

IMPLEMENTAÇÃO DE REGRESSÃO PARA ANÁLISE NUMÉRICA DE
ESCOAMENTO BIDIMENSIONAL EM UM DEGRAU

Heitor Gomes de Souza Batista

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
Setembro de 2025



**UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO**

**Politécnica
UFRJ**

Escola Politécnica

Departamento de Engenharia Mecânica

**IMPLEMENTAÇÃO DE REGRESSÃO PARA ANÁLISE NUMÉRICA DE
ESCOAMENTO BIDIMENSIONAL EM UM DEGRAU**

Heitor Gomes de Souza Batista

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 Pereira Duda, D.Sc.



Prof. Fernando Alves Rochinha, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2025

Gomes de Souza Batista, Heitor

Implementação de Regressão Para Análise Numérica de Escoamento Bidimensional em um Degrau/ Heitor Gomes de Souza Batista. – Rio de Janeiro: UFRJ/Escola Politécnica, 2025.

XII, 67 p.: il.; 29, 7cm.

Orientador: Gustavo Rabello dos Anjos

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Mecânica, 2025.

Referências Bibliográficas: p. 68 – 70.

1. Análise de Regressão. 2. *Python*. 3. Método de Elementos Finitos. 4. Dinâmica de Fluidos Computacional. I. Rabello dos Anjos, Gustavo. II. Universidade Federal do Rio de Janeiro, UFRJ, Curso de Engenharia Mecânica. III. Implementação de Regressão Para Análise Numérica de Escoamento Bidimensional em um Degrau.

*“Se vi mais longe, foi por estar
sobre os ombros de gigantes” -
Isaac Newton.*

Agradecimentos

Agradeço à minha família e amigos, que me apoiaram em cada passo deste projeto. Esta pesquisa decorreu durante um dos períodos mais turbulentos da minha vida, e seu suporte, compreensão e carinho me deram forças para finalizá-la.

Agradeço também ao meu professor orientador Gustavo Rabello do Anjos que sempre esteve disponível para qualquer dúvida e cujas contribuições no conteúdo apresentado aqui são inúmeras. Espero que possamos trabalhar juntos novamente.

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

IMPLEMENTAÇÃO DE REGRESSÃO PARA ANÁLISE NUMÉRICA DE ESCOAMENTO BIDIMENSIONAL EM UM DEGRAU

Heitor Gomes de Souza Batista

Setembro/2025

Orientador: Gustavo Rabello dos Anjos

Departamento: Engenharia Mecânica

O projeto descrito neste relatório visa utilizar Regressão para produzir previsões adequadas e consistentes quanto às dimensões de escoamento turbulento sobre um degrau a partir de seu Número de Reynolds. A análise é feita a partir de resultados computacionais baseados no Método de Elementos Finitos com discretização espacial e temporal. A Regressão é então validada quanto à sua precisão, visando determinar a viabilidade dessa abordagem. Bibliotecas *Python* foram utilizadas em todas as etapas deste estudo. Especificamente, *Gmsh* e *FeniCS* para a construção das Equações de Navier-Stokes, e *Sklearn* para a Análise de Regressão. Ao longo do estudo, múltiplas hipóteses são testadas quanto à relação entre a dimensão dos fenômenos fluidodinâmicos e o Número de Reynolds do escoamento. Para cada hipótese, foi criada uma nova Regressão, que é então validada quanto à precisão de seus resultados. Entre os modelos testados, foi concluído que a magnitude da Zona de Recirculação no interior da tubulação segue uma relação polinomial de ordem 3, em função do Número de Reynolds. A precisão final da Regressão foi de 82.96%, a partir do Coeficiente de Determinação R^2 .

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

IMPLEMENTATION OF REGRESSION FOR NUMERICAL ANALYSIS OF 2D
FLOW OVER A BACKWARD-FACING STEP

Heitor Gomes de Souza Batista

September/2025

Advisor: Gustavo Rabello dos Anjos

Department: Mechanical Engineering

The project described in this report aims to use Regression Analysis to produce adequate and consistent predictions in regards to the dimensions of a turbulent flow over a backwards step, based on its Reynolds Number. The analysis is based on computational simulations data, obtained from the Finite Element Method with spatial and temporal discretization. The Regression is then validated in its precision, such that it can be determined the viability of this approach. *Python* libraries were utilized for all steps of this project. Particularly, *Gmsh* and *FeniCS* were the base for the construction of the Navier-Stokes equations and proper meshing of the problem's domain, while *Sklearn* was chosen for the Regression Analysis. Throughout this study, multiple hypotheses were tested as to the relationship between the magnitude of the fluid dynamic phenomena and the Reynolds Number. For each hypothesis, a new regression model is developed, which is then analyzed in its accuracy. From the tested models, it is concluded that the vortices' magnitude inside the pipe follows a polynomial function of order 3, based on the Reynolds Number. The final accuracy for the regression was approximately 82.96%, based on the Coefficient of Determination R^2 .

Sumário

Lista de Figuras	x
Lista de Tabelas	xii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Descrição dos Capítulos	3
2 Revisão Bibliográfica	4
2.1 Método de Elementos Finitos Computacional	4
2.2 Regressão em Dinâmica de Fluidos	6
2.3 Escoamentos em um Degrau	8
2.4 Objetivos deste Projeto	10
3 Fluidodinâmica	11
3.1 Definição do Problema Hidrodinâmico	11
3.2 Equações de Navier-Stokes para Fluido Incompressível	12
3.2.1 Incógnitas e Graus de Liberdade	12
3.2.2 Conservação de Massa	12
3.2.3 Incompressibilidade do Fluido	12
3.2.4 Momento Linear do Fluido	13
3.3 Condições de Contorno	13
3.3.1 Velocidade do Fluido no Interior da Tubulação no Instante Inicial	14
3.3.2 Velocidade do Fluido na Entrada da Tubulação	14

