

ALGORITMO DE REDE NEURAL PARA CÁLCULO DE FORÇAS
AERODINÂMICAS EM AEROFÓLIOS TREINADO COM MODELO DE
DINÂMICA DOS FLUIDOS COMPUTACIONAL POR ELEMENTOS FINITOS

Leonardo da Costa Lima Xavier de Mendonça

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.

Orientador: Gustavo Rabello dos Anjos

Rio de Janeiro

Março de 2024



*UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO*

Politécnica
UFRJ

Escola Politécnica

Departamento de Engenharia Mecânica

ALGORITMO DE REDE NEURAL PARA CÁLCULO DE FORÇAS
AERODINÂMICAS EM AEROFÓLIOS TREINADO COM MODELO DE
DINÂMICA DOS FLUIDOS COMPUTACIONAL POR ELEMENTOS FINITOS

Leonardo da Costa Lima Xavier de Mendonça

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.

Prof. Fernando Alves Rochinha, Ph.D.

Prof. Marcelo Amorim Savi, Ph.D.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2024

da Costa Lima Xavier de Mendonça, Leonardo

Algoritmo de rede neural para cálculo de forças aerodinâmicas em aerofólios treinado com modelo de dinâmica dos fluidos computacional por elementos finitos/
Leonardo da Costa Lima Xavier de Mendonça. – Rio de Janeiro: UFRJ/Escola Politécnica, 2024.

XII, 75 p.: il.; 29,7cm.

Orientador: Gustavo Rabello dos Anjos

Projeto de Graduação – UFRJ/ Escola Politécnica/
Curso de Engenharia Mecânica, 2024.

Referências Bibliográficas: p. 64 – 68.

1. Dinâmica dos Fluidos Computacional. 2. Elementos Finitos. 3. Aprendizado de máquina. 4. Redes neurais. I. Rabello dos Anjos, Gustavo. II. Universidade Federal do Rio de Janeiro, UFRJ, Curso de Engenharia Mecânica. III. Algoritmo de rede neural para cálculo de forças aerodinâmicas em aerofólios treinado com modelo de dinâmica dos fluidos computacional por elementos finitos.

Agradecimentos

Devo minha gratidão acima de tudo a minha família: a meus pais, a quem devo tudo que sou hoje; a meus avós, que me apoiaram em toda oportunidade que tiveram; a meus padrinhos, que sempre se preocuparam comigo; a meus irmãos, que suportaram minha companhia durante toda a sua vida, incluindo o período estressante de composição de um projeto de graduação.

Agradeço a inúmeros amigos, mas especialmente ao Gabriel, que me ajudou muito no processo de compor este trabalho, com referências e conselhos.

Também sou grato aos professores Fábio Ramos, Daniel Cruz, Thiago Ritto, Átila Pantaleão e principalmente ao meu orientador Gustavo Rabello, por me despertarem o interesse e me mostrarem o caminho nos temas que busquei explorar neste trabalho.

Sou muito grato à Bárbara, minha maior ouvinte, que já me viu trabalhando e ouviu falar tanto sobre este trabalho que já pode se considerar uma especialista em mecânica dos fluidos, e que me ajudou a encontrar forças em cada etapa deste processo.

Todos têm minha gratidão e meu afeto.

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

ALGORITMO DE REDE NEURAL PARA CÁLCULO DE FORÇAS
AERODINÂMICAS EM AEROFÓLIOS TREINADO COM MODELO DE
DINÂMICA DOS FLUIDOS COMPUTACIONAL POR ELEMENTOS FINITOS

Leonardo da Costa Lima Xavier de Mendonça

Março/2024

Orientador: Gustavo Rabello dos Anjos

Departamento: Engenharia Mecânica

O projeto iterativo de superfícies aerodinâmicas, como as pás de turbinas eólicas, exige a capacidade de estimar as propriedades do escoamento, como a sustentação e o arrasto. O Método de Elementos Finitos permite o cálculo dessas propriedades com precisão, porém a um grande custo computacional. Por isso, foi desenvolvida uma metodologia de aprendizado de máquina a fim de estimar as forças aerodinâmicas em aerofólios, obtendo uma série de resultados por elementos finitos, que foram dados como entrada para treinamento de uma rede neural profunda densamente conectada. A rede neural se mostrou capaz de calcular acuradamente os coeficientes de sustentação, arrasto e momento de aerofólios bidimensionais sujeitos a escoamentos laminares incompressíveis.

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

NEURAL NETWORK ALGORITHM FOR THE CALCULATION OF
AERODYNAMIC FORCES IN AN AIRFOIL TRAINED WITH A FINITE
ELEMENT CFD MODEL

Leonardo da Costa Lima Xavier de Mendonça

March/2024

Advisor: Gustavo Rabello dos Anjos

Department: Mechanical Engineering

The iterative design of aerodynamic surfaces, such as wind turbine blades, demands the ability to estimate flow properties, such as lift and drag. The Finite Element Method can accurately calculate these properties, but at a great computational cost. Therefore, a machine learning methodology was designed with the goal of estimating aerodynamic forces in airfoils, by obtaining a series of results via the finite element method, which would be used as input for the training of a fully connected neural network. The neural network was found capable of accurately calculate the lift, drag and moment coefficients for two-dimensional airfoils under laminar incompressible flow.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xii
1 Introdução	1
1.1 Motivação: Projeto de turbinas eólicas	1
1.2 Teoria de Aerofólio Fino	3
1.3 Redes Neurais	5
2 Metodologia	8
2.1 Equações de Navier-Stokes	8
2.2 Considerações sobre análise dimensional	9
2.2.1 Grupos adimensionais	9
2.2.2 Equação de movimento adimensional	11
2.3 Teoria de Aerofólio Fino	12
2.3.1 Premissas e teoria	12
2.3.2 Limitações	16
2.3.3 Aerofólios NACA	16
2.4 Método de Elementos Finitos	18
2.4.1 Teoria: Método de Elementos Finitos de Galerkin	19
2.4.2 Aplicação a escoamentos laminares	23
2.4.3 Cálculo dos coeficientes aerodinâmicos	28
2.4.4 Implementação	30
2.5 Rede Neural	32
2.5.1 Descida estocástica do gradiente	34
2.5.2 Retropropagação	35

2.5.3	Aplicação	36
2.5.4	Estratégias de treino	38
3	Resultados	41
3.1	Método de Elementos Finitos	41
3.1.1	Validação do Método de Elementos Finitos	41
3.1.2	Escolha de parâmetros de cálculo	50
3.2	Rede Neural	56
3.2.1	Seleção de hiperparâmetros	56
3.3	Desempenho e comparação dos métodos	59
4	Conclusão	62
	Referências Bibliográficas	64
A	Montagem das matrizes do Método de Elementos Finitos	69
A.1	Integração em elementos triangulares	69
A.2	Matriz de massa \mathbf{M}	71
A.3	Matriz elástica \mathbf{K}	72
A.3.1	Ordem 1: $\mathbf{M}_Q^{\hat{Q}}$	72
A.3.2	Ordem 2: $\mathbf{M}_V^{\hat{V}}$	72
A.4	Matriz gradiente \mathbf{G}	73
A.4.1	Ordem 1 e 2: $\mathbf{G}_Q^{\hat{Q}}$	74
A.4.2	Ordem 2: $\mathbf{G}_V^{\hat{V}}$	74

Lista de Figuras

1.1	Parque eólico de Gargaú, no Estado do Rio de Janeiro. Fonte: (1) . . .	2
1.2	Forças de sustentação (F_L), arrasto (F_D) e momento (M) atuantes sobre um aerofólio. D é dito comprimento da corda do aerofólio. . . .	2
1.3	Parâmetros geométricos de um aerofólio fino da série NACA de 4 dígitos (seção 2.3.3). T define a espessura, m o <i>camber</i> e p a forma da curva de <i>camber</i>	3
1.4	Representação esquemático de um único neurônio que compõe uma rede neural	6
1.5	Ilustração da rede neural treinada nesse trabalho, com uma arquitetura simplificada para fins de clareza da figura. Os parâmetros de entrada m , p , T , α e Re são discutidos na seção 2.3.3, e a arquitetura da rede, na seção 2.5	7
2.1	Esquema do aerofólio fino, e das curvas definidas para cálculo da circulação	13
2.2	Aerofólio NACA-7311, com <i>camber</i> máximo $m = 7\%D$, posição de maior <i>camber</i> $p = 30\%D$ e espessura $T = 11\%D$. Na representação acima, a linha tracejada marca a linha média, e D denota o comprimento de corda do aerofólio.	17
2.3	Elemento triangular linear, com 3 nós (vértices), para interpolação da função $T(x, y)$. Fonte: (2)	20
2.4	Elemento triangular quadrático, com 6 nós definidos: 3 nos vértices e 3 nos pontos médios das arestas. Fonte: (2)	23
2.5	Exemplo de rede neural <i>feed-forward</i> com uma única camada oculta. Fonte: adaptado de (3).	32

2.6	Representação do i -ésimo neurônio da k -ésima camada oculta de uma rede neural. À saída do neurônio, é aplicada uma função de ativação σ . (repetição da figura 1.4)	33
2.7	Função $y = \tanh z$	37
3.1	Resultados do escoamento de Poiseuille, simulado com o Método de Elementos Finitos, para $Re = 50$, com $h = 0.05$, $\Delta t = 0.01$, $T = 5$. . .	43
3.2	Perfil vertical de velocidade ao longo do duto, comparado com a solução analítica para o perfil desenvolvido. Note-se que as curvas a partir de $x = 4$ se sobrepõem, porque o perfil já se estabilizou. . . .	44
3.3	Perfil pressão ao longo do duto, comparado com a solução analítica para o perfil desenvolvido	44
3.4	Representação do escoamento em uma cavidade quadrada. Fonte: adaptado de (4)	45
3.5	Perfil de velocidade horizontal em $x=0,5$ obtido pelo código para diversos números de Reynolds, e comparação com a literatura (4). Os valores para $Re = 0,01$ e $Re = 10$ se sobrepõem, tanto na referência quanto na saída do código.	46
3.6	Perfil de velocidade vertical em $y=0,5$ obtido pelo código para diversos números de Reynolds, e comparação com a literatura (4). Os valores para $Re = 0,01$ e $Re = 10$ se sobrepõem, tanto na referência quanto na saída do código.	47
3.7	Resultados do escoamento no interior da Cavidade, simulado com o Método de Elementos Finitos, para $Re = 100$, com $h = 0.01$, $\Delta t = 0.01$, $T = 30$	48
3.8	Perfil de velocidade horizontal em diversos pontos do duto no escoamento sobre um degrau, e comparação com (5). $Re = 50$, $\Delta t = 0,01$, $T = 30$, $h_1 = 0,1$, $h_2 = 0,01$	49
3.9	Resultados do escoamento sobre um degrau, simulado com o Método de Elementos Finitos, para $Re = 50$, $\Delta t = 0,01$, $T = 30$, $h_1 = 0,1$, $h_2 = 0,01$	50

3.10	Coeficientes aerodinâmicos e tempo computacional para o cálculo do aerofólio NACA-4412 variando a folga. $n = 50, h = 1, 0, \Delta t = 0, 01, T = 30$	52
3.11	Coeficientes aerodinâmicos e tempo computacional para o cálculo do aerofólio NACA-4412 variando o número de pontos no contorno do aerofólio n . $folga = 10, h = 1, 0, \Delta t = 0, 01, T = 50$	53
3.12	Coeficientes aerodinâmicos e tempo computacional para o cálculo do aerofólio NACA-4412 variando o passo temporal Δt . $n = 100, folga = 6, h = 1, 0, T = 30$	54
3.13	Coeficientes aerodinâmicos e tempo computacional para o cálculo do aerofólio NACA-4412 variando o tamanho dos elementos de malha h . $n = 100, folga = 6, \Delta t = 0, 05, T = 30$	55
3.14	Erro quadrático médio obtido pela rede neural nos dados de validação e tempo computacional gasto em seu treinamento em função do número de neurônios e de camadas.	57
3.15	Erro quadrático médio e tempo de treino da rede em função do parâmetro de regularização λ	58
3.16	Erro quadrático médio obtido pela rede neural nos dados de validação e tempo computacional gasto em seu treinamento em função da taxa inicial de aprendizado de sua taxa de decaimento.	58
3.17	Coeficientes de sustentação, arrasto e momento calculados com cada um dos métodos para o aerofólio NACA4412. $Re=100$	61

Lista de Tabelas

2.1	Dimensão das grandezas envolvidas na força fluidodinâmica de sustentação de um objeto	10
3.1	Parâmetros selecionados para cálculo das forças dinâmicas em aerofólios, que serão dadas como entrada para a rede neural.	55
3.2	Valores iniciais dos hiperparâmetros utilizados para comparação das possíveis arquiteturas de rede.	57
3.3	Conjunto de hiperparâmetros selecionados para o treinamento da rede após as análises desta seção	59
3.4	Resultados obtidos para a rede selecionada, usando os parâmetros da tabela 3.3	59
3.5	Erro máximo absoluto, erro quadrático médio e tempo de cálculo da rede nos dados de teste; comparação com a teoria de aerofólio fino.	60

Capítulo 1

Introdução

1.1 Motivação: Projeto de turbinas eólicas

Em 2022, o Balanço Energético Nacional levantado pela EPE (6) apurou que a fonte eólica correspondia a 11,8% da oferta de energia primária na matriz elétrica brasileira. Com a tendência global à descarbonização do setor industrial, que pode dar-se tanto através da eletrificação quanto através do hidrogênio verde, espera-se que a importância das fontes renováveis para a composição energética do país seja ampliada durante os próximos anos.

Essa expectativa justifica esforços em maximizar o desempenho de máquinas e parques eólicos no clima local, que pode divergir em certos aspectos daquele encontrado no hemisfério Norte, onde se localizam a maioria dos desenvolvedores de aerogeradores.

Uma turbina eólica, ou aerogerador, atua convertendo a energia cinética translacional do vento em energia cinética rotacional das pás, que a transmitem através do rotor rígido da turbina até o eixo, montado na nacele. Dentro da nacele, pode haver uma caixa de transmissão mecânica a fim de aumentar a velocidade angular para que esteja compatível com a frequência do sistema elétrico nacional, que opera a 60 Hz. Na sequência, um gerador elétrico converte a energia cinética rotacional em energia elétrica, que é então transmitida para a subestação e finalmente despachada para a rede, chegando ao consumidor.

Todo o código desenvolvido para este trabalho é original, utiliza ferramentas *open source*, e está disponível em um repositório público no endereço <https://github.com/leo-mendonca/aerofolios>



Figura 1.1: Parque eólico de Gargaú, no Estado do Rio de Janeiro. Fonte: (1)

O foco do presente trabalho será na primeira etapa dessa cadeia: a conversão de energia cinética translacional do vento em energia cinética rotacional. A superfície aerodinâmica das pás é projetada de tal forma que, diante de um vento frontal, é produzida uma força normal à direção do vento e à direção longitudinal da turbina. Sendo assim, é induzido um momento em relação ao eixo do rotor, que realiza trabalho e portanto transfere energia à máquina.

Em duas dimensões, a seção reta da pá corresponde a um aerofólio, no qual a força de sustentação realiza trabalho útil, ao passo que a força de arrasto e o momento apenas aumentam a carga que a pá e a estrutura do aerogerador deverão suportar. Uma representação das forças atuantes sobre um aerofólio está exibido na figura 1.2. Desse modo, o projetista eólico buscará uma forma que maximize o coeficiente de sustentação e minimize o arrasto e o momento. O mesmo princípio de projeto ocorre em aplicações aeronáuticas, turbomáquinas, ventiladores, entre outros.

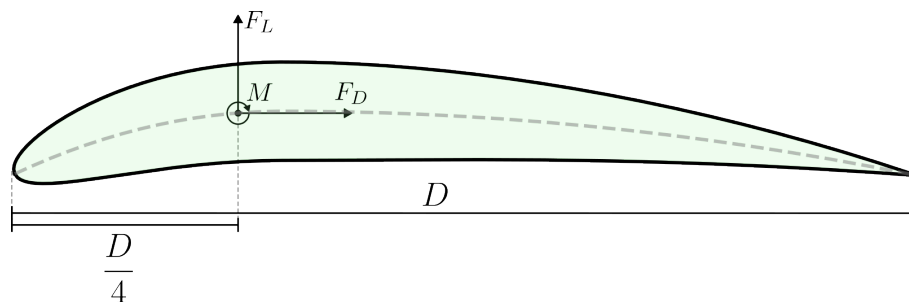


Figura 1.2: Forças de sustentação (F_L), arrasto (F_D) e momento (M) atuantes sobre um aerofólio. D é dito comprimento da corda do aerofólio.

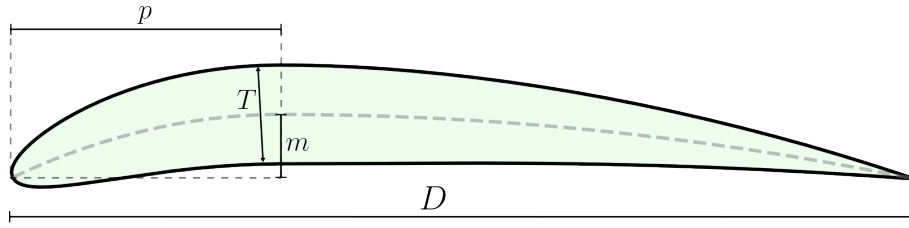


Figura 1.3: Parâmetros geométricos de um aerofólio fino da série NACA de 4 dígitos (seção 2.3.3). T define a espessura, m o *camber* e p a forma da curva de *camber*.

A otimização do projeto de um aerofólio exige do projetista a capacidade de desenhar e calcular as forças sobre a peça de forma iterativa, permitindo o ajuste de todas as restrições de projeto. Em um contexto de otimização multi-critérios, no qual pode ser necessário, por exemplo, maximizar a sustentação e minimizar o custo simultaneamente, é interessante a execução sucessiva do algoritmo a fim de identificar a frente de ótimos de Pareto, pontos de projeto que podem ser considerados otimizados por algum critério (7). Contudo, os métodos de dinâmica dos fluidos computacional (CFD) disponíveis para análise do escoamento exigem um tempo computacional significativo e uma grande disponibilidade de memória, o que tira agilidade do processo de projeto.

Por isso, o presente trabalho propõe uma metodologia para estimar rapidamente as propriedades aerodinâmicas de um aerofólio. Para fins de otimização, também pode ser interessante calcular os gradientes dessas propriedades, o que pode ser fornecido por métodos analíticos aproximados ou por ferramentas de machine learning. Vemos a seguir um sumário de ambas as abordagens.

1.2 Teoria de Aerofólio Fino

A teoria de aerofólio fino apresenta uma primeira aproximação do escoamento sobre um aerofólio. Para tanto, a viscosidade do fluido é desconsiderada, a espessura do aerofólio é suposta desprezível e seu *camber* e ângulo de ataque são considerados pequenos (ver fig. 1.3). Assim, é possível obter equações analíticas que relacionam diretamente a geometria do aerofólio com a distribuição de pressão sobre sua superfície e, portanto, com a força de sustentação e o momento aerodinâmico transmitidos para o perfil (8).

No problema clássico de escoamento sobre um aerofólio, os esforços são aplicados localmente ao longo da superfície aerodinâmica, porém, pela teoria de corpo rígido, em problemas bidimensionais, qualquer carregamento pode ser substituído por uma força (vetorial) e um momento perpendicular ao plano (9). Sendo assim, o problema se resume a determinar 3 esforços principais: arrasto, sustentação e momento.

- **Arrasto**

A força de arrasto é definida como a componente da força que o fluido imprime sobre o aerofólio *na direção do escoamento*. Isto é, na fig. 1.2, corresponde à força horizontal. Como será discutido na seção 2.2, essa força pode ser escrita da seguinte forma (10):

$$F_D = c_D \cdot \frac{1}{2} \rho U^2 S \quad (1.1)$$

onde ρ é a massa específica, U a velocidade do fluido, S a área de ataque e c_D é um parâmetro adimensional, o *coeficiente de arrasto*, função da geometria e do número de Reynolds $Re = \frac{\rho U L}{\mu}$.

- **Sustentação**

Analogamente, a força de sustentação é a força projetada sobre a direção normal ao escoamento. Na fig. 1.2, é a força vertical. Assim como o arrasto, a sustentação também pode se escrever em função de um adimensional, o *coeficiente de sustentação*:

$$F_L = c_L \cdot \frac{1}{2} \rho U^2 S \quad (1.2)$$

- **Momento**

O momento aerodinâmico é definido em torno da posição de quarto de corda, ou seja, em $x = L/4$, onde L é o comprimento do aerofólio. As razões para essa convenção resultam da teoria de aerofólio fino, como expresso na seção 2.3. Na fig. 1.2, o momento é definido como positivo quando entra no plano, ou seja, quando tende a aumentar o ângulo de ataque. Seu valor é dado em função do *coeficiente de momento* c_M :

$$M = c_M \cdot \frac{1}{2} \rho U^2 L S \quad (1.3)$$

Na teoria de aerofólio fino, os coeficientes aerodinâmicos são dados por:

$$c_L = 2\pi \left(A_0 + \frac{A_1}{2} \right) \quad (1.4a)$$

$$c_M = \frac{\pi}{2} \left(A_0 + A_1 + \frac{A_2}{2} \right) \quad (1.4b)$$

$$c_D = 0 \quad (1.4c)$$

Onde A_0, A_1, A_2 são coeficientes adimensionais derivados da geometria do aerofólio, como será visto na seção 2.3. A eq. 1.4c evidencia a principal limitação dessa metodologia: ao suprimir os efeitos viscosos, ela se torna incapaz de estimar o coeficiente de arrasto, que será nulo de acordo com as premissas da teoria. Além disso, em escoamentos com número de Reynolds baixo, a aproximação de fluido invíscido deixa de ser válida, de modo que os valores dados para os coeficientes de momento e sustentação também serão inadequados.

1.3 Redes Neurais

Uma rede neural é um modelo matemático composto de neurônios: unidades de processamento que recebem como entrada um número arbitrário de valores e aplicam um funcional linear seguido de uma função não-linear (*função de ativação*), produzindo como saída um número que servirá como entrada para outros neurônios da rede (3). As entradas, saídas e parâmetros de um neurônio, bem como a função de ativação aplicada, são discutidos na seção 2.5.

Existem na literatura diversas arquiteturas diferentes para redes neurais, com aplicações em ciência, engenharia, segurança, gestão, educação, entre outros (11). Pode-se citar reconhecimento e classificação de imagens (12), previsão de dados meteorológicos (13), processamento de linguagem natural (14) e resolução de equações diferenciais de origem física (15) (16).

O modelo aqui utilizado será uma rede neural *feed-forward* densamente conectada, treinada por aprendizado supervisionado para regressão. A rede receberá como entrada um certo conjunto de parâmetros geométricos de um dado aerofólio, bem como a velocidade relativa do fluido e seu ângulo de ataque. Já a saída da rede será composta por um vetor contendo os coeficientes de arrasto, sustentação e momento, como mostrado na figura 1.5.

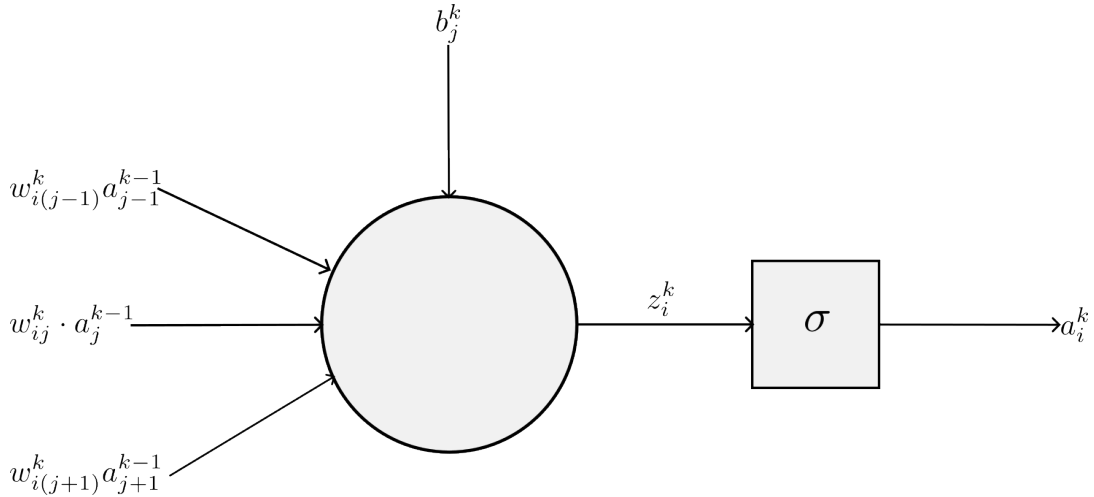


Figura 1.4: Representação esquemático de um único neurônio que compõe uma rede neural

Uma vez treinada a rede, o valor dos coeficientes dinâmicos de qualquer aerofólio poderá ser calculado rapidamente, e os gradientes de cada coeficiente em relação aos parâmetros geométricos, necessários para certas formas de otimização do projeto, são obtidos, também a um custo computacional baixo, através da retropropagação (17), uma forma de diferenciação automática (18), descrita na seção 2.5.

