o domínio da Figura 5.49 como referência, foi construída uma malha computacional com 9978 elementos triangulares, apresentada na Figura 5.68.

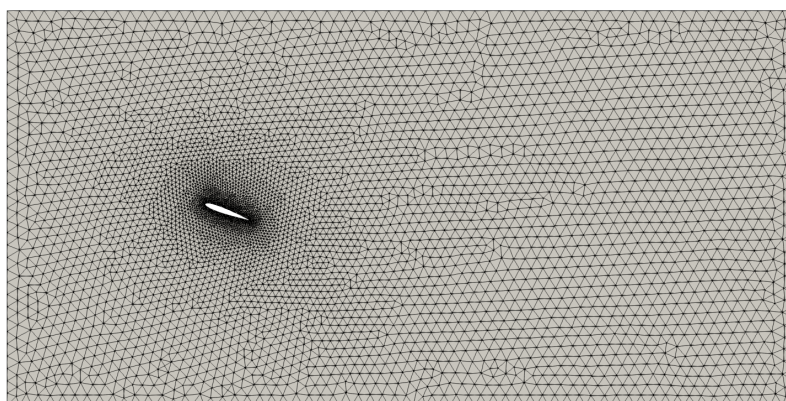


Figura 5.68: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região

com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.69.

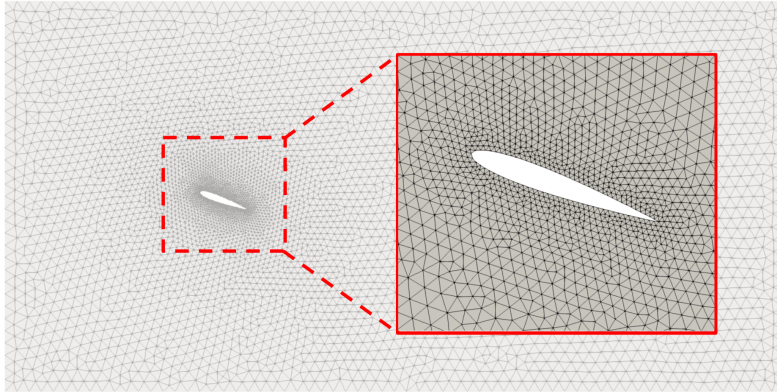


Figura 5.69: Detalhe da malha computacional do caso na região de interesse

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.70.

É interessante notar como a velocidade, principalmente na direção horizontal, varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.71. Os resultados das grandezas podem ser observados nas Figuras 5.72 e 5.73.

Os resultados de ψ , em conformidade com os da velocidade e da vorticidade, sugerem um princípio de aparição de uma zona no bordo de fuga onde pontos de circulação poderiam ser formados. Para avaliar esse comportamento, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.74.

Esse resultado não confirma a aparição de vórtices na região posterior da geometria simulada, porém sugere o início do descolamento do escoamento. Esse é um resultado esperado, tendo em vista o aumento do ângulo de ataque e uma possível entrada na região de estol do aerofólio (BOTTASSO, 2019). Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores,

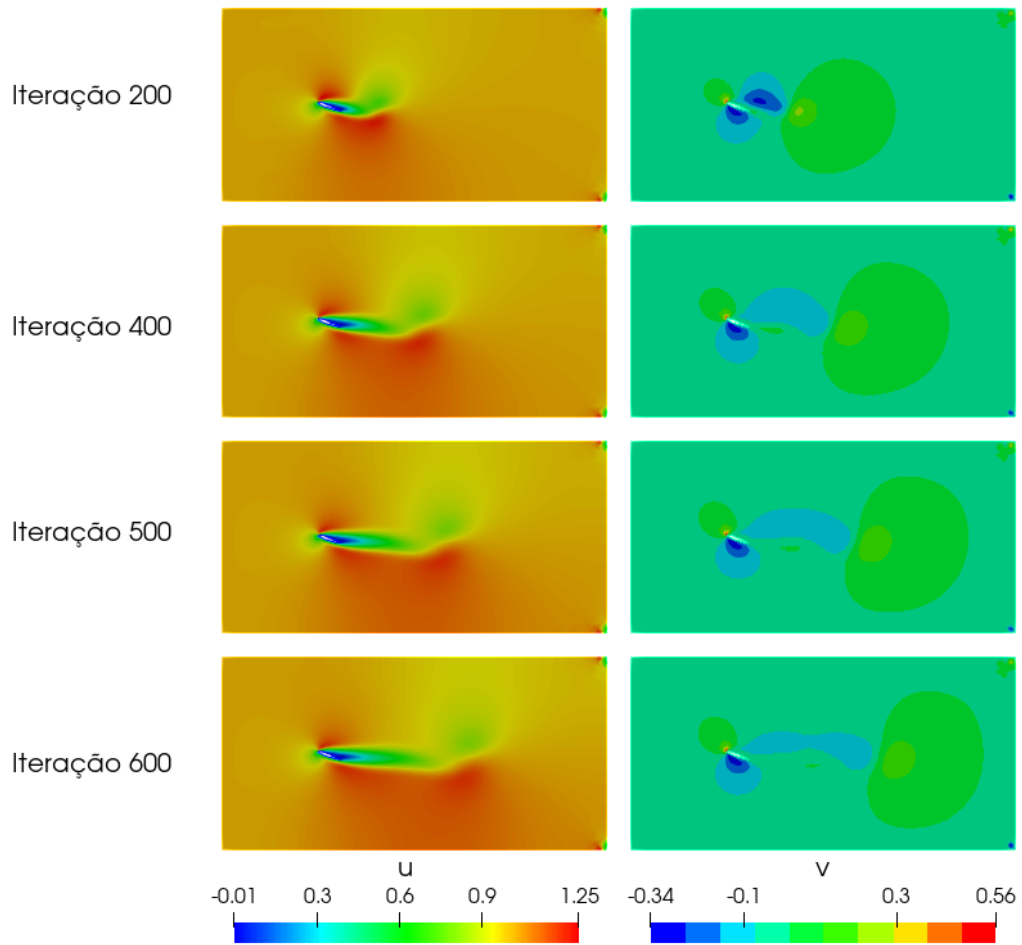


Figura 5.70: Evolução temporal dos campos de velocidade u e v

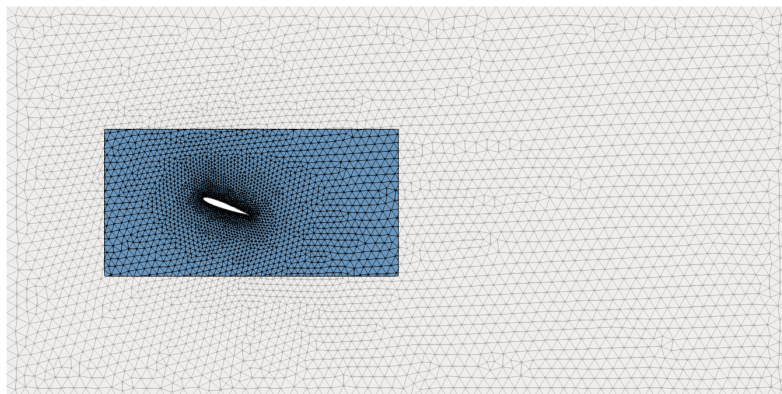


Figura 5.71: Destaque da região onde o corte foi executado

podem ser encontrados na Tabela 5.9.

De maneira análoga a comentada nos casos anteriores, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as

