

**UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
INSTITUTO DE ENGENHARIA MECÂNICA**

Ferramentas de Aprendizado de Máquina
aplicadas à Engenharia: Uma análise de
soluções numéricas de equações da Mecânica
dos Fluidos

Marcos Henrique Morbidelli Junior

Itajubá, 11 de abril de 2022

**UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
INSTITUTO DE ENGENHARIA MECÂNICA**

**Ferramentas de Aprendizado de Máquina
aplicadas à Engenharia: Uma análise de
soluções numéricas de equações da Mecânica
dos Fluidos**

Marcos Henrique Morbidelli Junior

Trabalho submetido ao Instituto de Engenharia
Mecânica com finalidade de aprovação no Traba-
lho Final de Graduação

**Orientador: Prof. Dr. Alan Bendasoli Pavan
Coorientador: Prof. Dr. João Andrade de
Carvalho Junior - UNESP/Guaratinguetá**

**11 de abril de 2022
Itajubá**

Dedico este trabalho à minha mãe Gilmara e ao meu pai Marcos.

Agradecimentos

Em primeiro lugar agradeço aos meus pais, Gilmara e Marcos, por todo apoio e incentivo que me deram e pelo esforço que dedicaram para eu poder estudar e me graduar em uma universidade. Estará eternamente gravada em minha memória a felicidade que nós três sentimos no dia da matrícula e o peso que é apresentar este trabalho sem a presença da minha mãe.

Agradeço ao meu orientador Prof. Dr. Alan Bendasoli Pavan por toda ajuda e orientação ao longo deste trabalho, e também pelo auxílio na minha busca por objetivo profissional.

Agradeço ao meu co-orientador Prof. Dr. João Andrade de Carvalho Junior por se disponibilizar a orientar e acompanhar este trabalho.

Agradeço à todas amigas que participaram da minha vida até aqui. A presença, apoio e influência de vocês foi fundamental de alguma forma para meu desenvolvimento e para que o tempo ao longo da graduação fosse mais significativo.

“O maior perigo para a maioria de nós não está em definir o nosso objetivo muito alto e ficarmos aquém, está na definição do nosso objetivo muito baixo, e alcançarmos a meta.”

– Michelangelo

Resumo

Este trabalho final de graduação aborda a aplicação da técnica chamada Redes Neurais Vinculadas pela Física, do inglês PINN (*Physics-informed Neural Networks*) em problemas de Engenharia. A PINN é uma técnica de aprendizado de máquina profundo que usa leis da física como vínculos para a obtenção de soluções ou coeficientes de equações diferenciais que modelam diversos problemas de interesse. Entre eles estão a modelagem das equações de Navier-Stokes e da equação de Burgers, que são importantes na Mecânica dos Fluidos. A rede neural foi treinada a partir de dados referentes a campos de velocidades e quando foi aplicada à equação de Burgers foi capaz de prever o coeficiente de difusão. Já para as equações de Navier-Stokes, a PINN foi capaz de prever o número de Reynolds associado a um escoamento incompressível, bidimensional, externo a um cilindro e reconstruir o campo de pressões, mesmo sem conhecimento de dados iniciais referentes à pressão. Os resultados obtidos mostram a efetividade e acurácia da aplicação da técnica em problemas de interesse para a Mecânica dos Fluidos.

Palavras-chaves: Equação de Navier-Stokes, Aprendizado de Máquina, Equação de Burgers, Redes Neurais, PINN

Abstract

This undergrad term paper addresses the application of the PINN (Physics-informed Neural Networks) to engineering problems. The PINN is a Deep Learning tool that employs laws of physics as links for obtaining solutions or coefficients of differential equations that model a plethora of useful problems. Among them are the Navier-Stokes equations and the Burgers equation, which are important to Fluid Mechanics. The network was trained from the velocity field data and when applied to the Burgers equation was able to predict the diffusion coefficient. For the Navier-Stokes equations, the PINN was successful in predicting the Reynolds number of an incompressible, two-dimensional, external flow to a cylinder and even reconstruct the pressure field without any prior knowledge of pressure data. The results achieved show the effectiveness and accuracy of the technique's application to useful problems in Fluid Mechanics.

Key-words: Navier-Stokes, Machine Learning, Deep Learning, Neural Networks, PINN

Lista de ilustrações

Figura 1 – Relações entre a rede, camadas, função de perda e otimizador. A pontuação de perda é usada como sinal de <i>feedback</i> para ajustar os pesos [CHOLLET, 2021].	23
Figura 2 – Representação em grafo computacional das operações matemáticas realizadas em uma iteração [CHOLLET, 2021].	24
Figura 3 – Gráfico da solução $u(t, x)$ da equação de Burgers para $\nu = 0.01/\pi$ gerado no Mathematica®.	30
Figura 4 – Distribuição espaço-temporal do campo de velocidade $u(t, x)$ da equação de Burgers para $\nu = 0.01/\pi$ gerada no Mathematica®.	31
Figura 5 – Malha empregada para a solução das equações de Navier-Stokes com 15894 nós.	32
Figura 6 – Região do domínio que corresponde ao local das soluções importadas pela PINN.	33
Figura 7 – Resultado obtido pela PINN para o ajuste dos parâmetros λ_1 e λ_2 da Equação de Burgers para $\nu = 0,01/\pi$	34
Figura 8 – Comparação entre os erros de λ_1 da equação de Burgers sem ruído para cada ν	35
Figura 9 – Comparação entre os erros de λ_2 da equação de Burgers sem ruído para cada ν	36
Figura 10 – Comparação entre os erros de λ_1 da equação de Burgers com 1% de ruído para cada valor de coeficiente de difusão, ν	37
Figura 11 – Comparação entre os erros de λ_2 da equação de Burgers com 1% de ruído para cada valor de coeficiente de difusão, ν	37
Figura 12 – Resultado obtido pela PINN para identificação das equações de Navier-Stokes para um escoamento incompressível, bidimensional, externo a um cilindro com fluido newtoniano para $Re = 500$	39
Figura 13 – Resultado obtido pela PINN para identificação das equações de Navier-Stokes para um escoamento incompressível, bidimensional, externo a um cilindro com fluido newtoniano para $Re = 1000$	40

Lista de tabelas

Tabela 1	– Erros na identificação de λ_1 e λ_2 da equação de Burgers para dados sem ruído.	35
Tabela 2	– Erros na identificação de λ_1 e λ_2 da equação de Burgers com 1% de ruído.	36
Tabela 3	– Comparação entre os erros de λ_1 da equação de Burgers para os dados sem ruído para redes com oito, dez e doze camadas de neurônios, respectivamente.	38
Tabela 4	– Comparação entre os erros de λ_2 da equação de Burgers para os dados sem ruído para redes com oito, dez e doze camadas de neurônios, respectivamente.	38
Tabela 5	– Erros na identificação de λ_1 e λ_2 da equação de Burgers para valores de ν intermediários a $\nu = 0,001/\pi$ e $\nu = 0,01/\pi$ para dados sem ruído.	38
Tabela 6	– Erros na identificação de λ_1 e λ_2 das equações de Navier-Stokes para escoamento incompressível, bidimensional, externo a um cilindro com fluido newtoniano. As faixas de Reynolds analisadas foram 100, 500 e 1000.	40

Siglas

PINN - Redes Neurais Vinculadas pela Física, do inglês *Physics-informed Neural Networks*

CFD - Dinâmica de fluidos computacional, do inglês *Computational fluid dynamics*

N-S - Equações de Navier-Stokes

VC - Volume de controle

SC - Superfície de controle

IA - Inteligência Artificial

AM - Aprendizado de Máquina

AP - Aprendizado Profundo

RGB - Vermelho-verde-azul, do inglês *Red-green-blue*

HSV - Valor de saturação de matiz, do inglês *Hue-saturation-value*

MSE - Erro quadrático médio, do inglês *Mean squared error*

GPU - Unidade de Processamento Gráfico, do inglês *Graphics Processing Unit*

VBA - *Visual Basic for Applications*[®]

EDP - Equação diferencial parcial

Símbolos

ρ - Massa específica

μ - Coeficiente de viscosidade dinâmica

ν - Coeficiente de viscosidade cinemática ou coeficiente de difusão

U - Velocidade representativa

L - Comprimento representativo

Re - Número de Reynolds, $Re = \rho UL/\mu = UL/\nu$

τ - Tensão de cisalhamento

t - tempo

p - campo de pressões

\vec{V} - Vetor velocidade

u - Componente da velocidade na direção do escoamento

v - Componente da velocidade na direção transversal ao escoamento

x - Coordenada espacial na direção do escoamento

y - Coordenada espacial na direção transversal ao escoamento

∇ - Operador nabla

ψ - Função corrente

\tanh - tangente hiperbólica

W - Tensor de pesos da rede neural

b - Tensor de bias da rede neural

\mathcal{N} - Operador não linear da rede neural

λ - Parâmetro a ser ajustado pela rede neural

Ω - Subconjunto do conjunto dos números reais em D dimensões

\mathbb{R}^D - Conjunto dos números reais em D dimensões

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.2	Justificativa	15
1.3	Organização do Trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Mecânica dos Fluidos	16
2.1.1	Introdução	16
2.1.2	Forma integral da lei de conservação da massa	17
2.1.3	Forma integral da lei de conservação da quantidade de movimento linear	17
2.1.4	A equação diferencial da conservação da massa	18
2.1.5	A equação diferencial da quantidade de movimento linear	18
2.1.6	Função corrente	18
2.1.7	Análise dimensional	19
2.1.8	Equação de Burgers	20
2.2	Inteligência Artificial	21
2.2.1	Introdução	21
2.2.2	Aprendizado de Máquina	21
2.2.3	Aprendizado de Máquina Profundo	22
2.2.4	Redes Neurais Vinculadas pela Física - PINN	25
2.2.5	Equação de Burgers	26
2.2.6	Equações de Navier-Stokes	27
3	METODOLOGIA	29
3.1	Metodologia para a equação de Burgers	29
3.2	Metodologia para as equações de Navier-Stokes	31
4	RESULTADOS	34
4.1	Resultados para a equação de Burgers	34
4.2	Resultados para as equações de Navier-Stokes	39
5	CONCLUSÃO	42
	REFERÊNCIAS	43

1 Introdução

Diversas formas de tecnologia interagem de alguma forma com algum fluido, isto é, um gás ou um líquido. Projetos de meios de transporte como carros, aviões e navios têm seções destinadas ao estudo da aerodinâmica para economia de combustível; ventiladores e sistemas de ar condicionado trabalham com escoamento de ar; motores e processadores de computador necessitam de sistema de arrefecimento; e em âmbito municipal, o abastecimento de água e disposição de esgotos também trabalham com escoamento de fluidos.

Cerca de 75% da Terra está coberta por água e 100% por ar. Áreas de estudo como meteorologia, oceanografia e hidrologia trabalham com escoamentos naturais de fluidos; ciências médicas também têm relações com escoamentos vista a natureza de processos como a respiração e circulação sanguínea; e no setor de energia, a maior parte da energia elétrica gerada no mundo depende do escoamento de água ou vapor através de uma turbina [WHITE, 2011].

Vista a presença e importância dos fluidos na vida humana, surgiu a necessidade de um desenvolvimento mais elaborado e sofisticado de nosso conhecimento a respeito deste tema, chamado Mecânica dos Fluidos. Cientistas como Leonhard Euler, Osborne Reynolds, Daniel Bernoulli, Claude-Louis Navier e George Stokes realizaram contribuições que nos permitiram construir desde carros e navios mais economicamente eficientes até mesmo máquinas extraordinariamente complexas como foguetes e ônibus espaciais [WHITE, 2011].