3.3.3	Pressão do Fluido na Entrada da Tubulação	15
3.3.4	Velocidade do Fluido nas Paredes da Tubulação	15
3.3.5	Velocidade na Saída da Tubulação	15
3.3.6	Pressão do Fluido na Saída da Tubulação	16
3.4	Análise de Elementos Finitos	16
3.4.1	Discretização de Domínio	16
3.4.2	Espaços das Funções	17
3.4.3	Formulação Variacional (Forma Fraca)	17
3.4.4	Ordem das Funções Testes	20
3.5	Resultados Extraídos	21
4	Análise de Regressão	23
4.1	Construção do Modelo de Regressão	23
4.2	Validação da Modelo de Regressão	24
5	Implementação Computacional	26
5.1	Dinâmica de Fluidos Computacional	26
5.1.1	Discretização do Espaço	27
5.1.2	Definindo o Passo de Tempo	29
5.1.3	Escrevendo o Código <i>Python</i>	30
5.1.4	Extração de Resultados	32
5.1.5	Verificação do Método Numérico	34
5.2	Regressão	35
6	Discussão de Resultados	37
6.1	Hipótese 1: Relação Polinomial de Ordem 2	39
6.2	Hipótese 2: Relação Polinomial de Ordem 3	42
6.3	Hipótese 3: Relação Polinomial de Ordem 4	44
6.4	Hipótese 4: Relação Logarítmica	46
7	Conclusão	48
7.1	Retorno de Experiência	48
7.2	Limitações e Sugestões para Projetos Futuros	49

Lista de Figuras

3.1	Dimensões do Domínio da Análise Computacional	11
3.2	Condições de Contorno	14
3.3	Perfil de Velocidade do Fluido na Entrada do Fluido	15
3.4	Principais medições realizadas	21
4.1	Exemplo de Sobreajuste	25
5.1	Malha discretizando o Espaço (1)	27
5.2	Malha discretizando o Espaço (2)	27
5.3	Exemplo de Elemento de Malha de Ordem 1 e 2	28
5.4	Exemplo de Perfil de Velocidade do Fluido Obtido por <i>FeniCS</i>	32
5.5	Medições da Recirculação	32
5.6	Comprimento da Zona de Recirculação para $Re = 400$ (Resultado Numérico)	34
6.1	Resultados das Medições do Comprimento da Zona de Recirculação em função do Número de Reynolds	38
6.2	Resultados Obtidos pela Regressão para Comprimento da Zona de Recirculação em função do Número de Reynolds - Polinômio de Ordem 2	39
6.3	Comparação entre as Previsões da Regressão Polinomial e o Valores Reais	41
6.4	Resultados obtidos pela Análise de Regressão para o Comprimento da Zona de Recirculação em função do Número de Reynolds - Polinômio de Ordem 3	42

6.5	Comparação entre as Previsões da Regressão Polinomial e o Valores Reais	43
6.6	Resultados obtidos pela Análise de Regressão para Comprimento da Zona de Recirculação em função do Número de Reynolds - Polinômio de Ordem 4	44
6.7	Comparação entre as Previsões da Regressão Polinomial e o Valores Reais	45
6.8	Resultados obtidos pela Regressão Logarítmica para comprimento da Zona de Recirculação em função do Número de Reynolds	46
6.9	Comparação entre as Previsões da Regressão Logarítmica e o Valores Reais	47

Lista de Tabelas

5.1	Relação das Medições com o Número de Reynolds do Fluido	33
5.2	Verificação da Análise Numérica	34
6.1	Coefficientes Polinomiais de Ordem 2 Determinados pela Regressão . .	39
6.2	Coefficientes Polinomiais de Ordem 3 Determinados pela Regressão . .	42
6.3	Coefficientes Polinomiais de Ordem 4 Determinados pela Curva de Regressão	44
6.4	Coefficientes Logarítmicos Determinados pela Análise de Regressão . .	46

Capítulo 1

Introdução

1.1 Motivação

O estudo do escoamento de partículas é crítico em múltiplos domínios acadêmicos e econômicos. Para isso, foram desenvolvidas equações diferenciais que representam analiticamente o comportamento da fase contínua de um escoamento como, por exemplo, as equações de Navier-Stokes.

Porém, à medida que novos estudos e experimentos aumentam em complexidade, torna-se inviável a obtenção de uma solução analítica capaz de representar o desenvolvimento de um escoamento através do espaço e tempo. Nesse contexto, temos o desenvolvimento de Simulações Computacionais de Dinâmica de Fluidos. Modelos computacionais utilizam técnicas numéricas, discretizando espaço e tempo, para transformar equações diferenciais parciais em comparativamente simples sistemas lineares. Além disso, essas simulações são facilmente parametrizáveis quanto à geometria do escoamento e propriedades físicas do fluido, permitindo que múltiplas permutações diferentes da análise possam ser realizadas. Por esses motivos, simuladores numéricos foram amplamente adotados em pesquisas em hidrodinâmica e aerodinâmica.

A performance de simulações de dinâmica de fluidos é diretamente limitada pelo hardware onde elas são feitas. O aumento na complexidade de uma simulação, como, por exemplo, ao utilizar uma discretização mais fina da geometria, induz diretamente ao aumento no tempo necessário para um computador resolver o sistema linear associado ao problema. Devido à natureza de métodos numéricos, toda variação nas

propriedades do fluido ou nas condições de contorno requer que uma nova análise seja realizada. Conseqüentemente, o tempo de simulação pode rapidamente expandir quanto maior for o escopo de um estudo.

Atualmente a única alternativa para reduzir o tempo de análise é a utilização de *hardware* mais computacionalmente eficiente, porém existe a constante possibilidade dessa solução tornar-se inviável devido a fatores econômicos ou mesmo pela disponibilidade de componentes.

Nesse contexto, vemos a oportunidade para o estudo e desenvolvimento de novas ferramentas e técnicas capazes de auxiliar na análise de fluidos que possam reduzir o tempo de análise necessário, sem requerer a adoção de novo *hardware*. Mais precisamente, neste estudo, o interesse é o desenvolvimento de uma abordagem computacional baseada em análise de regressão, capaz de fornecer uma previsão acurada quanto à evolução do campo de velocidade de um escoamento ao longo de um canal, correlacionado com as propriedades físicas do fluido.