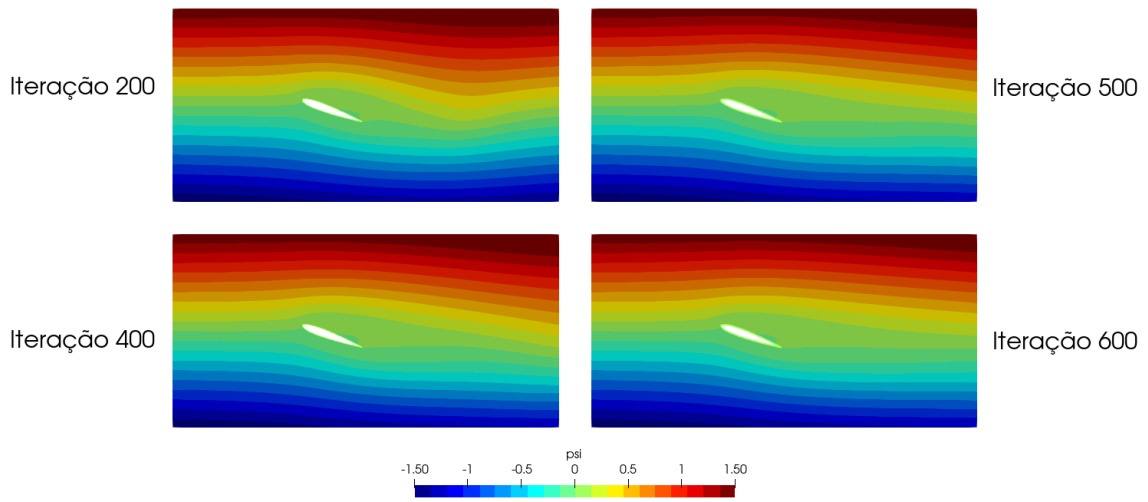


Figura 5.72: Evolução temporal da função corrente ψ

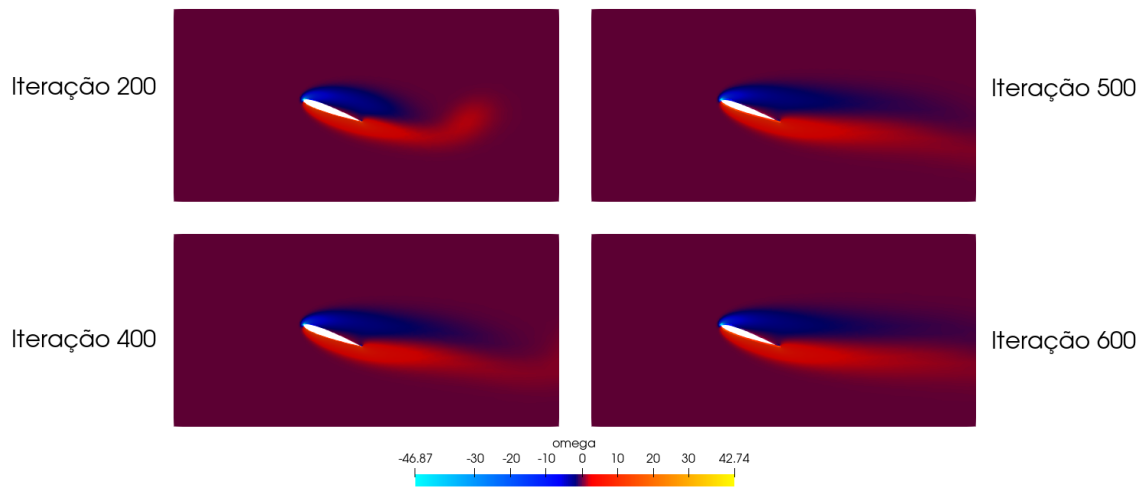


Figura 5.73: Evolução temporal do campo de vorticidade ω

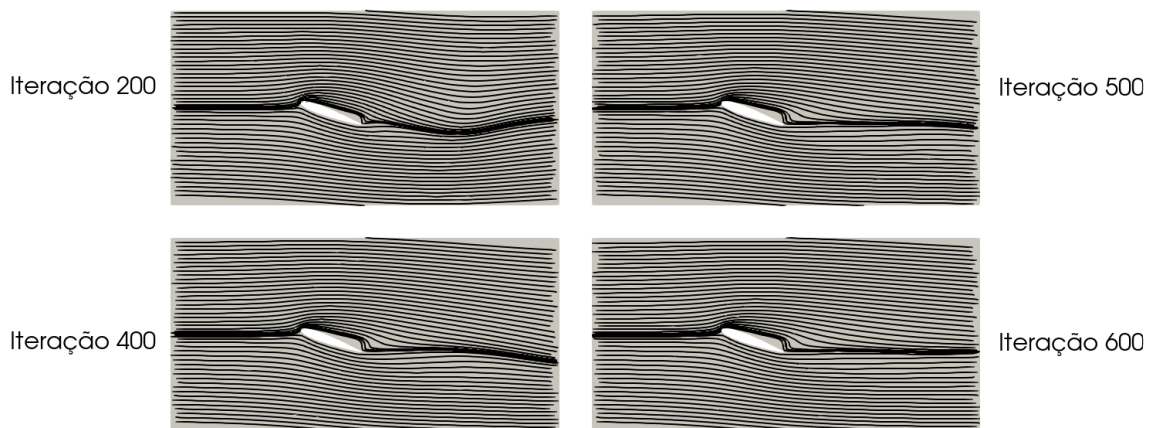


Figura 5.74: Evolução temporal das linhas de corrente

Variável	Min.	Máx.
u	-0.01	1.25
v	-0.34	0.56
$ \nabla U $	0.0	41.61
ω	-43.63	16.28

Tabela 5.9: Avaliação das variáveis para comparação quantitativa

esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.75 e 5.76.

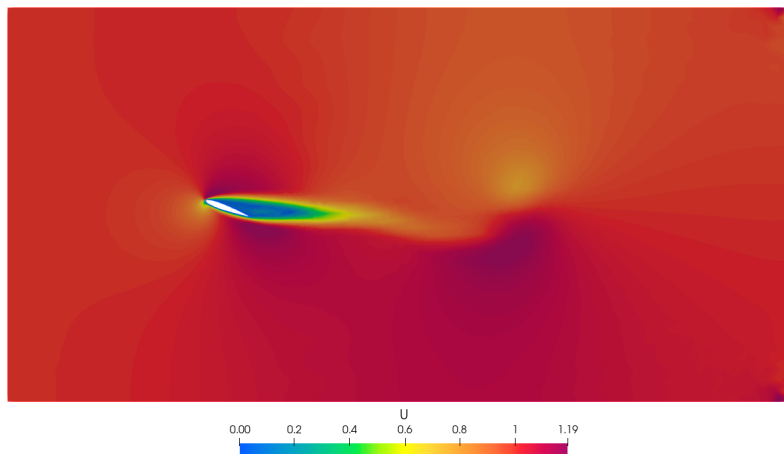


Figura 5.75: Velocidade do escoamento para Reynolds elevado

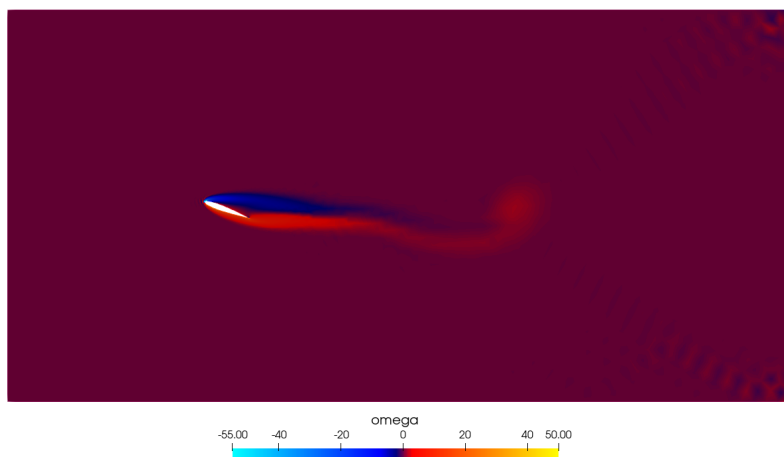


Figura 5.76: Vorticidade do escoamento para Reynolds elevado

Para o caso do NACA 0012 com maior ângulo de ataque (20 graus de inclinação), é possível observar um aumento na velocidade e na vorticidade em uma região supe-

rior do escoamento, como apresentado nas Figuras 5.66 e 5.67. Esse fato é agravado com o aumento do número de Reynolds, tornando claramente observável a formação de um vórtice abandonando o aerofólio, sendo esse o mesmo mecanismo presente na simulação com o baixo ângulo de ataque (8 graus).

5.4 Escoamento ao redor do aerofólio NACA 4415

Para a simulação desses casos, foi discretizado um domínio ao redor de um aerofólio NACA 4415, posicionado na parte anterior e permitindo uma melhor observação do comportamento do fluido na parte posterior. O aerofólio foi posicionado no centro, variando o seu ângulo de ataque para três situações distintas:

- Caso A: NACA 4415 com ângulo de ataque 0°
- Caso B: NACA 4415 com ângulo de ataque 8°
- Caso C: NACA 4415 com ângulo de ataque 20°

Tomando o caso A como referência, o domínio e suas faces de contorno são representadas na Figura 5.77.

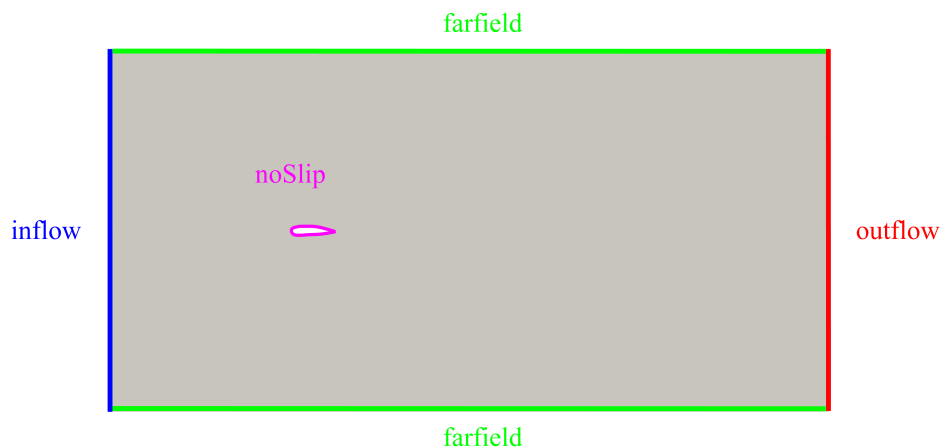


Figura 5.77: Domínio e condições de contorno do caso

5.4.1 Caso A: NACA 4415 com ângulo de ataque 0°

Tomando o domínio da Figura 5.77 como referência, foi construída uma malha computacional com 9824 elementos triangulares, apresentada na Figura 5.78.

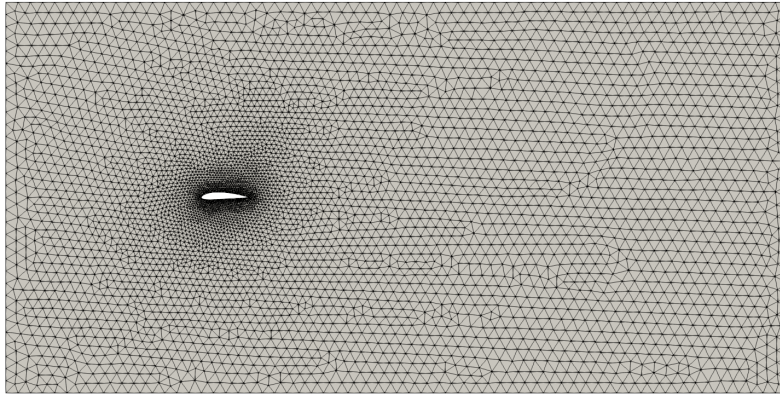


Figura 5.78: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.79.

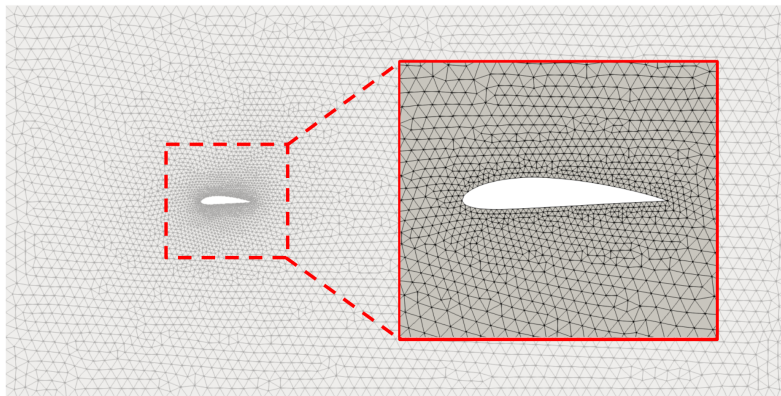


Figura 5.79: Detalhe da malha computacional do caso na região de interesse

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.80.

É interessante notar como a velocidade, principalmente na direção horizontal, varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.81. Os resultados das grandezas podem ser observados nas Figuras 5.82 e 5.83.