A rede neural exige, contudo, uma grande massa de dados para realizar seu treinamento, e que deverá ser mais significativa quanto maior for a expressividade e granularidade da rede. Para a finalidade desse trabalho, de estimativa das forças dinâmicas em aerofólios, esses dados podem ser obtidos por medições em túneis de vento ou por simulações numéricas de dinâmica dos fluidos computacional (CFD). A segunda abordagem será utilizada, por meio do método de Elementos Finitos, descrito na seção 2.4, para resolução das equações de Navier-Stokes.

A metodologia aqui desenvolvida se assemelha ao desenvolvido em (19). Nesse artigo, foi implementada uma rede neural convolucional que recebia como entrada uma imagem descrevendo a forma do aerofólio, e retornava a distribuição de pressão na superfície do aerofólio, utilizando resultados de simulações numéricas para treinamento. (20) e (21) ambos empregaram redes neurais com o objetivo de otimizar o projeto de aerofólios baseado em uma descrição paramétrica dos mesmos, com o objetivo de maximizar a razão entre sustentação e arrasto c_L/c_D .

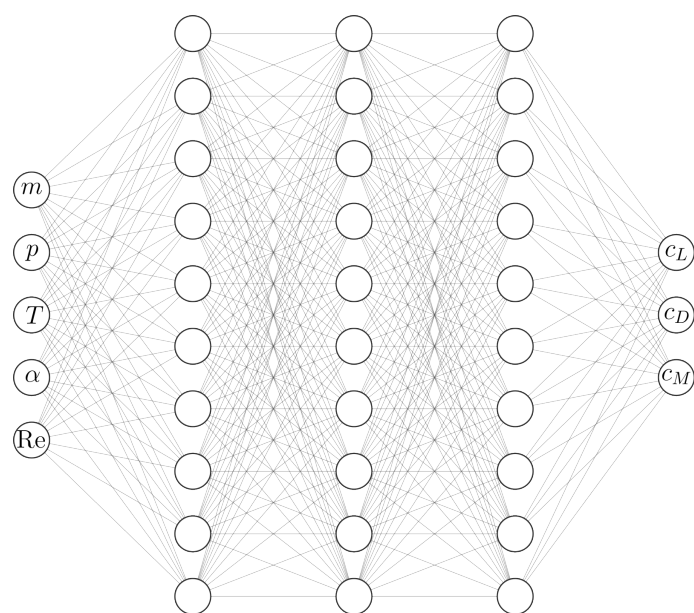


Figura 1.5: Ilustração da rede neural treinada nesse trabalho, com uma arquitetura simplificada para fins de clareza da figura. Os parâmetros de entrada m , p , T , α e Re são discutidos na seção 2.3.3, e a arquitetura da rede, na seção 2.5

Capítulo 2

Metodologia

2.1 Equações de Navier-Stokes

Na aplicação do Método de Elementos Finitos, o objetivo será resolver as equações diferenciais de movimento a fim de modelar o escoamento e consequentemente calcular os esforços atuando sobre aerofólios finos. As equações que regem um fluido newtoniano em escoamento compressível são (10):

$$\nabla \cdot \mathbf{u} = 0 \quad (2.1a)$$

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{u} \quad (2.1b)$$

Onde a eq. 2.1a diz respeito à conservação de massa associada à incompressibilidade, ao passo que a eq. 2.1b diz respeito à conservação da quantidade de movimento.

O termo $D\mathbf{u}/Dt$ é a dita *derivada material* de \mathbf{u} . A derivada material de uma dada propriedade $\lambda(\mathbf{x}, t)$ é a derivada no tempo de λ para uma determinada partícula inserida no escoamento, ou seja, a derivada material acompanha o fluido e, portanto, depende da convecção (22):

$$\frac{D\lambda}{Dt} = \frac{\partial \lambda}{\partial t} + (\mathbf{u} \cdot \nabla)\lambda = \frac{\partial \lambda}{\partial t} + \sum_{i=1}^3 u_i \frac{\partial \lambda}{\partial x_i} \quad (2.2)$$

Já o termo $\rho \mathbf{g}$ diz respeito a forças volumétricas, que não são consideradas neste trabalho, ou seja, $\mathbf{g} = 0$.

2.2 Considerações sobre análise dimensional

2.2.1 Grupos adimensionais

O objetivo central deste trabalho é desenvolver uma metodologia para calcular de forma eficiente as forças dinâmicas atuantes sobre um aerofólio. Contudo, sabe-se que essas forças dependem não apenas da geometria da peça, mas também da velocidade relativa do fluido, do ângulo de ataque do aerofólio e das propriedades materiais do fluido: massa específica e viscosidade dinâmica.

Torna-se interessante, portanto, relacionar essas grandezas de modo a reduzir a dimensão da entrada do modelo de *machine learning*, reduzindo, portanto, seu grau de complexidade e a massa de dados necessária para treiná-lo.

O Teorema Pi de Buckingham (23) oferece uma forma direta para reduzir o número de variáveis envolvidas, sem perda de informação. O teorema afirma que, dada uma relação:

$$\Phi(x_1, x_2, \dots, x_n) = 0 \quad (2.3)$$

entre n grandezas dimensionais, que possuam entre si r dimensões independentes, é possível reescrever Φ na forma de uma relação entre $n - r$ variáveis adimensionais independentes entre si:

$$\Phi(x_1, x_2, \dots, x_n) = \Psi(\pi_1, \pi_2, \dots, \pi_{n-r}) = 0 \quad (2.4)$$

O trabalho girará em torno das forças aerodinâmicas sobre a seção reta de uma aerofólio, no qual, por simplicidade, considera-se que o escoamento é bidimensional. Nesse problema, seja F_L a força de sustentação por unidade de comprimento; D o comprimento característico da seção; U a magnitude da velocidade do fluido; ρ a massa específica; μ a viscosidade da água; α o ângulo de ataque e $\boldsymbol{\beta} = (\beta_1, \dots, \beta_k)$ um vetor de parâmetros adimensionais puramente geométricos que definem a forma da peça (um exemplo para definição de aerofólios com 3 parâmetros é dado na seção 2.3.3). Desenvolvemos aqui o raciocínio para a força de sustentação, mas conclusões análogas são válidas para o arrasto e para o momento. Imediatamente observa-se que α e $\boldsymbol{\beta}$ já são grandezas adimensionais e, portanto, não podem ser reduzidos por análise dimensional. As dimensões das demais grandezas são dadas na tabela 2.1.

Dadas $n = 5$ variáveis dimensionais com $r = 3$ dimensões independentes, é

Grandeza	Dimensão
F_L	$[M] \cdot [T]^{-2}$
D	$[L]$
U	$[L] \cdot [T]^{-1}$
ρ	$[L]^{-3} \cdot [M]$
μ	$[L]^{-1} \cdot [M] \cdot [T]^{-2}$

Tabela 2.1: Dimensão das grandezas envolvidas na força fluidodinâmica de sustentação de um objeto

possível reduzir o problema a $n - r = 2$ variáveis adimensionais, além dos adimensionais α e β . Um adimensional natural é o número de Reynolds:

$$\pi_1 = Re = \frac{\rho U D}{\mu} \quad (2.5)$$

Resta construir um adimensional envolvendo a força por unidade de comprimento F_L , o que pode ser feito multiplicando essa grandeza pelas potências adequadas de ρ , U e D . Tem-se então:

$$\pi_2 = F_L \cdot \rho^x \cdot U^y \cdot D^z \quad (2.6)$$

Onde os valores de x , y e z podem ser estabelecidos impondo a condição de que π_2 seja adimensional, ou seja:

$$[\pi_2] = [F_L] \cdot [\rho]^x \cdot [U]^y \cdot [D]^z = [L]^{3x+y+z} \cdot [M]^{1+x} \cdot [T]^{-2-y} = 1 \quad (2.7a)$$

$$x = -1 \quad (2.7b)$$

$$y = -2 \quad (2.7c)$$

$$-3x + y + z = 0 \implies z = -1 \quad (2.7d)$$

$$\pi_2 = \frac{F_L}{\rho U^2 D} \quad (2.7e)$$

$$(2.7f)$$

Assim, a equação 2.4 resulta:

$$\Psi(Re, \pi_2, \alpha, \beta) = 0 \implies \pi_2 = \frac{F_L}{\rho U^2 D} = f(Re, \alpha, \beta) \quad (2.8)$$

Pode-se escrever, portanto:

$$c_L(Re, \alpha, \boldsymbol{\beta}) = 2f(Re, \alpha, \boldsymbol{\beta}) \quad (2.9a)$$

$$F_L = \frac{1}{2}\rho U^2 D c_L(Re, \alpha, \boldsymbol{\beta}) \quad (2.9b)$$

$$(2.9c)$$

Onde c_L é o coeficiente de arrasto na sua definição usual, com o comprimento característico igual à corda do aerofólio.

Um raciocínio análogo permite definir os coeficientes de arrasto e momento:

$$F_D = \frac{1}{2}\rho U^2 D c_D(Re, \alpha, \boldsymbol{\beta}) \quad (2.10)$$

$$M = \frac{1}{2}\rho U^2 D^2 c_M(Re, \alpha, \boldsymbol{\beta}) \quad (2.11)$$

Note-se que tradicionalmente o coeficiente de momento é definido como positivo no sentido *horário*.

2.2.2 Equação de movimento adimensional

A seção 2.2.1 acima evidencia que é possível calcular os coeficientes aerodinâmicos de arrasto, sustentação e momento em função apenas da geometria (representada por α e $\boldsymbol{\beta}$) e do número de Reynolds. Com isso, cada resultado pode ser extrapolado para diferentes condições, fluidos e escalas, desde que seja mantido o mesmo número de Reynolds. A fim de escrever as equações 2.1 de forma a refletir essa generalidade, elas podem ser adimensionalizadas através da introdução de variáveis adimensionais:

$$\mathbf{u} = U \mathbf{u}' \quad (2.12a)$$

$$\mathbf{x} = D \mathbf{x}' \quad (2.12b)$$

$$t = \frac{D}{U} t' \quad (2.12c)$$

$$p = \rho U^2 p' \quad (2.12d)$$

Da eq. 2.12b resulta a definição do operador nabla adimensional:

$$\nabla = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right] = \frac{1}{D} \left[\frac{\partial}{\partial x'}, \frac{\partial}{\partial y'}, \frac{\partial}{\partial z'} \right] = \frac{1}{D} \nabla' \quad (2.13)$$

Com as definições acima, reescreve-se a equação de conservação de massa:

$$\nabla \cdot \mathbf{u} = \frac{U}{D} \nabla' \cdot \mathbf{u}' = 0 \implies \nabla' \cdot \mathbf{u}' = 0 \quad (2.14)$$

E a conservação de quantidade de movimento:

$$\rho \frac{U^2}{D} \frac{\partial \mathbf{u}'}{\partial t'} + \rho \frac{U^2}{D} (\mathbf{u}' \cdot \nabla') \mathbf{u}' = -\frac{\rho U^2}{D} \nabla' p' + \mu \frac{U}{D^2} (\nabla')^2 \mathbf{u}' \quad (2.15)$$

Dividindo por $\rho U^2/D$:

$$\frac{\partial \mathbf{u}'}{\partial t'} + (\mathbf{u}' \cdot \nabla') \mathbf{u}' = -\nabla' p' + \frac{1}{Re} (\nabla')^2 \mathbf{u}' \quad (2.16)$$

Onde $Re = \rho U D / \mu$, e o termo gravitacional $\rho \mathbf{g}$ é considerado nulo.

Nas seções subsequentes, serão consideradas as equações adimensionalizadas acima, e, por simplicidade, será omitido o sobrescrito ', considerando que todas as variáveis já estão na forma adimensional, ou seja:

$$\nabla \cdot \mathbf{u} = 0 \quad (2.17a)$$

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} \quad (2.17b)$$

2.3 Teoria de Aerofólio Fino

2.3.1 Premissas e teoria

A teoria de aerofólio fino parte da premissa de que se pode modelar de forma aproximada o escoamento incompressível considerando um fluido invíscido, além de considerar que a espessura $t(x)$ do aerofólio é arbitrariamente pequena e que o camber $y(x)$ e o ângulo de ataque α são suficientemente próximos de 0 para fazer certas aproximações. Para calcular a força de sustentação, basta calcular a circulação da velocidade em torno do aerofólio. Pode-se demonstrar (24) (25) que, em um escoamento invíscido bidimensional, a magnitude da força de sustentação será dada por:

$$F_L = \rho U \Gamma = \rho U \oint_C \mathbf{u} \cdot d\mathbf{r} \quad (2.18)$$

Onde ρ e U são respectivamente a massa específica e a velocidade do fluido a montante, $\mathbf{u} = (u_x, u_y)$ é o campo vetorial de velocidade, Γ é a circulação e C é uma curva fechada no sentido horário que percorre a superfície do aerofólio, como se vê

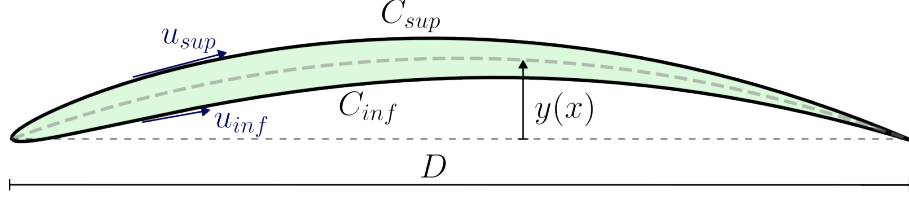


Figura 2.1: Esquema do aerofólio fino, e das curvas definidas para cálculo da circulação

na figura 2.1, sendo $d\mathbf{r} = (dx, dy)$ um deslocamento infinitesimal ao longo da curva.

Diante da hipótese de que a espessura do aerofólio é pequena, pode-se considerar que a circulação equivale a:

$$\Gamma = \int_{C_{sup}} \mathbf{u}_{sup} \cdot d\mathbf{r} - \int_{C_{inf}} \mathbf{u}_{inf} \cdot d\mathbf{r} \quad (2.19)$$

Onde \mathbf{u}_{sup} é a velocidade do fluido na superfície superior e \mathbf{u}_{inf} na inferior; e C_{sup} e C_{inf} são curvas percorrendo respectivamente a superfície superior e inferior. Fazendo mais uma aproximação, considera-se que o campo de velocidades pode ser modelado pela sobreposição de um campo uniforme $\mathbf{u} = U(\mathbf{n}_x \cos(\alpha) + \mathbf{n}_y \sin(\alpha))$ e de uma série de vórtices distribuídos ao longo da linha média do aerofólio, com intensidade $\gamma(x) = u_{sup}(x) - u_{inf}(x)$ (definida como positiva quando o vórtice está no sentido horário). Definimos os vetores unitários \mathbf{n}_x , \mathbf{n}_y e \mathbf{n}_z :

$$\mathbf{n}_x = (1, 0, 0) \quad (2.20a)$$

$$\mathbf{n}_y = (0, 1, 0) \quad (2.20b)$$

$$\mathbf{n}_z = (0, 0, 1) \quad (2.20c)$$

Assim, a velocidade induzida no ponto x pelo vórtice do ponto x' é igual a: (8)

$$d\mathbf{u}_\gamma(x) = \frac{\gamma(x')dx'}{2\pi[(x-x')^2 + (y(x)-y(x'))^2]}((y(x)-y(x'))\mathbf{n}_x - (x-x')\mathbf{n}_y) \quad (2.21)$$

Se o camber y for suficientemente pequeno, pode-se desprezar os termos $y - y'$ na equação acima, de modo que:

$$d\mathbf{u}_\gamma(x) = -\frac{\gamma(x')dx'}{2\pi(x-x')} \mathbf{n}_y \quad (2.22a)$$

$$\mathbf{u}_\gamma(x) = -\int_0^D \frac{\gamma(x')dx'}{2\pi(x-x')} \mathbf{n}_y \quad (2.22b)$$

Onde D é o comprimento da corda do aerofólio (maior distância entre dois pontos da linha média deste). O campo de velocidades total na superfície do aerofólio será portanto:

$$\mathbf{u}(x) = \mathbf{u}_\gamma(x) + U(\mathbf{n}_x \cos(\alpha) + \mathbf{n}_y \sin(\alpha)) \quad (2.23)$$

Pela condição de não-penetração do fluido, é preciso que a velocidade seja paralela à superfície, ou seja:

$$\frac{dy}{dx} = \frac{\mathbf{u} \cdot \mathbf{n}_y}{\mathbf{u} \cdot \mathbf{n}_x} = \frac{u_\gamma}{U \cos(\alpha)} + \tan(\alpha) \approx \frac{u_\gamma}{U} + \alpha \quad (2.24)$$

Onde são feitas aproximações de α pequeno. A equação 2.24 acima fornece, portanto, uma forma de determinar a expressão de γ , que é o que buscamos a fim de poder calcular a força de sustentação. É conveniente supor que γ tem a forma de uma série trigonométrica:

$$x = \frac{D}{2}(1 - \cos \theta) \quad (0 \leq \theta < \pi) \quad (2.25a)$$

$$dx = \frac{D}{2} \sin \theta d\theta \quad (2.25b)$$

$$\frac{\gamma(\theta)}{2U} = A_0 \cot \frac{\theta}{2} + \sum_{n=1}^{\infty} A_n \sin(n\theta) \quad (2.25c)$$

Donde, substituindo $\cot(\theta/2) = \frac{1+\cos\theta}{\sin\theta}$, temos:

$$\gamma dx = DU \left(A_0(1 + \cos \theta) + \sum_{n=1}^{\infty} A_n \sin(n\theta) \sin \theta \right) d\theta \quad (2.26a)$$

$$u_\gamma(x) = -\frac{U}{\pi} \int_0^\pi \frac{A_0(1 + \cos \theta) + \sum_{n=1}^{\infty} A_n \sin(n\theta) \sin \theta}{\cos \theta' - \cos \theta} d\theta' \quad (2.26b)$$

Manipulando as funções trigonométricas e calculando a integral da equação 2.26b, tem-se (8) :

$$\frac{u_\gamma(\theta)}{U} = -A_0 + \sum_{n=1}^{\infty} A_n \cos(n\theta) \quad (2.27)$$

Voltando à equação 2.24, obtém-se:

$$-A_0 + \sum_{n=1}^{\infty} A_n \cos(n\theta) = \frac{dy}{dx} - \alpha \quad (2.28)$$

Onde α é constante e dy/dx é conhecido, e depende apenas da função que define o camber. Finalmente, para encontrar os coeficientes A_m , basta multiplicar a equação

acima por $\cos(m\theta)$:

$$A_0 = \alpha - \frac{1}{\pi} \int_0^\pi \frac{dy}{dx}(\theta) d\theta \quad (2.29a)$$

$$A_m = \frac{2}{\pi} \int_0^\pi \frac{dy}{dx}(\theta) \cos(m\theta) d\theta \quad (2.29b)$$

Uma vez calculados os coeficientes acima, é possível retornar à equação 2.19, substituindo a 2.26a, e assim calculando a circulação:

$$\Gamma = \int_0^D \gamma(x) dx = DU \int_0^\pi \left(A_0(1 + \cos \theta) + \sum_{n=1}^{\infty} A_n \sin(n\theta) \sin \theta \right) d\theta \quad (2.30a)$$

$$\Gamma = DU\pi \left(A_0 + \frac{A_1}{2} \right) \quad (2.30b)$$

$$F_L = \rho U \Gamma = \rho D U^2 \pi \left(A_0 + \frac{A_1}{2} \right) \quad (2.30c)$$

$$c_L = \frac{2F_L}{\rho D U^2} = 2\pi \left(A_0 + \frac{A_1}{2} \right) \quad (2.30d)$$

O momento em relação ao ponto $x = 0$ pode ser calculado de forma similar (8):

$$M = -\rho U \int_0^D \gamma x dx = -\frac{\rho D^2 U^2}{2} \int_0^\pi (1 - \cos \theta) \left(A_0(1 + \cos \theta) + \sum_{n=1}^{\infty} A_n \sin(n\theta) \sin \theta \right) d\theta \quad (2.31a)$$

$$M = -\frac{\pi}{4} \rho D^2 U^2 \left(A_0 + A_1 - \frac{A_2}{2} \right) \quad (2.31b)$$

$$c_M = \frac{2M}{\rho D^2 U^2} = -\frac{\pi}{2} \left(A_0 + A_1 - \frac{A_2}{2} \right) \quad (2.31c)$$

As equações 2.29, 2.30 e 2.31 permitem calcular de forma eficiente a sustentação e o momento de um determinado aerofólio, além de permitir que se tire certas conclusões importantes para o projeto aerodinâmico, como:

- Inserindo a expressão de A_0 nas eqs. 2.30 e 2.31 e derivando em α , nota-se que a relação entre os coeficientes e o ângulo de ataque é aproximadamente linear, independentemente da forma do aerofólio.
- Reescrevendo a equação 2.31 para o momento em torno de $x = D/4$, nota-se que o momento nesse ponto é independente do ângulo α . Por isso, esse ponto é comumente definido como a referência em torno da qual se calcula o coeficiente de momento (8)(26), padrão que será seguido aqui.

2.3.2 Limitações

A principal limitação da teoria de aerofólio fino é que, ao desconsiderar os efeitos viscosos, torna-se impossível calcular o arrasto, uma vez que pode-se demonstrar que, para um corpo imerso em um fluido ideal (invíscido) com velocidade constante, a força de arrasto é nula (27). A eliminação da viscosidade também impede que se observe a variação dos esforços em função do número de Reynolds. Com efeito, esse parâmetro só possui significado em um fluido viscoso.

Outra limitação que aparece é o fato de se ter desconsiderado a espessura do aerofólio nos desenvolvimentos acima, de modo que não é possível computar a influência dessa característica. Além disso, quanto maior a espessura, mais distante a teoria de aerofólio fino estará de sua região de validade, de modo que os resultados devem ser analisados com cautela e não substituem uma análise rigorosa do escoamento via CFD ou ensaios experimentais.

A vantagem do método, contudo, está em permitir calcular rapidamente e de forma analítica uma aproximação dos coeficientes de sustentação e momento, o que pode ser vantajoso para a primeira etapa de um processo de otimização de projeto.

2.3.3 Aerofólios NACA

Na década de 1920, o Comitê Consultivo Nacional para Aeronáutica estadunidense (NACA), estabeleceu um método para definir a forma de um aerofólio a partir de um pequeno número de parâmetros, expressos através de 4 dígitos. Nesse modelo, o primeiro dígito define o camber máximo m , o segundo dá a posição de máximo camber p e os dois últimos fornecem a espessura máxima t , em função do comprimento de corda D (26). A figura 2.2 abaixo exemplifica um aerofólio definido dessa forma.

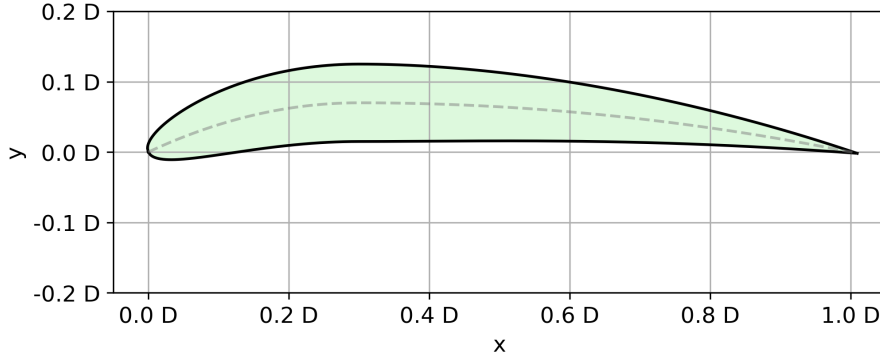


Figura 2.2: Aerofólio NACA-7311, com camber máximo $m = 7\%D$, posição de maior camber $p = 30\%D$ e espessura $T = 11\%D$. Na representação acima, a linha tracejada marca a linha média, e D denota o comprimento de corda do aerofólio.