1.2 Objetivos

Ao longo deste projeto, deseja-se estudar, desenvolver e avaliar um modelo de regressão que seja computacionalmente leve e, se possível, portátil para fácil implantação, capaz de acuradamente correlacionar o número de Reynolds de um escoamento com o campo de velocidades ao qual o fluido associado está submetido. A viabilidade do modelo será testada usando como referência os métodos numéricos tradicionais, visando reduzir ao mínimo a diferença entre os dois resultados. Por fim, serão discutidas as limitações presentes no modelo e quais medidas podem ser tomadas para que sua precisão seja mais confiável.

1.3 Descrição dos Capítulos

Nos capítulos a seguir, será feita uma revisão da bibliografia disponível para os principais tópicos abordados, seguida do desenvolvimento das Equações Diferenciais Parciais que governam o comportamento do escoamento. O conceito de Regressão é então introduzido, seu princípio de funcionamento, e como ela deve ser validada. Por fim, as ferramentas computacionais da análise tornam-se o foco, sendo descritos os passos necessários para utilizá-las.

Capítulo 2

Revisão Bibliográfica

No escopo deste projeto, 3 tópicos principais serão abordados: Método de Elementos Finitos Computacional, Análise de Regressão em Dinâmica de Fluidos e o fenômeno de Recirculação. Com isso em mente, neste capítulo faremos uma breve revisão sobre a história por trás desses temas.

2.1 Método de Elementos Finitos Computacional

Segundo LI *et al.* (2022), técnicas numéricas para resolução de sistemas de Equações Diferenciais Parciais datam desde os anos 1940, particularmente abordando problemas de transferência de calor e mecânica de sólidos. Apesar de existirem múltiplas abordagens diferentes para o método, um ponto comum pode ser encontrado com a discretização do domínio através de malhas.

Em análise computacional, malha é um conjunto de faces (geralmente triângulos ou quadriláteros), arestas e vértices que representam a superfície de um objeto ou espaço. A malha tem como função simplificar o domínio de um sistema de Equações Diferenciais, convertendo um espaço contínuo, com infinitos pontos, para um espaço discreto, com um número finito de pontos.

Os principais avanços na área ocorreram a partir dos anos 1970, quando a combinação Métodos de Elementos/Volumes Finitos e ferramentas computacionais aceleraram a compreensão dos principais desafios por trás da aplicabilidade de métodos numéricos e, conseqüentemente, novas estratégias foram criadas para contorná-los. Vale notar que, ao contrário da maioria dos movimentos matemáticos, o desenvolvi-

mento do Método de Elementos Finitos foi liderado principalmente por engenheiros, devido à necessidade de ferramentas para a resolução de equações diferenciais complexas.

Rapidamente, a implementação de simulações computacionais de fluidos foi adotada para diferentes campos da Engenharia como, por exemplo, os usos de métodos numéricos para análise Aerodinâmica. PERAIRE *et al.* (1989) apresenta o desenvolvimento de malhas triangulares e tetraédricas para domínios computacionais para o uso em análises numéricas de fluidos com viscosidades diversas, no contexto de aeronaves. SEIBERT *et al.* (1989) utiliza métodos numéricos computacionais para a resolução de sistemas de equações não lineares. Neste caso, malhas são refinadas localmente em regiões apresentando elevados gradientes de velocidade, oriundas de ondas de choque. GOUTINES *et al.* (1989) aplica elementos numéricos para o estudo de turbinas e compressores transônicos, uma vez que o comportamento do fluido nessas velocidades é melhor representado computacionalmente nas regiões de bordo de ataque/fuga e esteira de turbulência.

Atualmente, uma ampla gama de ferramentas foi desenvolvida para facilitar a análise numérica de mecânica de fluidos, criando uma forte indústria global de *Software* voltada ao suporte de outros setores industriais que realizam pesquisa quanto à fluidodinâmica. Em paralelo, projetos independentes introduziram novas opções ao mercado. Bases de código aberto que reduzem a barreira de entrada para a aplicação de ferramentas computacionais para problemas de dinâmica de fluidos. Dentre esses projetos, destaca-se *FeniCS*, uma biblioteca *Python* rica em recursos e de flexível aplicação para uma grande diversidade de estudos. Um exemplo é a análise realizada por MORTENSEN *et al.* (2011), que demonstra a simplicidade que o *FeniCS* permite ao pesquisador na composição de equações diferenciais parciais, acoplar equações, linearizar equações e construir algoritmos iterativos para a resolução de problemas não lineares, enquanto, simultaneamente, fornece um código de alto nível e fácil legibilidade.

Apesar das vantagens até aqui apresentadas, métodos computacionais ainda precisam considerar condições de convergência para que as simulações tragam resultados proveitosos. Existe um balanço entre a performance do algoritmo computacional, isto é, a rapidez com que os resultados são produzidos e a estabilidade da simulação.

Isso é claramente ilustrado no estudo de GNEDIN *et al.* (2018), que aborda especificamente a condição de Courant-Friedrich-Lewy para convergência de uma análise computacional de dinâmica de fluidos. De fato, o artigo revela o impacto negativo que o refinamento de malha e gradientes elevados de velocidade trazem para a estabilidade da simulação, e como a imposição de um passo de tempo adequado é capaz de combater a ocorrência de singularidades nos resultados.

2.2 Regressão em Dinâmica de Fluidos

A medida que avanços tecnológicos no campo de microprocessadores desaceleram, cresce o interesse em soluções alternativas, capazes de complementar as ferramentas atuais de análise numérica computacional. Dentre essas alternativas, o uso de modelos estatísticos, como Regressão, na análise de fluidos se mostra extremamente promissor no estudo de fluidodinâmica, por serem uma solução de resposta rápida e de fácil implementação em uma ampla gama de dispositivos.

Pesquisas quanto à adoção de ferramentas estatísticas na Análise Computacional de Fluidos são comparativamente recentes, uma vez que a literatura sobre modelos computacionais preditivos está estabelecida. Em particular, HENNING WESSELS (2020) estuda a aplicação capaz de associar as condições de contorno para problemas de Mecânica de Lagrange, a fim de obter uma ferramenta de simulação versátil para fluidos incompressíveis envolvendo superfícies livres. Como resultado, obteve-se um método capaz de realizar a correção de pressão em escoamentos incompressíveis de modo mais eficiente do que os métodos numéricos.