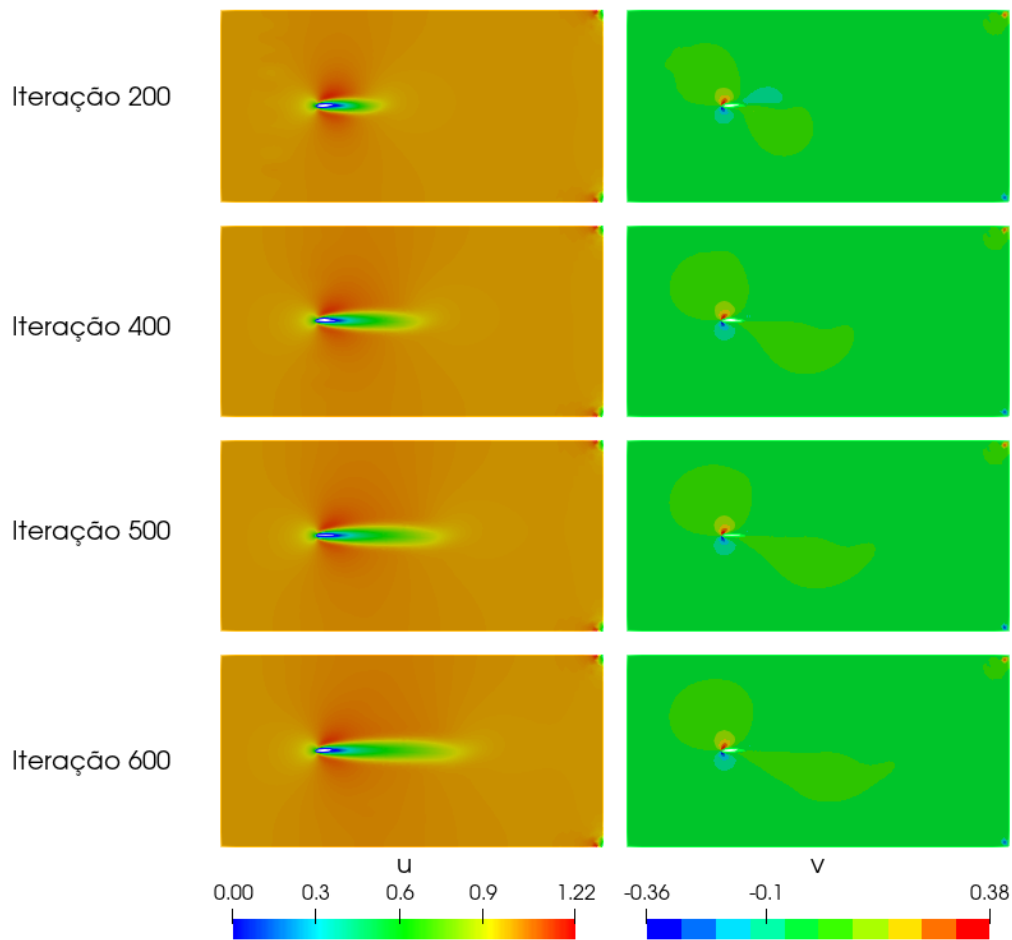


Figura 5.80: Evolução temporal dos campos de velocidade u e v

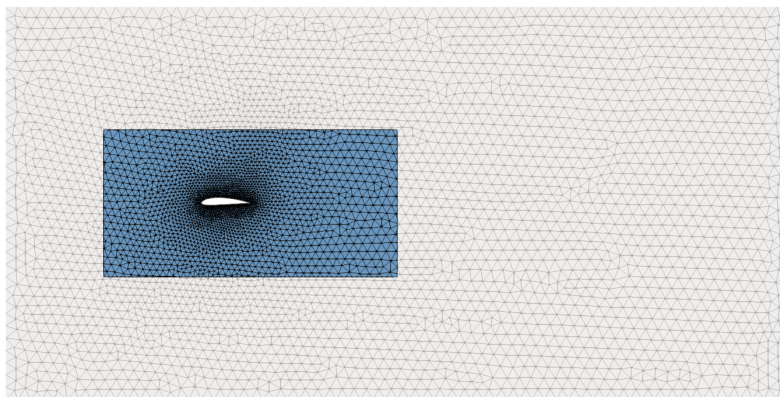


Figura 5.81: Destaque da região onde o corte foi executado

Embora os resultados de ψ , em conformidade com os da velocidade e da vorticidade, não indiquem diretamente o surgimento de zonas de circulação, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.84, com o objetivo de investigar o comportamento dessas linhas.

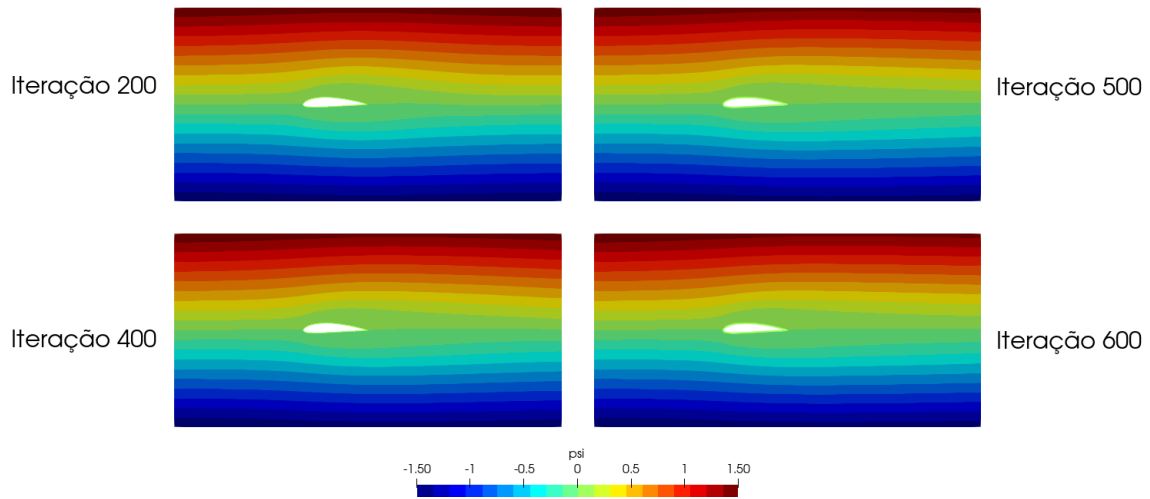


Figura 5.82: Evolução temporal da função corrente ψ

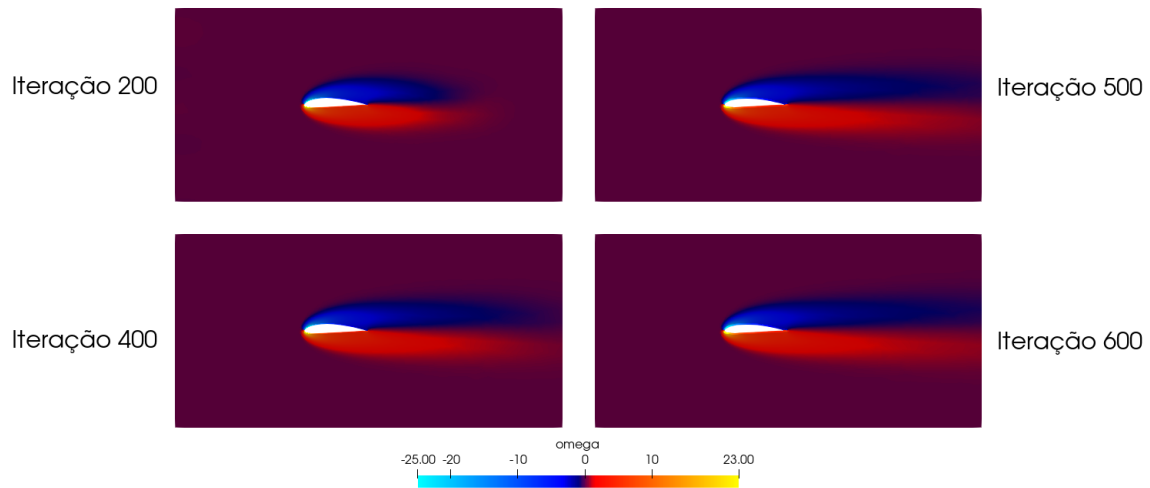


Figura 5.83: Evolução temporal do campo de vorticidade ω

Como ocorreu com o perfil NACA 0012, não houve a aparição de vórtices na região posterior desse perfil. De maneira análoga, esse é um resultado previsto para perfis de aerofólio, tendo em vista que o aparecimento de zonas de recirculação seria uma barreira para a geração de forças de sustentação. O comportamento apontado sugere, portanto, concordância com o que é observado na natureza. Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.10.

De maneira análoga a comentada nos casos anterior, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as esteiras

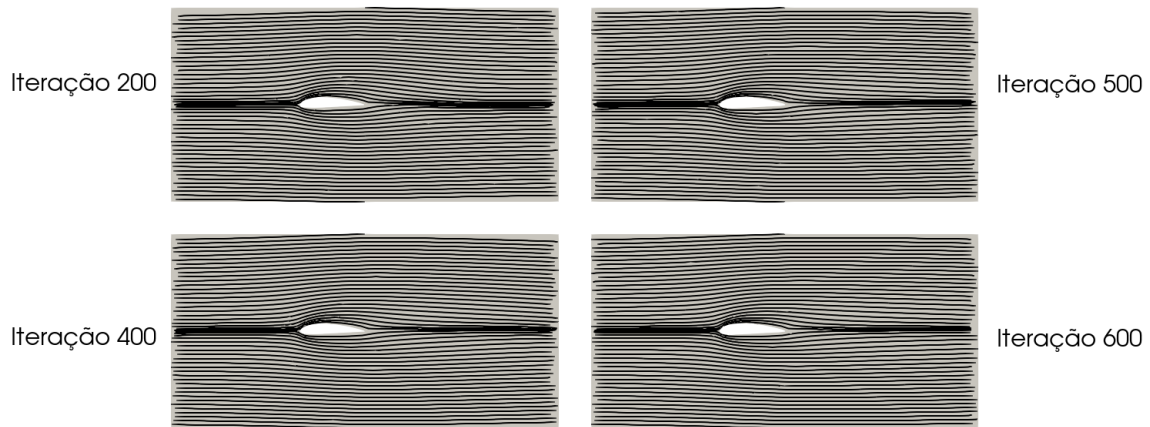


Figura 5.84: Evolução temporal das linhas de corrente

Variável	Min.	Máx.
u	0.00	1.22
v	-0.36	0.38
$ \nabla U $	0.0	29.81
ω	-24.35	35.93

Tabela 5.10: Avaliação das variáveis para comparação quantitativa

de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.85 e 5.86.

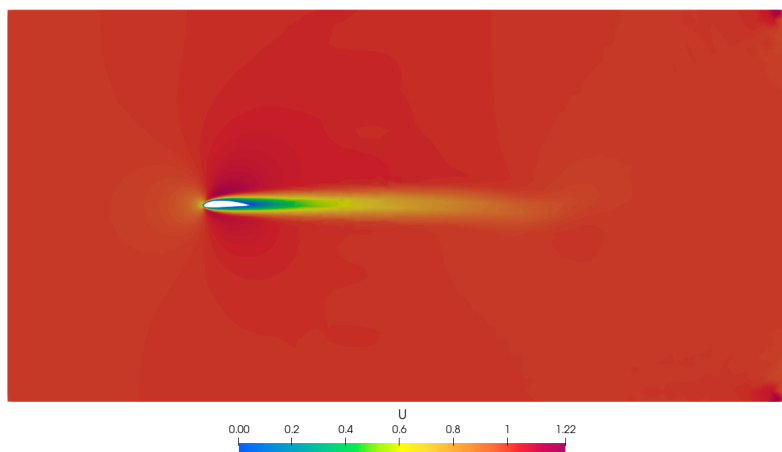


Figura 5.85: Velocidade do escoamento para Reynolds elevado

Para o caso do NACA 4415 totalmente alinhado com o escoamento, não há a formação aparente de vórtices, como apresentado na Figura 5.85. Como consequência desse fato, nota-se um escoamento similar ao observado para $Re = 100$.