Apesar de diversos avanços na Mecânica dos Fluidos, ainda há lacunas como o problema da turbulência. Seu aspecto não-linear e caótico torna difícil sua modelagem e praticamente impossível o desenvolvimento de solução analítica. Assim, modelos computacionais como CFD (dinâmica de fluidos computacional) surgiram como alternativa para contornar estes obstáculos. Entretanto, estes modelos enfrentam dificuldades quanto à quantidade de dados necessária para que uma simulação consiga um resultado preciso, de forma que o custo de aquisição torna-se proibitivo. Ainda sobre a aquisição de um conjunto de dados, tirar medidas de um escoamento pode ser uma tarefa difícil visto que inserir um instrumento no sistema pode alterar a própria natureza do escoamento [WHITE, 2011].

Alternativas para contornar a necessidade de um conjunto de dados extenso surgiram valendo-se de avanços da área da Inteligência Artificial. Seu princípio de funcionamento, que será mais bem explicado adiante, consiste em aprender a relacionar dados de entrada, como os valores de velocidade de uma amostra de pontos, com dados de saída,

como o número de Reynolds associado e o campo de pressões. Esta relação utiliza uma amostra dos pontos para treino e outra para validação do método. No entanto, o procedimento de relacionar dados de entrada com dados de saída sofre também, por vezes, de demorado tempo para executar o código com um número razoável de iterações para obter resultados confiáveis. [CHOLLET, 2021]

A técnica estudada neste trabalho, chamada Redes Neurais Vinculadas pela Física, ou PINN (*Physics-informed Neural Networks*) propõe justamente uma solução para diminuir o tempo de iteração e a quantidade de dados necessários, utilizando como auxílio leis físicas, como conservação de massa. A utilização de leis físicas neste método alcançou, inclusive, um resultado surpreendente que foi a reconstrução do campo de pressões de um escoamento incompressível, bidimensional, de um fluido newtoniano externo a um cilindro sem possuir informação inicial a respeito desta grandeza. [RAISSI; PERDIKARIS; KARNIADAKIS, 2019]

1.1 Objetivos

Os objetivos principais deste trabalho são aplicar a técnica PINN para predição do coeficiente de difusão na equação de Burgers a partir do campo de velocidade; identificar o número de Reynolds e reconstruir o campo de pressões na equação de Navier-Stokes sem dimensão para um escoamento incompressível, bidimensional, de um fluido newtoniano externo a um cilindro a partir do campo de velocidades.

Os objetivos específicos são os seguintes:

- Fazer uma breve recapitulação de Mecânica dos Fluidos a respeito do desenvolvimento analítico das equações de Navier-Stokes, de tópicos sobre redes neurais e do artigo que apresenta a técnica PINN;
- Aplicação do código da PINN para prever o coeficiente de difusão na equação de Burgers a partir do campo de velocidade;
- Aplicação do código da PINN para prever o número de Reynolds e obter a reconstrução do campo de pressões nas equações de Navier-Stokes adimensional para um escoamento incompressível, bidimensional, de um fluido newtoniano externo a um cilindro a partir do campo de velocidades.

1.2 Justificativa

Sistemas governados pela mecânica dos fluidos são complexos e nem sempre é simples realizar a medição de seus parâmetros, de forma que a aquisição de dados se torna custosa ou mesmo impossível. Assim, surge a demanda por alternativas que consigam extrair informações a respeito de um escoamento a partir de uma quantidade restrita de dados. A técnica estudada neste trabalho apresenta um potencial considerável de diminuir tempo e custo de análises fluidodinâmicas visto que foi capaz de recuperar informações de interesse como o número de Reynolds e o campo de pressões de um escoamento incompressível, bidimensional, externo a um cilindro a partir de apenas 5000 pontos da amostra de um milhão de dados fornecidos referentes ao campo de velocidade.

1.3 Organização do Trabalho

Este trabalho está dividido da seguinte forma. No Capítulo 1 há uma introdução sobre o tema e objetivos do trabalho. O Capítulo 2 destina-se à revisão da literatura de Mecânica dos Fluidos, Inteligência Artificial no contexto deste trabalho e apresentação da técnica PINN. No Capítulo 3 é apresentada a metodologia empregada para o ajuste dos parâmetros da equação de Burgers e das equações de Navier-Stokes. O Capítulo 4 apresenta os resultados obtidos da implementação da técnica para a equação de Burgers e para as equações de Navier-Stokes. Por fim, no Capítulo 5 estão as conclusões deste trabalho.

2 Fundamentação Teórica

2.1 Mecânica dos Fluidos

2.1.1 Introdução

A fundamentação teórica sobre Mecânica dos Fluidos será baseada no livro Mecânica dos Fluidos de Frank White [WHITE, 2011]. O objetivo desta seção é expor o desenvolvimento analítico das equações de Navier-Stokes (N-S) em sua forma adimensional para apresentar os parâmetros que serão previstos pela PINN. Problemas de mecânica dos fluidos são, em geral, analisados utilizando a descrição euleriana que lida com o campo de escoamento, ou seja, trabalha-se com o padrão de escoamento e não as variações instantâneas de cada partícula movendo-se pelo campo. Logo, as leis físicas são escritas para um volume de controle (VC). As equações de N-S advêm da análise diferencial das leis básicas de conservação da massa e conservação de quantidade de movimento linear aplicadas a um VC infinitamente pequeno, isto é, um sistema fluido infinitesimal.

Este trabalho aborda um escoamento incompressível, bidimensional de um fluido newtoniano externo a um cilindro. Tecnicamente um fluido é definido como uma substância que se deforma continuamente enquanto submetida a uma tensão cisalhante. O termo incompressível indica que a massa específica de cada partícula é invariante, admitindo que a massa se conserva. O termo newtoniano diz respeito ao comportamento reológico do fluido, isto é, o ângulo de deformação $\delta\theta$ devido ao cisalhamento cresce continuamente com o tempo enquanto a tensão de cisalhamento τ for mantida. Para fluidos newtonianos, a tensão de cisalhamento é proporcional à taxa de deformação angular que, por sua vez, equivale ao gradiente de velocidade. Esta constante de proporcionalidade é chamada de coeficiente de viscosidade dinâmica, μ :

$$\tau \propto \frac{\delta\theta}{\delta t} \quad \therefore \quad \tau = \mu \frac{d\theta}{dt} = \mu \frac{du}{dy}. \quad (2.1)$$

Uma grandeza associada e comumente empregada chama-se viscosidade cinemática, $\nu = \mu/\rho$. A viscosidade mede quantitativamente a resistência de um fluido ao escoamento e pode ser explicada por duas ideias: coesão entre moléculas e velocidade de transferência de quantidade de movimento molecular entre as camadas de um fluido. Essa grandeza aumenta a complexidade das equações básicas e tem um efeito desestabilizador, originando um fenômeno de natureza desordenada e aleatória chamado de turbulência. Ao longo do século XIX, Claude-Louis Navier e George Stokes acrescentaram os termos viscosos newtonianos às equações do movimento desenvolvidas por Euler no século XVIII.

As equações resultantes passaram então a ser chamadas de equações de Navier-Stokes.

Conforme mencionado anteriormente, a análise de fluidos não requer, em geral, que seja seguido o trajeto completo de uma partícula específica do fluido. Portanto as leis básicas devem ser reescritas para que se apliquem a um VC. A fronteira do volume chama-se superfície de controle (SC). A técnica matemática para transformar análise de sistema em volume de controle é o Teorema de Transporte de Reynolds.

Seja B uma propriedade extensiva qualquer do fluido (massa, quantidade de movimento, etc.), e seja $\beta = dB/dm$ a grandeza intensiva correspondente, definida pela quantidade de B por unidade de massa em qualquer porção pequena do fluido. Seja também o vetor unitário \vec{n} o vetor que aponta para fora, perpendicular em qualquer ponto, de uma superfície de controle de entrada ou saída. No limite quando $dt \rightarrow 0$, a variação instantânea de B no sistema é a soma de sua variação no interior do VC, mais o seu fluxo que sai, menos o seu fluxo que entra:

$$\frac{d}{dt}(B_{sist}) = \frac{d}{dt} \left(\int_{VC} \beta \rho dV \right) + \int_{SC} \beta \rho (\vec{V} \cdot \vec{n}) dA. \quad (2.2)$$

Este é o Teorema de Transporte de Reynolds para um volume de controle fixo arbitrário, onde V corresponde à notação utilizada para a grandeza volume.

2.1.2 Forma integral da lei de conservação da massa

Para a conservação da massa, $B = m$ e $\beta = dm/dm = 1$. Logo a Equação 2.2 torna-se, para um VC fixo, com número de entradas e saídas unidimensionais:

$$\left(\frac{dm}{dt} \right)_{sist} = \frac{d}{dt} \left(\int_{VC} \rho dV \right) + \sum_i (\rho_i A_i V_i)_{sai} - \sum_i (\rho_i A_i V_i)_{ent} = 0. \quad (2.3)$$

2.1.3 Forma integral da lei de conservação da quantidade de movimento linear

Para esta lei tem-se que $\vec{B} = m\vec{V}$ e que $\beta = d\vec{B}/dm = \vec{V}$. Portanto a Equação 2.2 torna-se para um VC fixo e entradas e saídas unidimensionais, onde $\dot{m} = \rho VA$:

$$\frac{d}{dt}(m\vec{V})_{sist} = \sum \vec{F} = \frac{d}{dt} \left(\int_{VC} \vec{V} \rho dV \right) + \sum_i (\dot{m}_i \vec{V}_i)_{sai} - \sum_i (\dot{m}_i \vec{V}_i)_{ent}, \quad (2.4)$$

em que o termo $\sum \vec{F}$ corresponde à soma vetorial das forças atuantes no VC.

2.1.4 A equação diferencial da conservação da massa

Aplicando a relação descrita na Equação 2.3 para um elemento pequeno o suficiente, a integral do volume será reduzida a um cubo diferencial. Os termos de fluxo ocorrem nas seis faces, de forma que, por exemplo, se ρu for conhecido na face esquerda do elemento diferencial, seu valor na face direita será $\rho u + (\partial \rho u / \partial x) dx$. Assumindo escoamento incompressível, isto é, com ρ constante e introduzindo estes termos na Equação 2.3 obtém-se a equação da continuidade para escoamento incompressível, ou seja, sob condição de incompressibilidade:

$$\vec{\nabla} \cdot \vec{V} = 0. \quad (2.5)$$

2.1.5 A equação diferencial da quantidade de movimento linear

As forças que agem sobre os VC consideradas neste trabalho serão a força de campo gravitacional e as forças de superfície originadas pelas tensões de compressão e tensões viscosas. Assume-se a hipótese de fluido newtoniano conforme discutido na seção 2.1.1 para escoamento incompressível de maneira a incluir o coeficiente de viscosidade dinâmica apresentado na equação 2.1 no termo de tensões cisalhantes. Aplicando um tratamento na Equação 2.4 similar ao aplicado na Equação 2.3 no item anterior, obtém-se o resultado:

$$\rho \frac{d\vec{V}}{dt} = \rho \vec{g} - \vec{\nabla} p + \mu \nabla^2 \vec{V}. \quad (2.6)$$

As equações 2.5 e 2.6 são as Equações de Navier-Stokes para o escoamento incompressível de um fluido newtoniano.