ZHANG e JANEWAY (2022), em sua pesquisa quanto ao uso de Redes Neurais para otimização de turbinas, aponta o longo tempo necessário para simulações computacionais como um fator motivador para o projeto. Métodos estatísticos com Análise de Regressão são essenciais na construção de Redes Neurais e sua adoção ao processo resultou numa redução de 80% no tempo de computação necessário para as avaliações quando comparado ao método tradicional, utilizando apenas análises numéricas computacionais.

A mesma justificativa e conclusão estão presentes no estudo de ZHOU e OOKA (2021) quanto à qualidade do ar em ambientes internos. Tradicionalmente, a

evolução da temperatura e umidade provém de simulações computacionais da temperatura do ar e umidade relativa. A incorporação de Análise de Regressão levou à redução do tempo de computação das análises em aproximadamente 94%.

Ferramentas estatísticas também observaram adoção na etapa de design de componentes mecânicos como bombas centrífugas, descritas por SAFIKHANI *et al.* (2011), veículos submarinos, descritos por TIAN *et al.* (2017), e bombas de mineração submarina, descritas por HU *et al.* (2022).

Não existe um único paradigma para a construção da base de dados. A opção mais popular, e que será utilizada para este estudo, trabalha com métodos numéricos (como, por exemplo, o método de Projeção de Chorin) para a resolução numérica de Equações Diferenciais Parciais para a construção da base de dados. Um exemplo dessa abordagem pode ser observado em PICHI *et al.* (2023), onde um escoamento em expansão súbita é modelado numericamente, e os fenômenos hidrodinâmicos, como o Efeito Coanda, podem então ser medidos. Esse processo é repetido até que se construa uma base de dados grande o bastante para associar as propriedades físicas do escoamento com o comportamento do fluido. Essa abordagem é intuitiva, pois a Análise de Regressão não demanda drásticas alterações aos algoritmos comuns na resolução numérica de Equações Diferenciais Parciais. Contudo, a dependência de uma base de dados derivada de resultados numéricos é a clara desvantagem da abordagem, que requer um considerável investimento computacional para obter um conjunto de treinamento. Em paralelo, alterações na geometria do problema ou nas condições de contorno das Equações Diferenciais Parciais requerem a completa reconstrução da base de dados e do Modelo de Regressão. Existem estudos, como o apresentado por PEHERSTORFER e WILLCOX (2015), que descrevem um modelo capaz de se adaptar dinamicamente às modificações das equações diferenciais que regem o problema. Porém, a aplicação desta estratégia ainda se inicia com uma base de dados de resultados numéricos na concepção da Regressão.

A base de dados para Análise de Regressão é representada majoritariamente por resultados numéricos na literatura. Isso não surpreende, uma vez que métodos numéricos podem ser facilmente iterados, gerando uma ampla base de dados para uso. Em contraste, resultados experimentais de análise de fluidos inerentemente demandam um investimento maior de tempo dos pesquisadores, dificultando a ob-

tenção de medições em escala comparável. Contudo, sem cuidados adequados, o risco de Sobreajuste aumenta consideravelmente, dado que a magnitude do conjunto de dados pode reduzir a acurácia do Modelo de Regressão para previsões.

2.3 Escoamentos em um Degrau

Estudos sobre Recirculação, particularmente em tubulações com geometria em “degrau”, ou *Backwards Facing Step*, são diversos, como explorado na revisão de MONTAZER *et al.* (2018). A expansão súbita de escoamentos é um fenômeno comum em aplicações de motores a combustão, sistemas de refrigeração, processamento de alimentos, preenchimento de moldes, entre outras, fazendo com que este tipo de escoamento seja extensivamente discutido na literatura, em particular, quanto à observação de uma Zona de Recirculação nas paredes da tubulação, e como fatores como velocidade e viscosidade podem influenciar nesse fenômeno. Além disso, esse tipo de escoamento é extremamente popular para testes de performance de ferramentas computacionais, devido à sua geometria simples e à facilidade de observação de Recirculação.

Como muitos fenômenos hidrodinâmicos, a ferramenta mais comum para o estudo de escoamentos em um degrau são análises computacionais por métodos de elementos finitos. O artigo de THABIT *et al.* (2018) fornece uma análise computacional detalhada do fenômeno e sua relação com seu Número de Reynolds. Contudo, o foco principal consiste em determinar as condições necessárias para o desprendimento nas paredes de tubulações ou métodos para a minimização da ocorrência de Recirculação.

GAUTIER e AIDER (2014) demonstra o uso de pulsos à montante da vazão capazes de reduzir o comprimento da zona. Já BISWAS *et al.* (2004) compreende uma série de análises sobre a vazão de um fluido, variando o número de Reynolds, até o limite de transição para turbulência, onde se espera a formação de uma Zona de Recirculação.

Segundo ANDERSON *et al.* (2020), Recirculação ocorre na presença de um grande gradiente de pressão na camada limite do escoamento, ou seja, na superfície que delimita ou restringe a passagem do fluido, como, por exemplo, as paredes de

uma tubulação. No contexto de Escoamentos em um Degrau, o gradiente de pressão elevado é consequência da súbita expansão do diâmetro da tubulação. Consequentemente, a velocidade do escoamento na camada limite inverte de sentido e induz a formação de uma Zona de Recirculação à parede da tubulação.

Análises experimentais sobre o comportamento de Escoamentos em um Degrau são particularmente complexas devido à sensibilidade da ocorrência de Recirculação para alterações no Número de Reynolds. Isso dificulta a replicabilidade dos experimentos, especialmente para escoamentos turbulentos, onde a magnitude do fenômeno é consideravelmente maior. Dado isso, experimentos costumam focar numa faixa pequena de número de Reynolds, ou mesmo em um único caso específico, com condições altamente controladas, tal como observado por JOVIC e DRIVER (1994).