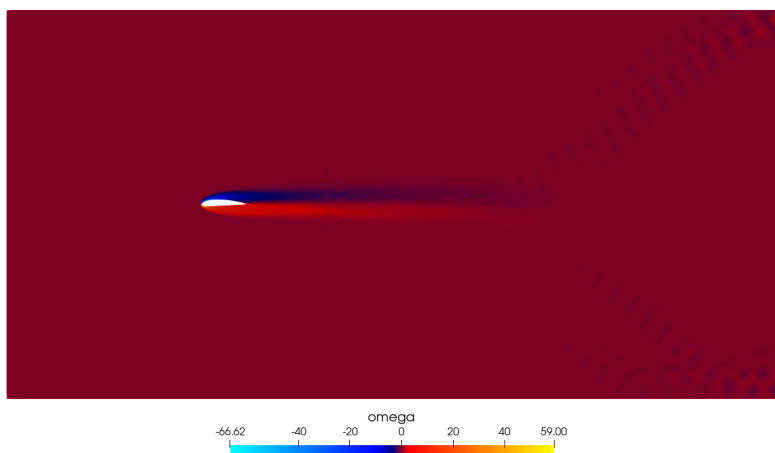


Figura 5.86: Vorticidade do escoamento para Reynolds elevado

5.4.2 Caso B: NACA 4415 com angulo de ataque 8°

Tomando o domínio da Figura 5.77 como referência, foi construída uma malha computacional com 9870 elementos triangulares, apresentada na Figura 5.87.

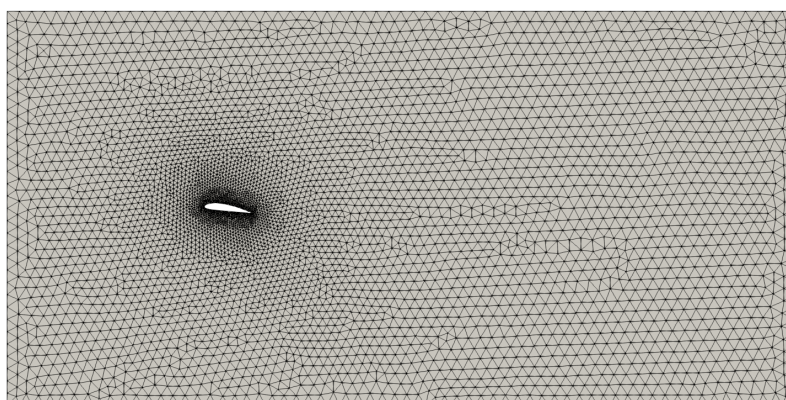


Figura 5.87: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.88.

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.89.

É interessante notar como a velocidade, principalmente na direção horizontal,

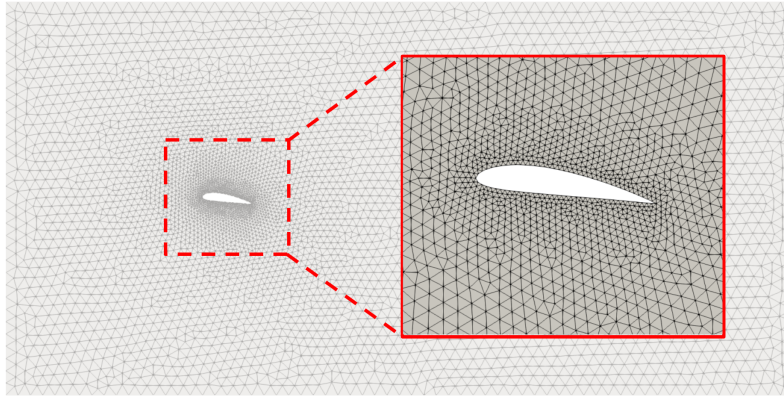


Figura 5.88: Detalhe da malha computacional do caso na região de interesse

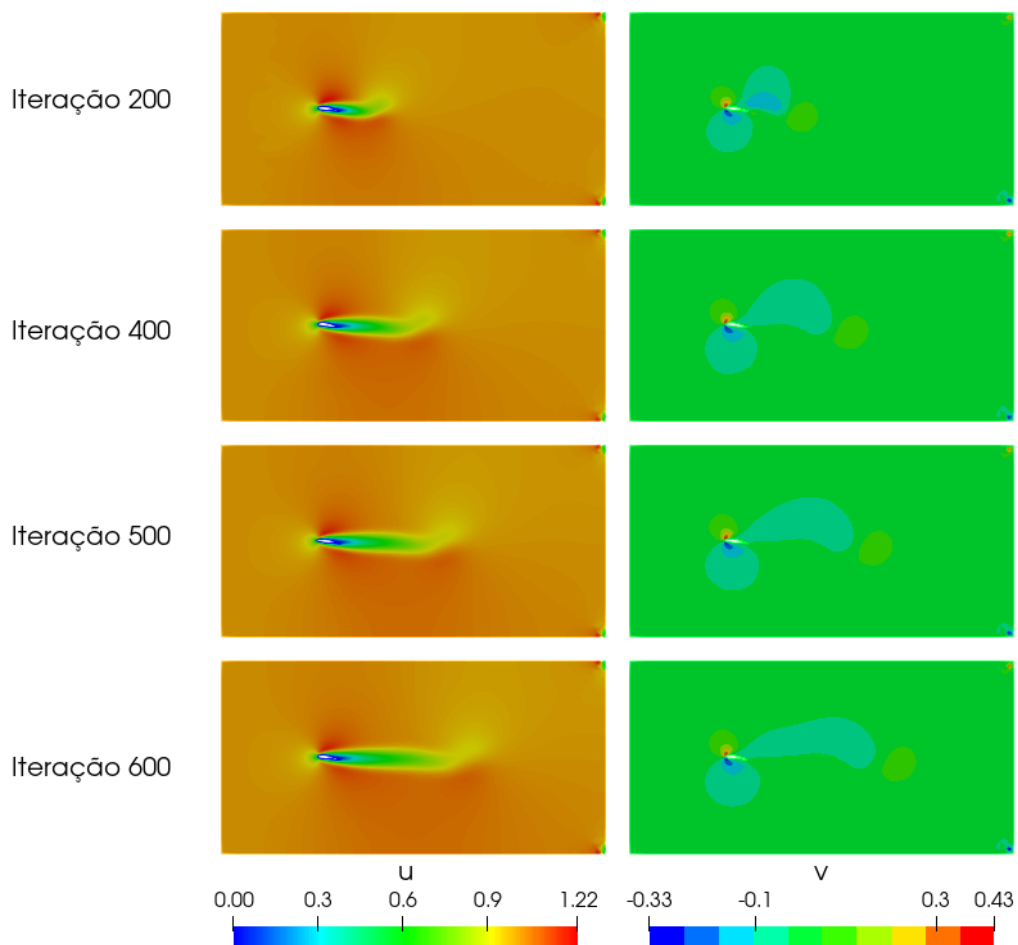


Figura 5.89: Evolução temporal dos campos de velocidade u e v

varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.90. Os resultados das grandezas podem ser observados nas Figuras 5.91 e 5.92.

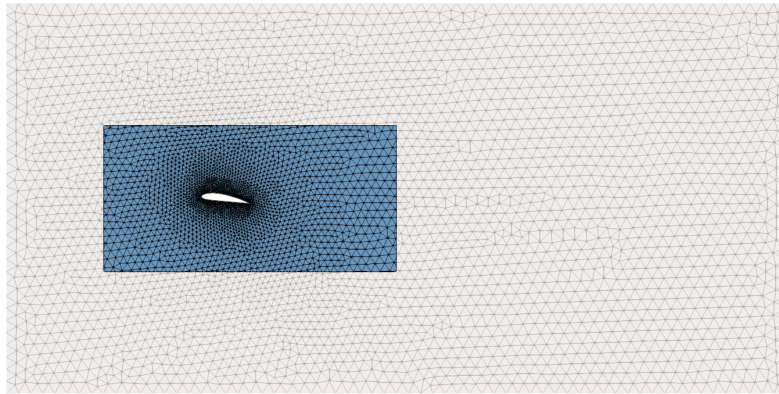


Figura 5.90: Destaque da região onde o corte foi executado

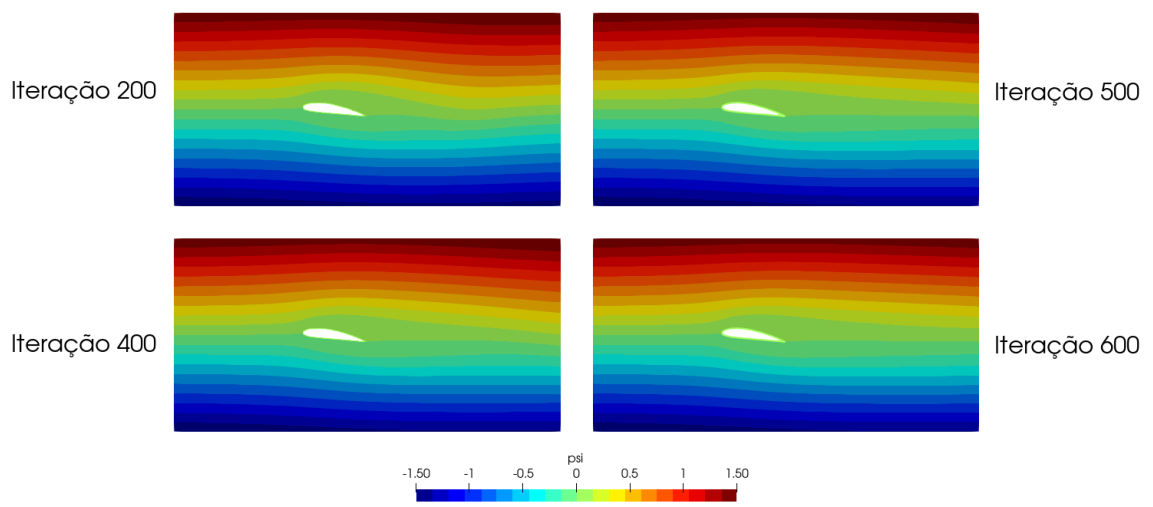


Figura 5.91: Evolução temporal da função corrente ψ

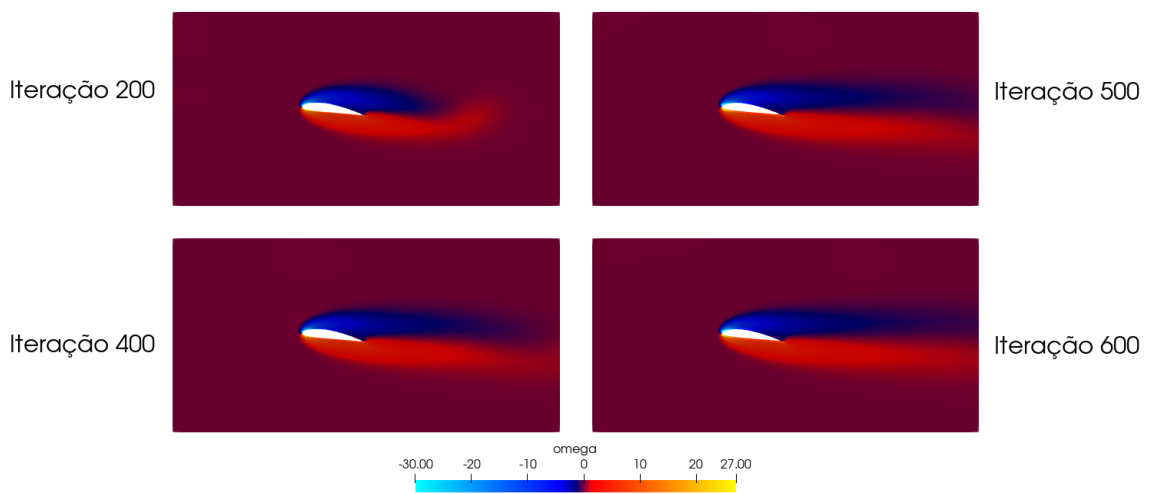


Figura 5.92: Evolução temporal do campo de vorticidade ω

Embora os resultados de ψ , em conformidade com os da velocidade e da vorticidade, não indiquem diretamente o surgimento de zonas de circulação, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.93, com o objetivo de investigar o comportamento dessas linhas.