2.1.6 Função corrente

A função corrente ψ é uma ferramenta que permite satisfazer a equação da continuidade 2.5 e resolver a equação da quantidade de movimento 2.6 diretamente para a variável ψ . Esta ferramenta, no entanto, funciona quando a equação da continuidade pode ser reduzida a apenas dois termos. É o caso de escoamento bidimensional em que a função $\psi(x, y)$ é definida da seguinte maneira:

$$u = \frac{\partial \psi}{\partial y} \quad v = -\frac{\partial \psi}{\partial x} \quad \text{ou} \quad \vec{V} = \vec{i} \frac{\partial \psi}{\partial y} - \vec{j} \frac{\partial \psi}{\partial x}. \quad (2.7)$$

A função corrente será utilizada pela PINN para satisfazer automaticamente a equação da continuidade 2.5.

2.1.7 Análise dimensional

A adimensionalização das equações básicas de conservação da massa e quantidade de movimento na forma diferencial é importante para o ganho de generalidade na formulação matemática dos problemas e para uma exposição mais compacta de dados experimentais e dados numéricos. Além disso há a identificação de parâmetros adimensionais relevantes para o fenômeno em análise, como é o caso do número de Reynolds $Re = \rho UL/\mu = UL/\nu$. Este parâmetro pode ser interpretado como a contraposição de fluxos de quantidade de movimento com forças viscosas, onde U e L correspondem, respectivamente, à velocidade representativa do fenômeno e comprimento representativo do fenômeno. Valores baixos costumam indicar escoamento laminar e valores altos indicam, em geral, escoamento turbulento. Desta forma, este parâmetro é de grande importância para a caracterização de escoamentos e este é justamente um dos resultados que a PINN pode fornecer.

Aqui serão expostos os resultados finais que são obtidos na adimensionalização das equações básicas, em que os termos com * indicam os termos adimensionalizados. Os detalhes a respeito do tratamento teórico e desenvolvimento das técnicas necessárias para a aplicação desta análise podem ser consultados na referência [WHITE, 2011].

$$\vec{\nabla}^* \cdot \vec{V}^* = 0, \quad (2.8)$$

$$\frac{\partial \vec{V}^*}{\partial t^*} + (\vec{V}^* \cdot \vec{\nabla}^*) \vec{V}^* = -\vec{\nabla}^* p^* + \frac{1}{Re} \nabla^{*2}(\vec{V}^*). \quad (2.9)$$

Será adotada a notação dos autores [RAISSI; PERDIKARIS; KARNIADAKIS, 2019], onde subscritos em u_x , por exemplo, indicam diferenciação em relação a espaço ou tempo. Desta forma, a equação da continuidade adimensional 2.8 será inserida na PINN com o seguinte formato:

$$u_x + v_y = 0. \quad (2.10)$$

Analogamente, as equações da quantidade de movimento sem dimensão 2.9 serão modeladas como,

$$\begin{aligned} u_t + (uu_x + vv_y) &= -p_x + \frac{1}{Re}(u_{xx} + u_{yy}), \\ v_t + (uv_x + vv_y) &= -p_y + \frac{1}{Re}(v_{xx} + v_{yy}). \end{aligned} \quad (2.11)$$

Assim, o alvo da PINN será encontrar qual é o número de Reynolds que melhor se ajusta aos dados relativos ao campo de velocidades inseridos no código como vetor de entrada.

2.1.8 Equação de Burgers

Antes de implementar o código para as N-S será ilustrado um exemplo de natureza mais simples, envolvendo a Equação de Burgers, por vezes também chamada de Bateman-Burgers. Esta equação, estudada por Harry Bateman em 1915 e posteriormente por J. M. Burgers em 1948 pode ser derivada a partir das N-S para o campo de velocidade através da remoção do termo de pressão. Trata-se de uma análise espacialmente unidimensional e se encontra em diversas aplicações como gasodinâmica, acústica não-linear e fluxo de tráfego. [BATEMAN, 1915]; [BURGERS, 1948].

A equação de Burgers em uma dimensão espacial é escrita da seguinte forma

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (2.12)$$

onde $u(t, x)$ é a velocidade ao longo da direção do escoamento e $\nu = \mu/\rho$ é a viscosidade cinemática, podendo também ser chamada de coeficiente de difusão.

Adotando a notação dos autores [RAISSI; PERDIKARIS; KARNIADAKIS, 2019], a Equação de Burgers pode ser escrita da seguinte maneira

$$u_t + uu_x - \nu u_{xx} = 0. \quad (2.13)$$

Para esta equação, o alvo da PINN será analisar um conjunto de dados referente ao campo de velocidade $u(t, x)$ e prever qual é o valor do coeficiente de difusão ν que poderia ter gerado tal campo de velocidade, além de identificar o valor correto do coeficiente que acompanha o termo convectivo, isto é, 1.

2.2 Inteligência Artificial

2.2.1 Introdução

A fundamentação teórica sobre Inteligência Artificial (IA) será feita de acordo com o livro *Deep Learning with Python* de François Chollet [CHOLLET, 2021]. Esta área nasceu nos anos 1950 propondo-se a estudar a possibilidade de computadores serem ou não capazes de pensar, sendo portanto uma sub-área da ciência da computação. Pode ser descrita como um esforço para automatizar tarefas intelectuais desempenhadas normalmente por humanos. Sendo assim, IA é um campo que engloba aprendizado de máquina (AM), e aprendizado profundo (AP), e inclusive outras abordagens que não envolvem aprendizado.

Dos anos 1950 até 1980, especialistas acreditavam que a única maneira de se alcançar uma inteligência artificial em um nível humano seria através da abordagem chamada IA Simbólica. Trata-se da criação de um conjunto de regras explícitas suficientemente grandes para manipular conhecimento armazenado em bancos de dados. Apesar da IA Simbólica ser adequada para a solução de problemas lógicos bem definidos tais como uma partida de xadrez, provou-se inviável para descobrir regras explícitas para a solução de problemas mais complexos, como reconhecimento de fala e classificação de imagens. A alternativa para essa limitação surgiu com a abordagem chamada Aprendizado de Máquina.

2.2.2 Aprendizado de Máquina

Sistemas de Aprendizado de Máquina não são explicitamente programados, mas sim treinados. Primeiramente, apresentam-se exemplos relevantes para a tarefa e, em seguida, o sistema encontra uma estrutura estatística ou padrão que o possibilita determinar regras para automatizar a tarefa. Um exemplo são sistemas que automatizam classificação de imagens, em que humanos rotulam os primeiros exemplos e depois a máquina procura regras estatísticas para associar as imagens aos rótulos.

Para se realizar AM são necessários três elementos:

- Dados de entrada, ou do inglês *input data*: Se a tarefa for catalogar imagens, por exemplo, estes dados podem ser fotografias;
- Exemplos de saída esperadas, ou do inglês *expected output*: Na tarefa sobre imagens, podem ser rótulos para as fotografias tais como, "cachorro", "gato", etc;
- Uma maneira de mensurar o desempenho do algoritmo: Este elemento determina o quão distante o *output* atual está do *output* esperado. Esta medida será usada como um sinal de *feedback* para ajustar o funcionamento do algoritmo. Esse passo de ajuste é o aprendizado.

Quando um modelo de AM recebe dados de *input* ele procura representar esses dados de maneira mais conveniente, um processo que é aprendido através da exposição a exemplos corretos de associação entre *inputs* e *outputs*. Por representação, pode-se entender como uma maneira de interpretar os dados, por exemplo, quando o modelo recebe imagens, estas podem ser codificadas de duas maneiras: formato RGB, do inglês *red-green-blue* ou no formato HSV, do inglês *hue-saturation-value*. Caso a tarefa seja selecionar todos os pixels de uma determinada cor, o formato RGB é mais adequado, enquanto se a tarefa for diminuir a saturação da imagem, é mais fácil trabalhar com o HSV. Essencialmente, um modelo de AM procura quais representações serão mais apropriadas para a tarefa presente.

2.2.3 Aprendizado de Máquina Profundo

Aprendizado profundo é uma sub-área específica de AM cuja ênfase está no aprendizado de sucessivas camadas de representações cada vez mais convenientes e adequadas. O termo *profundo* diz respeito à ideia de sucessivas camadas de representação. A profundidade do modelo indica a quantidade de camadas que contribuem com o modelo. Modelos atuais de AP envolvem frequentemente dezenas ou mesmo centenas de camadas sucessivas de representação e estas são aprendidas automaticamente através da exposição a dados de treinamento.

Estas representações em camadas são aprendidas através de modelos chamados de redes neurais, que são estruturadas em camadas uma seguida da outra. Apesar da terminologia, não há relação com a neurobiologia, isto é, não trata-se de um modelo do funcionamento do cérebro. Cada neurônio da rede neural pode ser entendido como uma função que recebe um *input* e gera um *output* de forma que a próxima camada receberá o *output* da camada anterior e irá transformá-lo sucessivamente.

Cada camada possui um conjunto de valores associados chamados pesos e diz-se que a transformação implementada entre cada camada é parametrizada pelos seus respectivos pesos. Os pesos agem nas ligações entre as camadas de forma a parametrizar as transformações dos dados. Desta maneira, o procedimento de aprendizagem implica em encontrar os valores dos pesos entre cada camada de forma que a rede mapeie corretamente cada *input* ao seu alvo correto. Para mensurar o quão distante o *output* está do valor esperado emprega-se a função de perda. Essa função basicamente mostra uma pontuação do quão longe foi a previsão da rede em relação ao valor correto. O cerne do funcionamento do AP está em usar esta pontuação como um sinal de *feedback* para ajustar os valores dos pesos na direção que vai diminuir a pontuação de perda. Esse ajuste é feito pelo otimizador que emprega um algoritmo chamado de retropropagação. Inicialmente, atribui-se valores aleatórios aos pesos da rede, de forma que o a pontuação de perda começa muita alta. No entanto, a cada exemplo que a rede processa, os pesos são ajustados aos poucos

e a pontuação diminui. Cada iteração, chamada de época, gera valores de pesos que minimizam a função de perda. Uma rede neural com perda mínima é aquela cujos *outputs* estão o mais próximo possível de seus alvos, isto é, uma rede treinada.

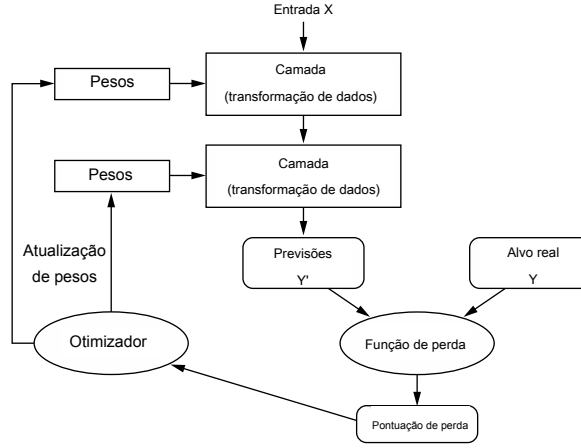


Figura 1 – Relações entre a rede, camadas, função de perda e otimizador. A pontuação de perda é usada como sinal de *feedback* para ajustar os pesos [CHOLLET, 2021].