O desenho do aerofólio é dado em função dos parâmetros m , p e T em (28) e (8). Em todas as equações a seguir, x , y , T , m e p representam as respectivas grandezas divididas pela corda D . Primeiramente, é preciso calcular o camber y , correspondente à linha média, e seu ângulo de inclinação φ :

$$y(x) = \begin{cases} \frac{m}{p^2}(2px - x^2), & \text{se } x \leq p \\ \frac{m}{(1-p)^2}[1 - 2p + 2px - x^2], & \text{se } x > p \end{cases} \quad (2.32a)$$

$$\tan \varphi = \frac{dy}{dx} = \begin{cases} \frac{2m}{p^2}(p - x), & \text{se } x \leq p \\ \frac{2m}{(1-p)^2}(p - x), & \text{se } x > p \end{cases} \quad (2.32b)$$

A espessura $t(x)$ é dada por:

$$t(x) = \frac{T}{2000}(2909\sqrt{x} - 1260x - 3516x^2 + 2843x^3 - 1015x^4) \quad (2.33)$$

As linhas superior (x_{sup}, y_{sup}) e inferior (x_{inf}, y_{inf}) são definidas como:

$$x_{sup} = x - t \sin \varphi \quad (2.34a)$$

$$y_{sup} = y + t \cos \varphi \quad (2.34b)$$

$$x_{inf} = x + t \sin \varphi \quad (2.34c)$$

$$y_{inf} = y - t \cos \varphi \quad (2.34d)$$

A facilidade com que se desenha os aerofólios NACA de 4 dígitos a partir de apenas 3 parâmetros torna-os adequados para as investigações deste trabalho. Como

se desenvolve na seção 2.2, sabe-se que os coeficientes aerodinâmicos dependerão apenas do ângulo de ataque, do número de Reynolds e do vetor de parâmetros $\beta = (m, p, T) = (\beta_1, \beta_2, \beta_3)$. O objetivo do método de Elementos Finitos, apresentado na seção 2.4 será, portanto, encontrar a função:

$$f(\beta_1, \beta_2, \beta_3, \alpha, Re) = (c_L, c_D, c_M) \quad (2.35)$$

E a rede neural será treinada a fim de aproximar essa função utilizando os resultados da análise de Elementos Finitos.

Como vimos acima, contudo, a teoria de aerofólio fino pode ser usada para encontrar f , embora dependa de uma série de aproximações. Substituindo dy/dx da equação 2.32b na eq. 2.29, obtém-se os coeficientes A_0 , A_1 e A_2 :

$$A_0 = \alpha - \frac{m}{\pi} [(2p - 1) \cdot (\pi + \beta\theta_p) + \beta \sin \theta_p] \quad (2.36a)$$

$$A_1 = \frac{m}{\pi} \left[2\beta(2p - 1) \sin \theta_p + \beta \left(\frac{1}{2} \sin 2\theta_p + \theta_p \right) + \frac{\pi}{(1 - p)^2} \right] \quad (2.36b)$$

$$A_2 = \frac{m\beta}{\pi} \left[(2p - 1) \sin 2\theta_p - \frac{2}{3} (\sin \theta_p \cos 2\theta_p - 2 \cos \theta_p \sin 2\theta_p) \right] \quad (2.36c)$$

Onde definimos:

$$\beta = \frac{1}{p^2} - \frac{1}{(1 - p)^2} \quad (2.37)$$

$$\theta_p = \arccos(2p - 1) \quad (2.38)$$

Assim, os coeficientes aerodinâmicos c_L , c_D e c_M são dados por:

$$c_L = 2\pi \left(A_0 + \frac{A_1}{2} \right) \quad (2.39a)$$

$$c_M = \frac{\pi}{2} \left(A_0 + A_1 - \frac{A_2}{2} \right) \quad (2.39b)$$

$$c_D = 0 \quad (2.39c)$$

2.4 Método de Elementos Finitos

Uma forma mais rigorosa de investigar o escoamento sobre um aerofólio é a resolução das equações de movimento de Navier-Stokes (eq. 2.17).

Entretanto, a não-linearidade da equação de quantidade de movimento, introduzida pelo termo convectivo $(\mathbf{u} \cdot \nabla)\mathbf{u}$, somada à complexidade geométrica do domínio,

que, no caso de um aerofólio, não necessariamente oferece simetrias que simplifiquem o problema, faz com que raramente seja possível resolver analiticamente as equações.

Desta forma, faz-se necessário buscar métodos numéricos de resolução, dentre os quais o Método de Elementos Finitos se destaca por sua flexibilidade, podendo ser aplicado sobre domínios de geometria complexa, desde que a malha de pontos seja suficientemente refinada (2). Por isso, essa técnica encontra aplicações em biomedicina (29); transferência de calor (2); mecânica dos sólidos e resistência dos materiais (30) (31); acústica (32) e, naturalmente, dinâmica dos fluidos (33) (34) (35) (36).

2.4.1 Teoria: Método de Elementos Finitos de Galerkin

O método de Galerkin, empregado nesse trabalho, consiste em buscar uma solução para a equação diferencial parcial em questão dentro de um espaço de funções de dimensão finita. No desenvolvimento a seguir, seja $u(x, y)$ a solução buscada, F e g funções contínuas, Ω o domínio espacial de (x, y) e $\partial\Omega$ seu contorno, no qual são impostas condições de Dirichlet (em $\partial\Omega_d$) e de von Neumann homogêneas (em $\partial\Omega_N$, com vetor normal \mathbf{n}). Suponha que se tem uma equação diferencial expressa por:

$$F(x, y, u, \nabla u, \nabla^2 u) = 0 \quad \forall (x, y) \in \Omega \quad (2.40a)$$

$$u(x_d, y_d) = g(x_d, y_d) \quad \forall (x_d, y_d) \in \partial\Omega_d \quad (2.40b)$$

$$\nabla u(x_N, y_N) \cdot \mathbf{n} = 0 \quad \forall (x_N, y_N) \in \partial\Omega_N \quad (2.40c)$$

As equações acima compõem a *forma forte* da equação diferencial. A fim de encontrar uma solução numérica para o problema acima pelo Método de Elementos Finitos, reescrevemos a equação 2.40a para obter a chamada *forma fraca*:

$$\int_{\Omega} F(x, y, u, \nabla u, \nabla^2 u) \cdot v(x, y) = 0 \quad \forall v \in \hat{V} \quad (2.41a)$$

$$u(x_d, y_d) = g(x_d, y_d) \quad \forall (x_d, y_d) \in \partial\Omega_d \quad (2.41b)$$

Onde v é chamada *função teste*, pertencente ao espaço \hat{V} , e V é o espaço de funções no qual buscamos u (também chamado *função tentativa*). A função tentativa deverá ser nula na região do contorno onde se aplica condições de Dirichlet. No método de Galerkin, a função teste deve estar no mesmo espaço da função tentativa, logo \hat{V} é o subconjunto de V contendo as funções que se anulam no contorno $\partial\Omega_d$.

As condições de contorno de von Neumann não aparecem de forma explícita na equação acima, mas serão consideradas indiretamente, como visto mais adiante na eq. 2.52.

Se V for o espaço das funções arbitrárias $u : \Omega \rightarrow \mathbb{R}$, a forma fraca é equivalente à forte. Entretanto, o Método de Elementos Finitos se baseia em uma escolha vantajosa de V de modo que seja um espaço de dimensão finita, ao mesmo tempo que permite o cálculo da integral da eq. 2.41a. Para isso, o espaço Ω é dividido em elementos, e o valor da função de interesse u ou v é definido a partir de uma função de interpolação contínua no interior de cada elemento, mas que pode ser descontínua entre um elemento e outro. O tipo de função de interpolação mais simples é linear, o que em um domínio bidimensional corresponde a um elemento com 3 nós, como mostrado na figura 2.3.

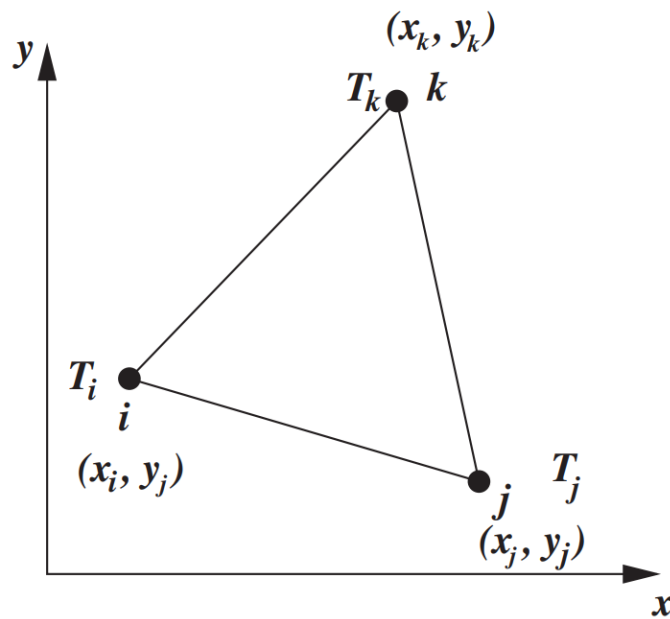


Figura 2.3: Elemento triangular linear, com 3 nós (vértices), para interpolação da função $T(x, y)$. Fonte: (2)

No elemento da figura 2.3, o valor de T no interior do elemento é dado por:

$$T(x, y) = a + bx + cy \quad (2.42)$$

Onde a , b e c são definidos unicamente de modo que:

$$T(x_q, y_q) = T_q \quad (q = i, j, k) \quad (2.43)$$

Assim, ao definir o valor de T nos 3 nós, fica definido seu valor em todo o elemento. Ou seja, o espaço de funções contendo T nesse elemento linear com 3 nós tem dimensão 3, em vez de ter dimensão infinita, que seria o caso do espaço de funções C^0 , por exemplo.

Uma descrição equivalente é dada a partir das funções de forma lineares N :

$$N_q(x, y) = a_q + b_q x + c_q y \quad (2.44a)$$

$$N_q(x_p, y_p) = \delta_{qp} \quad (2.44b)$$

$$T(x, y) = \sum_{q \in (i, j, k)} T_q \cdot N_q(x, y) \quad (2.44c)$$

Os coeficientes a_q, b_q, c_q da função de forma N_q são obtidos aplicando a eq. 2.44b em $q = i, j, k$. Abaixo exemplificamos para $q = i$:

$$\begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{bmatrix} \cdot \begin{pmatrix} a_i \\ b_i \\ c_i \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \implies \begin{cases} a_i = (x_j y_k - x_k y_j)/2A \\ b_i = (y_j - y_k)/2A \\ c_i = (x_k - x_j)/2A \end{cases} \quad (2.45)$$

Onde A é a área do elemento, igual a:

$$A = \frac{1}{2} \det \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} \quad (2.46)$$

Com a estrutura do elemento linear acima definido, o espaço Ω pode ser dividido inteiramente em elementos triangulares, e, em cada elemento $e_\ell \subset \Omega$, temos:

$$T(x, y) = \sum_{q=1}^3 {}^\ell T_q \cdot {}^\ell N_q(x, y) \quad \forall (x, y) \in e_\ell \quad (2.47)$$

Onde ${}^\ell T_q$ denota o valor da função no q -ésimo nó do ℓ -ésimo elemento, e ${}^\ell N_q(x, y)$ a função de forma linear correspondente.

Definimos o espaço V da função tentativa u como o espaço das funções $u : \Omega \rightarrow \mathbb{R}$ descritas pela eq. 2.47.

$$V = \left\{ u : \Omega \rightarrow \mathbb{R} \mid (x, y) \in e_\ell \Rightarrow u(x, y) = \sum_{q=1}^3 {}^\ell u_q \cdot {}^\ell N_q(x, y) \right\} \quad (2.48)$$

Portanto, o espaço \hat{V} da função teste é:

$$\hat{V} = \{v \in V \mid v_i = 0 \quad \forall (x_i, y_i) \in \partial\Omega_d\} \quad (2.49)$$

Os espaços V e \hat{V} são definidos de forma única pelos valores nodais u_i e v_i . Conseqüentemente, se temos n nós ao todo, e n_d nós no contorno com condições de Dirichlet, o espaço V tem dimensão n e o espaço \hat{V} tem dimensão $n - n_d$.

A integral da equação 2.41a se torna, portanto, uma soma de integrais em cada elemento da malha:

$$\begin{aligned} \int_{\Omega} F(x, y, u, \nabla u, \nabla^2 u) \cdot v &= \sum_{\ell} \int_{e_{\ell}} F(x, y, u, \nabla u, \nabla^2 u) \cdot \sum_{q=1}^3 {}^{\ell}v_q {}^{\ell}N_q(x, y) \\ &= \sum_{\ell} \sum_{q=1}^3 {}^{\ell}v_q \int_{e_{\ell}} F(x, y, u, \nabla u, \nabla^2 u) \cdot {}^{\ell}N_q(x, y) \end{aligned} \quad (2.50)$$

No Método de Elementos Finitos de Galerkin, a eq. 2.50 é aplicada para todas as funções $v \in \hat{V}$ tais que se tenha $v_i = 1$ em um único nó i , e $v_j = 0$, $\forall j \neq i$, ou seja, $v(x, y) = N_i(x, y)$. Sendo assim, se obtém $n - n_d$ equações algébricas, que, unidas às n_d equações das condições de contorno de Dirichlet, compõem um sistema de n equações, suficiente para determinar os n valores nodais u_i , que são as incógnitas do problema. Na aplicação do método às equações de Navier-Stokes, usaremos certas estratégias para garantir que essas equações componham um sistema linear, reduzindo significativamente o tempo computacional da resolução.

Além disso, caso a expressão envolva derivadas segundas, é necessário aplicar a integração por partes para reduzir a ordem da derivada, como mostrado abaixo para o termo laplaciano:

$$\int_{\Omega} \nabla^2 u v d\Omega = \int_{\partial\Omega} v \nabla u \cdot \mathbf{n} d\partial\Omega - \int_{\Omega} \nabla u \cdot \nabla v d\Omega \quad (2.51)$$

Observemos que, em $\partial\Omega_d$, temos $v = 0$, e, em $\partial\Omega_N$, temos $\nabla u \cdot \mathbf{n} = 0$, ou seja, a integral no contorno desaparece, de modo que:

$$\int_{\Omega} \nabla^2 u v d\Omega = - \int_{\Omega} \nabla u \cdot \nabla v d\Omega \quad (2.52)$$

Caso haja um problema com condições de contorno de von Neumann não-homogêneas, a integral deverá ser calculada considerando esses valores.

O método descrito nesta seção foi exemplificado para um domínio Ω bidimensional e elementos triangulares lineares, mas a metodologia se generaliza para espaços de dimensão qualquer e para diversos outros tipos de elemento. Neste trabalho, trataremos apenas de escoamentos bidimensionais, e utilizaremos elementos triangulares lineares (3 nós) e quadráticos (6 nós), como será discutido a seguir.

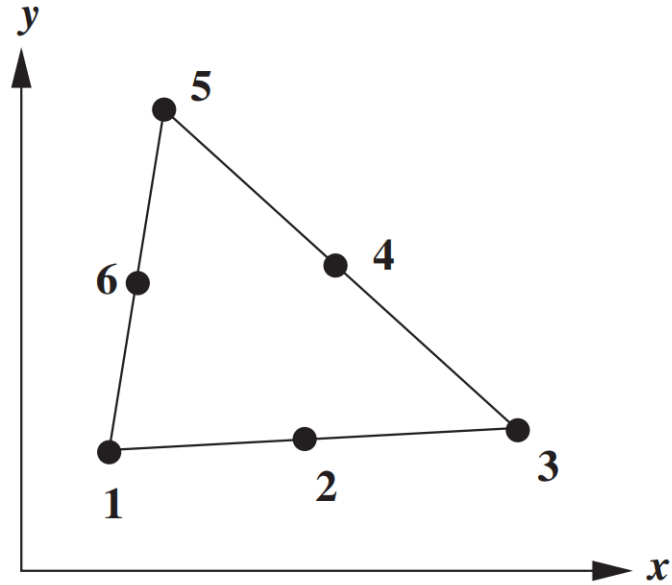


Figura 2.4: Elemento triangular quadrático, com 6 nós definidos: 3 nos vértices e 3 nos pontos médios das arestas. Fonte: (2)

2.4.2 Aplicação a escoamentos laminares

Condição de Ladyzhenskaya-Babuska-Brezzi

Em um escoamento incompressível, caso se utilize elementos finitos iguais para interpolação da velocidade e da pressão, produz-se um problema de ponto de sela, no qual a solução não é estável numericamente. A fim de evitar tal instabilidade, é necessário empregar elementos de maior dimensão para a velocidade do que para a pressão, na dita condição de Ladyzhenskaya-Babuška-Brezzi (30). Por isso, nos cálculos de elementos finitos deste trabalho, são definidos um espaço de funções V para a velocidade, com funções de forma quadráticas (ordem 2); e um espaço de funções Q para a pressão, com funções de forma lineares (ordem 1). Com essa combinação de elementos, garante-se que, se o problema original for bem-posto, o sistema linear produzido pelo Método de Elementos Finitos terá solução única.

Com esse fim, definimos o elemento triangular quadrático (ilustrado na fig. 2.4), associado à função de forma de ordem 2, $M(x, y)$. Note-se que, por M possuir 6 coeficientes, são necessários 6 nós por elemento.

$$M(x, y) = a + bx + cy + dx^2 + ey^2 + fxy \quad (2.53a)$$

$${}^\ell M_i(x_j, y_j) = \delta_{ij} \quad (i, j = 1, \dots, 6) \quad (2.53b)$$

A equação 2.53b fornece um sistema linear 6×6 , permitindo definir os coeficientes a, b, c, d, e e f em cada nó do elemento l , como mostrado abaixo para o nó i :

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2^2 & y_2^2 & x_2 y_2 \\ 1 & x_3 & y_3 & x_3^2 & y_3^2 & x_3 y_3 \\ 1 & x_4 & y_4 & x_4^2 & y_4^2 & x_4 y_4 \\ 1 & x_5 & y_5 & x_5^2 & y_5^2 & x_5 y_5 \\ 1 & x_6 & y_6 & x_6^2 & y_6^2 & x_6 y_6 \end{bmatrix} \cdot \begin{pmatrix} a_i \\ b_i \\ c_i \\ d_i \\ e_i \\ f_i \end{pmatrix} = \begin{pmatrix} \delta_{i1} \\ \delta_{i2} \\ \delta_{i3} \\ \delta_{i4} \\ \delta_{i5} \\ \delta_{i6} \end{pmatrix} \quad (2.54)$$

Os coeficientes de cada função de forma e cada elemento são calculados na inicialização do problema, resolvendo numericamente a equação 2.54 para $i = 1, \dots, 6$.

Define-se então os espaços Q para a pressão p e V para a velocidade \mathbf{u} :

$$Q = \left\{ p : \Omega \rightarrow \mathbb{R} \mid (x, y) \in e_\ell \Rightarrow p(x, y) = \sum_{q=1}^3 {}^\ell p_q \cdot {}^\ell N_q(x, y) \right\} \quad (2.55a)$$

$$\hat{Q} = \{ q \in Q \mid q_i = 0 \quad \forall (x_i, y_i) \in \partial\Omega_d \} \quad (2.55b)$$

$$V = \left\{ \mathbf{u} : \Omega \rightarrow \mathbb{R}^2 \mid (x, y) \in e_\ell \Rightarrow \mathbf{u}(x, y) = \sum_{\lambda=x,y} \sum_{q=1}^6 {}^\ell u_{\lambda q} \cdot {}^\ell M_q(x, y) \mathbf{n}_\lambda \right\} \quad (2.56a)$$

$$\hat{V} = \{ \mathbf{v} \in V \mid \mathbf{v}(x_d, y_d) = 0 \quad \forall (x_d, y_d) \in \partial\Omega_d \} \quad (2.56b)$$

Onde \mathbf{n}_x e \mathbf{n}_y são os vetores unitários nas direções x e y , respectivamente.

A velocidade é interpolada em um número de nós maior do que a pressão, por isso a dimensão de V , denotada n_V , é maior que $\dim Q = n_Q$.

Desacoplamento entre velocidade e pressão

As equações de Navier-Stokes (eq. 2.17) apresentam um acoplamento entre velocidade e pressão. É possível, contudo, desacoplar as duas grandezas através de uma resolução em múltiplas etapas, proposta em (37). Nessa abordagem, a conservação de quantidade de movimento e a conservação de massa são tratadas sequencialmente a cada passo de tempo. Essa separação permite uma redução do custo computacional relativo sem perda de acurácia (33).

Primeiramente, devemos discretizar no tempo a equação 2.17b:

$$\frac{\partial \mathbf{u}}{\partial t} \approx \frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\Delta t} \implies \frac{\mathbf{u}_{n+1}}{\Delta t} = \frac{\mathbf{u}_n}{\Delta t} - (\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla p_n + \frac{1}{Re} \nabla^2 \mathbf{u} \quad (2.57)$$

No lado direito da equação acima, os termos contendo \mathbf{u} e p podem ser tomados no instante t_n , de modo que já serão conhecidos quando se estiver calculando \mathbf{u}_{n+1} , configurando uma forma *explícita*, ou no instante $t_n + 1$, de modo que esses termos entrarão na matriz do sistema linear que corresponde a essa equação, em uma forma *implícita*. Neste trabalho, consideramos a forma implícita, porém, para que o sistema seja linear, o termo convectivo é calculado como $(\mathbf{u}_n \cdot \nabla)\mathbf{u}_{n+1}$. Desse modo, ele se torna uma função linear de \mathbf{u}_{n+1} .

Na técnica de separação velocidade-pressão, introduzimos um campo intermediário de velocidades \mathbf{u}^* , que não respeita a condição de incompressibilidade, e um campo auxiliar de variação de pressão p^* . Reescreve-se a equação 2.57 da seguinte forma::

$$\frac{\mathbf{u}^* - \mathbf{u}_n}{\Delta t} = -(\mathbf{u}_n \cdot \nabla)\mathbf{u}^* - \nabla p_n + \frac{1}{Re}\nabla^2\mathbf{u}^* \implies \quad (2.58a)$$

$$\implies \left[\frac{1}{\Delta t} + (\mathbf{u}_n \cdot \nabla) - \frac{1}{Re}\nabla^2 \right] \cdot \mathbf{u}^* = \frac{\mathbf{u}_n}{\Delta t} - \nabla p_n \quad (2.58b)$$

$$\frac{\mathbf{u}_{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla p^* \quad (2.58c)$$

$$p^* = p_{n+1} - p_n \quad (2.58d)$$

A eq. 2.58a pode ser resolvida apenas com conhecimento de \mathbf{u}_n e p_n . A eq. 2.58c a princípio contém duas incógnitas, porém, se aplicarmos o operador divergência a ambos os lados da equação, temos:

$$\nabla^2 p^* = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (2.59)$$

Pois, pela condição de incompressibilidade, $\nabla \cdot \mathbf{u}_{n+1} = 0$. O campo \mathbf{u}^* , por outro lado, não é um campo de velocidades real, e por isso não precisa ser incompressível. Após obter o valor de p^* , basta substituí-lo nas eqs. 2.58c e 2.58d para encontrar \mathbf{u}_{n+1} e p_{n+1} , respectivamente. Sendo assim, o método de separação velocidade-pressão pode ser escrito passo a passo:

$$\left[\frac{1}{\Delta t} + (\mathbf{u}_n \cdot \nabla) - \frac{1}{Re}\nabla^2 \right] \cdot \mathbf{u}^* = \frac{\mathbf{u}_n}{\Delta t} - \nabla p_n \quad (2.60a)$$

$$\nabla^2 p^* = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (2.60b)$$

$$\mathbf{u}_{n+1} = \mathbf{u}^* - \Delta t \nabla p^* \quad (2.60c)$$

$$p_{n+1} = p_n + p^* \quad (2.60d)$$

Onde o lado direito das equações contém apenas grandezas já conhecidas em cada passo. Uma vantagem significativa de tratar o problema dessa forma, em vez da expressão original (eq. 2.17), é que todas as etapas acima são equações lineares em relação às incógnitas, podendo ser resolvidas por algoritmos numéricos usuais.

A seguir, escreve-se as equações 2.60 na forma fraca, de modo a transformá-las em equações lineares.

Montagem do sistema linear

Reiterando, o objetivo é escrever as equações 2.60 na forma fraca, como mostrado na eq. 2.41a. O espaço de funções da velocidade V emprega polinômios de ordem 2, e o da pressão, Q , polinômios de ordem 1.