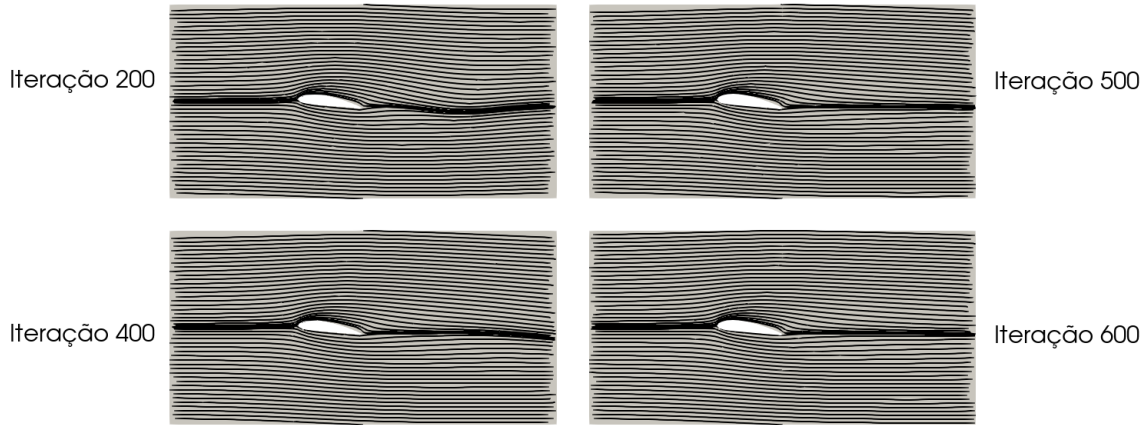


Figura 5.93: Evolução temporal das linhas de corrente

Como esperado, para uma pequena angulação, não houve a aparição de zonas de recirculação na região posterior desse perfil. De maneira análoga ao caso anterior, esse é um resultado previsto para perfis de aerofólio, sustentando a tese de que para pequenas angulações, o aerofólio mantém o bom desempenho, sem a aparição de zonas de recirculação. O comportamento apontado, portanto, sugere concordância com o que é observado na natureza. Alguns resultados numéricos de cada variável, conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.11.

Variável	Min.	Máx.
u	0.00	1.22
v	-0.33	0.43
$ \nabla U $	0.0	27.46
ω	-33.40	24.94

Tabela 5.11: Avaliação das variáveis para comparação quantitativa

De maneira análoga a comentada nos casos anteriores, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as

esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.94 e 5.95.

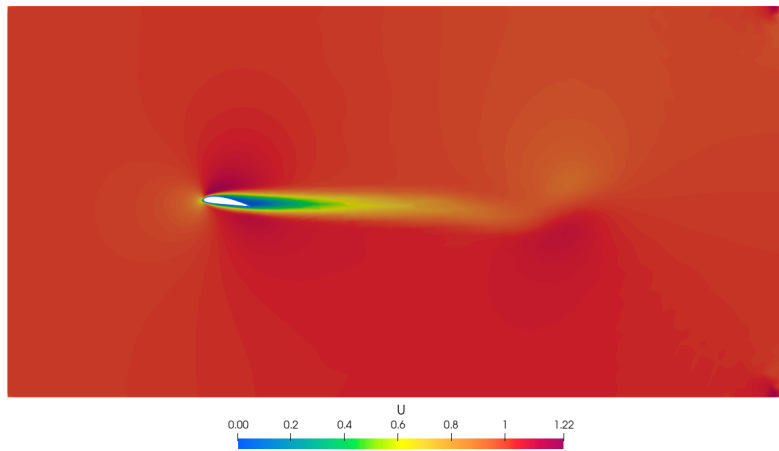


Figura 5.94: Velocidade do escoamento para Reynolds elevado

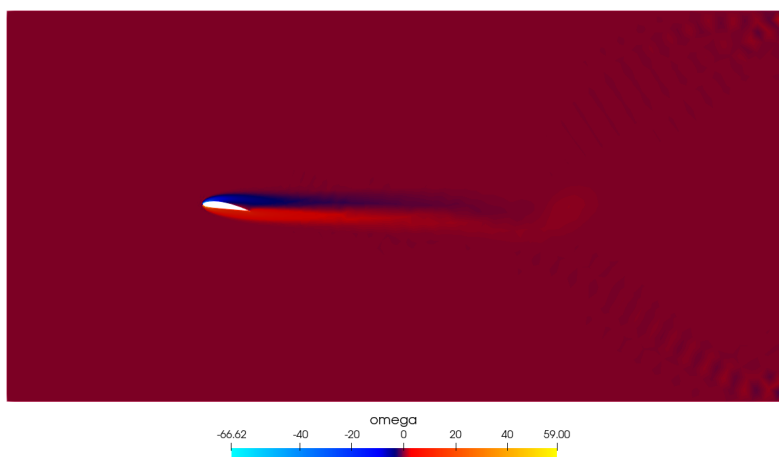


Figura 5.95: Vorticidade do escoamento para Reynolds elevado

Para o caso do NACA 4415 com baixo ângulo de ataque (8 graus de inclinação), embora não haja aparecimento da esteira de von Kármán, é possível observar um aumento na velocidade e na vorticidade em uma região superior do escoamento, como apresentado nas Figuras 5.66 e 5.67. Esse fato é agravado com o aumento do número de Reynolds. A explicação para isso é que a diferença de velocidades na parte superior de aerofólios, sob a condição de Kutta (na qual os fluidos devem deixar o aerofólio suavemente), pressupõe a criação de um vórtice que abandona o bordo de fuga no início do escoamento. Esse mecanismo é responsável pela geração das forças de sustentação e pode ser amplamente observado na literatura (BOTTASSO, 2019) e em aeronaves, o que valida o comportamento observado.

5.4.3 Caso C: NACA 4415 com ângulo de ataque 20°

Tomando o domínio da Figura 5.77 como referência, foi construída uma malha computacional com 9864 elementos triangulares, apresentada na Figura 5.96.

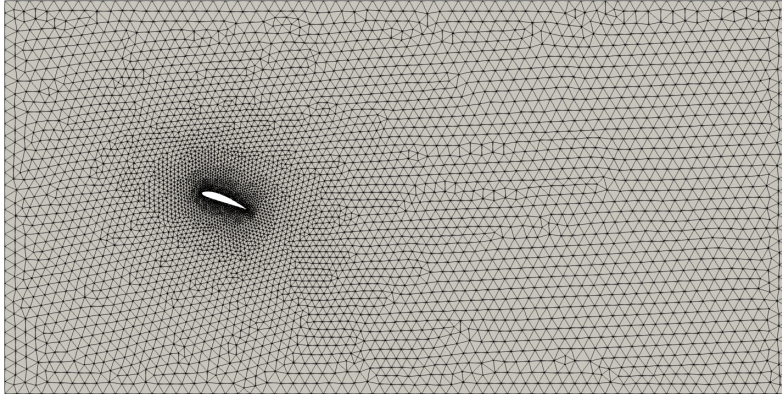


Figura 5.96: Malha computacional do caso

A zona de interesse, localizada ao redor da geometria discretizada, é uma região com grandes gradientes das variáveis calculadas, uma vez que a condição de *noSlip* atribuí velocidade zero àqueles pontos. Com o objetivo de obter uma maior precisão no cálculo das variáveis nessa zona, é interessante que haja maior refino na malha computacional, como apresentado na Figura 5.97.

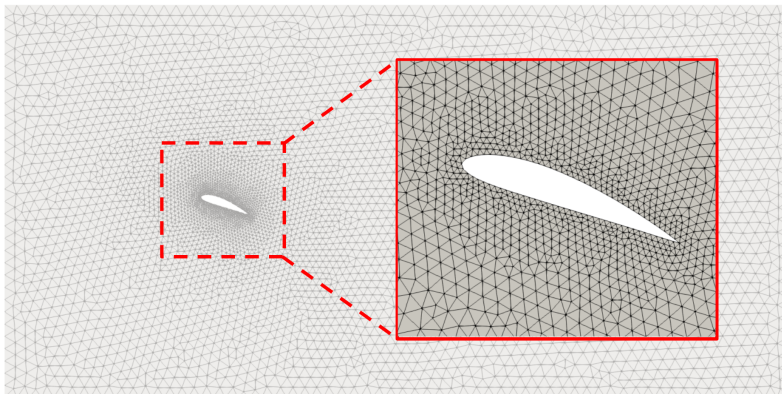


Figura 5.97: Detalhe da malha computacional do caso na região de interesse

Após o término da simulação, os resultados obtidos foram pós-processados e avaliados em diferentes passos de tempo para cada variável de interesse. Os valores de velocidade são apresentados na Figura 5.98.

É interessante notar como a velocidade, principalmente na direção horizontal, varia em todo espaço ao redor do corpo simulado. Contudo, essa característica é