Matematicamente, em um modelo de AP ocorre uma operação matricial entre os dados de *input* e os pesos, podendo-se enunciar-la da seguinte forma:

$$output = \tanh\left([input] \cdot [W] + [b]\right), \quad (2.14)$$

onde:

- W e b são tensores atributos da camada. Esses são os pesos ou parâmetros treináveis da camada, em que o tensor b chama-se *bias*;
- \tanh é uma função de ativação, que no caso da PINN é a função tangente hiperbólica. Uma função de ativação entra em uma rede neural para ajudá-la a aprender padrões complexos. Ela recebe o sinal de *output* da camada anterior e o converte em um formato que possa ser recebido como um *input* pela camada posterior. Além disso, a função de ativação inclui o elemento não linear, uma vez que os neurônios só são capazes de executar operações lineares.

Inicialmente, os valores de W e b são aleatórios e em seguida são gradualmente ajustados baseado no sinal de *feedback* fornecido pela função de perda e pelo algoritmo de retropropagação. Esse é, em essência, o processo de treinamento. A Figura 2 ilustra esse processo. A matriz de *inputs* X é multiplicada pela matriz de pesos W e somada à matriz de *bias* b . Em seguida, uma função de ativação, que no caso da ilustração é a função *relu*, age sobre esse *output*, conforme ilustrado pela equação 2.14. O valor resultante entra como *input* na próxima camada e sofre operações com a matriz de pesos análogos à

camada anterior. Por fim, o *output* da última camada entra na última função de ativação, que no caso do exemplo é a *softmax*, e finalmente é comparado com o valor esperado para a computação da pontuação de perda. Existem diversos algoritmos que fazem esse trabalho de retropropagação e no caso da PINN é utilizada a diferenciação automática. Para uma discussão mais aprofundada a respeito da diferenciação automática, recomenda-se a leitura dos seguintes trabalhos: [BAYDIN; PEARLMUTTER; SISKIND, 2018] e [LU et al., 2021].

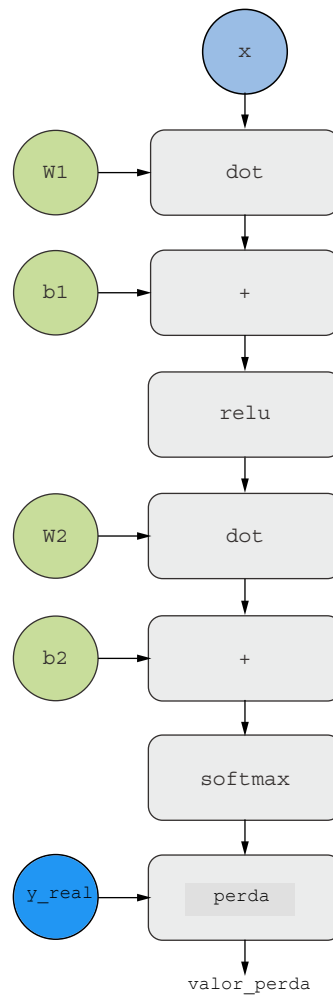


Figura 2 – Representação em grafo computacional das operações matemáticas realizadas em uma iteração [CHOLLET, 2021].

2.2.4 Redes Neurais Vinculadas pela Física - PINN

As PINN, *Physics-informed Neural Networks*, são redes neurais treinadas para solucionar tarefas supervisionadas respeitando leis da física descritas em geral por equações diferenciais parciais não-lineares. Existem duas classes principais de problemas: obtenção da solução ou ajuste dos coeficientes de equações diferenciais parciais. Para contornar a limitação de trabalhar com poucos dados, empregam-se leis físicas que governam o fenômeno observado, como por exemplo no caso de escoamento incompressível em que descarta-se soluções de escoamento não realísticas que violam o princípio da conservação da massa. Desta forma, codificar esta informação na rede capacita o algoritmo a se direcionar mais rapidamente para a solução correta e inclusive generalizar mesmo quando poucos exemplos de treinamento são disponíveis. [RAISSI; PERDIKARIS; KARNIADAKIS, 2019].

Estas redes neurais são limitadas a respeitar quaisquer leis de simetria, invariância ou princípios de conservação originados das leis físicas que governam os dados observados, modelados por equações diferenciais parciais não lineares dependentes do tempo. De acordo com a notação dos autores [RAISSI; PERDIKARIS; KARNIADAKIS, 2019], as equações diferenciais parciais não lineares serão escritas no seguinte formato:

$$u_t + \mathcal{N}[u; \lambda] = 0, \quad x \in \Omega, \quad t \in [0, T], \quad (2.15)$$

onde $u(t, x)$ denota a solução latente, $\mathcal{N}[\cdot; \lambda]$ é um operador não linear parametrizado por λ , e Ω é um subconjunto de \mathbb{R}^D . Esta configuração engloba uma gama de problemas em física envolvendo leis de conservação, processos de difusão, etc.

O presente trabalho consiste na implementação da segunda classe de problemas, isto é, a obtenção dos coeficientes dos termos da equação de Burgers e das equações de Navier-Stokes que melhor se ajustam aos dados fornecidos. Os códigos utilizados para a geração dos resultados estão disponíveis no repositório on-line GitHub: <<https://github.com/Morbidelli/TFG-PINN-TermPaper>>.

2.2.5 Equação de Burgers

A primeira aplicação da PINN será na equação de Burgers, ou seja, um exemplo espacialmente unidimensional. O objetivo é obter valores dos parâmetros λ que melhor modelam um campo de velocidade $u(t, x)$ fornecido. Para uma dimensão espacial, a equação de Burgers será descrita no seguinte formato:

$$u_t + \lambda_1 u u_x - \lambda_2 u_{xx} = 0. \quad (2.16)$$

Define-se a função $f(t, x)$ dada por:

$$f := u_t + \lambda_1 u u_x - \lambda_2 u_{xx}, \quad (2.17)$$

e prosseguimos aproximando $u(t, x)$ através de uma rede neural. Isto resulta na rede neural vinculada pela física $f(t, x)$. Os parâmetros das operações lineares realizadas pelas redes neurais $u(t, x)$ e $f(t, x)$ junto com λ_1 e λ_2 do operador diferencial podem ser aprendidos através da minimização do erro quadrático médio, MSE , do inglês *mean squared error*, da função de perda descrita pela equação 2.18:

$$MSE = MSE_u + MSE_f, \quad (2.18)$$

onde

$$MSE_u = \frac{1}{N} \sum_{i=1}^N |u(t_u^i, x_u^i) - u^i|^2, \quad (2.19)$$

e

$$MSE_f = \frac{1}{N} \sum_{i=1}^N |f(t_u^i, x_u^i)|^2. \quad (2.20)$$

Aqui, $\{t_u^i, x_u^i, u^i\}_{i=1}^N$ denotam os dados de treinamento a respeito de $u(t, x)$. A perda MSE_u descrita pela equação 2.19 corresponde ao dados de treinamento de $u(t, x)$ enquanto MSE_f descrita pela equação 2.20 impõe a estrutura estabelecida pela equação 2.16. Esta imposição é feita verificando se os *outputs* da rede neural conseguem recuperar os valores fornecidos como *input*, ou seja, através da validação dos resultados. Nota-se que comparando a equação 2.16 com a equação de Burgers descrita pelas equações 2.12 e 2.13, o valor de λ_1 deve convergir para 1, enquanto o valor de λ_2 deve corresponder ao valor do coeficiente de difusão ν .

Os testes referentes à equação de Burgers serão feitos criando-se um conjunto de treino aleatório de $N = 2000$ pontos ao longo de todo o domínio espaço-temporal do

conjunto de dados fornecidos, correspondentes ao parâmetro $\lambda_1 = 1$ que multiplica o termo convectivo e quatro valores para o parâmetro λ_2 que multiplica o termo difusivo, de forma que $\lambda_2 = n \cdot 0.001/\pi$, $n \in \{1, 10, 100, 1000\}$. Além disso, para cada exemplo são analisados dois casos: um sem ruído nos dados e outro em que os dados são corrompidos com um ruído de 1%.

2.2.6 Equações de Navier-Stokes

Para a segunda aplicação da PINN será abordado um exemplo espacialmente bi-dimensional envolvendo as equações de Navier-Stokes para o escoamento incompressível de um fluido newtoniano. As N-S nesse contexto são escritas no seguinte formato:

$$\begin{aligned} u_t + \lambda_1(uu_x + vv_y) &= -p_x + \lambda_2(u_{xx} + u_{yy}), \\ v_t + \lambda_1(uv_x + vv_y) &= -p_y + \lambda_2(v_{xx} + v_{yy}), \end{aligned} \quad (2.21)$$

$$u_x + v_y = 0,$$

onde $u(t, x, y)$ denota a componente x do campo de velocidades, $v(t, x, y)$ a componente y , e $p(t, x, y)$ a pressão. Aqui λ_1 e λ_2 são os parâmetros desconhecidos. As soluções para esses parâmetros são procuradas no conjunto de funções sem divergência, isto é, respeitando a lei de conservação da massa descrita pela equação 2.10. Assume-se agora a representação expressa na equação 2.7, escrita com a notação de subscritos:

$$u = \psi_y, \quad v = -\psi_x, \quad (2.22)$$

para uma função latente $\psi(t, x, y)$, isto é, faz-se uso da função corrente descrita na seção 2.1.6 para satisfazer automaticamente a equação da continuidade 2.10.

Dadas as medições $\{t^i, x^i, y^i, u^i, v^i\}_{i=1}^N$ do campo de velocidades, o objetivo é ajustar os parâmetros λ assim como reconstruir o campo de pressões $p(t, x, y)$. Define-se portanto $f(t, x, y)$ e $g(t, x, y)$ dadas por:

$$\begin{aligned} f &:= u_t + \lambda_1(uu_x + vv_y) + p_x - \lambda_2(u_{xx} + u_{yy}), \\ g &:= v_t + \lambda_1(uv_x + vv_y) + p_y - \lambda_2(v_{xx} + v_{yy}), \end{aligned} \quad (2.23)$$

e prossegue-se aproximando $[\psi(t, x, y) \quad p(t, x, y)]$ usando uma única rede neural com dois *outputs*. Esta suposição junto com as equações 2.22 e 2.23 resultam na rede neural vinculada pela física $[f(t, x, y) \quad g(t, x, y)]$.

Os parâmetros λ do operador de Navier-Stokes juntos com os parâmetros das operações lineares realizados pela rede neural $[\psi(t, x, y) - p(t, x, y)]$ e $[f(t, x, y) - g(t, x, y)]$ podem ser aprendidos através da minimização do erro quadrático médio da função de perda descrita pela equação 2.24:

$$\begin{aligned} MSE := & \frac{1}{N} \sum_{i=1}^N \left(|u(t_u^i, x_u^i, y_u^i) - u^i|^2 + |v(t_u^i, x_u^i, y_u^i) - v^i|^2 \right) \\ & + \frac{1}{N} \sum_{i=1}^N \left(|f(t_u^i, x_u^i, y_u^i)|^2 + |g(t_u^i, x_u^i, y_u^i)|^2 \right). \end{aligned} \quad (2.24)$$

Para esse caso bidimensional será considerado o escoamento incompressível de um fluido newtoniano externo a um cilindro. Comparando a equação 2.11 com a equação 2.21, o objetivo é certificar que λ_1 converge para 1 enquanto λ_2 converge para o inverso do número de Reynolds associado ao escoamento.