O uso de métodos numéricos na representação do fenômeno é de elevado interesse, uma vez que flexibiliza a simulação do escoamento para uma variedade de perfis de velocidade e tipos de fluido. Ao mesmo tempo, é necessário garantir que quaisquer resultados numéricos estejam de acordo com a literatura disponível, ou seja, deseje-se sempre a redução do erro entre os dados numéricos e os resultados experimentais obtidos de outros projetos. Um exemplo disso é o estudo de ABDOLLAHPOUR *et al.* (2023), cuja análise numérica de um escoamento em um degrau apresentou um erro médio de menos de 20% quanto à magnitude da Zona de Recirculação no interior da tubulação.

Vale considerar que existem múltiplas abordagens possíveis quanto ao método numérico utilizado para a análise de um escoamento turbulento. Modelos numéricos para análise de Recirculação visam representar matematicamente o comportamento do escoamento, tendo em mente que o perfil de velocidade e viscosidade do fluido ao longo da tubulação é extremamente imprevisível. Por exemplo, o estudo de JEHAD *et al.* (2015) adota o modelo K-Epsilon para representar a variância das propriedades físicas do escoamento, obtendo resultados condizentes com análises experimentais disponíveis na literatura. No escopo deste estudo, contudo, será utilizado um modelo mais simples para a análise da Zona de Recirculação, especificamente o modelo de equações RANS.

Apesar de modelos analíticos sobre o fenômeno de Recirculação serem escassos,

análises experimentais desenvolvidas por AMEUR e MENNI (2019) demonstraram que o fenômeno não é completamente aleatório, e alterações nos parâmetros do escoamento (tais como o número de Reynolds e o fator de expansão do escoamento na saída do "degrau") resultam em transformações previsíveis na magnitude do desprendimento.

2.4 Objetivos deste Projeto

A bibliografia apresentada neste capítulo serve como fundamento para este relatório. Deseja-se aqui aplicar os conceitos introduzidos sobre análise numérica, Análise de Regressão e escoamentos em um degrau, e replicar os resultados de simulações anteriores. Por fim, o produto final deste estudo é um Modelo de Regressão capaz de descrever razoavelmente a evolução da Zona de Recirculação de um escoamento a partir do seu Número de Reynolds.

Capítulo 3

Fluidodinâmica

3.1 Definição do Problema Hidrodinâmico

Primeiro define-se o problema hidrodinâmico inicial, que servirá como base do estudo. Quer-se descrever a evolução da vazão de um fluido, ao longo do tempo, utilizando um sistema de equações diferenciais parciais. Para isso, precisamos primeiro impor o escopo da análise, isto é, como é o espaço sendo atravessado por um fluido incompressível, suas dimensões e quais os canais por onde o escoamento entra e sai.

Consideremos um caso no espaço bidimensional, onde o fluido atravessa uma tubulação de 10 metros de comprimento. A tubulação apresenta um diâmetro inicial de 1 metro, que abruptamente se expande para 2 metros. A figura 3.1 ilustra as dimensões do domínio.

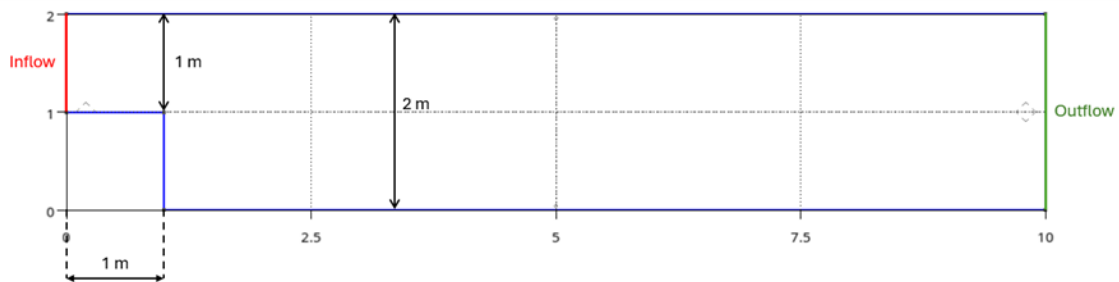


Figura 3.1: Dimensões do Domínio da Análise Computacional

3.2 Equações de Navier-Stokes para Fluido Incompressível

3.2.1 Incógnitas e Graus de Liberdade

A velocidade \mathbf{U} e a pressão P do escoamento são as duas incógnitas que serão calculadas pelo Método de Elementos Finitos. Portanto, é preciso definir quantos graus de liberdade as duas variáveis apresentam no domínio, ou mais especificamente, em quais dimensões no espaço a velocidade e a pressão evoluem; para este estudo, consideramos:

- A velocidade apresenta 2 graus de liberdade, variando verticalmente (entre as paredes da tubulação), e horizontalmente (ao longo da tubulação);
- A pressão apresenta 1 grau de liberdade, variando ao longo da tubulação.

3.2.2 Conservação de Massa

Primeiro é preciso descrever a Conservação de Massa ao longo da tubulação, dada em 3.1. Desta forma, impomos que a massa total de fluido injetado na entrada da tubulação deve ser igual à massa total na saída. A equação está em função de \mathbf{U} , P e ρ , a densidade do fluido.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (3.1)$$

3.2.3 Incompressibilidade do Fluido

Como o fluido estudado é incompressível, a derivada e gradiente da densidade ao longo do tempo são constantes. Logo, a equação de Conservação de massa é igual à Equação 3.2.

$$\nabla \cdot (\mathbf{U}) = 0 \quad (3.2)$$

3.2.4 Momento Linear do Fluido

A evolução do momento linear de escoamento incompressível é descrita pelas equações de Navier-Stokes, derivadas da Segunda Lei de Newton.

$$\rho\left(\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U}\right) = \nabla \cdot \boldsymbol{\sigma}(\mathbf{U}, P) + f \quad (3.3)$$

Onde \mathbf{U} é o vetor velocidade do fluido, P é a pressão, ρ é a densidade do fluido, e f representa as forças por unidade de volume atuando sob o fluido. O termo $\boldsymbol{\sigma}(\mathbf{U}, P)$ representa o tensor de estresse, que para um fluido Newtoniano é dado:

$$\boldsymbol{\sigma}(\mathbf{U}, P) = 2\mu\boldsymbol{\epsilon}(\mathbf{U}) - P\mathbf{I} \quad (3.4)$$

$$\boldsymbol{\epsilon}(\mathbf{U}) = \frac{1}{2}(\nabla \mathbf{U} + (\nabla \mathbf{U})^T) \quad (3.5)$$

Onde μ representa a viscosidade dinâmica do fluido.