Começamos multiplicando a eq. 2.60a por uma função teste $v \in \hat{V}$ e integrando:

$$\int_{\Omega} \left[\frac{\mathbf{u}^*}{\Delta t} + (\mathbf{u}_n \cdot \nabla) \mathbf{u}^* - \frac{1}{Re} \nabla^2 \mathbf{u}^* \right] \cdot \mathbf{v} d\Omega = \int_{\Omega} \left(\frac{\mathbf{u}_n}{\Delta t} - \nabla p_n \right) \cdot \mathbf{v} d\Omega \quad (2.61)$$

Observe-se que, para um dado ponto no interior do ℓ -ésimo elemento e_{ℓ} , temos:

$$u_x(x, y) = \sum_{q=1}^6 {}^{\ell} u_{xq} {}^{\ell} M_q(x, y) \quad (2.62a)$$

$$u_y(x, y) = \sum_{q=1}^6 {}^{\ell} u_{yq} {}^{\ell} M_q(x, y) \quad (2.62b)$$

$$p = \sum_{r=1}^3 {}^{\ell} p_r {}^{\ell} N_r(x, y) \quad (2.62c)$$

E que, no método de Galerkin, selecionamos a função teste igual a M_i ($i = 1, \dots, n_V$) \mathbf{n}_{λ} , com $\lambda = x$ ou y . Ou seja, num dado elemento e_{ℓ} :

$$\mathbf{v}(x, y) = M_i(x, y) \mathbf{n}_{\lambda} = \begin{cases} 0 & , \text{ se } i \notin e_{\ell} \\ {}^{\ell} M_{\gamma(\ell, i)}(x, y) \mathbf{n}_{\lambda} & , \text{ se } i \in e_{\ell} \end{cases} \quad (2.63)$$

Onde $\gamma(\ell, i)$ denota a posição relativa (1, 2, .. ou 6) do vértice i no elemento e_{ℓ} .

Voltando à eq. 2.61, os termos integrais podem ser escritos como somatórios:

$$\begin{aligned} \int_{\Omega} \frac{\mathbf{u}}{\Delta t} \cdot (M_i \mathbf{n}_{\lambda}) d\Omega &= \frac{1}{\Delta t} \sum_j \sum_{e_{\ell} \ni i, j} u_{\lambda j} \int_{e_{\ell}} M_j(x, y) M_i(x, y) de_{\ell} = \\ &= \frac{1}{\Delta t} [\mathbf{M}_V^{\hat{V}} \cdot \mathbf{u}_{\lambda}]_i \end{aligned} \quad (2.64)$$

$$\begin{aligned}
\int_{\Omega} \frac{1}{Re} \nabla^2 \mathbf{u} \cdot (M_i \mathbf{n}_{\lambda}) d\Omega &= -\frac{1}{Re} \int_{\Omega} \nabla u_{\lambda} \cdot \nabla M_i d\Omega = \\
&= \sum_j \sum_{e_{\ell} \ni i, j} -u_{\lambda j} \int_{e_{\ell}} \nabla M_i(x, y) \cdot \nabla M_j(x, y) de_{\ell} = \quad (2.65) \\
&= \frac{1}{Re} \left[\mathbf{K}_{\hat{V}} \cdot \mathbf{u}_{\lambda} \right]_i
\end{aligned}$$

$$\begin{aligned}
\int_{\Omega} -\nabla p \cdot (M_i \mathbf{n}_{\lambda}) d\Omega &= -\int_{\Omega} \frac{\partial p}{\partial x_{\lambda}} M_i d\Omega = \\
&= \sum_j \sum_{e_{\ell} \ni i, j} p_j \int_{e_{\ell}} \frac{\partial N_j}{\partial x_{\lambda}}(x, y) M_i(x, y) de_{\ell} = \quad (2.66) \\
&= -\left[\mathbf{G}_{\lambda \hat{Q}} \cdot \mathbf{p} \right]_i
\end{aligned}$$

$$\begin{aligned}
\int_{\Omega} (\mathbf{u}_n \cdot \nabla) \mathbf{u} \cdot (M_i \mathbf{n}_{\lambda}) &\approx \left[\mathbf{u}_{nx} \mathbf{G}_{x \hat{V}} + \mathbf{u}_{ny} \mathbf{G}_{y \hat{V}} \right] \cdot \mathbf{u}_{\lambda}^* = \\
&= \left[\mathbf{C}_{\hat{V}} \cdot \mathbf{u}_{\lambda}^* \right]_i \quad (2.67)
\end{aligned}$$

Onde o subscripto de cada matriz é o espaço da função tentativa, e o sobrescrito é o espaço da função teste. A matriz \mathbf{M} é identificada como matriz de massa, \mathbf{K} é dita matriz elástica, \mathbf{G} matriz gradiente e \mathbf{C} matriz de convecção. Note-se que apenas a matriz de convecção sofre alteração entre um passo temporal e outro. As demais matrizes podem ser inicializadas no início do processo de solução e mantidas na memória do programa, a fim de poupar esforço computacional. Todas as integrais que compõem as matrizes citadas são resolvidas analiticamente, e suas expressões estão exibidas no apêndice A.

O método de Elementos Finitos e a separação velocidade-pressão permitiram escrever a equação de quantidade de movimento como um sistema linear, no qual apenas \mathbf{u}^* é desconhecido. O mesmo pode ser feito nas outras equações da eq. 2.60. Na eq. 2.60b, multiplicamos cada termo por uma função teste $q \in \hat{Q}$ escalar, com uma função de forma linear, igual a N_i , onde i varrendo cada nó sem condição de Dirichlet da pressão (ou seja, apenas os vértices de cada elemento). Assim:

$$\int_{\Omega} \nabla^2 p N_i d\Omega = \left[\mathbf{K}_{\hat{Q}} \cdot \mathbf{p} \right]_i \quad (2.68)$$

$$\begin{aligned}
\int_{\Omega} (\nabla \cdot \mathbf{u}) q d\Omega &= \sum_{\lambda \in \{x, y\}} \sum_j \sum_{e_{\ell} \ni i, j} u_{\lambda j} \int_{e_{\ell}} \frac{\partial M_j}{\partial x_{\lambda}}(x, y) N_i(x, y) de_{\ell} = \\
&= \left[\mathbf{G}_{x \hat{V}} \right] \cdot \mathbf{u}_x + \left[\mathbf{G}_{y \hat{V}} \right] \cdot \mathbf{u}_y \quad (2.69)
\end{aligned}$$

Já na eq. 2.60c, multiplicamos novamente por $v \in \hat{V}$, resultando na eq. 2.71c.

Note-se que estamos fazendo a função teste assumir o valor da função de forma em cada nó no qual não há condição de Dirichlet. Nos nós onde há essa condição de contorno, contudo, a função teste deve ser obrigatoriamente nula. Sendo assim, se temos n_V nós para a velocidade, dos quais n_{Vd} têm condições de Dirichlet, dadas por $u(\mathbf{x}) = g(\mathbf{x})$, o sistema linear equivalente à eq. 2.60 terá $n_V - n_{Vd}$ equações e n_V incógnitas, ou seja, não teria solução única. Contudo, se impusermos o valor da condição de contorno em cada um dos n_{Vd} nós, produziremos mais n_{Vd} equações e, portanto, podemos resolver o sistema. Sendo assim, definimos a matriz \mathbf{D}_u e o vetor \mathbf{d}_u :

$$[\mathbf{D}_u]_{ij} = \begin{cases} \delta_{ij} & , \text{ se } i \text{ tem condição de Dirichlet} \\ 0 & , \text{ caso contrário} \end{cases} \quad (2.70a)$$

$$[\mathbf{d}_u]_i = \begin{cases} g(x_i, y_i) & , \text{ se } i \text{ tem condição de Dirichlet} \\ 0 & , \text{ caso contrário} \end{cases} \quad (2.70b)$$

Um procedimento idêntico define a matriz \mathbf{D}_p e o vetor \mathbf{d}_p para a pressão. Note-se que a pressão auxiliar p^* corresponde à variação de pressão entre passos temporais, de modo que será nula nos pontos onde é imposta uma condição de Dirichlet (que aqui supomos não variar no tempo).

Finalmente, obtemos um sistema linear para cada linha da eq. 2.60:

$$\left(\frac{1}{\Delta t} \mathbf{M}_V^{\hat{V}} + \mathbf{C}_V^{\hat{V}} - \frac{1}{Re} \mathbf{K}_V^{\hat{V}} + \mathbf{D}_u \right) \cdot \mathbf{u}^* = \frac{1}{\Delta t} \mathbf{M}_V^{\hat{V}} \cdot \mathbf{u}_n - \mathbf{G}_Q^{\hat{V}} \cdot \mathbf{p} + \mathbf{d}_u \quad (2.71a)$$

$$\left(\mathbf{K}_Q^{\hat{Q}} + \mathbf{D}_p \right) \cdot \mathbf{p}^* = \frac{1}{\Delta t} \left(\mathbf{G}_{xV}^{\hat{Q}} + \mathbf{G}_{yV}^{\hat{Q}} \right) \cdot \mathbf{u}^* + \mathbf{d}_{p^*} \quad (2.71b)$$

$$\left(\mathbf{M}_V^{\hat{V}} + \mathbf{D}_u \right) \cdot \mathbf{u}_{n+1} = \mathbf{M}_V^{\hat{V}} \cdot \mathbf{u}^* - \Delta t \mathbf{G}_Q^{\hat{V}} \cdot \mathbf{p} + \mathbf{d}_u \quad (2.71c)$$

$$\mathbf{I} \cdot \mathbf{p}_{n+1} = \mathbf{p}_n + \mathbf{p}^* \quad (2.71d)$$

2.4.3 Cálculo dos coeficientes aerodinâmicos

O objetivo do Método de Elementos Finitos neste trabalho é calcular os coeficientes de arrasto, sustentação e momento para um dado aerofólio, o que equivale a calcular a força e o momento resultantes. O momento é calculado em torno do

ponto de 1/4 de corda, ou seja, o ponto:

$$\mathbf{x}_O = \left(\frac{D}{4}, y\left(\frac{D}{4}\right), 0 \right) \quad (2.72)$$

Onde y denota o camber do aerofólio e D a corda, como na seção 2.3.

A força que o fluido impõe sobre o aerofólio é superficial, ou seja, atua ao longo da superfície da peça, podendo ser representada por um tensor de tensões:

$$\sigma_{ij} = -p\delta_{ij} + \mu \frac{\partial u_i}{\partial x_j} + \mu \frac{\partial u_j}{\partial x_i} \quad (2.73a)$$

$$\mathbf{F} = \int_S \boldsymbol{\sigma} \cdot \mathbf{n} dS \quad (2.73b)$$

$$\mathbf{M} = \int_S (\mathbf{x} - \mathbf{x}_O) \times (\boldsymbol{\sigma} \cdot \mathbf{n}) dS \quad (2.73c)$$

Onde \mathbf{n} é um vetor unitário normal à superfície do aerofólio, orientado para o exterior deste.

Basta, portanto, obter o tensor $\boldsymbol{\sigma}$ ao longo do contorno do aerofólio e integrá-lo, já que o escoamento é bidimensional. Como os elementos empregados são triangulares, o contorno do aerofólio é aproximado por uma sequência de arestas lineares. Caso a malha seja suficientemente fina, podemos supor a tensão aproximadamente constante ao longo de cada aresta a , de modo que basta calcular $\boldsymbol{\sigma} \cdot \mathbf{n}$ no ponto médio de cada aresta ${}^a\mathbf{x}_m$. Para isso, note-se que, para uma aresta a contida no elemento e_ℓ , a eq. 2.62 permite escrever \mathbf{u} e p através das funções de forma M e N , de modo que:

$$\sigma_{ij}({}^a\mathbf{x}_m) = - \sum_{r=1}^3 {}^\ell p_r {}^\ell N_r({}^a\mathbf{x}_m) \delta_{ij} + \mu \sum_{q=1}^6 {}^\ell u_{iq} \frac{\partial}{\partial x_j} {}^\ell M_q({}^a\mathbf{x}_m) + \mu \sum_{q=1}^6 {}^\ell u_{jq} \frac{\partial}{\partial x_i} {}^\ell M_q({}^a\mathbf{x}_m) \quad (2.74)$$

Assim:

$$\mathbf{F} \approx \sum_a s_a \boldsymbol{\sigma}_{ij}({}^a\mathbf{x}_m) \cdot \mathbf{n} \quad (2.75a)$$

$$\mathbf{M} \approx \sum_a ({}^a\mathbf{x}_m - \mathbf{x}_O) \times (s_a \boldsymbol{\sigma}_{ij}({}^a\mathbf{x}_m) \cdot \mathbf{n}) \quad (2.75b)$$

Onde s_a é o comprimento da aresta a .

Com isso, calcula-se os coeficientes de arrasto, sustentação e momento, conforme

foram definidos na seção 2.2.1:

$$c_D = 2 \frac{\mathbf{F} \cdot \mathbf{n}_x}{\rho U^2 D} \quad (2.76a)$$

$$c_L = 2 \frac{\mathbf{F} \cdot \mathbf{n}_y}{\rho U^2 D} \quad (2.76b)$$

$$c_M = 2 \frac{\mathbf{M} \cdot (-\mathbf{n}_z)}{\rho U^2 D^2} \quad (2.76c)$$

O sinal negativo no coeficiente de momento diz respeito à orientação escolhida: o momento é positivo no sentido horário e negativo no sentido anti-horário.

2.4.4 Implementação

O código para montagem do problema, resolução das equações e posterior visualização dos resultados foi desenvolvido em linguagem **Python**, em função da grande flexibilidade e disponibilidade de bibliotecas auxiliares nessa linguagem de programação. Todo o código foi desenvolvido exclusivamente utilizando bibliotecas de código aberto.

A cada passo de tempo, deve-se resolver os sistemas lineares de cada uma das eqs. 2.71. Uma característica vantajosa desses sistemas é que os mesmos são compostos de matrizes esparsas, que podem ser armazenadas na memória com um volume pequeno, além de que, através de algoritmos especializados, os sistemas lineares esparsos podem ser resolvidos rapidamente. Neste trabalho, foi utilizado o módulo **scipy.sparse**, da biblioteca de código aberto **scipy** (38), com o objetivo de criar as matrizes esparsas e em seguida resolver os sistemas lineares correspondentes.

O método depende da geração de uma malha de elementos triangulares que cubra toda a área do escoamento. Para produzir essa malha, foi utilizado o modelo computacional **Gmsh**, uma ferramenta *open-source* desenvolvida para a geração de malhas de elementos finitos em problemas uni, bi ou tridimensionais (39). Esse programa é chamado dentro do código Python através de uma API (*Application Programming Interface*).

A execução do programa para resolução de qualquer problema de escoamento depende da definição de 3 parâmetros:

- Passo de tempo Δt
- Tamanho do elemento de malha h , passado como argumento para o **Gmsh**

- Tempo máximo de escoamento T . A cada 10 iterações do passo de tempo, caso a variação da velocidade e da pressão seja inferior a 10^{-3} , considera-se que foi atingido o regime estacionário e a execução termina. Caso contrário, repete-se o processo até chegar ao tempo T

Quanto aos casos selecionados para dar como entrada à rede neural, foi realizada uma amostragem aleatória de valores dos parâmetros geométricos do aerofólio, do ângulo de ataque e do número de Reynolds. Como estamos trabalhando apenas com adimensionais, pode-se definir a viscosidade e o comprimento de corda do aerofólio como unitários, de modo que $Re = U$.

Considerando que o objetivo de aplicação do trabalho é modelar as pás de aerogeradores, que estão expostas a condições estocásticas de vento, consideramos que a velocidade (e, portanto, Re) seguirá uma distribuição de Weibull, comumente empregada para modelagem de vento (40), definida pela seguinte distribuição de probabilidade acumulada:

$$F(U) = 1 - \exp\left(-\frac{U^k}{c}\right) \quad (2.77)$$

Onde apenas valores positivos da velocidade são admitidos, c é dito *fator de escala*, aqui considerado $c = 100$, e k é dito *fator de forma*, aqui considerado $k = 2$. Já para a direção do vento, devido à predominância de sistemas de controle de *yaw* e *pitch* nos aerogeradores, consideramos que a variação de direção será relativamente contida, seguindo uma distribuição uniforme entre -15° e 15° .

O parâmetro de camber m foi tirado de modo que 20% dos casos têm camber nulo, ou seja, correspondem a aerofólios simétricos, e os demais seguem uma distribuição uniforme entre $m = 0\%$ e $m = 10\%$. A espessura T e a posição de camber p seguem também distribuições uniformes: $5\% \leq T \leq 30\%$ e $10\% \leq p \leq 90\%$.

Para a produção dos casos de treinamento, foram utilizadas 10 máquinas virtuais do tipo t3.micro, disponíveis através do serviço de computação em nuvem AWS¹. Cada uma das máquinas possui um processador de geração atual, que possui 1 núcleo com capacidade de 2 *threads*, e uma frequência de operação de 2,5 GHz, com acesso a 1 GB de memória. Desse modo, as máquinas virtuais foram utilizadas por um período de cerca de 4 dias, para gerar 12390 casos de treinamento, em cada um dos quais foram calculados os coeficientes de sustentação, arrasto e momento.

¹Amazon Web Services. Mais informações disponíveis em <https://aws.amazon.com/pt/ec2/>

2.5 Rede Neural

Uma rede neural, no sentido mais amplo, é um modelo matemático que consiste, de forma abstrata, em um conjunto de neurônios, cada um dos quais recebe um conjunto de entradas, nas quais é aplicada uma transformação linear, e produz uma única saída, que de modo geral será modificada por uma função de ativação (3).

Tornando a definição um pouco mais específica, neste trabalho são empregadas redes *feed-forward* densamente conectadas. Este tipo de rede é formado por uma sequência de camadas de neurônios, nas quais cada neurônio recebe como entrada a saída de todos os neurônios da camada anterior. Uma representação esquemática desse tipo de rede é exibida na figura 2.5. A camada mais à esquerda é chamada *camada de entrada*, e recebe como entrada um vetor \mathbf{x} de argumentos, ao passo que a camada à direita é a *camada de saída*, que produz um vetor \mathbf{y} correspondente à saída do modelo. Todas as camadas intermediárias são chamadas *camadas ocultas*, por não estarem diretamente "visíveis" na aplicação do modelo.

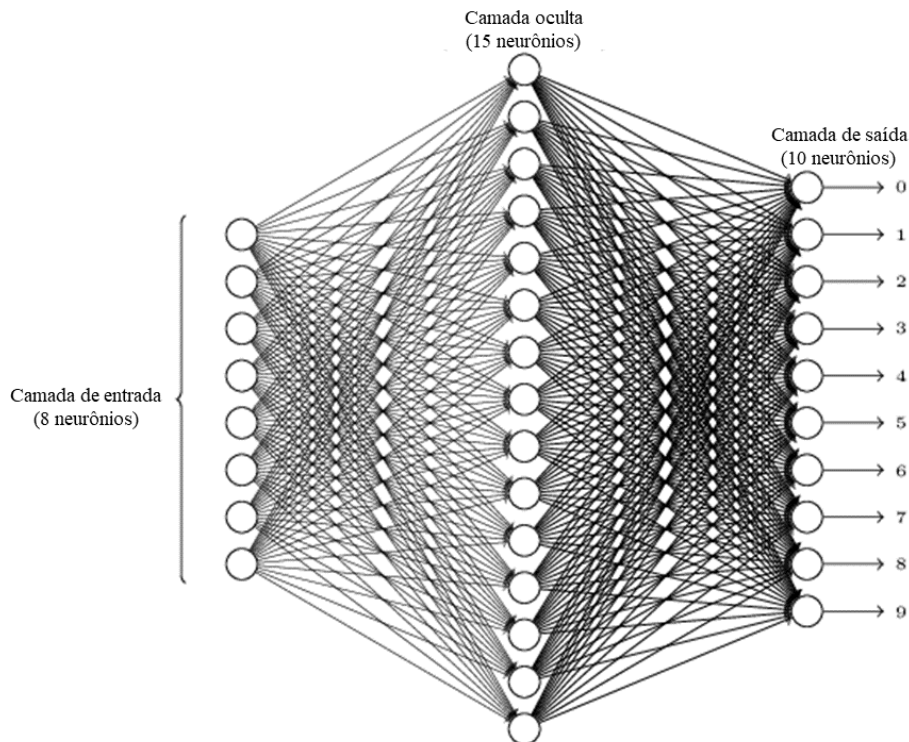


Figura 2.5: Exemplo de rede neural *feed-forward* com uma única camada oculta.
Fonte: adaptado de (3).

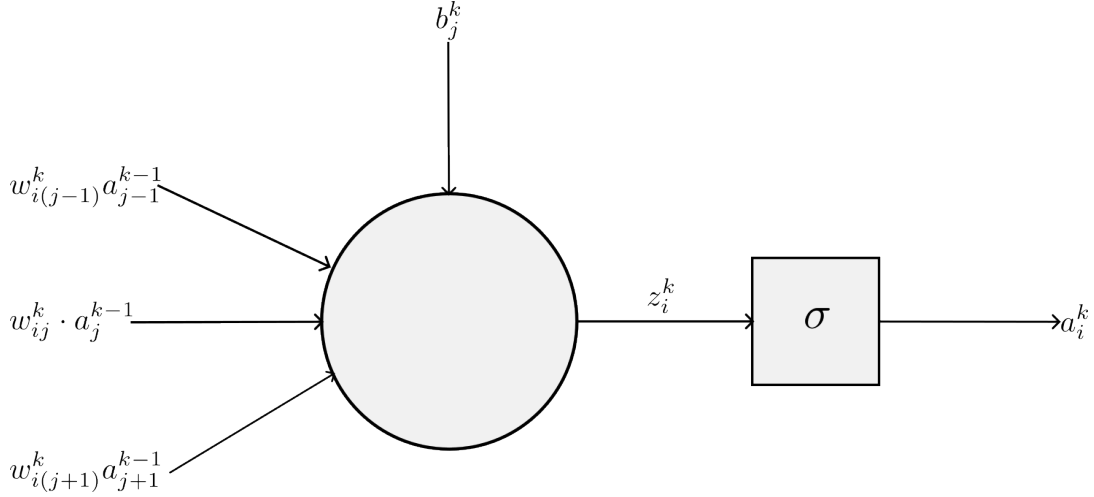


Figura 2.6: Representação do i -ésimo neurônio da k -ésima camada oculta de uma rede neural. À saída do neurônio, é aplicada uma função de ativação σ . (repetição da figura 1.4)

A figura 2.6 mostra que cada camada da rede é equivalente à aplicação de uma matriz de pesos \mathbf{W}^k ao vetor \mathbf{a}^{k-1} , acrescida da soma de um vetor de viés \mathbf{b}^k , produzindo o vetor \mathbf{z}^k . Em seguida, é aplicada uma função de ativação não-linear σ , produzindo a saída da camada \mathbf{a}^k . Escrito na forma de equação, para uma rede de L camadas ocultas:

$$\mathbf{a}^0 = \mathbf{x} \quad (2.78a)$$

$$\mathbf{a}^k = \sigma(\mathbf{z}^k) = \sigma(\mathbf{W}^k \cdot \mathbf{a}^{k-1} + \mathbf{b}^k) \quad (2.78b)$$

$$\mathbf{y} = \mathbf{a}^L = \sigma^S(\mathbf{W}^L \cdot \mathbf{a}^{L-1} + \mathbf{b}^L) \quad (2.78c)$$

Onde a função de ativação da saída é denotada por σ^S , uma vez que essa função tem caráter especial ao definir a forma da saída, de modo que comumente se emprega uma função diferente daquela usada para as camadas ocultas.

A principal vantagem das redes neurais é sua expressividade, ou universalidade: qualquer função $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ pode ser aproximada por uma rede neural de uma única camada oculta, desde que seu número de neurônios seja suficientemente grande (3). Essa universalidade naturalmente se estende à rede multicamadas, que pode, em tese, obter maior expressividade do que uma rede de uma camada única com o mesmo número de parâmetros.

As componentes da matriz \mathbf{W}^k e do vetor \mathbf{b}^k são parâmetros da rede. A cada camada, se a k -ésima camada tem n neurônios e a $(k+1)$ -ésima tem m , tem-se $m \cdot n$

pesos e m vieses. Mesmo para redes de porte relativamente pequeno, não é viável definir manualmente esses parâmetros. A grande vantagem dos métodos de *machine learning*, ou aprendizado de máquina, nos quais a rede neural se insere, é a possibilidade de calibrar o modelo automaticamente a partir dos dados disponíveis. Por isso, os pesos e os vieses são chamados de *parâmetros treináveis* da rede. Como veremos a seguir, a construção da rede neural fornece uma forma computacionalmente eficiente de treinar os parâmetros.