Os testes referentes às equações de Navier-Stokes serão feitos criando-se um conjunto de treino aleatório de $N = 5000$ pontos ao longo de todo o domínio espaço-temporal do conjunto de dados fornecidos, correspondentes ao parâmetro $\lambda_1 = 1$ que multiplica o termo convectivo e três valores para o parâmetro λ_2 que multiplica o termo viscoso, de forma que $\lambda_2 = 1/Re$, $Re \in \{100, 500, 1000\}$. Para as N-S será analisado apenas o caso com dados sem ruído.

Uma surpresa a respeito desse método é que o campo de pressões $p(t, x, y)$ é reconstruído sem conhecimento de quaisquer dados a respeito da pressão para o treinamento da rede. A única utilidade dos dados de pressão importados pela PINN é para o cálculo do erro de predição dos valores da pressão.

3 Metodologia

3.1 Metodologia para a equação de Burgers

A PINN foi implementada utilizando os códigos escritos na linguagem Python disponibilizados no repositório on-line GitHub do autor Maziar Raissi: <<https://github.com/maziarraissi/PINNs>>, e compilados na plataforma Google Colab sem emprego de GPU. Esses códigos fazem uso do TensorFlow®, que é a principal biblioteca de código aberto para desenvolver e criar modelos de AM. No entanto, alguns recursos específicos utilizados pelos autores da PINN requerem uma versão específica do TensorFlow®, de forma que foi necessário fazer um ajuste no código implementado neste trabalho para não utilizar a última versão disponível da biblioteca, mas sim a utilizada pelos autores da PINN, que é a versão 1.15.2. Para maiores informações sobre a biblioteca e utilização do TensorFlow®, recomenda-se a leitura da referência [CHOLLET, 2021].

Para a validação da técnica, foi gerado neste trabalho um conjunto de dados do campo de velocidade resolvendo a equação de Burgers (2.12) utilizando o software Wolfram Mathematica®. A função empregada pelo software para solução de equações diferenciais é a *NDSolve* que utiliza o método dos elementos finitos. A condição inicial definida foi $u(0, x) = -\sin(\pi x)$ e as condições de contorno foram do tipo Dirichlet $u(t, -1) = 0$ e $u(t, 1) = 0$. A abrangência do método foi testada resolvendo a equação de Burgers para quatro valores de coeficiente de difusão ν em quatro ordens de grandeza, isto é, $\nu = n \cdot 0.001/\pi$, $n \in \{1, 10, 100, 1000\}$. A Figura 3 e Figura 4 ilustram, respectivamente, a superfície referente à solução $u(t, x)$ para $\nu = 0.01/\pi$ gerada no Mathematica® e a distribuição espaço-temporal do campo de velocidade $u(t, x)$ para $\nu = 0.01/\pi$. O *notebook* com os códigos estão disponíveis no repositório on-line GitHub deste trabalho: <<https://github.com/Morbidelli/TFG-PINN-TermPaper>>.

As soluções obtidas no Mathematica® foram exportadas em três arquivos no formato .dat: *usol.dat*, *t.dat*, *x.dat* correspondendo respectivamente às grandezas velocidade, instante de tempo e posição. Tratam-se de 101 instantes de tempo de $t = 0$ até $t = 1$ para 511 posições de $x = -1$ até $x = 1$ totalizando 51611 valores de velocidade. Deste total, foram sorteados 2000 valores de velocidade para treinar a rede neural e identificar λ_1 e λ_2 . A rede é composta por dez camadas tendo dois neurônios na primeira camada, vinte neurônios em cada camada intermediária e um neurônio na última camada.

A inserção desses arquivos de dados no código foi feita unindo-os em um único arquivo no formato .mat através do software MathWorks MATLAB®. Esse formato trata-se de uma variável matricial capaz de incluir diversas outras variáveis. Também é necessário

incluir no código o arquivo *plotting.py*, disponível no repositório on-line GitHub do autor Maziar Raissi, que corresponde às configurações estabelecidas pelo autor para a geração dos gráficos. Foi estabelecido nos códigos de identificação dos parâmetros da equação de Burgers um máximo de 2000 iterações para treinamento, pois este valor já foi capaz de gerar resultados com baixo valor de erro associado, isto é, menor do que 3% em geral.

A Figura 3 ilustra a solução $u(t, x)$ da Equação de Burgers para $\nu = 0.01/\pi$. Esta superfície descreve como os valores da velocidade u variam com respeito ao tempo t e à posição x . Esses valores de u serão importados pela PINN para a predição do valor de ν .

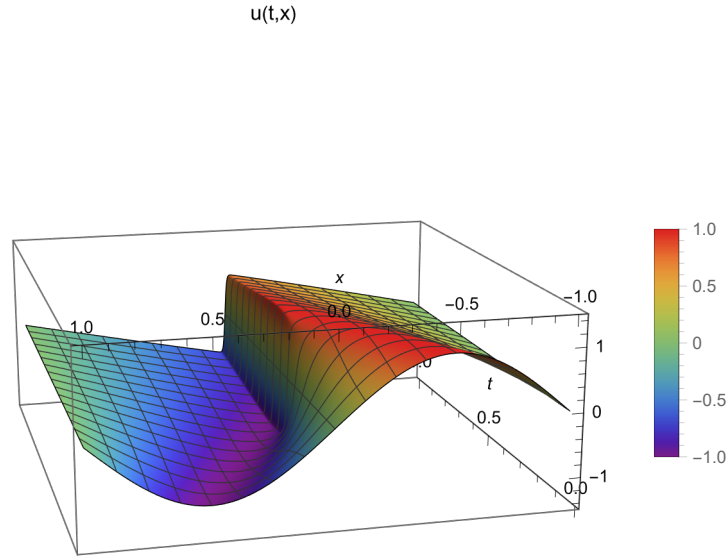


Figura 3 – Gráfico da solução $u(t, x)$ da equação de Burgers para $\nu = 0.01/\pi$ gerado no Mathematica®.

A Figura 4 descreve a distribuição espaço-temporal do campo de velocidade $u(t, x)$ mencionada anteriormente. Os resultados obtidos pela PINN ilustrarão quais pontos foram sorteados para o treinamento da rede, logo este gráfico será útil para a apresentação e identificação destes pontos sorteados.

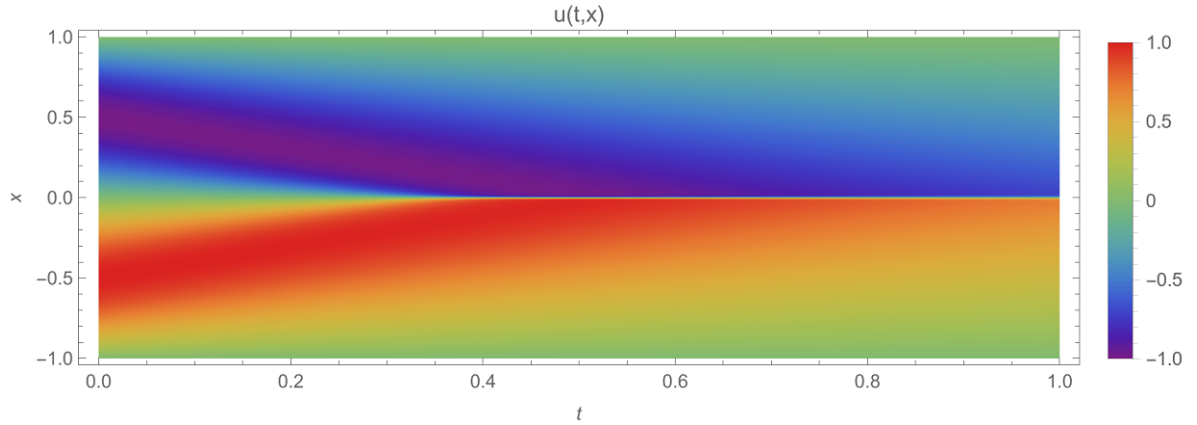


Figura 4 – Distribuição espaço-temporal do campo de velocidade $u(t, x)$ da equação de Burgers para $\nu = 0.01/\pi$ gerada no Mathematica®.

3.2 Metodologia para as equações de Navier-Stokes

A técnica foi aplicada para as N-S no Google Colab de maneira análoga à equação de Burgers. Foi utilizado o código também disponibilizado no repositório on-line GitHub do autor Maziar Raissi: <<https://github.com/maziarraissi/PINNs>>, fazendo o mesmo ajuste para implementar a biblioteca TensorFlow® adequada. As soluções das N-S para o campo de velocidades e campo de pressões foram geradas no software Wolfram Mathematica® para três escoamentos com número de Reynolds 100, 500 e 1000. Os três escoamentos foram incompressíveis, bidimensionais, externos a um cilindro e assumindo fluido newtoniano.

O código no Mathematica®, também disponível no repositório on-line GitHub deste trabalho em <<https://github.com/Morbidelli/TFG-PINN-TermPaper>>, simula um escoamento incompressível, bidimensional, externo a um cilindro de diâmetro adimensional 0.2 contido no domínio $[-1, 4] \times [-1, 1]$ conforme ilustrado na Figura 5. As N-S foram resolvidas empregando a função *NDSolve* que resolve equações diferenciais através do método dos elementos finitos empregando uma malha de 15894 nós gerada. Os tamanhos das áreas discretizadas foram estipulados pela função *MaxCellMeasure* em que foi atribuído o valor 0.0005. Este valor foi escolhido, pois foi capaz de gerar resultados tanto com precisão quanto com tempo de execução razoáveis. As condições iniciais foram $u(0, x, y) = 0$, $v(0, x, y) = 0$ e $p(0, x, y) = 0$. As condições de contorno foram do tipo Dirichlet, sendo um perfil de velocidade adimensional uniforme de corrente livre na fronteira esquerda $U_\infty = 1$, valor da pressão na fronteira direita $p = 0$ e condição de escorregamento nulo nas fronteiras superior e inferior, isto é, $\vec{V}(t, x, y) = 0$, $y \in \{-1, 1\}$, $-1 < x < 4$. Os instantes de tempo t analisados foram de 0 até 15, no entanto por questões numéricas foi necessário empregar uma função rampa no perfil de velocidades de forma a desenvolvê-lo ao longo dos 3 primeiros instantes de tempo. Não é exatamente claro o motivo do Mathematica® não aceitar que o perfil de velocidade já esteja desenvolvido no instante $t = 0$.

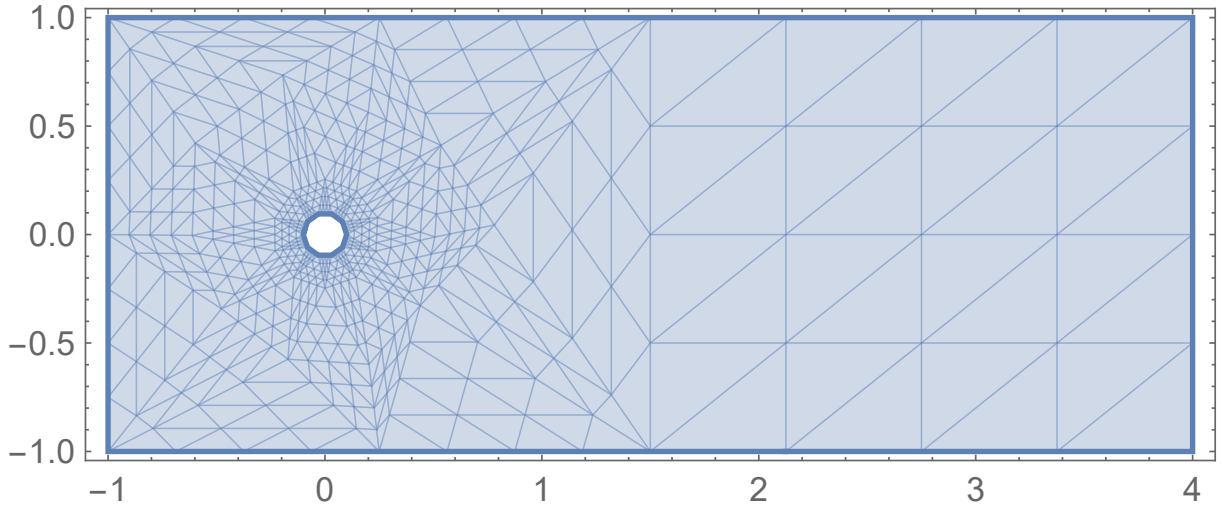


Figura 5 – Malha empregada para a solução das equações de Navier-Stokes com 15894 nós.