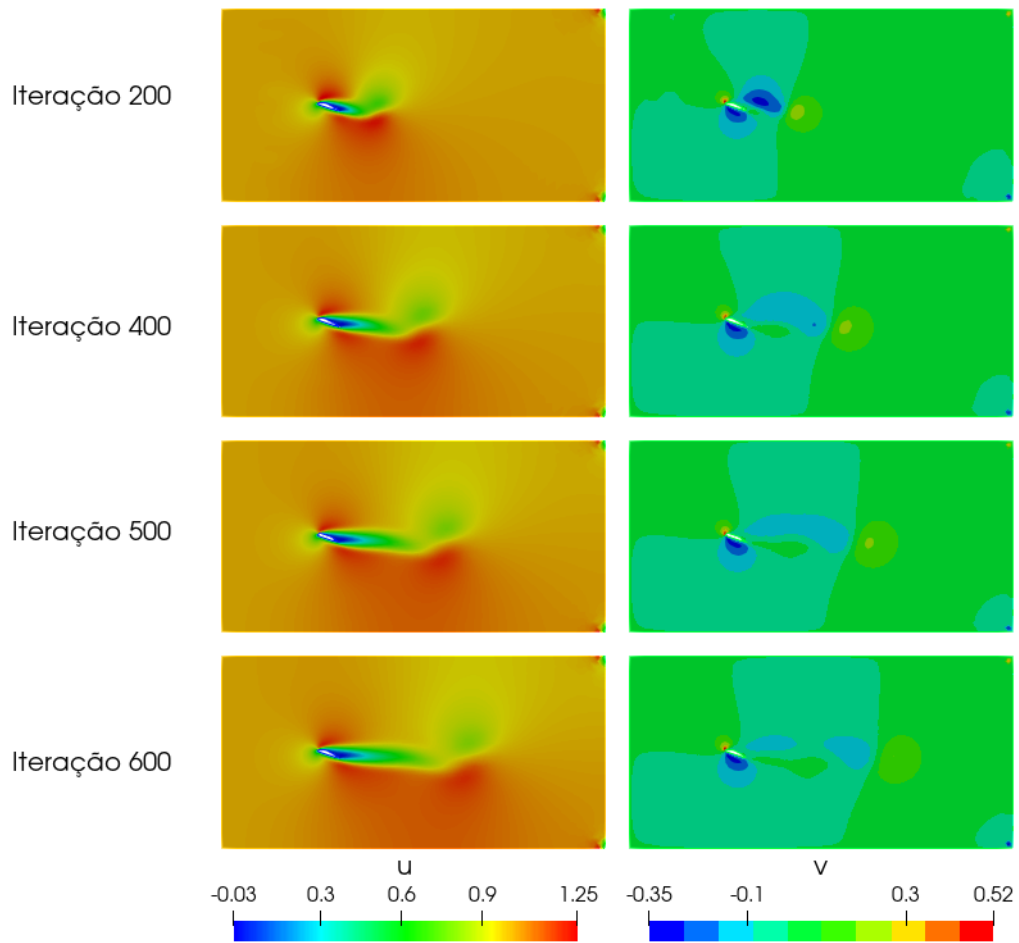


Figura 5.98: Evolução temporal dos campos de velocidade u e v

típica desse campo, não se repetindo para as demais grandezas. Dessa forma, para observá-las com mais detalhes, foi feito um corte ampliado ao redor da geometria, como apresentado na Figura 5.99. Os resultados das grandezas podem ser observados nas Figuras 5.100 e 5.101.

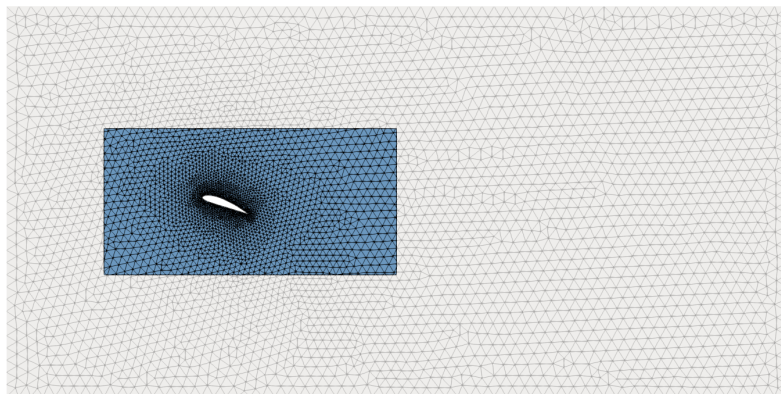


Figura 5.99: Destaque da região onde o corte foi executado

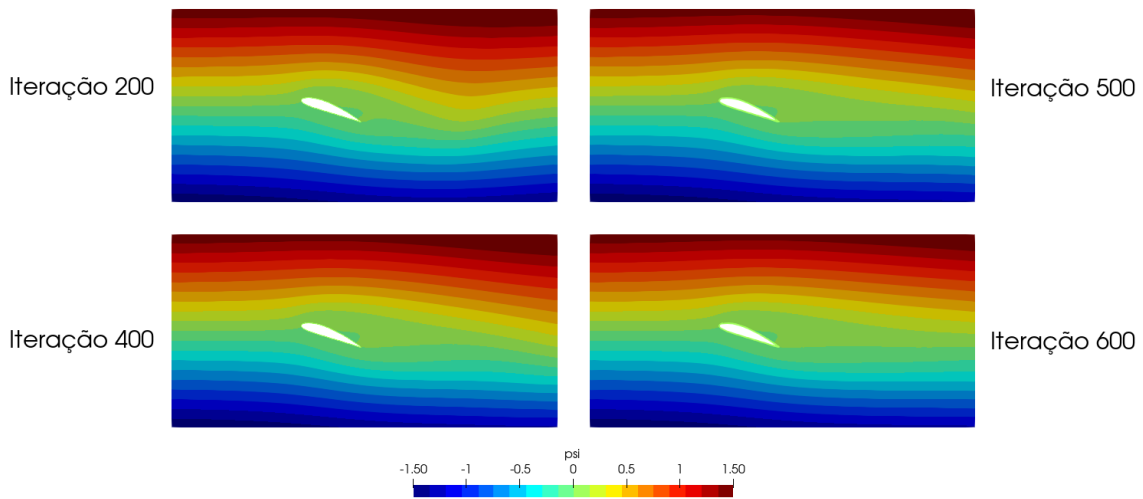


Figura 5.100: Evolução temporal da função corrente ψ

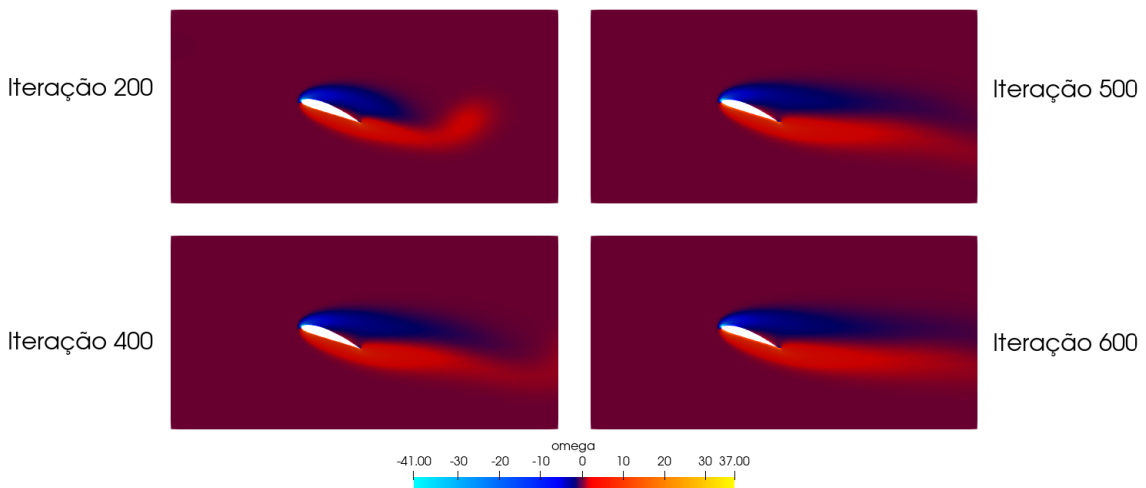


Figura 5.101: Evolução temporal do campo de vorticidade ω

Os resultados de ψ , em conformidade com os da velocidade e da vorticidade, sugerem um princípio de aparição de uma zona no bordo de fuga onde pontos de circulação poderiam ser formados. Para avaliar esse comportamento, foi gerado um pós-processamento que ilustra a evolução de linhas de corrente no domínio, apresentado na Figura 5.102.

Esse resultado confirma a aparição de uma pequena zona de estagnação na região posterior da geometria simulada, no bordo de fuga, além de apresentar o início do descolamento do escoamento. Esse é um resultado esperado, tendo em vista o aumento do ângulo de ataque e uma possível entrada na região de estol do aerofólio (BOTTASSO, 2019). Alguns resultados numéricos de cada variável,

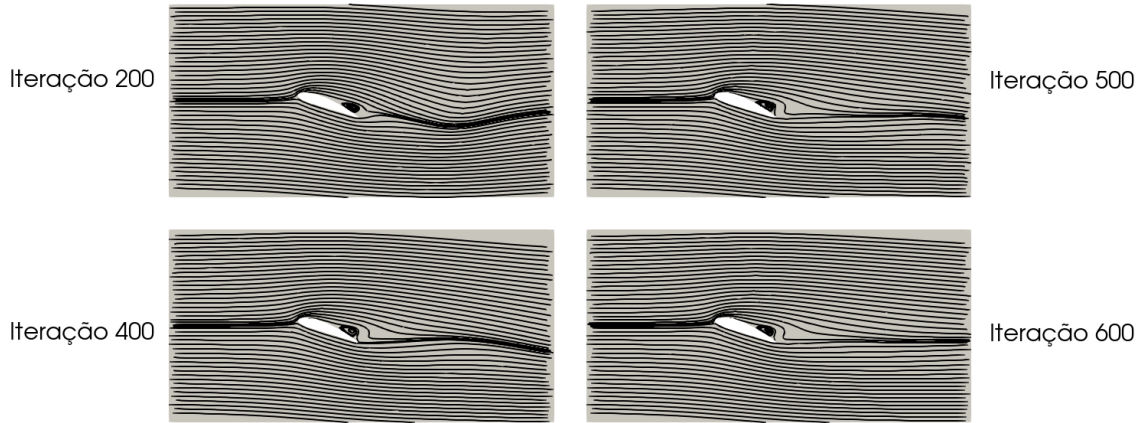


Figura 5.102: Evolução temporal das linhas de corrente

conforme comentado nos casos anteriores, podem ser encontrados na Tabela 5.12.

Variável	Min.	Máx.
u	-0.03	1.25
v	-0.35	0.52
$ \nabla U $	0.0	33.99
ω	-39.29	17.14

Tabela 5.12: Avaliação das variáveis para comparação quantitativa

De maneira análoga a comentada nos casos anteriores, foi feita uma nova simulação com elevado número de Reynolds ($Re = 300$) com o intuito de observar as esteiras de von Kármán. Os resultados obtidos podem ser vistos nas Figuras 5.103 e 5.104.

Para o caso do NACA 4415 com maior ângulo de ataque (20 graus de inclinação), é possível observar um aumento na velocidade e na vorticidade em uma região superior do escoamento, como apresentado nas Figuras 5.66 e 5.67. Esse fato é agravado com o aumento do número de Reynolds, tornando claramente observável a formação de um vórtice abandonando o aerofólio, sendo esse o mesmo mecanismo presente na simulação com o baixo ângulo de ataque (8 graus).