2.5.1 Descida estocástica do gradiente

A otimização dos parâmetros da rede é feita usualmente por descida estocástica do gradiente. No método de descida do gradiente convencional, dada uma função custo f e um vetor de argumentos \mathbf{w} , realizamos um processo iterativo partindo de um valor inicial \mathbf{w}_0 e, a cada passo, fazendo:

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \eta \nabla f(\mathbf{w}_i) \quad (2.79)$$

Onde η é a *taxa de aprendizado*, um parâmetro que determina quão agressiva será a otimização. Assim, a cada passo, avançamos na direção de maior diminuição da função custo, até chegar a um mínimo local, quando o gradiente se torna nulo e o processo termina.

No caso da descida do gradiente estocástico, calculamos a função custo da rede em função apenas de uma amostra aleatória dos dados de entrada, e fazemos um único passo de descida do gradiente com base nos resultados dessa amostra (*batch*, ou *lote*). Assim, se utilizamos uma amostra contendo n elementos x_1, \dots, x_n , e a função custo f é dada em função de x e \mathbf{w} , temos a expressão para o p -ésimo passo de treinamento:

$$g_p(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^n f(x_i, \mathbf{w}) \quad (2.80a)$$

$$\mathbf{w}_{p+1} = \mathbf{w}_p - \eta \nabla g_p(\mathbf{w}_p) \quad (2.80b)$$

Se a massa total de dados de teste corresponde a N entradas, repetimos o passo acima $\frac{N}{n}$ vezes sem repetir nenhuma amostra x_i até esgotar o conjunto de dados, o que configura uma *época* de treinamento. O processo se repete por um número

pré-definido de épocas, ou até que o desempenho da rede se estabilize, na estratégia chamada de *parada antecipada*.

Na descida estocástica do gradiente, garante-se apenas que, *na média*, um número suficientemente grande de iterações caminha na direção da diminuição global da função custo. Esse método, contudo, permite escapar de certos mínimos locais, em função do ruído introduzido pela amostragem aleatória de elementos a cada passo (41), além de ser computacionalmente mais eficiente que sua contraparte determinística (42).

No presente trabalho, empregou-se o método *Adam*, um algoritmo de descida do gradiente estocástica que se baseia em estimativas sucessivas e adaptativas dos momentos estatísticos do gradiente através do cálculo de suas médias móveis (43).

2.5.2 Retropropagação

O cálculo do gradiente da função custo em relação aos parâmetros da rede é feito através da retropropagação, uma forma de diferenciação automática na qual o gradiente é propagado desde a camada de saída até as primeiras camadas ocultas, por meio da aplicação sucessiva da regra da cadeia (18).

Retomando a notação da eq. 2.78, e dada uma função custo $f(\mathbf{y})$, pode-se calcular a derivada do custo em relação à saída dos neurônios da última camada oculta, denotado por δ^L : (3)

$$\delta_j^L = \frac{\partial f}{\partial z_j^L} = \frac{d\sigma^S}{dz}(z_j^L) \frac{\partial f}{\partial y_j} \quad (2.81)$$

Já a derivada do custo em relação às saídas da camada k é:

$$\delta_j^k = \frac{\partial f}{\partial z_j^k} = \sum_i \frac{\partial a_j^k}{\partial z_j^k} \cdot \frac{\partial z_i^{k+1}}{\partial a_j^k} \cdot \frac{\partial f}{\partial z_i^{k+1}} = \sum_i \frac{d\sigma}{dz}(z_j^k) \cdot w_{ij}^{k+1} \cdot \delta_i^{k+1} \quad (2.82)$$

Seguindo a mesma linha de raciocínio, se calcula as derivadas em relação aos pesos e vieses:

$$\frac{\partial f}{\partial w_{ij}} = \frac{\partial z_i^k}{\partial w_{ij}} \cdot \frac{\partial f}{\partial z_i^k} = a_j^{k-1} \cdot \delta_i^k \quad (2.83)$$

$$\frac{\partial f}{\partial b_j^k} = \frac{\partial z_j^k}{\partial b_j^k} \cdot \frac{\partial f}{\partial z_j^k} = \delta_j^k \quad (2.84)$$

Dessa forma, é possível calcular os termos δ de forma retrógrada, começando pela camada final e sucessivamente aplicando a equação 2.82 para obter os valores de δ

de cada neurônio. Em seguida, aplicamos as equações 2.83 e 2.84 para encontrar o gradiente da função custo em relação a todos os pesos e vieses, respectivamente. A informação do gradiente é passada para o algoritmo de descida do gradiente (*Adam*, nesse caso), e utilizada para gradativamente ajustar os pesos e vieses para minimizar a função custo, o que equivale a treinar a rede para melhor responder aos dados.

2.5.3 Aplicação

Conforme observado nas seções 2.2 e 2.3.3, os coeficientes de sustentação, arrasto e momento de um aerofólio da série NACA de 4 dígitos são função dos parâmetros geométricos (camber máximo, posição de camber máximo e espessura máxima), do ângulo de ataque e do número de Reynolds. Sendo assim, a rede neural deverá modelar uma função:

$$g : [\beta_1, \beta_2, \beta_3, \alpha, Re] \longrightarrow [c_L, c_D, c_M] \quad (2.85)$$

Desse modo, a camada de entrada deverá ter obrigatoriamente 5 neurônios, e a camada de saída deverá ter 3 neurônios. A quantidade de camadas escondidas e o número de neurônios em cada uma delas podem ser livremente definidos, e são hiperparâmetros, que devem ser ajustados modificando a rede iterativamente e observando seu desempenho, como será feito na seção 3.2.1.

Para a função custo, escolhemos o erro quadrático médio, que penaliza a distância quadrática entre o valor real e estimado da saída da rede. Para um lote de n dados:

$$f(\mathbf{x}, \mathbf{w}) = \frac{1}{n} \sum_{i=1}^n ((c_{Li} - \hat{c}_L(x_i))^2 + (c_{Di} - \hat{c}_D(x_i))^2 + (c_{Mi} - \hat{c}_M(x_i))^2) \quad (2.86)$$

Onde o sobrescrito $\hat{}$ denota o valor estimado pela rede. Com isso, o gradiente do custo em relação a um dos valores de saída, digamos aquele correspondente a c_L , é:

$$\frac{\partial f}{\partial \hat{c}_L} = 2(\hat{c}_L - c_L) \quad (2.87)$$

Para que seja possível calcular os gradientes, resta apenas definir a função de ativação. Nas camadas ocultas, é recomendado usar funções de ativação simétricas em torno de zero, como a função $\tanh z$, exibida na figura 2.7, pois sua convergência é mais rápida (42). Assim, essa será a função utilizada nas camadas ocultas. Já para a camada de saída, queremos que os neurônios sejam capazes de produzir os

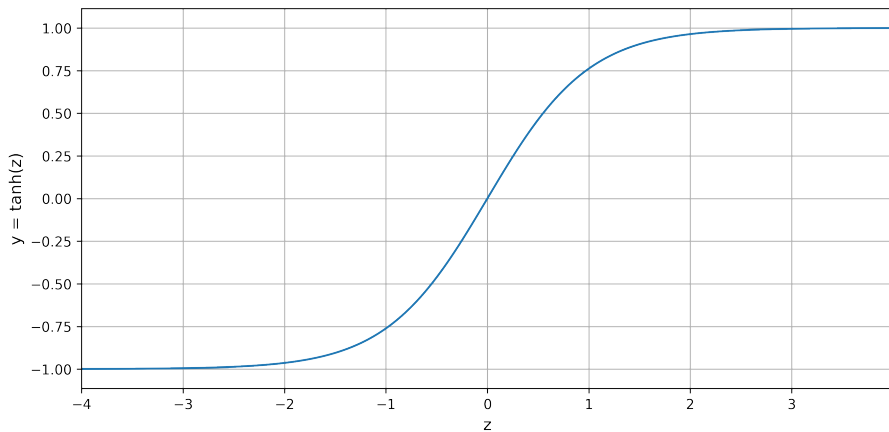


Figura 2.7: Função $y = \tanh z$

valores dos coeficientes dinâmicos. Na ausência de qualquer limitação *a priori* ou relação entre os coeficientes, queremos que a imagem da rede seja o \mathbb{R}^3 , por isso não empregamos função de ativação na saída ($\sigma^S = id$).

Uma dificuldade comumente enfrentada em aprendizado de máquina é o fenômeno denominado *overfitting*, ou *sobreajuste*, no qual a rede se especializa para maximizar o desempenho diante dos dados de treino, aprendendo suas peculiaridades, mas sem necessariamente ser capaz de compreender o comportamento geral do fenômeno real, de modo que o desempenho da rede diante de um dado inédito será inferior ao esperado. A fim de evitar essa dificuldade, dividimos os dados em 3 conjuntos:

- 50% de dados de treino
- 25% de dados de validação
- 25% de dados de teste

Nessa separação, apenas os dados de treino são usados para a descida do gradiente e aprendizado dos parâmetros. Os dados de validação são usados para selecionar os hiperparâmetros da rede: se houver *overfitting*, isso será identificado como uma piora no desempenho da rede para os dados de validação, ao mesmo tempo que se observa uma melhora para os dados de treino. Já os dados de teste são considerados apenas ao fim da calibração da rede, e sua função é testar o desempenho do modelo em um conjunto de dados completamente inédito, que não influenciou o treinamento

nem a escolha de parâmetros, o que se assemelha à situação que a rede encontrará na sua aplicação prática.

O modelo de rede neural, bem como todas as estratégias aqui definidas, é implementado através da biblioteca **keras**, voltada para o desenvolvimento de redes neurais em Python. Essa biblioteca apresenta como principais vantagens a sua alta eficiência computacional, que permite acelerar o aprendizado através de uma placa de vídeo (GPU), caso esteja disponível; e a grande variedade de ferramentas de rede neural já implementadas, o que permite simplificar significativamente o código escrito para esse trabalho.

A rede foi treinada e testada em um computador pessoal dotado de processador Intel Core i7 7500-U, com 4 núcleos operando à frequência de 3,3 GHz e placa de vídeo Nvidia GeForce 920MX, além de 8 GB de memória.

2.5.4 Estratégias de treino

No treinamento de uma rede neural, ou de qualquer algoritmo de aprendizado de máquina, deve-se tomar cuidados especiais para evitar o *overfitting*. Na literatura referente a redes neurais, encontram-se algumas metodologias de treinamento que podem garantir uma melhor generalização do modelo, o que pode ser verificado pela melhora do desempenho nos dados de validação. Há também estratégias que podem ampliar o desempenho da rede de forma geral, ou que reduzem o tempo computacional. Neste trabalho, são empregados alguns desses métodos, discutidos a seguir.

Parada antecipada

Conforme o treinamento da rede avança, chega-se a um ponto em que a função custo tende a estagnar: nesse momento, a rede já está efetivamente treinada. Caso se continue a ajustar os parâmetros da rede, o único efeito será um sobreajuste da rede aos dados de treino, ao passo que o desempenho do modelo nos dados de validação se mantém constante ou, no pior dos casos, aumenta, ou seja, o modelo se torna menos funcional em dados novos, desperdiçar tempo computacional com esse treinamento.

Para evitar essa patologia, se aplica o método de parada antecipada (44): A cada

época de treino, o programa verifica se a função custo nos *dados de validação* teve uma diminuição apreciável ao longo das últimas ν épocas. Senão, o treinamento é interrompido imediatamente. Uma vantagem adicional desse critério é que não há necessidade de estabelecer previamente um número de épocas de treinamento. Em vez disso, define-se apenas um número máximo (aqui considerado 100 épocas), e, caso o treinamento fique estagnado antes disso, ele será automaticamente interrompido.

O número de épocas n antes da parada, bem como a redução esperada do erro ξ nessas ν épocas, são hiperparâmetros, que podem ser definidos levando em consideração a função custo nos dados de validação e o tempo computacional total. Como os resultados finais não são muito sensíveis a esses hiperparâmetros, eles foram definidos após um processo de tentativa e erro:

$$\nu = 5 \tag{2.88a}$$

$$\xi = 10^{-4} \tag{2.88b}$$

Regularização

A estratégia de *decaimento de pesos*, ou *regularização L2*, apresenta uma outra forma de evitar o sobreajuste. Neste método, o módulo de cada peso w_{ij}^k da rede é penalizado, sendo agregado à função custo. Assim, o aumento de um determinado peso só ocorrerá se ocasionar uma redução suficiente da função custo, o que tende a evitar que a rede incorpore o ruído dos dados de treino.

Numericamente, se f é a função custo original, ela deve ser substituída por f' . Se \mathbf{b}^k e \mathbf{W}^k são respectivamente o vetor dos vieses e a matriz de pesos na k -ésima camada, e \mathbf{x} é o vetor de entrada da rede em um determinado passo de treinamento:

$$f'(\mathbf{x}, \mathbf{W}^1, \dots, \mathbf{W}^L, \mathbf{b}^1, \dots, \mathbf{b}^L) = f(\mathbf{x}, \mathbf{W}^1, \dots, \mathbf{W}^L, \mathbf{b}^1, \dots, \mathbf{b}^L) + \frac{\lambda}{2} \sum_{i,j,k} |w_{ij}^k|^2 \tag{2.89}$$

Essa abordagem introduz o hiperparâmetro λ , chamado de *parâmetro de regularização*, cujo valor ideal depende da arquitetura da rede e não pode ser determinado *a priori*. Por isso, ele será definido através dos dados de validação na seção 3.2.1.

Redução progressiva da taxa de aprendizado

O algoritmo Adam, usado para descida estocástica do gradiente, tem como parâmetro a taxa de aprendizado η . No início do treinamento, um valor elevado de η significa uma rápida evolução do desempenho. Contudo, conforme os parâmetros da rede se aproximam de um ponto de mínimo (local ou global), uma taxa de aprendizado muito alta amplifica o ruído aleatório da descida do gradiente, e pode impedir que nos aproximemos desse mínimo, já que isso requer um ajuste fino dos parâmetros.

A fim de possibilitar esse ajuste fino, a taxa de aprendizado é reduzida gradualmente a cada época de treino, de modo que iniciamos o processo com uma taxa elevada e um treinamento agressivo, mas torna-se mais sutil à medida que se progride no treinamento.

Se η_i é a taxa de aprendizado na i -ésima época de treino, fazemos:

$$\eta_{i+1} = \begin{cases} \gamma \cdot \eta_i, & \text{se } i \geq 10 \\ \eta_0, & \text{se } i < 10 \end{cases} \quad (2.90)$$

Tanto a taxa inicial de aprendizado η_0 quanto a taxa de decaimento γ são hiperparâmetros, e sua escolha será feita com base nos dados de validação, na seção 3.2.1.

Capítulo 3

Resultados

3.1 Método de Elementos Finitos

3.1.1 Validação do Método de Elementos Finitos

O código de elementos finitos desenvolvido nesse trabalho se baseia nas equações de Navier-Stokes e segue uma metodologia já bem estabelecida na literatura. Porém, antes de começar a empregá-lo para gerar uma base de dados de treinamento para a rede neural, é necessário garantir seu bom funcionamento. Para tal, reproduzimos certos escoamentos canônicos, cujo resultado esperado já é conhecido, seja por considerações teóricas, seja por experimentos numéricos ou empíricos. Após verificar que o resultado obtido por nosso modelo apresenta bom acordo com experiências anteriores, será possível considerá-lo válido.

Problema de Poiseuille

O escoamento bidimensional de Poiseuille é definido como o escoamento estacionário do fluido através de um duto de largura constante, igual a D , e comprimento L . Nesse problema, é possível relacionar a perda de carga (queda de pressão) e a vazão volumétrica. Consideramos que o fluido entra no duto com perfil de velocidades uniforme, igual a $U\mathbf{n}_x$, e sem gradiente de pressão, e que o duto é suficientemente longo para que o escoamento esteja completamente desenvolvido, de modo que, na saída, $\frac{\partial \mathbf{u}}{\partial x} = 0$. Na saída, a pressão é considerada constante e igual a zero, embora o valor absoluto da pressão não interfira de todo no resultado, já que

o escoamento é incompressível, ou seja, apenas a pressão relativa e seu gradiente influenciam no movimento. Nas paredes, a condição de não-deslizamento impõe velocidade nula, e considera-se que o gradiente de pressão na direção normal à parede também será nulo. Tomando U e D como velocidade e comprimento característicos, respectivamente, as condições de contorno na forma adimensional ficam:

$$\mathbf{u}(0, y) = 1\mathbf{n}_x \quad (3.1a)$$

$$\mathbf{u}(x, 0) = \mathbf{u}(x, 1) = 0 \quad (3.1b)$$

$$\frac{\partial \mathbf{u}}{\partial x} \left(\frac{L}{D}, y \right) = 0 \quad (3.1c)$$

$$p \left(\frac{L}{D}, y \right) = 0 \quad (3.1d)$$

$$\frac{\partial p}{\partial x}(0, y) = 0 \quad (3.1e)$$

$$\frac{\partial p}{\partial y}(x, 0) = \frac{\partial p}{\partial y}(x, 1) = 0 \quad (3.1f)$$

O perfil de velocidade do escoamento desenvolvido possui solução analítica. Começamos observando que, se o perfil é completamente desenvolvido, então $\frac{\partial u_x}{\partial x} = \frac{\partial u_y}{\partial x} = 0$. Então, da equação de continuidade 2.17a, tem-se:

$$\frac{\partial u_x}{\partial x} = -\frac{\partial u_y}{\partial y} = 0 \quad (3.2)$$

Mas, como sabemos que $u_y = 0$ nas paredes, então a velocidade vertical deve ser nula ao longo de todo o perfil. O termo convectivo da eq. 2.17b também desaparece:

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = u_x \frac{\partial \mathbf{u}}{\partial x} + u_y \frac{\partial \mathbf{u}}{\partial y} = u_x \cdot 0 + 0 \cdot \frac{\partial \mathbf{u}}{\partial y} = 0 \quad (3.3)$$

Assim, na região de escoamento desenvolvido, tem-se:

$$\nabla p = \frac{1}{Re} \nabla^2 \mathbf{u} = \frac{1}{Re} \frac{\partial^2 u_x}{\partial y^2} \mathbf{n}_x \quad (3.4)$$

Como $p = 0$ ao longo de toda a superfície de saída, resulta que o gradiente de pressão será constante em todo o perfil, $\nabla p = G\mathbf{n}_x$. Assim, integrando a eq. 3.4:

$$\frac{1}{Re} u_x(y) = \frac{G}{2} y^2 + c_1 y + c_2 \quad (3.5)$$

Notando que $u_x(0) = u_x(1) = 0$, resulta $c_1 = -\frac{G}{2}$. Finalmente, notando que, pela conservação de massa, a vazão deve ser constante e igual à da entrada:

$$\int_0^1 u_x(y) dy = \int_0^1 \frac{ReG}{2} (y^2 - y) dy = 1 \implies G = \frac{-12}{Re} \quad (3.6)$$

$$u_x(y) = 6(y - y^2) \quad (3.7)$$

$$p(x) = \frac{12}{Re} \left(\frac{L}{D} - x \right) \quad (3.8)$$

Com isso, simulamos o problema computacionalmente com a ferramenta de elementos finitos desenvolvida, fazendo $Re = 50$, $L = 10D$, a fim de comparar nossos resultados com o perfil analítico. Os resultados são representados graficamente nas figuras 3.1.

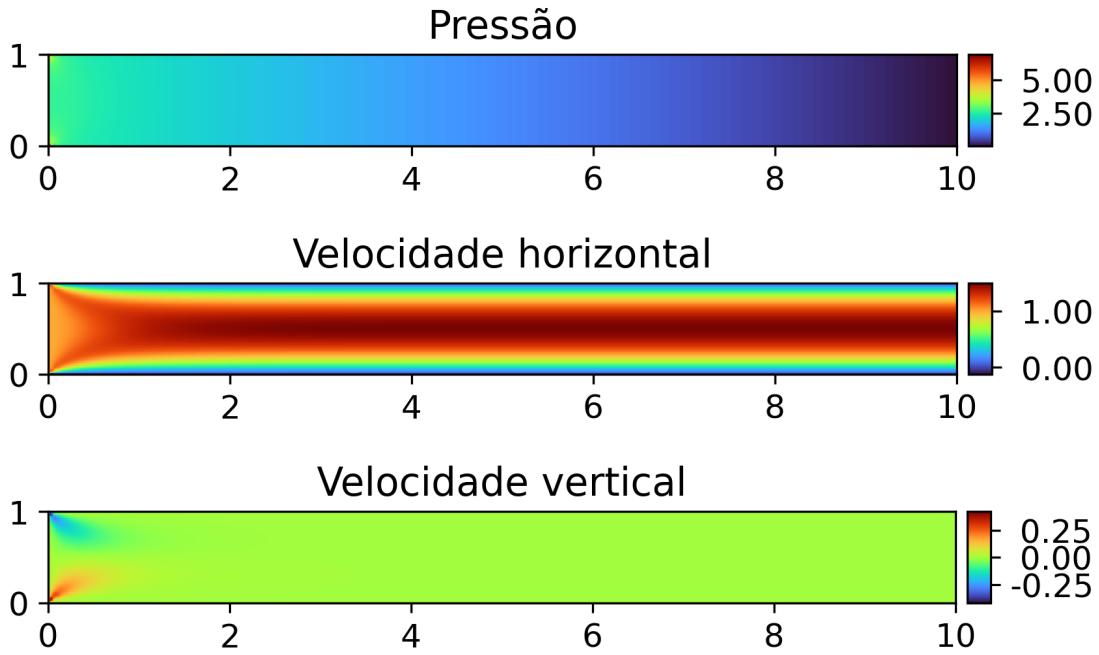


Figura 3.1: Resultados do escoamento de Poiseuille, simulado com o Método de Elementos Finitos, para $Re = 50$, com $h = 0.05$, $\Delta t = 0.01$, $T = 5$.

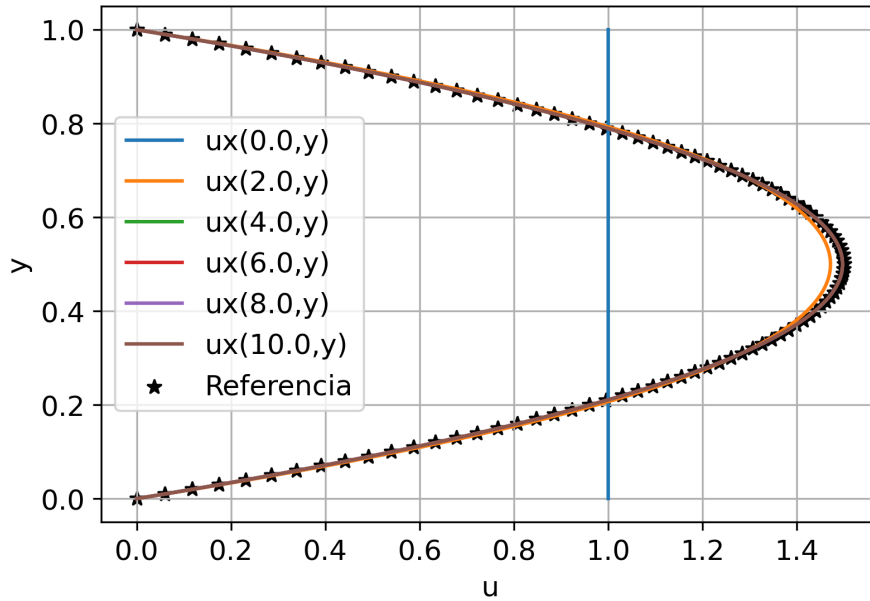


Figura 3.2: Perfil vertical de velocidade ao longo do duto, comparado com a solução analítica para o perfil desenvolvido. Note-se que as curvas a partir de $x = 4$ se sobrepõem, porque o perfil já se estabilizou.

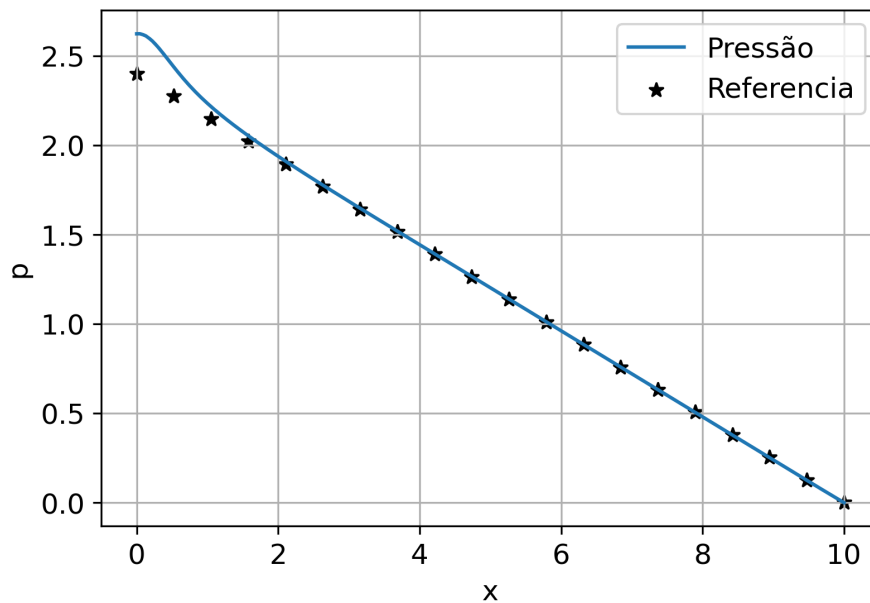


Figura 3.3: Perfil pressão ao longo do duto, comparado com a solução analítica para o perfil desenvolvido

Conforme se vê nas figuras 3.2 e 3.3, conforme nos aproximamos da saída do duto, a velocidade e a pressão se aproximam da solução analítica, que é o valor de referência. Esse resultado é uma evidência inicial do bom funcionamento do código,

contudo é importante notar que se trata de um caso simples, e no qual não há efeitos convectivos. No caso a seguir, os efeitos convectivos e o número de Reynolds se tornam importantes.