As soluções foram exportadas em quatro arquivos que são três planilhas no formato *.xlsx* correspondendo às duas componentes de velocidade u e v e à pressão p ; e um arquivo de texto no formato *.dat* correspondendo aos instantes de tempo t . As planilhas estão dispostas de forma que cada linha descreve um ponto de x , cada coluna um instante de tempo e cada aba um ponto de y .

Para a exportação das soluções foi considerado uma porção do domínio à jusante do cilindro, $[0.1, 1.5] \times [-0.5, 0.5]$, indicado na Figura 6. Foi estipulado que o software exportasse 100 pontos ao longo de x de 0.1 até 1.5 e 50 pontos ao longo de y de -0.5 até 0.5, isto é, 5000 pontos para 200 instantes de tempo t de 4 até 10. O total corresponde a um milhão de pontos de velocidade. Além disso, foi implementado nas planilhas um código na linguagem *Visual Basic for Applications*® (VBA) de forma a combinar os dados de todas as abas em apenas uma.

Para a PINN analisar corretamente esses dados, deve-se criar um arquivo que mapeia as grandezas u , v e p , isto é, que indica à qual posição (x, y) cada ponto diz respeito. Esse arquivo é chamado *X_star* e sua extensão é *.dat*. No repositório on-line GitHub desde trabalho há o código desenvolvido no Matlab® que gera o arquivo *X_star.dat* para quaisquer intervalos de tempo que se queira exportar as soluções.

Os arquivos referentes às velocidades u e v devem ser unidas em uma única variável chamada *U_star* e finalmente combinada com pressão p , instantes de tempo t e posições (x, y) em um único arquivo final com extensão *.mat*. Deve-se atentar que o nome escolhido para o arquivo final seja também ajustado no código na seção *Load Data*. Esses procedimentos devem ser seguidos para que a PINN importe corretamente cada variável. No GitHub deste trabalho também há o código escrito em Matlab® que cria o arquivo final que deve ser importado pela PINN.

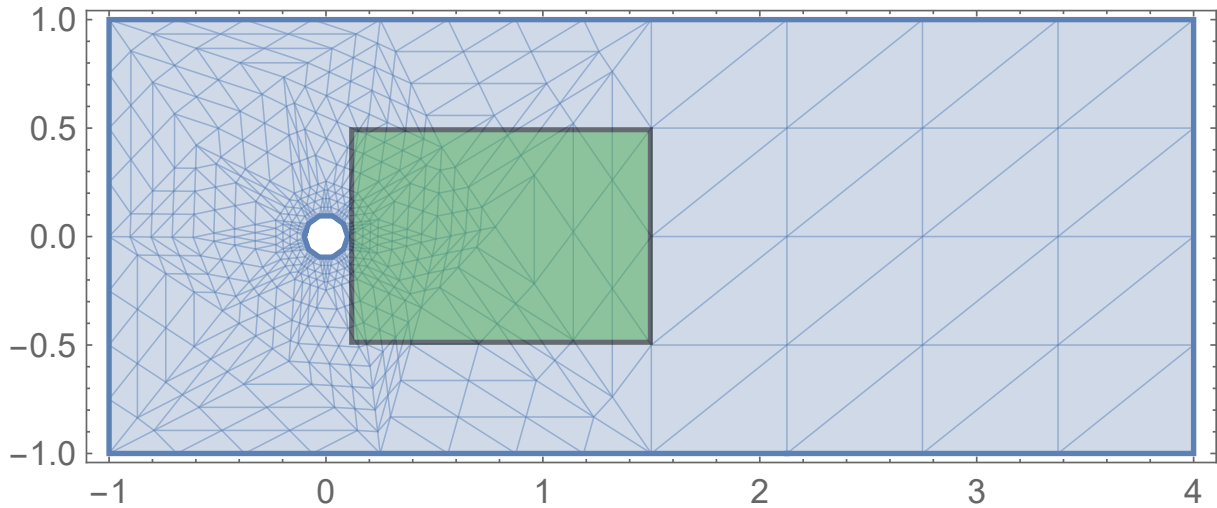


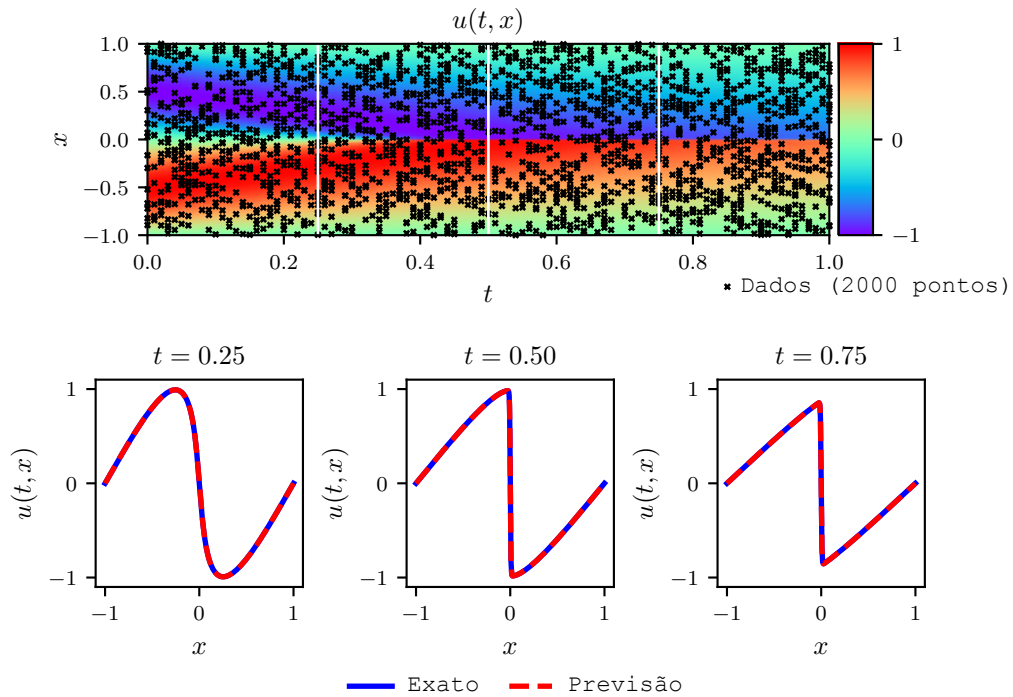
Figura 6 – Região do domínio que corresponde ao local das soluções importadas pela PINN.

Para as N-S foi estabelecido um limite de 10000 iterações para o treinamento da rede neural, pois foi um número capaz de gerar resultados com erros associados menores do que 2% para $Re = 500$. A rede é composta por dez camadas tendo três neurônios na primeira camada, vinte neurônios em cada camada intermediária e dois neurônios na última camada. A análise foi feita apenas dos dados sem ruído e do total de um milhão de pontos gerados no Mathematica® a rede neural sorteou 5000 para o seu treinamento.

4 Resultados

4.1 Resultados para a equação de Burgers

Os principais resultados obtidos pela PINN para o ajuste dos parâmetros λ a partir do campo de velocidade $u(t, x)$ referente à $\nu = 0.01/\pi$ são apresentados na Figura 7. Cada ponto preto na distribuição espaço-temporal do campo de velocidade $u(t, x)$ corresponde a um dos 2000 pontos sorteados do campo de velocidades utilizados no treino da rede neural. Abaixo, estão três perfis de velocidade correspondendo aos cortes transversais nos instantes de tempo $t = 0,25$, $t = 0,50$ e $t = 0,75$ sobrepondo os valores previstos em vermelho e os exatos em azul. Posteriormente, estão as tabelas que indicam qual é a equação diferencial parcial (EDP) correta que descreve os dados, junto com as que foram identificadas pelo código para os casos sem ruído e com ruído de 1%.



EDP Correta	$u_t + uu_x - 0.0031831u_{xx} = 0$
EDP Identificada (sem ruído)	$u_t + 0.99988uu_x - 0.0031826u_{xx} = 0$
EDP Identificada (1% de ruído)	$u_t + 0.99865uu_x - 0.0032198u_{xx} = 0$

Figura 7 – Resultado obtido pela PINN para o ajuste dos parâmetros λ_1 e λ_2 da Equação de Burgers para $\nu = 0,01/\pi$.

A PINN mostrou-se capaz de identificar os parâmetros λ da equação de Burgers para a maioria dos casos. A Tabela 1 descreve os valores dos erros de predição dos parâmetros λ_1 e λ_2 para os dados sem ruído. Os erros relacionados à identificação dos parâmetros podem ser interpretados como uma distância entre o valor correto que descreve os dados fornecidos e o valor identificado. Por exemplo, se o valor correto do parâmetro em análise é 100 e a rede neural identificá-lo como 98, o erro associado à essa identificação será de 2%. A fórmula 4.1, a seguir, descreve como é feito o cálculo do erro:

$$Erro(\%) = \frac{|(Y' - Y)|}{Y} \times 100, \quad (4.1)$$

onde: Y' é o valor do parâmetro identificado pela rede neural e Y é o valor real esperado do parâmetro.

Coeficiente de difusão, ν	Erro para λ_1	Erro para λ_2
$0,001/\pi$	0,03%	198%
$0,01/\pi$	0,01%	0,02%
$0,1/\pi$	0,06%	0,02%
$1/\pi$	2,06%	0,06%

Tabela 1 – Erros na identificação de λ_1 e λ_2 da equação de Burgers para dados sem ruído.

A Figura 8 ilustra a diferença entre os erros do parâmetro λ_1 para os dados sem ruído. Nota-se que o método foi mais eficiente para identificar os parâmetros a partir dos dados referentes aos coeficientes de difusão $\nu = 0,01/\pi$ e $\nu = 0,1/\pi$. Além disso, o pior resultado foi para o valor de maior coeficiente de difusão indicando um possível limite superior de aplicação da técnica.

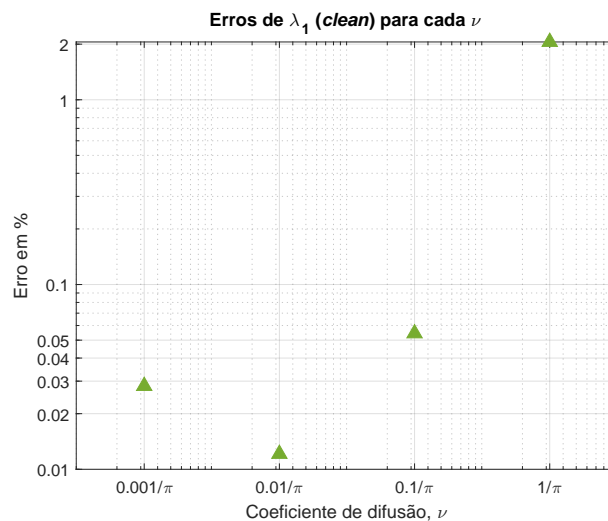


Figura 8 – Comparação entre os erros de λ_1 da equação de Burgers sem ruído para cada ν .