3.3 Condições de Contorno

Para que as equações de Navier-Stokes tenham uma solução definitiva, deve-se definir as condições iniciais e de contorno. Segundo HEATH (2002) e LEWIS *et al.* (2004), o número de condições de contorno necessárias para que uma EDP tenha uma única solução é igual à maior ordem de cada uma de suas derivadas. Observando a ordem das equações diferenciais apresentadas, podemos inferir que serão necessárias:

- 1 Condição Inicial para a velocidade do fluido,
- 2 Condições de Contorno para a velocidade do fluido,
- 1 Condição de Contorno para a pressão do fluido.

Lembrando que, como a velocidade \mathbf{U} é um vetor bidimensional, cada condição deve considerar a velocidade do escoamento na direção x e y .

As condições de contorno utilizadas neste problema estão resumidas na Figura 3.2, descrevendo o comportamento da velocidade e pressão do escoamento nas fronteiras do domínio do problema. Vejamos a seguir a explicação para cada uma delas.

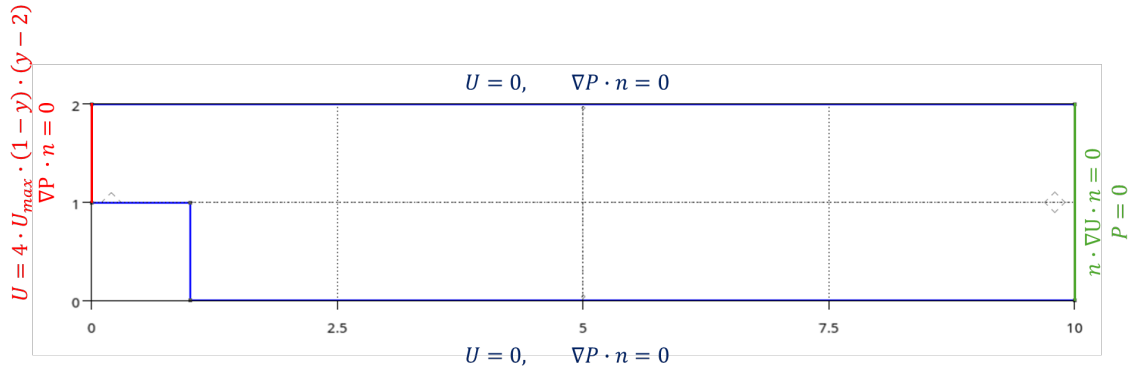


Figura 3.2: Condições de Contorno

3.3.1 Velocidade do Fluido no Interior da Tubulação no Instante Inicial

Aqui é feita a imposição de que o fluido se encontra em repouso no instante inicial $t = 0$ no interior da tubulação. Isso facilitará a visualização do escoamento ao longo do tempo e a Zona de Recirculação.

3.3.2 Velocidade do Fluido na Entrada da Tubulação

O fluido entra na tubulação pelo *Inflow*, com um perfil de velocidade parabólico. Aqui fazemos nossa primeira imposição sobre o comportamento do fluido nas paredes da tubulação: supondo que não haja deslizamento entre o fluido e as paredes da tubulação, a velocidade deverá ser nula nas paredes. Descrito em 3.6, a velocidade do fluido ao passar no *Inflow* alcança seu valor máximo U_{max} no centro da tubulação e reduz até zero quanto mais próximo das paredes da tubulação, como mostra a Figura 3.3.

$$\mathbf{U} = 4 \cdot \mathbf{U}_{max} \cdot (1 - y) \cdot (y - 2), \quad y \in [1, 2] \quad (3.6)$$

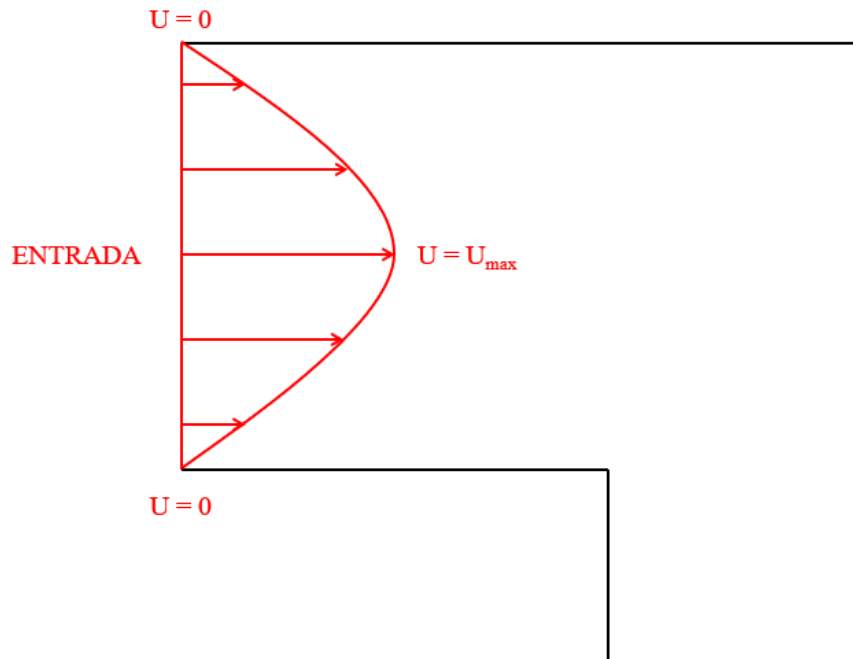


Figura 3.3: Perfil de Velocidade do Fluido na Entrada do Fluido

3.3.3 Pressão do Fluido na Entrada da Tubulação

Impomos neste problema que o gradiente de pressão na entrada da tubulação seja igual a zero. Isso deriva da assunção de que o perfil de velocidade do fluido está completamente desenvolvido na entrada do domínio.