Cavidade

O problema de escoamento no interior de uma cavidade é computacionalmente bem mais complexo do que o de Poiseuille. Não apenas o termo convectivo se torna importante, mas também ocorre a formação de vórtices próximo às paredes. Por isso, esse escoamento é uma boa forma de validar um código computacional. O escoamento é caracterizado por uma cavidade quadrada fechada, cuja tampa se move, transferindo quantidade de movimento ao fluido por meio do cisalhamento. A figura 3.4 exibe uma representação das condições desse escoamento.

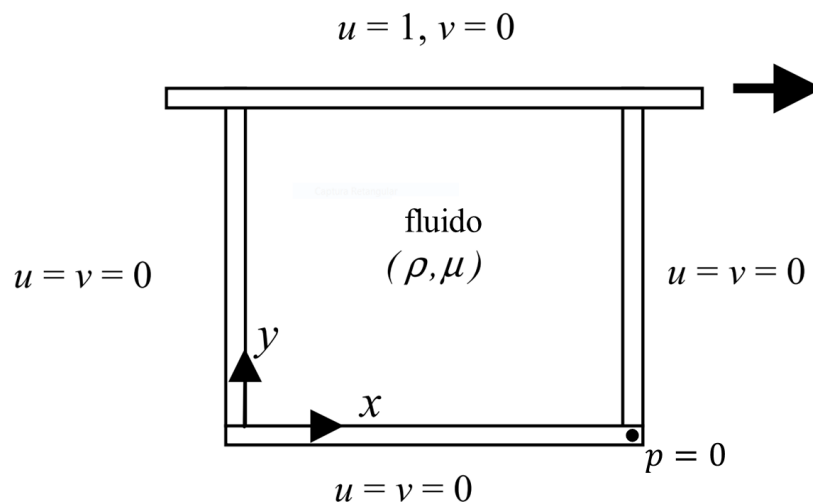


Figura 3.4: Representação do escoamento em uma cavidade quadrada. Fonte: adaptado de (4)

O problema da cavidade não possui solução analítica, por isso faz-se necessário validar nosso código contra resultados de simulação precisos, disponíveis em (4). Essa referência realizou uma simulação de volumes finitos com uma malha estruturada 1024×1024 , obtendo resultados precisos. Nas figuras 3.5 e 3.6, comparamos os valores de velocidade obtidos pelo nosso código ($h = 0,01, \Delta t = 0,01, T = 30$) com os da referência ao longo de uma linha vertical (em $x = 0,5$) e outra horizontal (em $y = 0,5$). Observa-se um bom acordo entre ambos, que começam a divergir apenas em números de Reynolds mais altos, entre $Re = 400$ e $Re = 1000$.

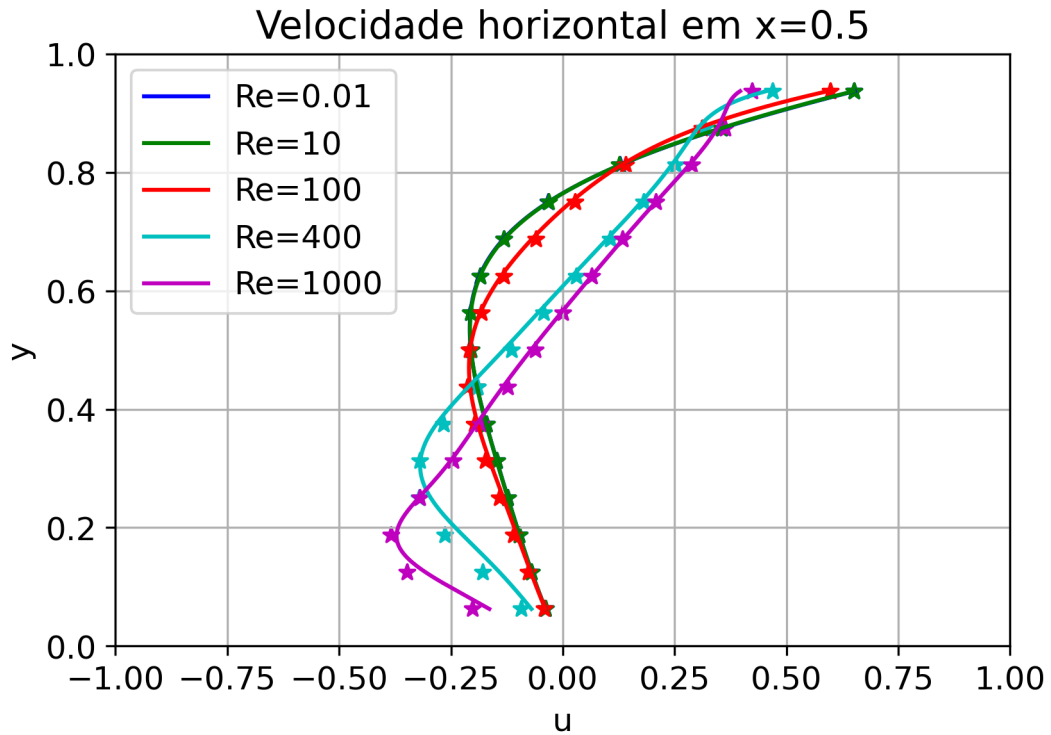


Figura 3.5: Perfil de velocidade horizontal em $x=0,5$ obtido pelo código para diversos números de Reynolds, e comparação com a literatura (4). Os valores para $Re = 0,01$ e $Re = 10$ se sobrepõem, tanto na referência quanto na saída do código.

A figura 3.7 exhibe um sumário dos resultados obtidos para $Re = 100$. Deve-se observar que a pressão atinge valores mais altos em torno dos vértices superiores, o que ocorre em função da descontinuidade da condição de contorno da velocidade, que é igual a 1 na parede superior, mas nula nas paredes laterais.

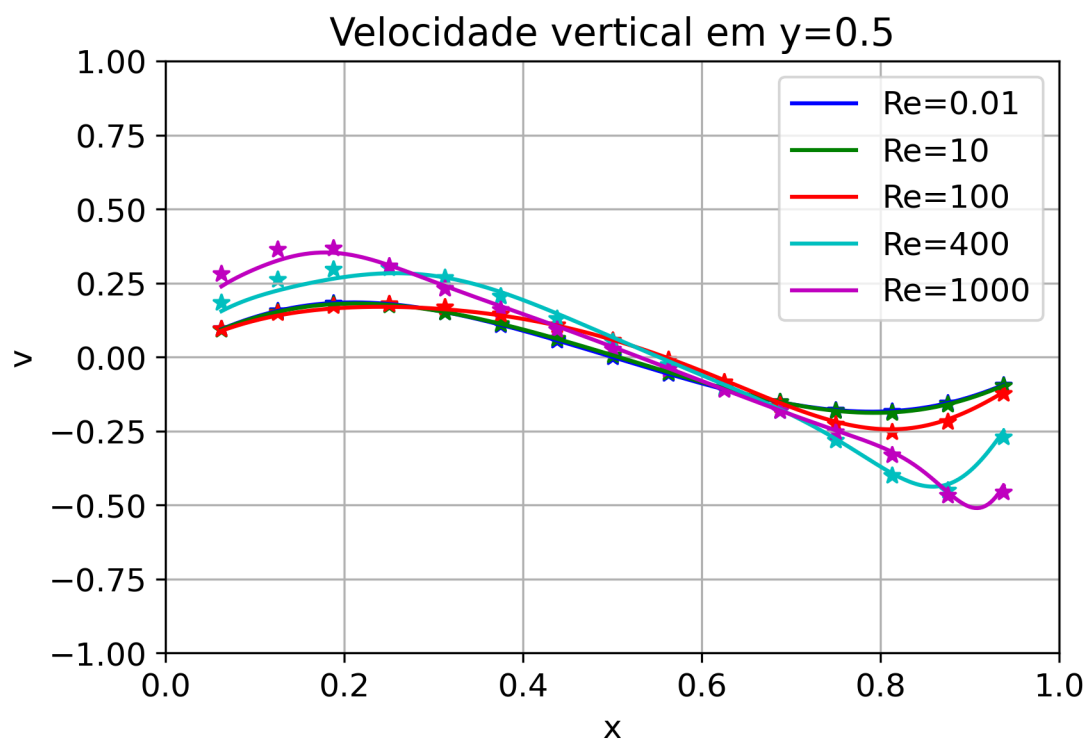


Figura 3.6: Perfil de velocidade vertical em $y=0,5$ obtido pelo código para diversos números de Reynolds, e comparação com a literatura (4). Os valores para $Re = 0,01$ e $Re = 10$ se sobrepõem, tanto na referência quanto na saída do código.

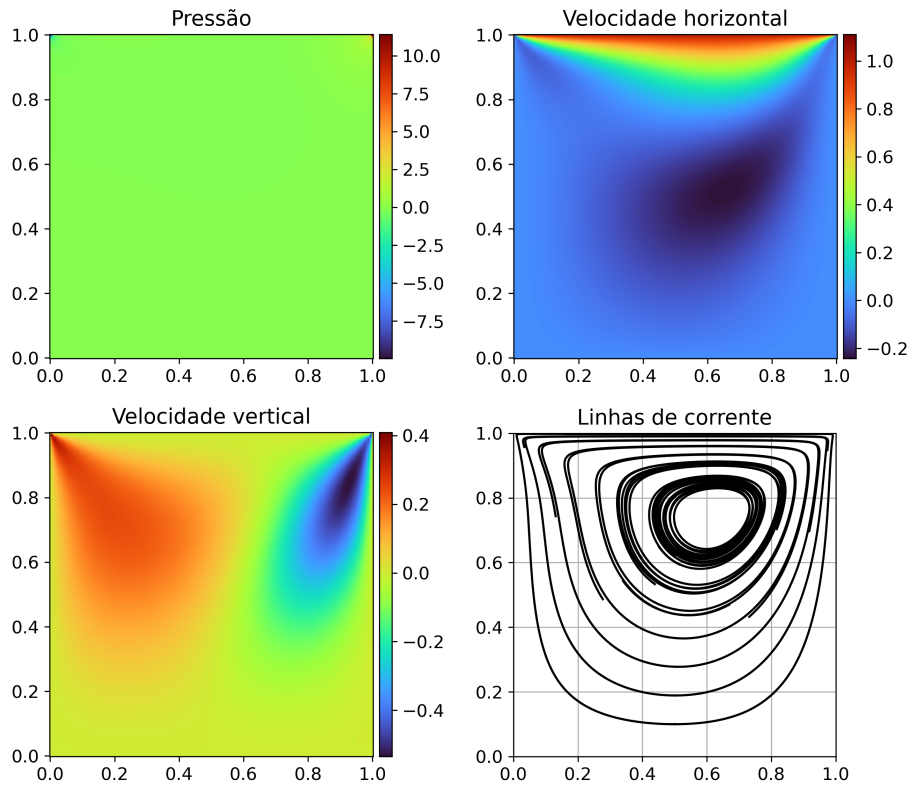


Figura 3.7: Resultados do escoamento no interior da Caverna, simulado com o Método de Elementos Finitos, para $Re = 100$, com $h = 0.01$, $\Delta t = 0.01$, $T = 30$.

Degrau

A última validação a ser feita é o escoamento em degrau (*backward-facing step flow*), no qual partimos de um escoamento desenvolvido no interior de um canal, que se expande rapidamente para um canal de maior largura. Nesse cenário, espera-se que ocorra recirculação na base do degrau, e que a jusante deste o perfil volte ao comportamento parabólico de um escoamento desenvolvido. Como referência para este caso, usamos o resultado de experimentos em laboratório, nos quais a velocidade foi medida em vários pontos com sensores laser-Doppler (5).

Foi simulado o escoamento para $Re = 50$, considerando como comprimento característico a largura do duto menor, e como velocidade característica a velocidade média na entrada. Exibimos na figura 3.8 a comparação das velocidades medidas experimentalmente com os nossos resultados, obtidos com $\Delta t = 0,01$, $T = 30$, e tamanho de malha $h_2 = 0,01$ na proximidade do vértice do degrau, e $h_1 = 0,1$ no resto do escoamento. Nota-se um bom acordo com o experimento, embora haja uma leve discrepância, que se explica pela incerteza dos dados experimentais e pelo fato

de que aqui usamos uma modelagem bidimensional, o que é intrinsecamente uma aproximação.

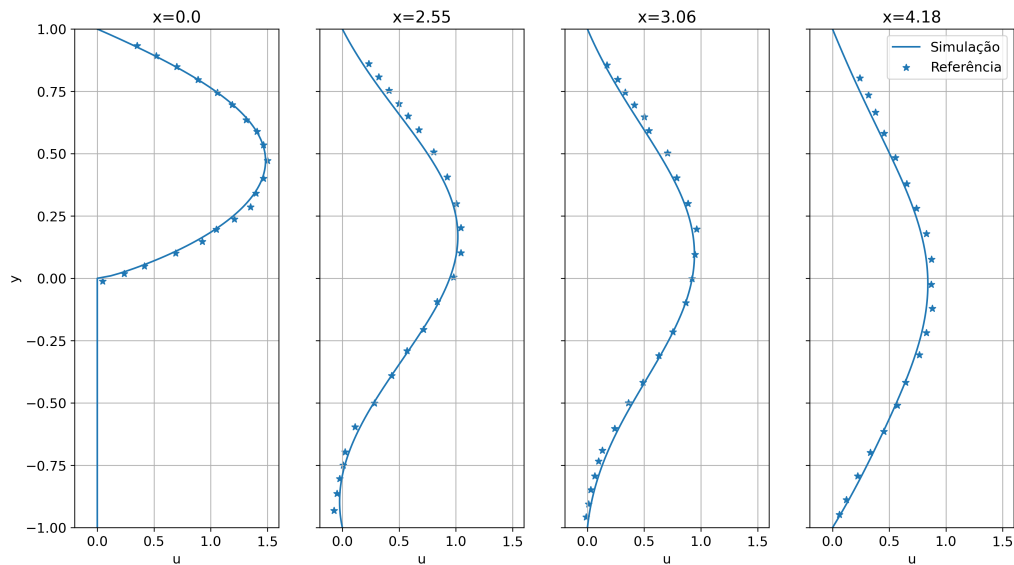


Figura 3.8: Perfil de velocidade horizontal em diversos pontos do duto no escoamento sobre um degrau, e comparação com (5). $Re = 50$, $\Delta t = 0,01$, $T = 30$, $h_1 = 0,1$, $h_2 = 0,01$.

As figuras 3.9 exibem os demais resultados do escoamento. Nota-se claramente que ocorre uma zona de recirculação próximo à base do degrau, como esperado.

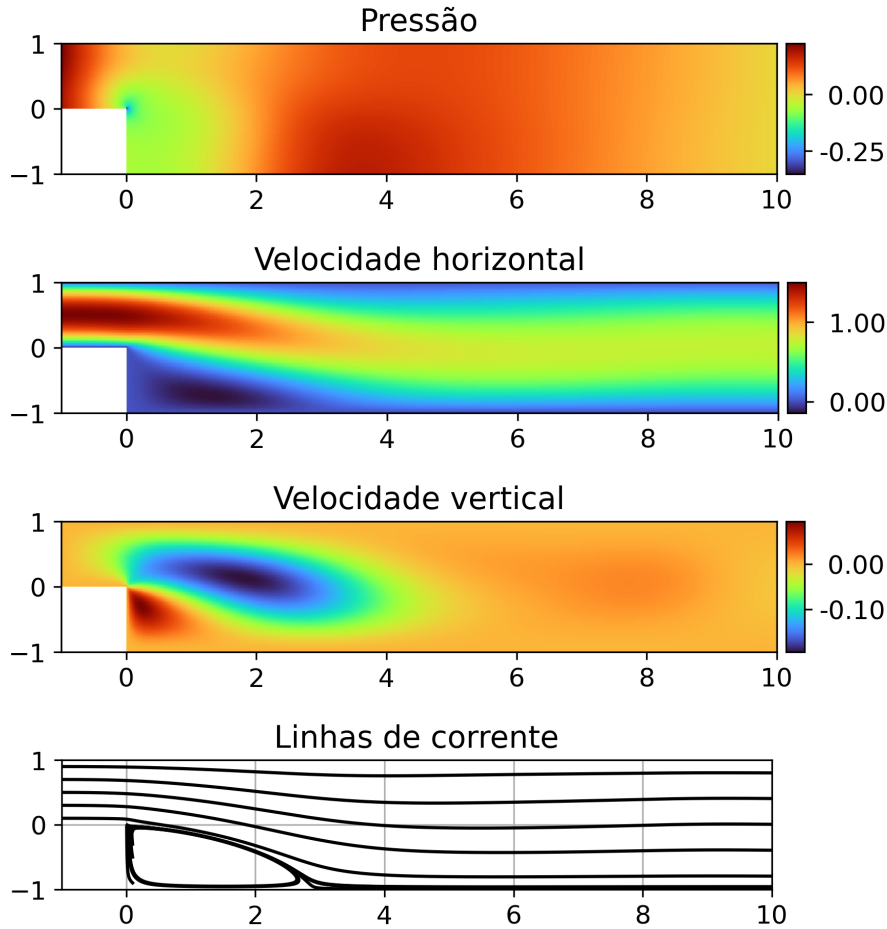


Figura 3.9: Resultados do escoamento sobre um degrau, simulado com o Método de Elementos Finitos, para $Re = 50$, $\Delta t = 0,01$, $T = 30$, $h_1 = 0,1$, $h_2 = 0,01$.

3.1.2 Escolha de parâmetros de cálculo

Uma vez validado o modelo, como visto na seção anterior, resta definir que parâmetros serão empregados na produção de resultados que serão usados para treinar a rede neural. Essa decisão é delicada, pois deve buscar ao mesmo tempo uma precisão fina o suficiente para obter resultados significativos e um tempo de cálculo reduzido, que permita a produção de uma grande massa de dados em tempo hábil, para o treinamento da rede neural.

As condições de contorno de Dirichlet para esse escoamento serão:

- $u = 1$ na superfície de entrada (esquerda)
- $u = 0$ na superfície do aerofólio, pela condição de não-deslizamento

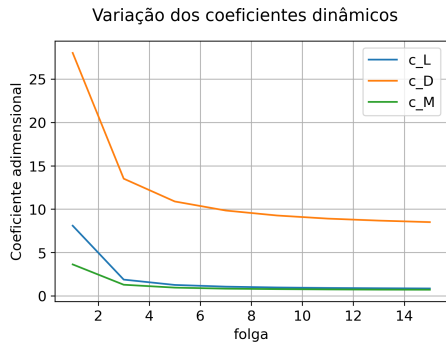
- $u = 1$ nas superfícies superior e inferior, que supomos estarem suficientemente afastadas do aerofólio
- $p = 0$ na superfície de saída

Todas as demais condições de contorno sobre velocidade e pressão são condições de von Neumann homogêneas.

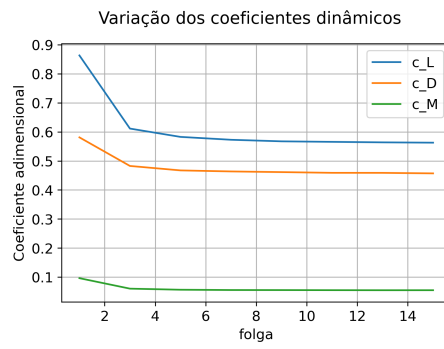
Definimos também quais parâmetros precisam ser determinados:

- *folga*: A distância entre o aerofólio e as superfícies superior e inferior.
- n_{cont} : Número de pontos no contorno superior e no contorno inferior do aerofólio usados para geração de malha. Quanto maior, mais precisa é a representação da peça.
- h : Tamanho dos elementos de malha distantes do aerofólio, passado para o Gmsh como argumento.
- Δt : Passo de tempo usado nas equações 2.71
- T : Tempo total de escoamento

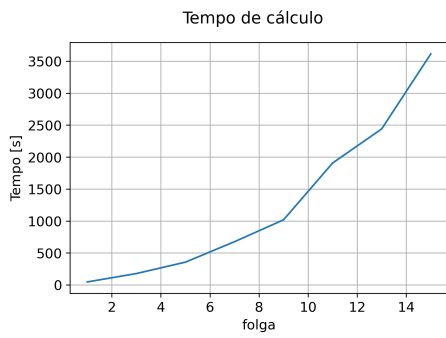
Para avaliar o impacto de cada um desses parâmetros nos resultados, executamos os cálculos para $Re = 1$ e $Re = 100$, variando gradativamente o valor de um dos parâmetros, mantendo os demais constantes, e calculamos os coeficientes aerodinâmicos obtidos. Também foi registrado o tempo computacional necessário, o que depende fortemente de cada um dos parâmetros. O aerofólio utilizado nessa análise foi o NACA-4412, com inclinação de 10° . As figuras a seguir exibem os resultados dessa análise. Note-se que o valor absoluto dos coeficientes encontrados não é importante no momento, e sim a sua sensibilidade aos parâmetros de cálculo.