A Figura 9 mostra a diferença entre os erros do parâmetro λ_2 para os dados sem ruído. Apesar do erro de λ_1 para $\nu = 0,001/\pi$ ter sido menor do que o erro de λ_1 para $\nu = 0,1/\pi$, o erro de λ_2 do primeiro foi de uma ordem de grandeza substancialmente superior aos demais com cerca de 200%. O contrário ocorreu com $\nu = 1/\pi$ em que seu erro para λ_2 não foi exatamente alto, mas a ordem de grandeza do erro de λ_1 foi consideravelmente superior aos demais valores de coeficiente de difusão. O erro de cerca de 200% associado à identificação de um valor de viscosidade extremamente baixo como foi para o caso de $\nu = 0,001/\pi$ pode indicar que as soluções numéricas de problemas convecção-difusão tornam-se muito instáveis para escoamentos altamente convectivos, isto é, com difusão baixa. Desta forma, buscando estabilidade, algum tipo de difusão numérica é introduzida que, por consequência, pode falsear a identificação de λ_2 . Entretanto, o coeficiente do termo convectivo ainda está sendo relativamente bem identificado com 0.03% de erro que, para o caso invíscido, é o único que teria sentido.

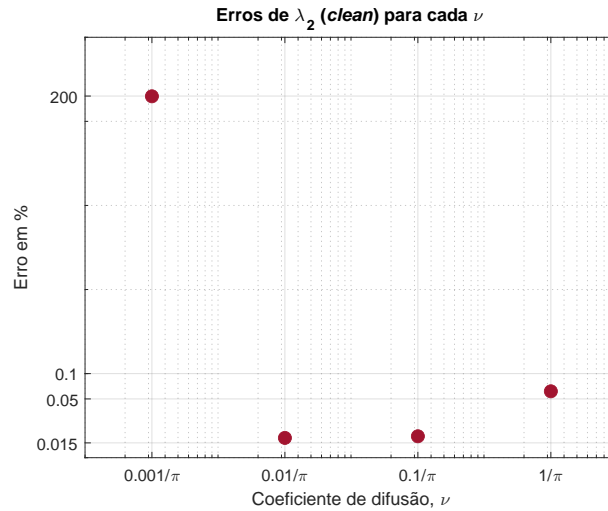


Figura 9 – Comparação entre os erros de λ_2 da equação de Burgers sem ruído para cada ν .

A Tabela 2 descreve os valores dos erros associados à predição dos parâmetros λ para os dados corrompidos com 1% de ruído. A análise dos dados corrompidos aponta queda de eficiência para quase todos os casos, exceto para a identificação dos parâmetros λ para $\nu = 0,1/\pi$.

Coeficiente de difusão, ν	Erro para λ_1	Erro para λ_2
$0,001/\pi$	0,19%	219%
$0,01/\pi$	0,13%	1,15%
$0,1/\pi$	0,01%	0,1%
$1/\pi$	0,95%	0,07%

Tabela 2 – Erros na identificação de λ_1 e λ_2 da equação de Burgers com 1% de ruído.

A Figura 10 ilustra a diferença entre os erros de λ_1 para os dados corrompidos. Dado o fato dos erros referente à $\nu = 0,1/\pi$ se comportarem de maneira diferente dos

demais, o efeito de ruído sobre os dados mostra-se imprevisível dependendo do valor de ν .

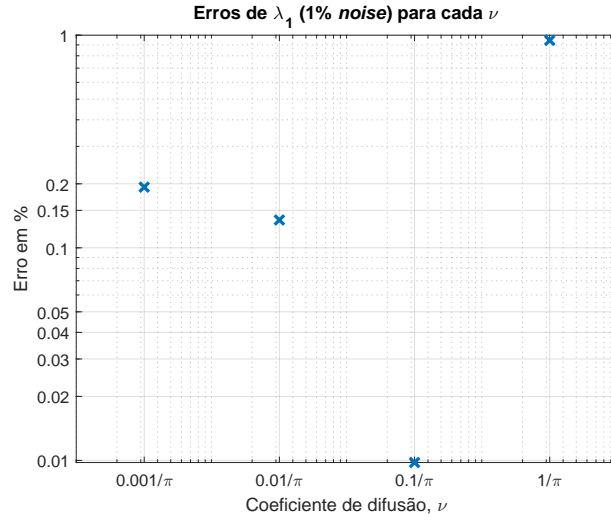


Figura 10 – Comparação entre os erros de λ_1 da equação de Burgers com 1% de ruído para cada valor de coeficiente de difusão, ν .

A Figura 11 mostra o comportamento dos erros para λ_2 para os dados corrompidos. Observa-se que houve diminuição da performance para todos os valores de coeficiente de difusão, ano entanto o aumento do erro de λ_2 para $\nu = 1/\pi$ foi mínimo, equivalente a 0.01%.

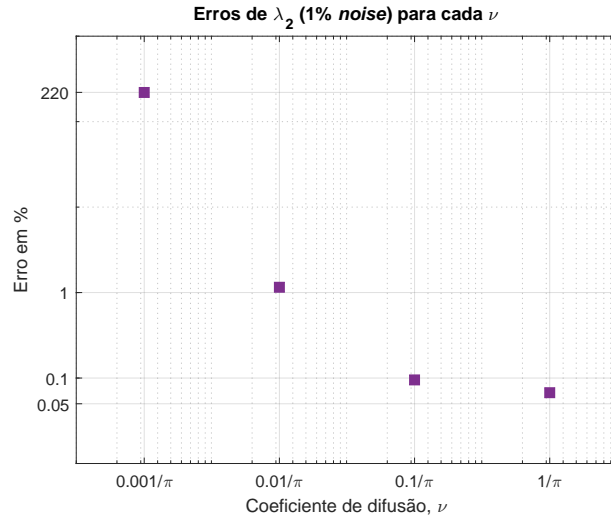


Figura 11 – Comparação entre os erros de λ_2 da equação de Burgers com 1% de ruído para cada valor de coeficiente de difusão, ν .

Outros dois testes de eficiência da aplicação da PINN à equação de Burgers foram realizados. O primeiro consistiu no estudo do efeito da remoção e adição de duas camadas de neurônios da rede neural nos erros de predição dos parâmetros λ para os dados sem ruído. A Tabela 3 ilustra o erro da rede em relação ao parâmetro λ_1 para oito, dez e doze camadas, respectivamente.

Efeito da adição e remoção de duas camadas no erro de λ_1			
Coeficiente de difusão, ν	8 camadas	10 camadas	12 camadas
$0,001/\pi$	0,02%	0,03%	0,19%
$0,01/\pi$	0,01%	0,01211%	0,01%
$0,1/\pi$	0,06%	0,05%	0,07%
$1/\pi$	25,4%	2,06%	61,6%

Tabela 3 – Comparação entre os erros de λ_1 da equação de Burgers para os dados sem ruído para redes com oito, dez e doze camadas de neurônios, respectivamente.

A Tabela 4 compara os valores dos erros para o parâmetro λ_2 quando se remove e adiciona duas camadas de neurônios da rede neural, respectivamente.

Efeito da adição e remoção de duas camadas no erro de λ_2			
Coeficiente de difusão, ν	8 camadas	10 camadas	12 camadas
$0,001/\pi$	185%	198%	309%
$0,01/\pi$	0,24%	0,02%	0,61%
$0,1/\pi$	0,08%	0,02%	0,02%
$1/\pi$	0,59%	0,06%	0,9%

Tabela 4 – Comparação entre os erros de λ_2 da equação de Burgers para os dados sem ruído para redes com oito, dez e doze camadas de neurônios, respectivamente.

O resultado desse teste mostra que o efeito da redução de camadas de neurônios foi eficaz em reduzir os erros dos parâmetros λ para o coeficiente de difusão $\nu = 0,001/\pi$. Além disso, também diminuiu o erro de λ_1 para $\nu = 0,01/\pi$, no entanto para todos os outros valores a quantidade original de dez camadas apresentou o melhor resultado.

O segundo teste explora o limite de aplicação da técnica ao modelar os parâmetros λ de outros valores de ν entre $\nu = 0,001/\pi$ e $\nu = 0,01/\pi$ para dados sem ruído. O objetivo é entender a diferença substancial dos erros de λ_2 entre esses valores de coeficiente de difusão. A Tabela 5 ilustra os valores dos erros de predição para quatro valores intermediários testados e mostra que de fato conforme os valores de ν aumentam os valores associados aos erros de predição dos parâmetros λ diminuem. Apesar da mudança entre os valores $\nu = 0,001/\pi$ e $\nu = 0,00175/\pi$ ser pequena, há uma melhora considerável nos valores dos erros o que indica que de fato o limite inferior de aplicação da PINN está em torno de $\nu = 0,001/\pi$.

Coeficiente de difusão, ν	Erro para λ_1	Erro para λ_2
$0,00175/\pi$	0,64%	81,6%
$0,0025/\pi$	0,45%	39,4%
$0,0050/\pi$	0,02%	1,99%
$0,0075/\pi$	0,09%	0,29%

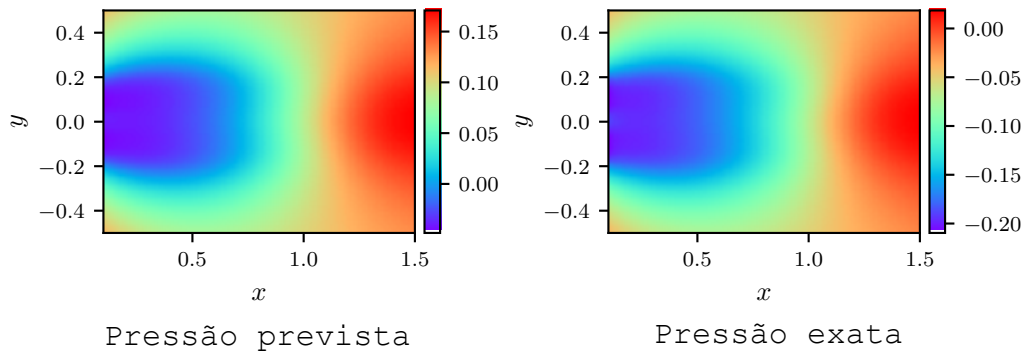
Tabela 5 – Erros na identificação de λ_1 e λ_2 da equação de Burgers para valores de ν intermediários a $\nu = 0,001/\pi$ e $\nu = 0,01/\pi$ para dados sem ruído.

4.2 Resultados para as equações de Navier-Stokes

A Figura 12 ilustra o resultado obtido pela PINN na modelagem das N-S para um escoamento incompressível, bidimensional, externo a um cilindro com fluido newtoniano para um número de Reynolds = 500. O quadro à esquerda descreve o campo de pressões $p(t, x, y)$ previsto, enquanto o quadro da direita mostra o campo de pressões exato do escoamento.

A PINN foi capaz de reconstruir o campo de pressões sem ter acesso aos dados de pressão do escoamento. Esse feito deve-se ao fato da rede neural ter empregado a lei de conservação de massa durante o treinamento utilizando o campo de velocidades. O único propósito dos dados sobre a pressão na rede foi para calcular, no final, o erro relativo à predição da pressão e construir o campo de pressões exato mostrado à direita. Nota-se que há diferença na magnitude das escalas das legendas entre os campos de pressão previsto e exato. Esse fenômeno ocorre porque nas N-S temos apenas um termo de gradiente da pressão de modo que o seu valor pode ser recuperado a menos de uma constante, justificando essas diferenças.