Além disso, ao contrário do que ocorreu no perfil NACA 0012, é possível observar certa oscilação no escoamento, como apresentado na Figura 5.75. Essa oscilação é uma consequência direta do aumento do número de Reynolds, sugerindo um com-

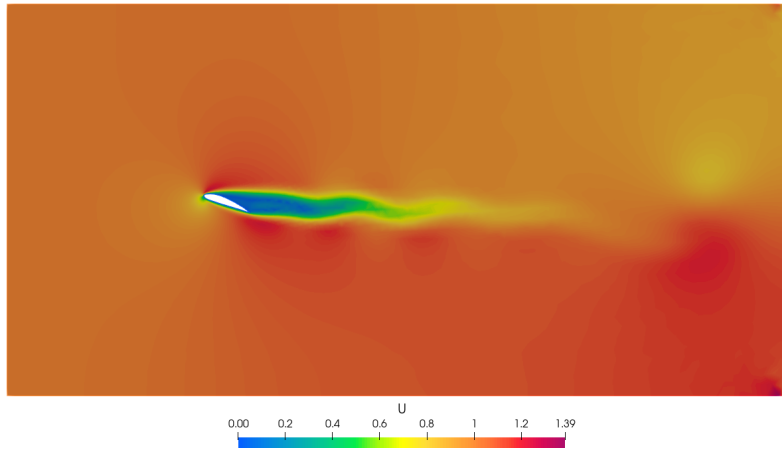


Figura 5.103: Velocidade do escoamento para Reynolds elevado

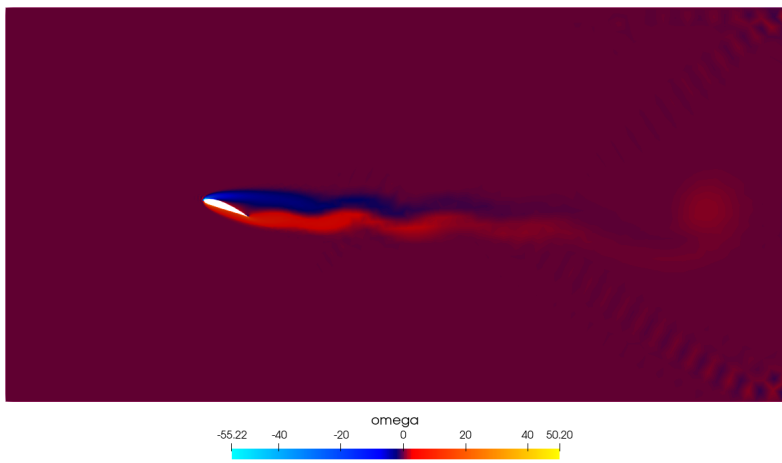


Figura 5.104: Vorticidade do escoamento para Reynolds elevado

portamento semelhante ao observado nas simulações anteriores, no qual as esteiras de von Kármán puderam ser observadas.

5.5 Discussão dos resultados

Para viabilizar uma visualização mais integrada dos resultados, foi gerada a Tabela 5.16, que unifica as principais variáveis de cada simulação.

Variáveis (Máx.)	Quad.	Triângulo				NACA 0012			NACA 4415		
		A	B	C	D	A	B	C	A	B	C
$ \nabla U $	39.13	46.70	32.70	20.72	15.82	26.87	29.74	41.61	29.81	27.46	33.99
$ \omega $	49.72	44.24	41.34	19.87	18.00	33.43	37.72	43.63	35.93	33.40	39.29

Tabela 5.13: Avaliação comparativa das variáveis em cada caso

- Influência de parede normal ao escoamento na parte anterior da geometria

Variáveis (Máx.)	Quad.	Triângulo				NACA 0012			NACA 4415		
		A	B	C	D	A	B	C	A	B	C
$ \nabla U $	39.13	46.70	32.70	20.72	15.82	26.87	29.74	41.61	29.81	27.46	33.99
$ \omega $	49.72	44.24	41.34	19.87	18.00	33.43	37.72	43.63	35.93	33.40	39.29

Tabela 5.14: Avaliação comparativa com destaque em laranja

Os casos que se encaixam nessa modalidade são: Quadrado, Triângulo A e Triângulo B. Como mostram as células da Tabela 5.14 iluminadas em laranja, fica perceptível que esses casos apresentaram valores para vorticidade elevados, em especial quando comparados com os casos onde há uma transição mais suave entre escoamento e bordo de ataque (Triângulos C e D). Esse resultado sugere que um escoamento que encontra uma parede normal resulte em grandes valores para a vorticidade, em especial quando os cantos vivos apresentam ângulos retos (Quadrado).

- Influência de parede normal ao escoamento na parte posterior da geometria

Os casos que se encaixam nessa modalidade são: Quadrado, Triângulo C e Triângulo D. Embora quantitativamente não se possa atribuir uma diferença considerável, resultados qualitativos apresentaram destaque na formação de vórtices e,

em especial, na formação da esteira de von Kármán para maiores números de Reynolds. Os resultados sugerem que geometrias com essa característica apresentam um desprendimento de vórtices mais bem definidos, vide Figuras 5.48 e 5.39.

- Influência de geometria do triângulo

Variáveis (Máx.)	Quad.	Triângulo				NACA 0012			NACA 4415		
		A	B	C	D	A	B	C	A	B	C
$ \nabla U $	39.13	46.70	32.70	20.72	15.82	26.87	29.74	41.61	29.81	27.46	33.99
$ \omega $	49.72	44.24	41.34	19.87	18.00	33.43	37.72	43.63	35.93	33.40	39.29

Tabela 5.15: Avaliação comparativa com destaque em ciano

Como mostram as células da Tabela 5.15 iluminadas em ciano, pode-se observar que os triângulos com uma transição mais abrupta (Triângulos A e C) apresentam valores maiores tanto do gradiente de velocidade quanto da vorticidade quando comparados com os triângulos com transição mais suave (Triângulos B e D). Esse resultado sugere que transições mais suaves auxiliam na redução da variação brusca de velocidades e de circulação no escoamento ao redor de geometrias.

- Influência de geometria do aerofólio e dos ângulos de ataque

Alguns comentários gerais:

- Ambos os aerofólios apresentaram, em geral, linhas de corrente sem pontos de estagnação para $Re = 100$, provando sua melhor eficiência aerodinâmica em comparação com as demais geometrias
- Os aerofólios alteram sua superioridade quantitativa a depender da situação, o que sustenta a tese de que pequenas mudanças na geometria de corpos similares pode alterar seu comportamento
- NACA 0012 gerou menos oscilações que o NACA 4415 em maior Reynolds, o que sugere que esse aerofólio possua um perfil aerodinâmico mais estável nesse cenário.

Além disso, também é interessante observar que, ao contrário do que acontece nas demais geometrias, o valor mínimo da vorticidade supera consideravelmente o valor máximo. Os valores negativos de vorticidade foram iluminados de azul na Tabela 5.15 para destaque:

Variáveis (Máx.)	Quad.	Triângulo				NACA 0012			NACA 4415		
		A	B	C	D	A	B	C	A	B	C
$ \nabla U $	39.13	46.70	32.70	20.72	15.82	26.87	29.74	41.61	29.81	27.46	33.99
$ \omega $	49.72	44.24	41.34	19.87	18.00	33.43	37.72	43.63	35.93	33.40	39.29

Tabela 5.16: Avaliação comparativa com destaque em azul

Como é possível observar nas Tabelas 5.7 - 5.12, esse efeito se agrava com o aumento do ângulo de ataque. A provável justificativa para esse fenômeno é que o aumento do ângulo de ataque faz com que a diferença entre as velocidades (e, conseqüentemente, pressões) na região superior e inferior dos aerofólios aumente (até certo limite, denominado estol). Essa diferença de pressões é o que gera a força de sustentação, principal objetivo no uso de aerofólios, de forma que esse é um resultado esperado.

Muito embora o aumento do módulo da vorticidade seja um indício do aumento da força de sustentação, seria necessário avaliar a distribuição global dessa variável ao redor da geometria e aplicar outros modelos matemático para concluir com precisão a relação entre a vorticidade e a força de sustentação, fugindo do escopo do atual trabalho.

Capítulo 6

Conclusões

Nesta monografia foi apresentada a solução computacional de um escoamento livre ao redor de diversas geometrias utilizando o Método de Elementos Finitos. A Formulação Corrente-Vorticidade foi utilizada como modelo físico para a Equação de Navier-Stokes. A discretização espacial contou com o Método de Resíduos Ponderados utilizando o Método de Galerkin com interpolação de primeira ordem, enquanto a discretização temporal utilizou um esquema de diferenças finitas progressivo de primeira ordem. O código desenvolvido encontra-se disponível no Apêndice A deste trabalho e poderá ser utilizado livremente por toda a comunidade científica.

Para a validação do código, foram utilizados dois casos com soluções conhecidas: o Escoamento entre Placas Paralelas (*Poiseuille*) e o Escoamento em Cavidade (*Lid-Driven*). As respostas obtidas pelo código e as presentes na literatura foram comparadas e foi obtida concordância satisfatória.

Em todas as simulações, o código permitiu observar o comportamento esperado de um escoamento ao redor das geometrias. A influência de paredes, de geometrias com melhor aerodinâmica e seu alinhamento com o escoamento pode ser facilmente observada tanto qualitativamente quanto quantitativamente, sugerindo concordância entre os resultados encontrados e os fenômenos físicos representados.

Para baixos número de Reynolds, o código foi capaz de identificar zonas de recirculação de fluidos na zona posterior às geometrias. Em simulações para um número de Reynolds maior, foi possível observar a formação de esteiras de von Kármán. Nos aerofólios, foi possível detectar a presença do vórtice inicial abandonando o bordo de fuga, concordando com a condição de Kutta. Esses resultados confirmam a validade

do código e sua capacidade de identificar fenômenos físicos de maior complexidade.

Paredes normais ao escoamento na região anterior das geometrias geraram elevados valores de vorticidade. Já as paredes normais na região posterior influenciaram na forma da esteira de von Kármán. A geometria dos corpos triangulares demonstrou que os corpos mais alongados e, portanto, com uma transição mais suave, apresentaram variações menos bruscas no escoamento. Por fim, testes com aerofólios mostraram que pequenas diferenças no formato do corpo influencia de maneira global seu comportamento. O perfil NACA 0012 sugeriu maior estabilidade aerodinâmica, apresentando menor oscilação dos campos de velocidade e vorticidade a maior número de Reynolds. Com relação à força de sustentação, uma conclusão concreta demandaria a utilização de outros modelos matemáticos, fugindo do escopo deste trabalho.

Para realização de estudos futuros, sugere-se uma atualização do modelo matemático e do código numérico para calcular não só os campos, mas as forças atuantes nos objetos simulados. Tal ação tornaria possível calcular variáveis aerodinâmicas de grande valia, como as forças de arrasto e sustentação. Além disso, sugere-se a implementação de equações termodinâmicas, permitindo a utilização do código para estudos em trocadores de calor, refrigeração, e demais objetos de estudo térmico.

Por fim, simulações numéricas sofrem influência de oscilações numéricas. Para trabalhos futuros, seria interessante obter redução das oscilações espúrias através de métodos de estabilização, permitindo simulações com grandes números de Reynolds e maiores passos de tempo.

Referências Bibliográficas

“SU2 Code Webpage”. <https://su2code.github.io/>. Acessado em: 01/06/2022.

ANJOS, G. R. *Computação Científica para Engenheiros*. PEM/COPPE/UFRJ, 2021.

LEWIS, R. W.; NITHIARASU, P. S. K. N. *Fundamentals of the Finite Element Method for Heat and Fluid Flow*. John Wiley & Sons Ltd, 2004.

PONTES, J.; MANGIAVACCHI, N. *Fenômenos de Transferência com Aplicações a Ciências Físicas e a Engenharia*. Apostila do Curso-UFRJ, 2009.

CARLOS MARCHI, ROBERTA SUERO, L. A. “The Lid-Driven Square Cavity Flow: Numerical Solution with a 1024 x 1024 Grid”, *Journal of The Brazilian Society of Mechanical Sciences and Engineering - J BRAZ SOC MECH SCI ENG*, v. 31, 07 2009.

YUNUS A. ÇENGEL, J. M. C. *Mecânica dos Fluidos: Fundamentos e Aplicações*. McGraw-Hill Education, 2017.