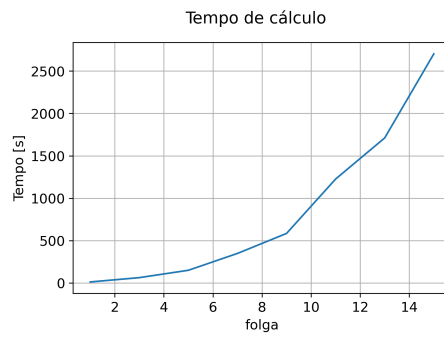
(a) $Re = 1$



(b) $Re = 100$

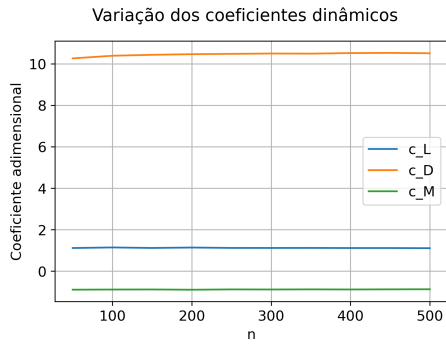


(c) $Re = 1$

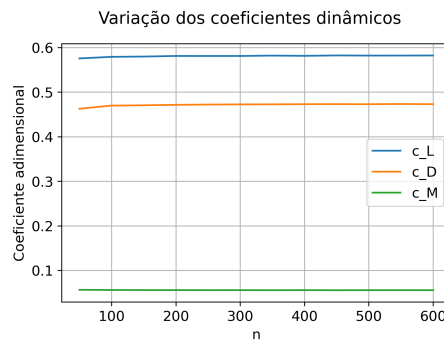


(d) $Re = 100$

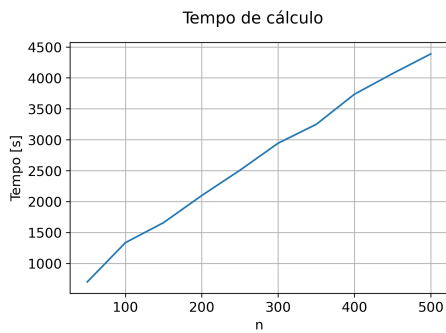
Figura 3.10: Coeficientes aerodinâmicos e tempo computacional para o cálculo do aerofólio NACA-4412 variando a folga. $n = 50$, $h = 1,0$, $\Delta t = 0,01$, $T = 30$



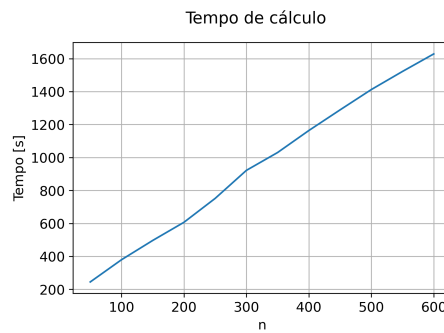
(a) $Re = 1$



(b) $Re = 100$

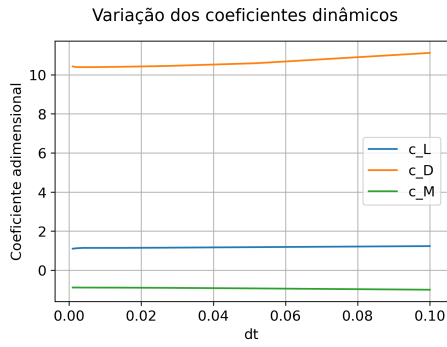


(c) $Re = 1$

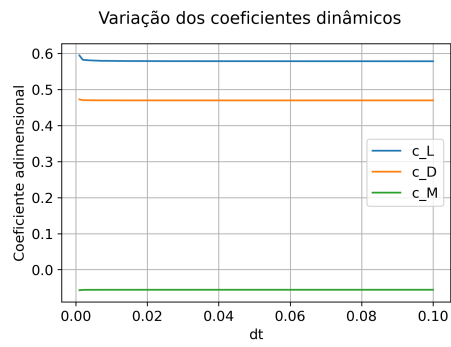


(d) $Re = 100$

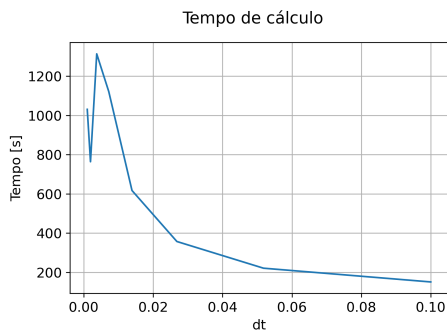
Figura 3.11: Coeficientes aerodinâmicos e tempo computacional para o cálculo do aerofólio NACA-4412 variando o número de pontos no contorno do aerofólio n . $folga = 10$, $h = 1,0$, $\Delta t = 0,01$, $T = 50$



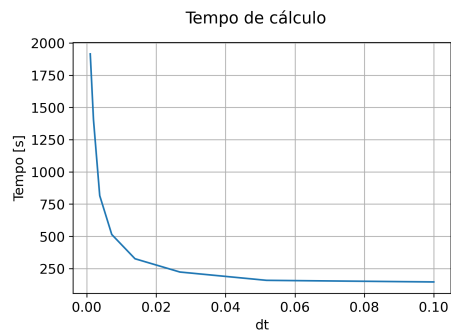
(a) $Re = 1$



(b) $Re = 100$



(c) $Re = 1$



(d) $Re = 100$

Figura 3.12: Coeficientes aerodinâmicos e tempo computacional para o cálculo do aerofólio NACA-4412 variando o passo temporal Δt . $n = 100$, $folga = 6$, $h = 1, 0$, $T = 30$

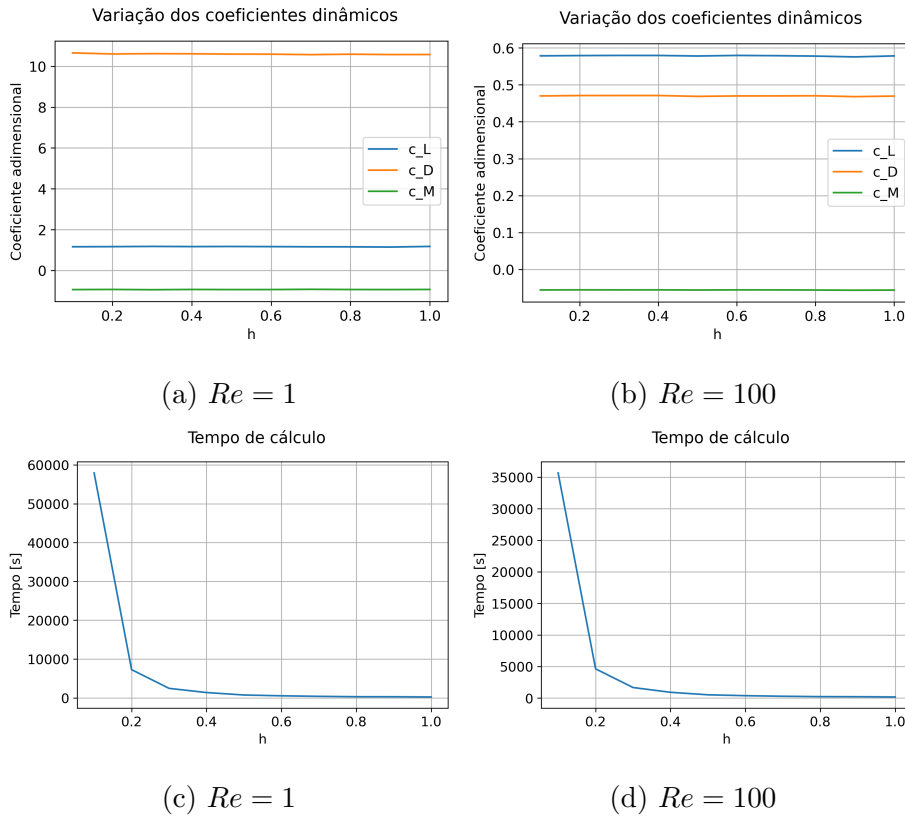


Figura 3.13: Coeficientes aerodinâmicos e tempo computacional para o cálculo do aerofólio NACA-4412 variando o tamanho dos elementos de malha h . $n = 100$, $folga = 6$, $\Delta t = 0,05$, $T = 30$

Com base nos resultados exibidos nas figuras 3.10 a 3.13, foram selecionados os parâmetros descritos na tabela 3.1, que apresentam um erro relativamente pequeno e demandam um tempo computacional moderado. Todos os resultados dados como entrada no treinamento da rede neural foram obtidos com esse conjunto de parâmetros.

n	100
folga	6,0
Δt	0,05
h	1,0
T	50

Tabela 3.1: Parâmetros selecionados para cálculo das forças dinâmicas em aerofólios, que serão dadas como entrada para a rede neural.

3.2 Rede Neural

3.2.1 Seleção de hiperparâmetros

É preciso determinar a arquitetura da rede, definida pelo número de camadas ocultas L e pelo número de neurônios por camada n_n . Além disso, as estratégias descritas na seção 2.5.4 dependem da taxa de aprendizado η_0 , da taxa de redução γ e do parâmetro de regularização λ . Esses 5 hiperparâmetros são anteriores ao início do treinamento, e por isso não podem ser treinados por descida do gradiente, como fazemos com os parâmetros da rede. Assim, a fim de definir os valores empregados, realizamos o treinamento variando sistematicamente os valores de L , n_n , η_0 , γ e λ , e avaliamos o erro quadrático médio obtido com esses valores nos dados de validação e o tempo computacional gasto no treinamento. Note-se que, embora o tempo de treino aqui encontrado seja pequeno, em função do número relativamente pequeno de dados em cada conjunto, ele pode crescer rapidamente em outras aplicações de nosso modelo, de forma que o tempo deve ser levado em consideração.

Arquitetura

Com base em testes preliminares, foram definidos valores tentativos para η_0 , γ e λ , listados na tabela 3.2. Usando esses valores, variamos o número de camadas ocultas entre 1 e 10, e o número de neurônios por camada de 10 a 100, de modo a determinar qual combinação apresentaria o melhor equilíbrio entre erro nos dados de validação e tempo computacional total. A figura 3.14a mostra que o erro quadrático médio dos dados de validação é menor quando usamos um grande número de neurônios por camada, e um maior número de camadas. A fig. 3.14b, contudo, evidencia que o tempo computacional é minimizado para redes com um número intermediário de camadas, entre 4 e 6. Sendo assim, visando ao equilíbrio entre precisão e tempo computacional, escolhemos a rede com $L = 4$ camadas e $n_n = 80$ neurônios por camada, que obteve um erro quadrático médio igual a $9,92 \times 10^4$, e levou 40,9s para ser treinada.

hiperparâmetro	valor
η_0	1×10^{-3}
γ	0,95
λ	1×10^{-5}

Tabela 3.2: Valores iniciais dos hiperparâmetros utilizados para comparação das possíveis arquiteturas de rede.

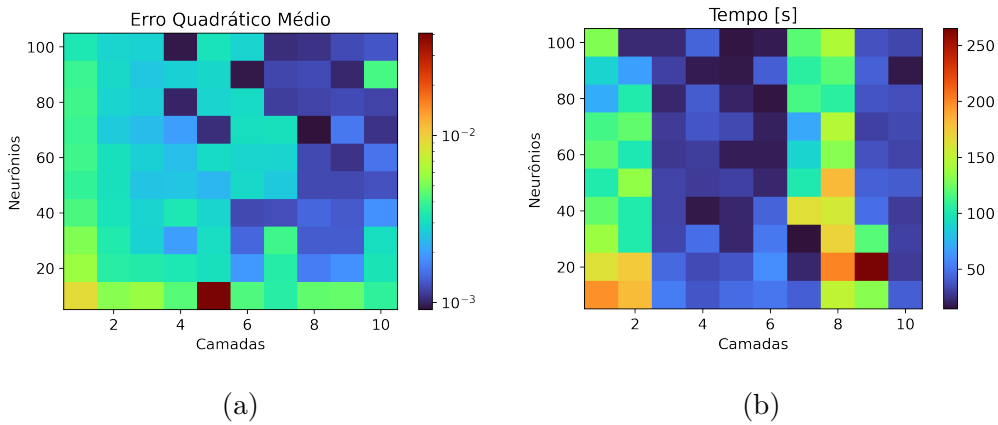


Figura 3.14: Erro quadrático médio obtido pela rede neural nos dados de validação e tempo computacional gasto em seu treinamento em função do número de neurônios e de camadas.

Parâmetro de regularização

Para selecionar o parâmetro λ , usado no método de regularização L2, fazemos esse parâmetro variar entre 10^{-6} e 10^{-1} , e treinamos uma rede com a arquitetura selecionada acima, e com os mesmos valores de η e γ . Os resultados obtidos para cada valor de λ são exibidos na figura 3.15. Nota-se que o tempo de treino é fracamente influenciado por esse parâmetro, de modo que selecionamos o valor que obtém o menor erro quadrático médio: $\lambda = 2,5 \times 10^{-6}$.

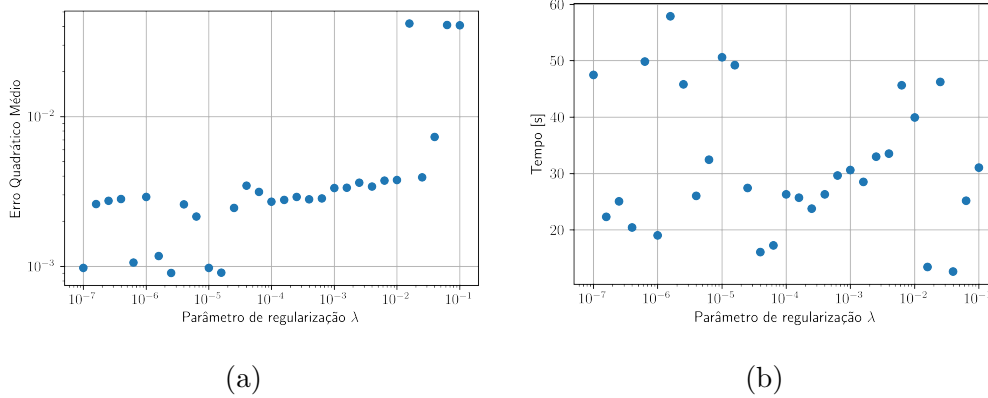


Figura 3.15: Erro quadrático médio e tempo de treino da rede em função do parâmetro de regularização λ

Taxa de aprendizado e redução

Finalmente, resta definir os parâmetros η_0 e γ . Como ambos estão intimamente relacionados, eles são avaliados simultaneamente: variamos η_0 de 10^{-5} a 10^{-2} , e γ de 70% a 98%. Como se vê na figura 3.16, o erro e o tempo de treino são minimizados para taxas de aprendizado da ordem de 10^{-2} . Assim, selecionamos a combinação que obteve o menor erro: $\gamma = 96\%$, $\eta_0 = 4,6 \times 10^{-3}$.

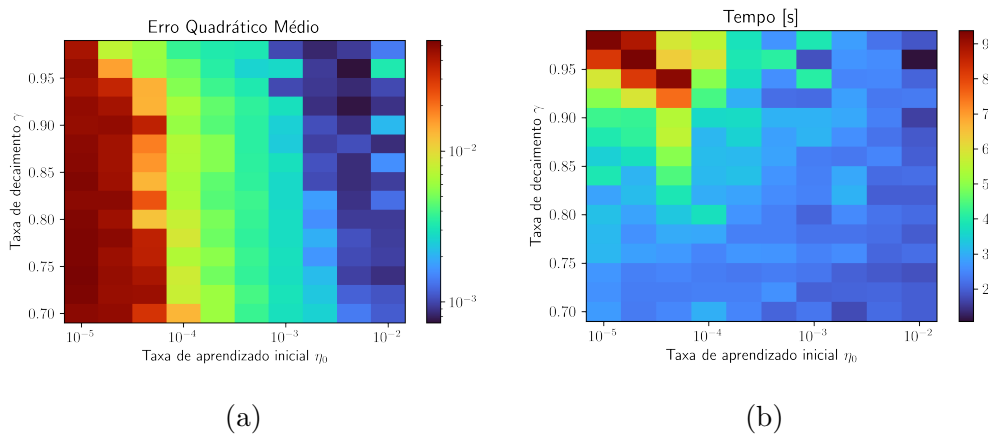


Figura 3.16: Erro quadrático médio obtido pela rede neural nos dados de validação e tempo computacional gasto em seu treinamento em função da taxa inicial de aprendizado de sua taxa de decaimento.

Os hiperparâmetros selecionados ao longo desta seção são resumidos na tabela 3.3. A partir daqui, será modificada levemente a estratégia de parada antecipada: aumentaremos o parâmetros ν de 5 para 10, ou seja, o treino se encerra caso não

haja melhora após 10 épocas. Essa mudança aumenta o tempo computacional, mas permite um desempenho superior, como ficou evidente após um processo de tentativa e erro. O desempenho obtido pela rede após a mesma ser aplicada sobre os dados de validação está exibido na tabela 3.4. Na seção a seguir, essa é a rede que será utilizada.

hiperparâmetro	valor
L	4
n_n	80
λ	$2,5 \times 10^{-5}$
η_0	$4,6 \times 10^{-3}$
γ	0,96

Tabela 3.3: Conjunto de hiperparâmetros selecionados para o treinamento da rede após as análises desta seção

tempo de treino	EQM sustentação	EQM arrasto	EQM momento	EQM geral
42 s	$9,70 \times 10^{-4}$	$1,59 \times 10^{-3}$	$3,30 \times 10^{-4}$	$9,00 \times 10^{-4}$

Tabela 3.4: Resultados obtidos para a rede selecionada, usando os parâmetros da tabela 3.3

3.3 Desempenho e comparação dos métodos

A seguir, comparamos os coeficientes obtidos pela rede neural, pela teoria de aerofólio fino e pelo método de elementos finitos, em um mesmo caso padrão: o aerofólio NACA4412, no qual fazemos variar o ângulo de ataque. A figura 3.17 exhibe os resultados encontrados, obtendo-se um excelente acordo entre a rede neural e a análise de elementos finitos. Já a teoria de aerofólio fino diverge em muito da realidade. Com efeito, essa teoria pressupõe um escoamento invíscido, o que, em números de Reynolds baixos, não é realista.

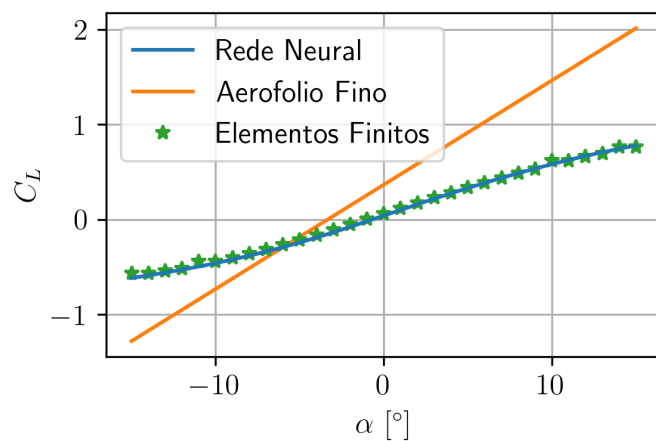
Avaliação nos dados de teste

Finalmente, resta avaliar o desempenho da rede em um conjunto de dados inteiramente novo, disjunto daquele com o qual ela foi treinada e também do conjunto utilizado para definir os hiperparâmetros. Com essa finalidade, foram separados 25% da massa de dados produzidos pelo método de elementos finitos, identificados como dados de teste. A tabela 3.5 mostra que a rede se saiu tão bem nos dados de teste quanto nos de validação, o que atesta a generalidade do modelo treinado.

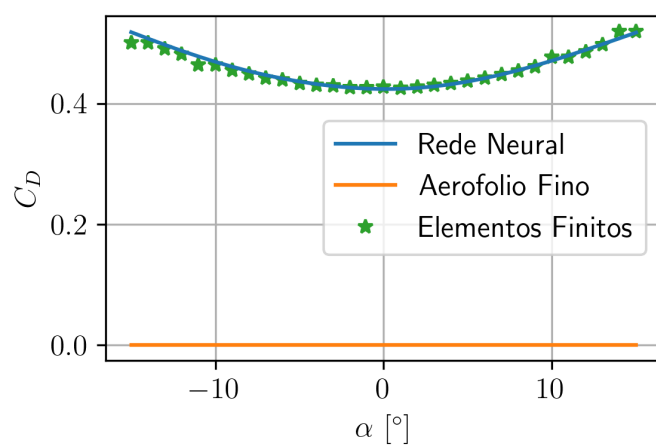
Pode-se observar também que o erro obtido pela teoria analítica de aerofólio fino é ordens de grandeza superior ao da rede. Seu tempo computacional também é mais elevado, embora isso possa se dever ao fato de esses cálculos estarem implementados em linguagem Python sem otimizações, ao passo que a rede neural é executada pelo módulo keras, que acelera consideravelmente os cálculos. Deve-se ressaltar, também, que o erro quadrático médio da ordem de 10^{-4} se mostra adequado para a etapa de pré-projeto e dimensionamento inicial de um aerofólio, que é o objetivo principal da metodologia desenvolvida neste trabalho.

Método	$ E_{max} $	$EQM(c_L)$	$EQM(c_D)$	$EQM(c_M)$	EQM	t
Aerofólio Fino	49,2	27,4	0,46	2,13	10,0	0,17 s
Rede Neural	0,238	$9,5 \times 10^{-4}$	$9,5 \times 10^{-5}$	$1,1 \times 10^{-4}$	$3,8 \times 10^{-4}$	0,047 s

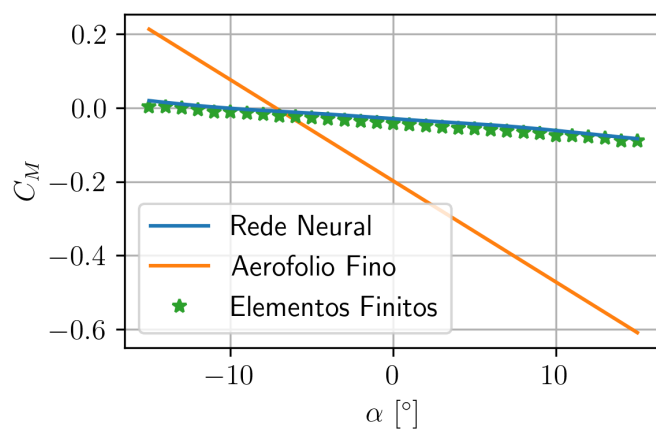
Tabela 3.5: Erro máximo absoluto, erro quadrático médio e tempo de cálculo da rede nos dados de teste; comparação com a teoria de aerofólio fino.



(a) Sustentação



(b) Arrasto



(c) Momento

Figura 3.17: Coeficientes de sustentação, arrasto e momento calculados com cada um dos métodos para o aerofólio NACA4412. $Re=100$.

Capítulo 4

Conclusão

Foi desenvolvida uma metodologia de rede neural que permite avaliar rapidamente os coeficientes de arrasto, sustentação e momento de um aerofólio, e que mostrou uma acurácia comparável ao método de elementos finitos com o qual a rede foi treinada. Apesar do tempo computacional significativo envolvido em gerar os casos de treino por elementos finitos, uma vez treinada a rede, os resultados são obtidos quase instantaneamente, o que se mostra vantajoso no caso em que se deseja otimizar as dimensões do aerofólio.

A teoria analítica de aerofólio fino, embora encontre aplicações na engenharia aeronáutica, não exibiu bons resultados nos casos estudados, sugerindo que a combinação entre dinâmica dos fluidos computacional e aprendizado de máquina aqui desenvolvida é preferível aos métodos analíticos aproximados.

Uma continuação natural deste trabalho seria o desenvolvimento de um modelo de otimização que utilizasse a rede neural como entrada, e que poderia ser adaptado para otimizar o aerofólio para diferentes condições de escoamento (por exemplo, caso se deseje projetar as pás de uma turbina eólica para locais com diferentes escalas de velocidade de vento), e para diferentes objetivos, como por exemplo maximizar a razão entre sustentação e arrasto, maximizar a sustentação total ou minimizar as tensões mecânicas e o custo de fabricação da peça, o que exigiria outras análises além do escopo deste trabalho. Ainda, alterando esses objetivos, é possível construir a frente de Pareto (7) do projeto, informando o engenheiro projetista quanto aos limites do desempenho do aerofólio, mas exigindo múltiplas iterações do processo de otimização, de modo que a rápida execução da rede neural se mostra vantajosa.

Outra possível extensão seria a modelagem de escoamentos turbulentos, modificando as equações resolvidas no método de elementos finitos para as equações médias de Reynolds, o que permite não apenas abordar uma classe de escoamentos mais próximos das aplicações práticas da engenharia, mas também avaliar a própria turbulência como um parâmetro adicional que pode ser modelado pela rede neural, uma vez que tem importância para o projeto de certos tipos de estrutura.

Por uma questão de simplicidade, os desenvolvimentos aqui exibidos foram feitos com aerofólios bidimensionais, da série NACA, definidos por apenas 3 parâmetros. Não há, contudo, impedimento para que se utilize outros conjuntos de parâmetros, ou mesmo abordagens não-paramétricas, para modelar um aerofólio arbitrário. Também é possível estender a metodologia a outros tipos de superfícies aerodinâmicas, bi ou tridimensionais, embora tanto o aumento do número de parâmetros de entrada quanto a passagem para 3 dimensões espaciais venham acompanhados de uma maior complexidade computacional.

Referências Bibliográficas

- [1] BOENTE, A. N. P., SOUZA, C. M. A. D., NUNES, E. C. N., et al. “Produção de energia limpa para redução dos impactos ambientais: estudo de caso do parque eólico de Gargaú no estado do Rio de Janeiro”, *SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA–SEGET*, v. 7, 2015.
- [2] NITHIARASU, P., LEWIS, R. W., SEETHARAMU, K. N. *Fundamentals of the Finite Element Method for Heat and Mass Transfer*. John Wiley & Sons, 2016. ISBN: 9780470756256.
- [3] NIELSEN, M. *Neural Networks and Deep Learning*. 2019. Disponível em: <<http://neuralnetworksanddeeplearning.com/>(acessoem01/2024)>.
- [4] MARCHI, C. H., SUERO, R., ARAKI, L. K. “The lid-driven square cavity flow: numerical solution with a 1024 x 1024 grid”, *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, v. 31, pp. 186–198, 2009.
- [5] ARMALY, B. F., DURST, F., PEREIRA, J., et al. “Experimental and theoretical investigation of backward-facing step flow”, *Journal of fluid Mechanics*, v. 127, pp. 473–496, 1983.
- [6] EPE. “Balanço Energético Nacional 2023: Ano base 2022”, 2023.
- [7] VAN VELDHUIZEN, D. A., LAMONT, G. B., OTHERS. “Evolutionary computation and convergence to a pareto front”. In: *Late breaking papers at the genetic programming 1998 conference*, pp. 221–228. Citeseer, 1998.
- [8] ABBOT, I., VON DOENHOFF, A. *Theory of Wing Sections: Including a summary of airfoil data*. New York, Dover Publications, 1959.