3.3.4 Velocidade do Fluido nas Paredes da Tubulação

Como mencionado antes, será adotada a hipótese de que não haverá deslizamento entre o fluido e as paredes da tubulação, o que significa que a velocidade do fluido é zero nesses pontos. No contexto do método de elementos finitos, pode-se dizer que a velocidade do fluido é nula na fronteira do sistema, com exceção das regiões de *Inflow* e *Outflow*.

3.3.5 Velocidade na Saída da Tubulação

Aqui impomos que a saída da tubulação localiza-se a uma distância grande o bastante do degrau, onde o perfil de velocidade está completamente desenvolvido, fazendo com que o gradiente de velocidade na direção de saída seja nulo.

3.3.6 Pressão do Fluido na Saída da Tubulação

No escopo desta análise, iremos considerar que o fluido sai da tubulação sem resistência do meio externo, representando assim uma pressão $P = 0$ no *Outflow*.

3.4 Análise de Elementos Finitos

O método de elementos finitos traz consigo excelente versatilidade para computação de dinâmica de fluidos. Em destaque, é a sua capacidade de resolver as equações não lineares presentes em problemas de Navier-Stokes.

Além disso, a análise com elementos finitos pode ser realizada com malhas não estruturadas, isto é, onde a discretização do domínio pode ser feita em intervalos irregulares, facilitando a concepção de malhas que se adequem a geometrias complexas. A teoria apresentada a seguir é derivada do Método de Galerkin, detalhada na síntese de ANJOS (2024).

3.4.1 Discretização de Domínio

Apesar do domínio do problema ser relativamente simples, visto que a tubulação é composta exclusivamente de retas formando ângulos retos, ainda existem vantagens em utilizar uma malha irregular para discretizar o problema.

Ao invés de utilizar o mesmo grau de refinamento de malha para todos os pontos da tubulação, podemos controlar a discretização do domínio em diferentes regiões da tubulação, dependendo da criticidade para a análise.

Por exemplo, regiões onde espera-se que o vetor velocidade do fluido varie drasticamente em magnitude e direção, uma malha mais fina é desejável para evitar a ocorrência de singularidades durante a análise de elementos finitos. Já regiões de pouca relevância para a análise, ou onde as propriedades do fluido não variam drasticamente, permitem o uso de uma malha contendo menos elementos, reduzindo o custo computacional da simulação. No capítulo de Implementação Computacional, será discutida a discretização do domínio e a escolha de malha realizada.

3.4.2 Espaços das Funções

A solução das equações de Navier-Stokes consiste de 2 funções: \mathbf{U} e P . Como veremos a seguir, uma das estratégias para determinar essas soluções consiste em alterar a Equação Diferencial Parcial para a sua forma fraca ou variacional. Contudo, para que a forma variacional possua uma solução única e estável, será definido precisamente o espaço de funções ao qual \mathbf{U} e P pertencem. Para que a equação variacional apresente uma solução, essas funções devem ser:

- Respeitar as condições de contorno para a Forma Fraca;
- São integráveis para o Domínio Ω do problema;
- Suas derivadas são integráveis para todo o domínio Ω do problema.

3.4.3 Formulação Variacional (Forma Fraca)

Quando comparado ao método numérico utilizado para resolver a Equação de Poisson ou a Equação de Transferência de Calor, solução numérica das equações de Navier-Stokes requer certas alterações para que haja a convergência adequada dos resultados.

Se o Método de Euler for utilizado para remover as derivadas parciais das equações de Navier-Stokes, o resultado obtido é um sistema de equações não lineares, com um ponto de sela. Em prática é possível trabalhar com sistemas não-lineares, mas seria necessário formular condições de contorno e métodos iterativos adicionais para que sejam eficientes.

A estratégia utilizada para solucionar o sistema de equações será abordar as equações diferenciais parciais separadamente, utilizando o Método de CHORIN (1968). O método consiste em transformar as equações de Navier-Stokes em um sistema de equações lineares. As incógnitas do sistema são os valores de velocidade e pressão em cada nó da malha que discretiza o domínio.

Primeiro começamos os passos padrões da formulação variacional, multiplicando ambos os lados da equação de conservação de momento por uma função teste v . Utilizaremos a notação abaixo para simplificar a visualização da equação:

$$\langle v, w \rangle = \int_{\Omega} v w dx, \quad \langle v, w \rangle_{\partial\Omega} = \int_{\partial\Omega} v w ds \quad (3.7)$$

Depois aplicamos o Método de Euler para um passo de tempo Δt , com uma função Velocidade Estimada \mathbf{U}^* , substituindo a derivada no tempo da velocidade, mantendo a pressão do passo anterior P^n .

$$\begin{aligned} & \langle \rho \frac{(\mathbf{U}^* - \mathbf{U}^n)}{\Delta t}, \mathbf{v} \rangle + \langle \rho \mathbf{U}^n \cdot \nabla \mathbf{U}^n, \mathbf{v} \rangle + \langle \sigma(\mathbf{U}^{n+\frac{1}{2}}, P^n), \epsilon(\mathbf{v}) \rangle + \langle P^n \mathbf{n}, \mathbf{v} \rangle_{\partial\Omega} \\ & - \langle \mu \nabla \mathbf{U}^{n+\frac{1}{2}} \cdot \mathbf{n}, \mathbf{v} \rangle_{\partial\Omega} = \langle f^{n+1}, \mathbf{v} \rangle \end{aligned} \quad (3.8)$$

Onde $\mathbf{U}^{n+\frac{1}{2}}$ é dado por:

$$\mathbf{U}^{n+\frac{1}{2}} \approx \frac{(\mathbf{U}^n + \mathbf{U}^{n+1})}{2} \quad (3.9)$$

Ao final desta etapa, obtém-se o valor da velocidade estimada \mathbf{U}^* .