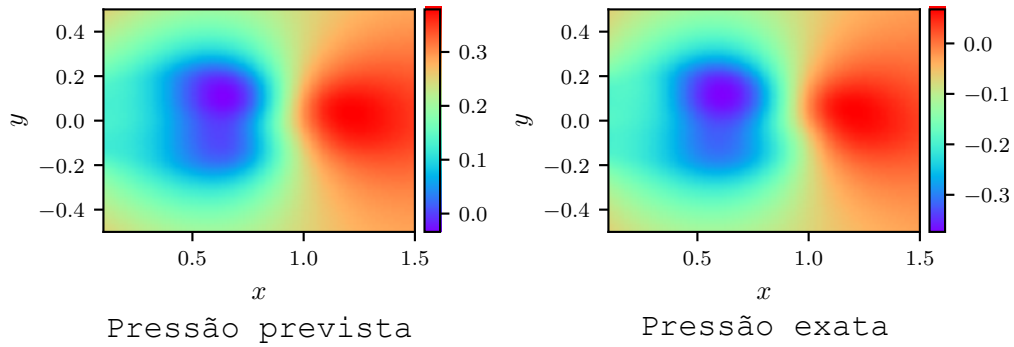
Abaixo dos campos de pressão há a comparação entre a equação diferencial parcial correta que modela o escoamento e qual foi a identificada pela PINN, isto é, quais foram os parâmetros λ ajustados pela rede neural. A performance da rede neural foi notável com erro de menos de 2% para os dois parâmetros mostrando-a eficiente para trabalhar nesta faixa de número de Reynolds.



EDP Correta	$u_t + (uu_x + vu_y) = -p_x + 0.002(u_{xx} + u_{yy})$ $v_t + (uv_x + vv_y) = -p_y + 0.002(v_{xx} + v_{yy})$
EDP Identificada (sem ruído)	$u_t + 0.983(uu_x + vu_y) = -p_x + 0.00198(u_{xx} + u_{yy})$ $v_t + 0.983(uv_x + vv_y) = -p_y + 0.00198(v_{xx} + v_{yy})$

Figura 12 – Resultado obtido pela PINN para identificação das equações de Navier-Stokes para um escoamento incompressível, bidimensional, externo a um cilindro com fluido newtoniano para $Re = 500$.

A Figura 13 mostra o resultado da identificação das N-S nas mesmas configurações de escoamento, no entanto para um número de Reynolds = 1000. A performance é menor em comparação com $Re = 500$, especialmente para o parâmetro λ_2 . Esta faixa de Reynolds pode indicar um limite superior para este método, ainda mais pela própria dificuldade de se gerar soluções no Mathematica[®] para serem testadas. Esta dificuldade deve-se à natureza do termo $1/Re$, isto é, valores muito altos de Reynolds implicam em frações menores gerando complicações numéricas na geração das soluções.



EDP Correta	$u_t + (uu_x + vu_y) = -p_x + 0.001(u_{xx} + u_{yy})$ $v_t + (uv_x + vv_y) = -p_y + 0.001(v_{xx} + v_{yy})$
EDP Identificada (sem ruído)	$u_t + 0.969(uu_x + vu_y) = -p_x + 0.00113(u_{xx} + u_{yy})$ $v_t + 0.969(uv_x + vv_y) = -p_y + 0.00113(v_{xx} + v_{yy})$

Figura 13 – Resultado obtido pela PINN para identificação das equações de Navier-Stokes para um escoamento incompressível, bidimensional, externo a um cilindro com fluido newtoniano para $Re = 1000$.

Além dos valores 500 e 1000 a PINN também foi testada para Reynolds = 100. Este valor sugere um limite inferior de aplicação da técnica, pois obteve os piores valores de erros associados ao ajuste dos parâmetros λ das N-S. A Tabela 6 descreve os erros associados. Nota-se também a diminuição da eficiência para o escoamentos com número de Reynolds = 1000, indicando um possível limite superior da técnica.

Número de Reynolds, Re	Erro para λ_1	Erro para λ_2
100	23,3%	24%
500	1,65%	1,22%
1000	3,07%	13,2%

Tabela 6 – Erros na identificação de λ_1 e λ_2 das equações de Navier-Stokes para escoamento incompressível, bidimensional, externo a um cilindro com fluido newtoniano. As faixas de Reynolds analisadas foram 100, 500 e 1000.

Um segundo teste foi feito para as N-S em que foi adicionado um terceiro termo às equações de conservação de quantidade de movimento acompanhado de um parâmetro λ_3 . O objetivo deste teste foi observar se a PINN seria capaz de prever o valor correto deste parâmetro, isto é, atribuir o valor zero ao coeficiente que acompanha o termo extra já que este não está presente na equação diferencial parcial que modela o fenômeno. O teste foi repetido duas vezes, cada uma para um termo extra diferente. O valor do número de Reynolds associado ao escoamento deste teste foi 500, visto que foi o valor com melhor performance no teste principal e também foram estipuladas cinco mil iterações para o treinamento da rede neural.

O primeiro termo extra trata-se de uma derivada de terceira ordem, de forma que as equações da conservação da quantidade de movimento foram inseridas na PINN de acordo com a equação 4.2:

$$\begin{aligned} u_t + \lambda_1(uu_x + vu_y) &= -p_x + \lambda_2(u_{xx} + u_{yy}) + \lambda_3(u_{xxx}^2 + u_{yyy}^2), \\ v_t + \lambda_1(uv_x + vv_y) &= -p_y + \lambda_2(v_{xx} + v_{yy}) + \lambda_3(v_{xxx}^2 + v_{yyy}^2). \end{aligned} \quad (4.2)$$

O segundo termo extra trata-se da multiplicação dos termos de derivada de segunda ordem, gerando a equação 4.3:

$$\begin{aligned} u_t + \lambda_1(uu_x + vu_y) &= -p_x + \lambda_2(u_{xx} + u_{yy}) + \lambda_3(u_{xx}^2 u_{yy}^2), \\ v_t + \lambda_1(uv_x + vv_y) &= -p_y + \lambda_2(v_{xx} + v_{yy}) + \lambda_3(v_{xx}^2 v_{yy}^2). \end{aligned} \quad (4.3)$$

Assim que a rede neural foi treinada, esta foi capaz de atribuir corretamente o valor zero ao parâmetro λ_3 acompanhado dos termos extras em cerca de 250 iterações. No entanto, assim que o valor convergiu o procedimento foi encerrado. Este encerramento prematuro comprometeu a determinação dos parâmetros λ_1 e λ_2 , pois suas convergências exigiram cerca de dez a onze mil iterações nos testes em que apenas estes dois parâmetros estavam presentes. Isto indica que ainda havia iterações para serem feitas para os outros dois parâmetros apesar do procedimento ser encerrado sem causa aparente.

O resultado deste teste sugere que a rede é capaz de identificar com considerável rapidez termos que não deveriam estar presentes na equação diferencial parcial que está sendo ajustada ao fenômeno em análise. Logo há a possibilidade de submeter um conjunto de dados à uma equação com termos extras e verificar a existência de outros fatores em ação no fenômeno. Após a exclusão dos termos cujos coeficientes convergem a zero, a rede pode enfim ser aplicada para a modelagem dos coeficientes restantes.

5 Conclusão

Este trabalho apresentou uma breve recapitulação de Mecânica dos Fluidos a respeito do desenvolvimento analítico das equações de Navier-Stokes e também revisou tópicos de Inteligência Artificial sobre redes neurais. Além disso, estudou a técnica Redes Neurais Vinculadas pela Física com ênfase no ajuste de coeficientes de equações da Mecânica dos Fluidos, que são de grande interesse para a Engenharia Mecânica.

A técnica estudada neste trabalho mostrou-se eficaz no ajuste dos coeficientes da equação de Burgers com menos de 0,1% de erro para uma faixa de valores de coeficiente de difusão entre $0,01/\pi$ e $0,1/\pi$. Para as equações de Navier-Stokes a técnica apresentou menos de 2% de erro para o ajuste dos coeficientes para a faixa de número de Reynolds igual a 500. Além disso, o fato da técnica ter sido capaz de prever com pequeno número de iterações que algum termo não deveria estar presente na equação descrevendo o fenômeno sugere a possibilidade de modelagem de outros problemas de natureza complexa.

Conclui-se que há faixas de valores em que a técnica alcança melhores resultados, indicando limitações. Para a equação de Burgers, a diminuição da eficiência na identificação do parâmetro que acompanha o termo convectivo para $\nu = 1/\pi$ sugere um limite superior de aplicação. O erro substancial na identificação do parâmetro que acompanha o termo difusivo para $\nu = 0.001/\pi$ sugere que o conjunto de dados oferecido para o treinamento da rede não seja adequado, pois possivelmente existe algum tipo de difusão numérica sendo introduzida para buscar estabilidade. Tal fenômeno pode ocorrer visto que soluções numéricas de problemas convecção-difusão tornam-se instáveis para escoamentos altamente convectivos. Uma alternativa é gerar um conjunto de dados oriundos de soluções numéricas invíscidas e manter o termo de difusão multiplicado por um parâmetro incógnito, de forma que a rede então poderia ser testada para identificar o valor zero do coeficiente acompanhando o termo de difusão.

Para as equações de Navier-Stokes a faixa ideal de trabalho mostrou-se ainda mais restrita apresentando diminuição considerável de eficiência tanto para valores de números de Reynolds maiores que 500 quanto para menores. Além da própria limitação da técnica existe a questão da limitação dos métodos empregados para a geração dos conjuntos de dados uma vez que o método numérico empregado pelo software Wolfram Mathematica® apresenta complicações para a solução de equações diferenciais parciais com coeficientes pequenos. Isto ocorre para números de Reynolds maiores do que 1000 vista a natureza do termo $1/Re$ que surge na adimensionalização das equações de Navier-Stokes. Trabalhos futuros podem explorar estas limitações a fim de aprimorar a técnica ou identificar quais as características dos fenômenos estudados que causam queda de eficiência.

Referências

- BATEMAN, H. Some recent researches on the motion of fluids. *Monthly Weather Review*, p. 163–170, 4 1915. 20
- BAYDIN, A. G.; PEARLMUTTER, B. A.; SISKIND, J. M. *Automatic Differentiation in Machine Learning: a Survey*. 2018. 1-43 p. 24
- BURGERS, J. M. A mathematical model illustrating the theory of turbulence. *Advances in Applied Mechanics*, v. 1, p. 171–199, 1948. 20
- CHOLLET, F. *Deep Learning with Python*. 2. ed. [S.l.]: Manning, 2021. ISBN 9781617296864. 8, 14, 21, 23, 24, 29
- LU, L. et al. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, Society for Industrial and Applied Mathematics Publications, v. 63, p. 208–228, 2021. ISSN 00361445. 24
- RAISSI, M.; PERDIKARIS, P.; KARNIADAKIS, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, Academic Press Inc., v. 378, p. 686–707, 2 2019. ISSN 10902716. 14, 19, 20, 25
- WHITE, F. M. *Mecânica dos Fluidos*. 6. ed. [S.l.]: McGraw Hill, 2011. ISBN 978-85-8055-009-2. 13, 16, 19