- [9] TENENBAUM, R. A. *Dinâmica Aplicada*. 3^a ed. Barueri, Editora Manole Ltda., 2006.
- [10] HANSEN, A. *Fluid Mechanics*. John Wiley & Sons Inc., 1967.
- [11] ABIODUN, O. I., JANTAN, A., OMOLARA, A. E., et al. “State-of-the-art in artificial neural network applications: A survey”, *Heliyon* 4, 2018. doi: <https://doi.org/10.1016/j.heliyon.2018.e00938>.
- [12] TRAORE, B. B., KAMSU-FOGUEM, B., TANGARA, F. “Deep convolution neural network for image recognition”, *Ecological informatics*, v. 48, pp. 257–268, 2018.
- [13] GHANBARZADEH, A., NOGHREHABADI, A., BEHRANG, M., et al. “Wind speed prediction based on simple meteorological data using artificial neural network”. In: *2009 7th IEEE International Conference on Industrial Informatics*, pp. 664–667, 2009. doi: 10.1109/INDIN.2009.5195882.
- [14] GOLDBERG, Y. *Neural network methods for natural language processing*. Springer Nature, 2022.
- [15] CAI, S., MAO, Z., WANG, Z., et al. “Physics-informed neural networks (PINNs) for fluid mechanics: A review”, *Acta Mechanica Sinica*, v. 37, n. 12, pp. 1727–1738, 2021.
- [16] EIVAZI, H., TAHANI, M., SCHLATTER, P., et al. “Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations”, *Physics of Fluids*, v. 34, n. 7, 2022.
- [17] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. “Learning representations by back-propagating errors”, *nature*, v. 323, n. 6088, pp. 533–536, 1986. doi: <https://doi.org/10.1038/323533a0>.
- [18] BAYDIN, A. G., PEARLMUTTER, B. A., RADUL, A. A., et al. “Automatic Differentiation in machine Learning: a Survey”, *Journal of Machine Learning Research*, v. 18, pp. 1–43, 2018.

- [19] HUI, X., BAI, J., WANG, H., et al. “Fast pressure distribution prediction of airfoils using deep learning”, *Aerospace Science and Technology*, v. 105, pp. 105949, 2020.
- [20] KOSTAS, K. V., MANOUSARIDOU, M. “Machine-Learning-Enabled Foil Design Assistant”, *Journal of Marine Science and Engineering*, v. 11, n. 7, pp. 1470, 2023.
- [21] SONG, X., WANG, L., LUO, X. “Airfoil optimization using a machine learning-based optimization algorithm”. In: *Journal of Physics: Conference Series*, v. 2217, p. 012009. IOP Publishing, 2022.
- [22] CHORIN, A. E., MARSDEN, J. E. *A Mathematical Introduction to Fluid Mechanics*. Springer, 1992.
- [23] ZOHURI, B. “Similitude Theory and Applications”. In: *Dimensional Analysis and Self-Similarity Methods for Engineers and Scientists*, pp. 93–193, Cham, Springer International Publishing, 2015. ISBN: 978-3-319-13476-5. doi: 10.1007/978-3-319-13476-5_2.
- [24] GENNARETTI, M., GIANANTE, R. “Kutta–Joukowski Theorem for Unsteady Linear Aerodynamics”, *AIAA Journal*, v. 60, n. 10, pp. 5779–5790, 2022. doi: 10.2514/1.J061894.
- [25] LAMB, S. H. *Hydrodynamics*. Cambridge University, 1975. ISBN: 0-521-05515-6.
- [26] MARCHMAN, J. *Aerodynamics and Aircraft Performance*. Virginia Tech, 2021. Disponível em: <<http://hdl.handle.net/10919/96525>>.
- [27] GRIMBERG, G., PAULS, W., FRISCH, U. “Genesis of d’Alembert’s paradox and analytical elaboration of the drag problem”, *Physica D: Nonlinear Phenomena*, v. 237, n. 14, pp. 1878–1886, 2008. ISSN: 0167-2789. doi: <https://doi.org/10.1016/j.physd.2008.01.015>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167278908000225>>. Euler Equations: 250 Years On.

- [28] ABBOTT, I. H., VON DOENHOFF, A. E., STIVERS JR, L. *Summary of airfoil data*. Relatório técnico, 1945. Disponível em: <<https://ntrs.nasa.gov/citations/19930090976>>.
- [29] SRIREKHA, A., BASHETTY, K. “Infinite to finite: an overview of finite element analysis”, *Indian Journal of Dental Research*, v. 21, n. 3, pp. 425–432, 2010.
- [30] ZIENKIEWICZ, O. C., TAYLOR, R. L., ZHU, J. Z. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, 2013. ISBN: 978-1-85617-633-0.
- [31] DAVID MÜZEL, S., BONHIN, E. P., GUIMARÃES, N. M., et al. “Application of the Finite Element Method in the Analysis of Composite Materials: A Review”, *Polymers*, v. 12, n. 4, 2020. ISSN: 2073-4360. doi: 10.3390/polym12040818. Disponível em: <<https://www.mdpi.com/2073-4360/12/4/818>>.
- [32] PRINN, A. G. “A Review of Finite Element Methods for Room Acoustics”, *Acoustics*, v. 5, n. 2, pp. 367–395, 2023. ISSN: 2624-599X. doi: 10.3390/acoustics5020022. Disponível em: <<https://www.mdpi.com/2624-599X/5/2/22>>.
- [33] ZIENKIEWICZ, O. C., TAYLOR, R. L., NITHIARASU, P. *The Finite Element Method for Fluid Dynamics*. Butterworth-Heinemann, 2014. ISBN: 978-1-85617-635-4.
- [34] RHEBERGEN, S., WELLS, G. N. “Analysis of a Hybridized/Interface Stabilized Finite Element Method for the Stokes Equations”, *SIAM Journal on Numerical Analysis*, v. 55, n. 4, pp. 1982–2003, 2017. doi: 10.1137/16M1083839. Disponível em: <<https://doi.org/10.1137/16M1083839>>.
- [35] CHAO, T., CHOW, W. K., KONG, H. “A review on the applications of finite element method to heat transfer and fluid flow”, *International Journal on Architectural Science*, v. 3, n. 1, pp. 1–19, 2002.

- [36] GLOWINSKI, R., PIRONNEAU, O. “Finite Element Methods for Navier-Stokes Equations”, *Annual Review of Fluid Mechanics*, v. 24, n. 1, pp. 167–204, 1992. doi: 10.1146/annurev.fl.24.010192.001123.
- [37] GODA, K. “A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows”, *Journal of computational physics*, v. 30, n. 1, pp. 76–95, 1979.
- [38] VIRTANEN, P., GOMMERS, R., OLIPHANT, T. E., et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”, *Nature Methods*, v. 17, pp. 261–272, 2020. doi: 10.1038/s41592-019-0686-2.
- [39] GEUZAINÉ, C., REMACLE, J.-F. “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities”, *International journal for numerical methods in engineering*, v. 79, n. 11, pp. 1309–1331, 2009.
- [40] CHAUHAN, A., SAINI, R. “Statistical analysis of wind speed data using Weibull distribution parameters”. In: *2014 1st International Conference on Non Conventional Energy (ICONCE 2014)*, pp. 160–163. IEEE, 2014.
- [41] KLEINBERG, B., LI, Y., YUAN, Y. “An alternative view: When does SGD escape local minima?” In: *International conference on machine learning*, pp. 2698–2707. PMLR, 2018.
- [42] LECUN, Y., BOTTOU, L., ORR, G. B., et al. “Efficient backprop”. In: *Neural networks: Tricks of the trade*, Springer, pp. 9–50, 2002.
- [43] KINGMA, D. P., BA, J. “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.
- [44] PRECHELT, L. “Early stopping-but when?” In: *Neural Networks: Tricks of the trade*, Springer, pp. 55–69, 2002.

Apêndice A

Montagem das matrizes do Método de Elementos Finitos

A.1 Integração em elementos triangulares

Todas as matrizes a seguir são compostas de integrais sobre elementos triangulares. Seja e_ℓ um tal elemento, com vértices $\mathbf{x}_0 = (x_0, y_0)$, $\mathbf{x}_1 = (x_1, y_1)$ e $\mathbf{x}_2 = (x_2, y_2)$, e seja $f(\mathbf{x})$ uma função que se deseja integrar. Definimos as variáveis auxiliares α e β tal que:

$$\mathbf{x}(\alpha, \beta) = \mathbf{x}_0 + (\mathbf{x}_1 - \mathbf{x}_0)\alpha + (\mathbf{x}_2 - \mathbf{x}_0)\beta \quad (\text{A.1})$$

Note-se que, ao variar β de 0 a 1, e α de 0 a $1 - \beta$, varre-se a área do triângulo. Assim, fazemos a mudança de variáveis $x, y \rightarrow \alpha, \beta$, multiplicando pelo jacobiano:

$$\left| \frac{\partial(x, y)}{\partial(\alpha, \beta)} \right| = \begin{vmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{vmatrix} = 2A_\ell \quad (\text{A.2})$$

$$\int_{e_\ell} f(x, y) dx dy = 2A_\ell \int_0^1 d\beta \int_0^{1-\beta} f(x(\alpha, \beta), y(\alpha, \beta)) d\alpha \quad (\text{A.3})$$

Onde A_ℓ corresponde à área do elemento.

Nos casos a seguir, f será uma função polinomial em α e β , logo é conveniente definir uma regra para a integração de termos $\alpha^a \beta^b$, como fazemos abaixo. Seja J_{ab} essa integral:

$$\begin{aligned}
J_{ab} &= \int_0^1 d\beta \int_0^{1-\beta} \alpha^a \beta^b d\alpha = \frac{1}{a+1} \int_0^1 (1-\beta)^{a+1} \beta^b d\beta = \\
&= \frac{1}{a+1} \left[(1-\beta)^{a+1} \frac{\beta^{b+1}}{b+1} \Big|_0^1 - \int_0^1 -(a+1)(1-\beta)^a \frac{\beta^{b+1}}{b+1} d\beta \right] = \quad (\text{A.4}) \\
&= \frac{a}{b+1} J_{(a-1)(b+1)}
\end{aligned}$$

Logo, sabendo que:

$$J_{0b} = \int_0^1 d\beta \int_0^{1-\beta} \beta^b d\alpha = \int_0^1 (1-\beta) \beta^b d\beta = \frac{\beta^{b+1}}{b+1} - \frac{\beta^{b+2}}{b+2} \Big|_0^1 = \frac{1}{(b+1)(b+2)} \quad (\text{A.5})$$

Temos, para qualquer par a, b :

$$\begin{aligned}
J_{ab} &= \frac{a}{b+1} \cdot \frac{a-1}{b+2} \cdot \dots \cdot \frac{1}{b+a} J_{0(b+a)} = \\
&= \frac{a!b!}{(b+a)!} \cdot \frac{1}{(b+a+1)(b+a+2)} = \quad (\text{A.6}) \\
&= \frac{a!b!}{(a+b+2)!}
\end{aligned}$$

Em suma:

$$J_{ab} = \int_0^1 d\beta \int_0^{1-\beta} \alpha^a \beta^b d\alpha = \frac{a!b!}{(a+b+2)!} \quad (\text{A.7})$$

Para facilitar a integração, as funções de forma podem ser escritas em função de α e β . Por simplicidade, fazemos uma translação no espaço de modo que $x_0, y_0 \rightarrow 0$, sem perda de generalidade:

$$N(x, y) = a + bx + cy = a + b(\alpha x_1 + \beta x_2) + c(\alpha y_1 + \beta y_2) = a' + b'\alpha + c'\beta \quad (\text{A.8})$$

Onde:

$$a' = a \quad (\text{A.9a})$$

$$b' = bx_1 + cy_1 \quad (\text{A.9b})$$

$$c' = bx_2 + cy_2 \quad (\text{A.9c})$$

O mesmo pode ser feito com as funções de forma quadráticas $M(x, y)$:

$$M(x, y) = a + bx + cy + dx^2 + ey^2 + fxy = a' + b'\alpha + c'\beta + d'\alpha^2 + e'\beta^2 + f'\alpha\beta \quad (\text{A.10})$$

Onde:

$$a' = a \quad (\text{A.11a})$$

$$b' = bx_1 + cy_1 \quad (\text{A.11b})$$

$$c' = bx_2 + cy_2 \quad (\text{A.11c})$$

$$d' = dx_1^2 + ey_1^2 + fx_1y_1 \quad (\text{A.11d})$$

$$e' = dx_2^2 + ey_2^2 + fx_2y_2 \quad (\text{A.11e})$$

$$f' = 2dx_1x_2 + 2ey_1y_2 + fx_1y_2 + fx_2y_1 \quad (\text{A.11f})$$

A.2 Matriz de massa M

Precisamos calcular a matriz de massa $\mathbf{M}_{\mathbf{V}}^{\hat{\mathbf{V}}}$, para função teste e tentativa ambas de ordem 2. Segundo a eq. 2.64, essa matriz é definida por:

$$[\mathbf{M}]_{ij} = \sum_{e_\ell \ni i,j} \int_{e_\ell} M_i M_j de_\ell \quad (\text{A.12})$$

Assim, escrevendo em função de α e β a fim de facilitar a integração:

$$\begin{aligned} M_i(\alpha, \beta) \cdot M_j(\alpha, \beta) = & \rho_1 + \rho_2\alpha + \rho_3\beta + \rho_4\alpha^2 + \\ & + \rho_5\beta^2 + \rho_6\alpha\beta + \rho_7\alpha^3 + \rho_8\beta^3 + \rho_9\alpha^2\beta + \rho_{10}\alpha\beta^2 + \\ & + \rho_{11}\alpha^4 + \rho_{12}\beta^4 + \rho_{13}\alpha^3\beta + \rho_{14}\alpha^2\beta^2 + \rho_{15}\alpha\beta^3 \end{aligned} \quad (\text{A.13})$$

Onde:

$$\rho_1 = a'_i a'_j \quad (\text{A.14a})$$

$$\rho_2 = a'_i b'_j + a'_j b'_i \quad (\text{A.14b})$$

$$\rho_3 = a'_i c'_j + a'_j c'_i \quad (\text{A.14c})$$

$$\rho_4 = a'_i d'_j + a'_j d'_i + b'_i b'_j \quad (\text{A.14d})$$

$$\rho_5 = a'_i e'_j + a'_j e'_i + c'_i c'_j \quad (\text{A.14e})$$

$$\rho_6 = a'_i f'_j + a'_j f'_i + b'_i c'_j + b'_j c'_i \quad (\text{A.14f})$$

$$\rho_7 = b'_i d'_j + b'_j d'_i \quad (\text{A.14g})$$

$$\rho_8 = c'_i e'_j + c'_j e'_i \quad (\text{A.14h})$$

$$\rho_9 = c'_i d'_j + c'_j d'_i + b'_i f'_j + b'_j f'_i \quad (\text{A.14i})$$

$$\rho_{10} = b'_i e'_j + b'_j e'_i + c'_i f'_j + c'_j f'_i \quad (\text{A.14j})$$

$$\rho_{11} = d'_i d'_j \quad (\text{A.14k})$$

$$\rho_{12} = e'_i e'_j \quad (\text{A.14l})$$

$$\rho_{13} = d'_i f'_j + d'_j f'_i \quad (\text{A.14m})$$

$$\rho_{14} = d'_i e'_j + d'_j e'_i + f'_i f'_j \quad (\text{A.14n})$$

$$\rho_{15} = e'_i f'_j + e'_j f'_i \quad (\text{A.14o})$$

Com auxílio da eq. A.7, calcula-se a integral sobre o elemento:

$$\begin{aligned} \int_{e_\ell} M_i(x, y) \cdot M_j(x, y) de_\ell &= 2A_\ell \int_0^1 d\beta \int_0^{1-\beta} M_i(\alpha, \beta) \cdot M_j(\alpha, \beta) d\alpha = \\ &= 2A_\ell \left(\frac{\rho_1}{2} + \frac{\rho_2}{6} + \frac{\rho_3}{6} + \frac{\rho_4}{12} + \frac{\rho_5}{12} + \frac{\rho_6}{24} + \frac{\rho_7}{20} + \frac{\rho_8}{20} + \frac{\rho_9}{60} + \frac{\rho_{10}}{60} + \frac{\rho_{11}}{30} + \frac{\rho_{12}}{30} + \frac{\rho_{13}}{120} + \frac{\rho_{14}}{180} + \frac{\rho_{15}}{120} \right) \end{aligned} \quad (\text{A.15})$$

A.3 Matriz elástica K

A matriz elástica deve ser calculada no espaço da velocidade ($\mathbf{M}_V^{\hat{V}}$) e da pressão ($\mathbf{M}_Q^{\hat{Q}}$). Para ambos os casos, será necessário calcular o produto das derivadas parciais em x e y .

A.3.1 Ordem 1: $\mathbf{M}_Q^{\hat{Q}}$

Com base na eq. 2.65, tem-se:

$$\left[\mathbf{M}_Q^{\hat{Q}} \right]_{ij} = \sum_{e_\ell \ni i, j} \int_{e_\ell} \nabla N_i \cdot \nabla N_j de_\ell \quad (\text{A.16})$$

Derivando $N(x, y)$, obtém-se os gradientes:

$$\nabla N_i \cdot \nabla N_j = \begin{pmatrix} b_i \\ c_i \end{pmatrix} \cdot \begin{pmatrix} b_j \\ c_j \end{pmatrix} = b_i b_j + c_i c_j \quad (\text{A.17})$$

Como o integrando é constante, a integração é simples:

$$\int_{e_\ell} \nabla N_i \cdot \nabla N_j de_\ell = (b_i b_j + c_i c_j) \int_{e_\ell} de_\ell = A_\ell (b_i b_j + c_i c_j) \quad (\text{A.18})$$

A.3.2 Ordem 2: $\mathbf{M}_V^{\hat{V}}$

De forma análoga ao caso acima, a matriz é dada por:

$$\left[\mathbf{M}_V^{\hat{V}} \right]_{ij} = \sum_{e_\ell \ni i, j} \int_{e_\ell} \nabla M_i \cdot \nabla M_j de_\ell \quad (\text{A.19})$$

Calculam-se os gradientes:

$$\nabla M_i \cdot \nabla M_j = \begin{pmatrix} b_i + 2d_i x + f_i y \\ c_i + 2e_i y + f_i x \end{pmatrix} \cdot \begin{pmatrix} b_j + 2d_j x + f_j y \\ c_j + 2e_j y + f_j x \end{pmatrix} = q + rx + sy + tx^2 + uy^2 + vxy \quad (\text{A.20})$$

Onde:

$$q = b_i b_j + c_i c_j \quad (\text{A.21a})$$

$$r = 2b_i d_j + 2b_j d_i + c_i f_j + c_j f_i \quad (\text{A.21b})$$

$$s = 2c_i e_j + 2c_j e_i + b_i f_j + b_j f_i \quad (\text{A.21c})$$

$$t = 4d_i d_j + f_i f_j \quad (\text{A.21d})$$

$$u = 4e_i e_j + f_i f_j \quad (\text{A.21e})$$

$$v = 2d_i f_j + 2d_j f_i + 2e_i f_j + 2e_j f_i \quad (\text{A.21f})$$

E, aplicando a eq. A.11a, resulta:

$$\nabla M_i \cdot \nabla M_j = \rho_1 + \rho_2 \alpha + \rho_3 \beta + \rho_4 \alpha^2 + \rho_5 \beta^2 + \rho_6 \alpha \beta \quad (\text{A.22})$$

$$\rho_1 = q \quad (\text{A.23a})$$

$$\rho_2 = rx_1 + sy_1 \quad (\text{A.23b})$$

$$\rho_3 = rx_2 + sy_2 \quad (\text{A.23c})$$

$$\rho_4 = tx_1^2 + uy_1^2 + vx_1 y_1 \quad (\text{A.23d})$$

$$\rho_5 = tx_2^2 + uy_2^2 + vx_2 y_2 \quad (\text{A.23e})$$

$$\rho_6 = 2tx_1 x_2 + 2uy_1 y_2 + vx_1 y_2 + vx_2 y_1 \quad (\text{A.23f})$$

Finalmente, aplicando a eq. A.7:

$$\int_{e_\ell} \nabla M_i \cdot \nabla M_j de_\ell = 2A_\ell \left(\frac{\rho_1}{2} + \frac{\rho_2}{6} + \frac{\rho_3}{6} + \frac{\rho_4}{12} + \frac{\rho_5}{12} + \frac{\rho_6}{24} \right) \quad (\text{A.24})$$

A.4 Matriz gradiente \mathbf{G}

Como visto na seção 2.4.2, será preciso obter a matriz gradiente para função tentativa de ordem 1 e teste ordem 2 ($\mathbf{G}_Q^{\hat{V}}$); tentativa ordem 2 e teste ordem 1 ($\mathbf{G}_V^{\hat{Q}}$) e tentativa e teste de ordem 2 ($\mathbf{G}_V^{\hat{V}}$). O cálculo das matrizes $\mathbf{G}_Q^{\hat{V}}$ e $\mathbf{G}_V^{\hat{Q}}$ é similar, por isso só calcularemos a primeira.

A.4.1 Ordem 1 e 2: $\mathbf{G}_Q^{\hat{V}}$

Os termos da matriz do gradiente na direção x são dados por:

$$\left[\mathbf{G}_{\mathbf{x}Q}^{\hat{V}} \right]_{ij} = \sum_{e_\ell \ni i,j} \int_{e_\ell} \frac{\partial N_j}{\partial x} M_i de_\ell \quad (\text{A.25})$$

Desenvolvendo o integrando, tem-se:

$$\frac{\partial N_j}{\partial x} M_i = (b_j) \cdot (a'_i + b'_i \alpha + c'_i \beta + d'_i \alpha^2 + e'_i \beta^2 + f'_i \alpha \beta) \quad (\text{A.26})$$

Logo, pela eq. A.7:

$$\int_{e_\ell} \frac{\partial N_j}{\partial x} M_i de_\ell = 2A_\ell b_j \left(\frac{a'_i}{2} + \frac{b'_i}{6} + \frac{c'_i}{6} + \frac{d'_i}{12} + \frac{e'_i}{12} + \frac{f'_i}{24} \right) \quad (\text{A.27})$$

De forma análoga, calcula-se o gradiente em y : Logo, pela eq. A.7:

$$\int_{e_\ell} \frac{\partial N_j}{\partial y} M_i de_\ell = 2A_\ell c_j \left(\frac{a'_i}{2} + \frac{b'_i}{6} + \frac{c'_i}{6} + \frac{d'_i}{12} + \frac{e'_i}{12} + \frac{f'_i}{24} \right) \quad (\text{A.28})$$

A.4.2 Ordem 2: $\mathbf{G}_V^{\hat{V}}$

No caso de as funções tentativa e teste serem de ordem 2, tem-se:

$$\left[\mathbf{G}_{\mathbf{x}V}^{\hat{V}} \right]_{ij} = \sum_{e_\ell \ni i,j} \int_{e_\ell} \frac{\partial M_j}{\partial x} M_i de_\ell \quad (\text{A.29})$$

$$\frac{\partial M_j}{\partial x} M_i = (b_j + 2d_j x + f_j y) \cdot (a'_i + b'_i \alpha + c'_i \beta + d'_i \alpha^2 + e'_i \beta^2 + f'_i \alpha \beta) \quad (\text{A.30})$$

Podemos escrever:

$$b_j + 2d_j x + f_j y = q_j + r_j \alpha + s_j \beta \quad (\text{A.31a})$$

$$q_j = b_j \quad (\text{A.31b})$$

$$r_j = 2d_j x_1 + f_j y_1 \quad (\text{A.31c})$$

$$s_j = 2d_j x_2 + f_j y_2 \quad (\text{A.31d})$$

De modo que:

$$\frac{\partial M_j}{\partial x} M_i = \rho_1 + \rho_2 \alpha + \rho_3 \beta + \rho_4 \alpha^2 + \rho_5 \beta^2 + \rho_6 \alpha \beta + \rho_7 \alpha^3 + \rho_8 \beta^3 + \rho_9 \alpha^2 \beta + \rho_{10} \alpha \beta^2 \quad (\text{A.32})$$

Onde:

$$\rho_1 = q_j a_i \quad (\text{A.33a})$$

$$\rho_2 = q_j b_i + r_j a_i \quad (\text{A.33b})$$

$$\rho_3 = q_j c_i + s_j a_i \quad (\text{A.33c})$$

$$\rho_4 = q_j d_i s_j a_i \quad (\text{A.33d})$$

$$\rho_5 = q_j e_i + s_j c_i \quad (\text{A.33e})$$

$$\rho_6 = r_j c_i + s_j b_i + q_j f_i \quad (\text{A.33f})$$

$$\rho_7 = r_j d_i \quad (\text{A.33g})$$

$$\rho_8 = s_j e_i \quad (\text{A.33h})$$

$$\rho_9 = r_j f_i + s_j d_i \quad (\text{A.33i})$$

$$\rho_{10} = s_j f_i + r_j e_i \quad (\text{A.33j})$$