PANTON, R. L. *Incompressible Flow*. John Wiley & Sons, Ltd, 2013.

FOX, ROBERT W.; MCDONALD, A. T. P. P. J. *Introdução a Mecânica dos Fluidos*. Editora LTC, 2010.

CHRISTOPHE GEUZAIN, J.-F. R. “Gmsh Reference Manual”, 2017.

ANJOS, G. R. “Projeto Final: Mecânica dos Fluidos e Transmissão de Calor Computacional”, *apostila do Curso-UFRJ*, 2022.

DEVELOPERS, P. “ParaView Reference Manual”, 2020.

- DA CUNHA, L. H. C. “ALE Finite Element Method for simulating flows with the streamfunction-vorticity formulation”, *Universidade do Estado do Rio de Janeiro*, 2020.
- BOTTASSO, C. L. “Introduction to Wind Turbine Aerodynamics”, *Wind Energy Institute, Technische Universität München*, 2019.
- ABRAHAM, R., MARSDEN, J. E., RATIU, T. *Manifolds, Tensor Analysis, and Applications*. 2 ed. New York, Springer-Verlag, 1988.
- ARNOLDI, W. E. “The principle of minimized iteration in the solution of the matrix eigenvalue problem”, *Quart. Appl. Math.*, v. 9, pp. 17–29, 1951.
- LANCZOS, C. “An iteration method for the solution of the eigenvalue problem of linear differential and integral operators”, *J. Research Nat. Bur. Standards*, v. 45, pp. 255–282, 1950.

Apêndice A

Código Fonte

```
1 # ----- FORMULAÇÃO CORRENTE-VORTICIDADE EM ESCOAMENTO LIVRE ----- #
2 # Autor: Juan de Oliveira dos Santos
3 # Engenharia Mecânica - Universidade Federal do Rio de Janeiro
4
5 # importando bibliotecas
6 import numpy as np
7 import meshio
8
9 # Definição das constantes do problema
10 nu = 0.01 # o termo "nu" foi utilizado para variar o número de Reynolds,
11 dt = 0.01 # uma vez que os demais termos característicos são unitários,
12 passos = 600 # resultando em nu = 1/Re
13
14
15
16 # Importação e tratamento da malha
17 msh = meshio.read('malha.msh') #importa malha computacional
18
19 X = msh.points[:,0] #lista com coordenada X dos pontos
20 Y = msh.points[:,1] #lista com coordenada Y dos pontos
21 IEN = msh.cells["triangle"] # matriz IEN
22 ne = IEN.shape[0] # define número de elementos
23
24 IENbound = msh.cells["line"] # lista de vértices nas faces do contorno
25 IENboundTypeElem = list(msh.cell_data['line']['gmsh:physical']-1) # pega índice
26 # de cada cc do physical group
27 # pra cada par de vértices (face) do IENbound
28 boundNames = list(msh.field_data.keys()) # pega o nome das cc's
29 # definidas no Gmsh
30 IENboundElem = [boundNames[elem] for elem in IENboundTypeElem] # relaciona elem
31 # ao índice da cc
32
```

```

33 # criando lista das faces (par de vértices) das células do contorno
34 cc = np.unique(IENbound.reshape(IENbound.size))
35
36 # atribuindo cada cc ao nome do contorno, definido no physical group do Gmsh
37 ccName = [[] for i in range(len(X))]
38 for elem in range(0,len(IENbound)):
39     ccName[IENbound[elem][0]] = IENboundElem[elem]
40     ccName[IENbound[elem][1]] = IENboundElem[elem]
41
42
43 # inicializando as matrizes
44 K = np.zeros( (npoints,npoints), dtype='float')
45 M = np.zeros( (npoints,npoints), dtype='float')
46 Gx = np.zeros( (npoints,npoints), dtype='float')
47 Gy = np.zeros( (npoints,npoints), dtype='float')
48 Kx = np.zeros( (npoints,npoints), dtype='float')
49 Ky = np.zeros( (npoints,npoints), dtype='float')
50 Kxy = np.zeros( (npoints,npoints), dtype='float')
51 Kestx = np.zeros( (npoints,npoints), dtype='float')
52 Kesty = np.zeros( (npoints,npoints), dtype='float')
53
54
55 # loop dos elementos da malha
56 for e in range(0,ne):
57     v = IEN[e]
58     # area do elemento
59     det = X[v[2]]*( Y[v[0]]-Y[v[1]]) \
60           + X[v[0]]*( Y[v[1]]-Y[v[2]]) \
61           + X[v[1]]*(-Y[v[0]]+Y[v[2]])
62
63     area = det/2.0
64
65     # matrizes do elemento linear
66     m = (area/12.0) * np.array([ [2.0, 1.0, 1.0],
67                                 [1.0, 2.0, 1.0],
68                                 [1.0, 1.0, 2.0] ])
69
70     # formula do elemento triangular linear
71     b1 = Y[v[1]]-Y[v[2]]
72     b2 = Y[v[2]]-Y[v[0]]
73     b3 = Y[v[0]]-Y[v[1]]
74
75     c1 = X[v[2]]-X[v[1]]
76     c2 = X[v[0]]-X[v[2]]
77     c3 = X[v[1]]-X[v[0]]
78
79     # matriz do gradiente
80     B = (1.0/(2.0*area)) * np.array([ [b1, b2, b3],

```

```

81         [c1, c2, c3] ])
82     # matriz do divergente
83     BT = B.transpose()
84
85     kxele = (1.0/(4.0*area))*np.array([ [b1**2, b1*b2, b1*b3],
86                                         [b2*b1, b2**2, b2*b3],
87                                         [b3*b1, b3*b2, b3**2] ])
88
89     kyele = (1.0/(4.0*area))*np.array([ [c1**2, c1*c2, c1*c3],
90                                         [c2*c1, c2**2, c2*c3],
91                                         [c3*c1, c3*c2, c3**2] ])
92
93     kele = kxele + kyele
94
95     gxele = (1.0/6.0)*np.array([ [b1, b2, b3],
96                                   [b1, b2, b3],
97                                   [b1, b2, b3] ])
98     gyele = (1.0/6.0)*np.array([ [c1, c2, c3],
99                                   [c1, c2, c3],
100                                  [c1, c2, c3] ])
101
102     for i in range(0,3):
103         ii = IEN[e,i]
104         for j in range(0,3):
105             jj = IEN[e,j]
106
107             # montagem (assembling) das matrizes K, M, Gx e Gy
108             K[ii,jj] = K[ii,jj] + kele[i,j]
109             M[ii,jj] = M[ii,jj] + m[i,j]
110             Gx[ii,jj] = Gx[ii,jj] + gxele[i,j]
111             Gy[ii,jj] = Gy[ii,jj] + gyele[i,j]
112
113
114     # Inicialização das variáveis
115     u = np.zeros( (npoints,1),dtype='float') #
116     v = np.zeros( (npoints,1),dtype='float') # Variáveis inicializadas
117     omega = np.zeros( (npoints,1),dtype='float') # com valor nulo
118     psi = np.zeros( (npoints,1),dtype='float') #
119
120     psicc = np.zeros( (npoints,1),dtype='float') # Vetor auxiliar para guardar
121                                                # informações de cc para psi
122     for i in cc:
123         if ccName[i] == 'inflow': # cc para a entrada do domínio
124             u[i] = 1.0 # Dirichlet para u
125             v[i] = 0.0 # Dirichlet para v
126             psicc[i] = Y[i] # Dirichlet para psi
127         if ccName[i] == 'farfield':
128             v[i] = 0.0

```

```

129     psicc[i] = Y[i]
130     if ccName[i] == 'noSlip':
131         u[i] = 0.0
132         v[i] = 0.0
133         psicc[i] = 0.0
134
135     # Calculo da condicao inicial de omega (M \omega = Gx * v - Gy * u)
136     b = np.dot(Gx,v) - np.dot(Gy,u)
137     omega = np.linalg.solve(M,b)
138
139
140     # Início do loop iterativo no tempo
141     for t in range(passos):
142         print(t) # Controle da iteração atual
143
144         # Calculo da condicao de contorno de omega (M omegacc = Gx * v - Gy * u)
145         b = np.dot(Gx,v) - np.dot(Gy,u) # Resolução do sistema linear Ax = b,
146         omegacc = np.linalg.solve(M,b) # onde A = M, x = omegacc e b = (Gx v - Gy u)
147
148         # Resolvendo o transporte da vorticidade
149         A_1 = (1.0/dt) * M + nu*K
150
151         # Utilizando o modo IMPLÍCITO: vgo entrando na matriz A
152         uI = np.diag(u[:,0]) # Transforma o vetor u em uma matriz diagonal
153         vI = np.diag(v[:,0]) # Transforma o vetor v em uma matriz diagonal
154         vgoI = np.dot(uI,Gx) + np.dot(vI,Gy) # Constrói matriz vgo = Gx u + Gy v
155
156         A_1 += vgoI # Adiciona a matriz vgo à matriz A
157
158         # Vetor solução b para o sist. linear do transporte da vorticidade
159         b_1 = (1.0/dt) * np.dot(M,omega)
160
161         # Imposição das cc's para omega
162         for i in cc:
163             A_1[i,:] = 0.0 # zerando toda a linha
164             A_1[i,i] = 1.0 # impondo 1 na diagonal
165             b_1[i] = omegacc[i] # cc definida no início do loop com base
166                 # nas velocidades do passo anterior
167
168         # Resolve transporte de vorticidade, atualizando o valor de omega
169         omega = np.linalg.solve(A_1,b_1)
170
171
172         # Resolvendo função corrente K*psi = M*omega, inicializando vetor solução b
173         b_2 = np.dot(M,omega)
174
175         # Inicializando matriz A
176         A_2 = K

```

```

177
178 # Impondo ccs de psi
179 for i in cc:
180     A_2[i,:] = 0.0 # zerando toda a linha
181     A_2[i,i] = 1.0 # impondo 1 na diagonal
182     b_2[i] = psicc[i]
183
184 psi = np.linalg.solve(A_2,b_2)
185
186
187 # Resolvendo velocidades  $M*u = Gy*psi$  e  $M*v = -Gx*psi$ 
188 b_3 = np.dot(Gy,psi)
189 b_4 = np.dot(-Gx,psi)
190
191 u = np.linalg.solve(M,b_3)
192 v = np.linalg.solve(M,b_4)
193
194 for i in cc:
195     if ccName[i] == 'inflow':
196         u[i] = 1.0
197         v[i] = 0.0
198     if ccName[i] == 'farfield':
199         v[i] = 0.0
200     if ccName[i] == 'noSlip':
201         u[i] = 0.0
202         v[i] = 0.0
203
204
205 # Exportando dados para pós-processamento
206 if t in range(0,passos):
207     point_data={'u':u}
208     data_vy={'v':v}
209     data_psi={'psi':psi}
210     data_omega={'omega':omega}
211     point_data.update(data_v)
212     point_data.update(data_psi)
213     point_data.update(data_omega)
214     meshio.write_points_cells('Iter'+str(t)+'.vtk',msh.points,
215                               msh.cells,point_data=point_data,)

```