Vale mencionar que a função teste \mathbf{v} pertence ao mesmo espaço de funções que o campo de velocidade \mathbf{U} , garantindo a unicidade e a estabilidade da solução.

$$\mathbf{v} \in V = \{\mathbf{v} \in [H^1(\Omega)]^2 : \mathbf{v} = 0 \quad \forall x \in \partial\Omega\} \quad (3.10)$$

Passamos então para o segundo passo do Método, onde focaremos na segunda incógnita das Equações de Navier-Stokes, a pressão P . Especificamente, vamos definir a pressão P^{n+1} após um intervalo de tempo Δt .

Devido à definição de \mathbf{U}^* , introduzimos um erro de incompressibilidade ao sistema de equações, pois ela está associada à pressão P^n ao invés de P^{n+1} . Consequentemente, o divergente da velocidade deixa de ser igual a zero.

$$\nabla \cdot \mathbf{U}^* \neq 0 \quad (3.11)$$

Para corrigir este erro, é necessário introduzir uma variação de pressão ΔP , definida como:

$$P^{n+1} = P^n + \Delta P \quad (3.12)$$

$$\nabla^2 \Delta P = \frac{(\nabla \cdot \mathbf{U}^*)}{\Delta t} \quad (3.13)$$

ΔP aqui representa uma função “variação de pressão” que atua em todos os pontos do domínio e fornece a variação de pressão necessária para corrigir o erro de incompressibilidade

Calculando o divergente de (3.12), obtemos:

$$\nabla^2 P^{n+1} = \nabla^2 P^n + \nabla^2 \Delta P = \nabla^2 P^n + \frac{(\nabla \cdot \mathbf{U}^*)}{\Delta t} \quad (3.14)$$

Obtemos aqui uma EDP. Com isso, podemos obter a sua forma fraca introduzindo uma função de teste q :

$$\langle \nabla P^{n+1}, \nabla q \rangle = \langle P^n, \nabla q \rangle - \Delta t^{-1} \langle \nabla \cdot \mathbf{U}^*, q \rangle \quad (3.15)$$

O resultado da forma fraca é a pressão P^{n+1} .

Pelo mesmo raciocínio explicado para a função \mathbf{v} , a função teste q pertence ao mesmo espaço de funções que o campo de velocidade P , garantindo a unicidade e a estabilidade da solução.

$$q \in Q = \{q \in [H^1(\Omega)]\} \quad (3.16)$$

Por fim, falta apenas determinar a verdadeira função do Campo de Velocidade para o domínio. Retornamos novamente para a equação de conservação de momento. Esta equação é verdadeira tanto se utilizarmos o par Campo de Velocidade \mathbf{U}^* com a pressão P^n , descrito na Equação (3.8), como também se utilizarmos o par Campo de Velocidade \mathbf{U}^{n+1} e pressão P^{n+1} , extraída de (3.15).

$$\begin{aligned} & \left\langle \rho \frac{(\mathbf{U}^{n+1} - \mathbf{U}^n)}{\Delta t}, \mathbf{v} \right\rangle + \langle \rho \mathbf{U}^n \cdot \nabla \mathbf{U}^n, \mathbf{v} \rangle + \langle \sigma(\mathbf{U}^{n+\frac{1}{2}}, P^{n+1}), \epsilon(\mathbf{v}) \rangle + \langle P^{n+1} \mathbf{n}, \mathbf{v} \rangle_{\partial\Omega} \\ & - \langle \mu \nabla \mathbf{U}^{n+\frac{1}{2}} \cdot \mathbf{n}, \mathbf{v} \rangle_{\partial\Omega} = \langle f^{n+1}, \mathbf{v} \rangle \end{aligned} \quad (3.17)$$

A forma fraca da diferença dessas duas Equações de Conservação de Momento (3.8) e (3.17) é apresentada abaixo.

$$\langle \mathbf{U}^{n+1}, \mathbf{v} \rangle = \langle \mathbf{U}^*, \mathbf{v} \rangle - \Delta t \langle \nabla(P^{n+1} - P^n), \mathbf{v} \rangle \quad (3.18)$$

A solução da forma fraca fornece o Campo de Velocidade no instante $n+1$, \mathbf{U}^{n+1} .

3.4.4 Ordem das Funções Testes

As formas fracas das equações de velocidade e pressão precisam ser representadas na forma de um sistema de equações lineares para que possam ser resolvidas numericamente. Isso é possível se utilizarmos Funções Solução \mathbf{U} e P e Funções de Teste \mathbf{v} e q adequadas; especificamente, \mathbf{U} , P , \mathbf{v} e q são somas de polinômios.

Em resumo, ordem das Funções de Solução e Teste afetará o gradiente de velocidade e pressão no resultado do método de elementos finitos. Pode-se utilizar qualquer ordem de polinômio, com polinômios de ordem maior resultando em resultados numéricos mais precisos e reduzindo o risco de singularidades. Contudo, a complexidade do sistema linear a ser resolvido é diretamente proporcional à ordem da função de teste.

Para a resolução da equação de Navier-Stokes, é recomendado que o Campo de Velocidade seja associado a uma soma de polinômios de, no mínimo, ordem 2 (Quadrática), tendo em mente os efeitos que a Recirculação têm no gradiente de velocidade. Já para a pressão, pode-se utilizar uma soma de polinômios de ordem 1 (Linear). A ordem da soma de polinômios também determina que tipo de elemento será utilizado para a malha. Como o Campo de Velocidade é de ordem 2, devem ser utilizados elementos quadráticos.

3.5 Resultados Extraídos

O Método de Elementos Finitos retorna como resultado a velocidade do fluido para cada nó da malha e para cada instante de tempo simulado. O escoamento é, então, representado graficamente para facilitar a visualização das Zonas de Recirculação que ocorrem devido à expansão da tubulação.

Como recirculação origina de grandes gradientes de pressão no escoamento, o foco é observar regiões da tubulação onde o escoamento varia de direção. No caso do escoamento em um degrau, é interessante observar o comportamento do fluido na região imediatamente após o degrau, próximo às paredes. Contudo, espera-se que recirculação também ocorra na parede superior da tubulação. Na Figura 3.4 estão apresentadas as principais regiões onde espera-se a ocorrência de recirculação. As Zonas de Recirculação são separadas do escoamento principal por uma "Camada Cisalhante", facilitando suas medições.

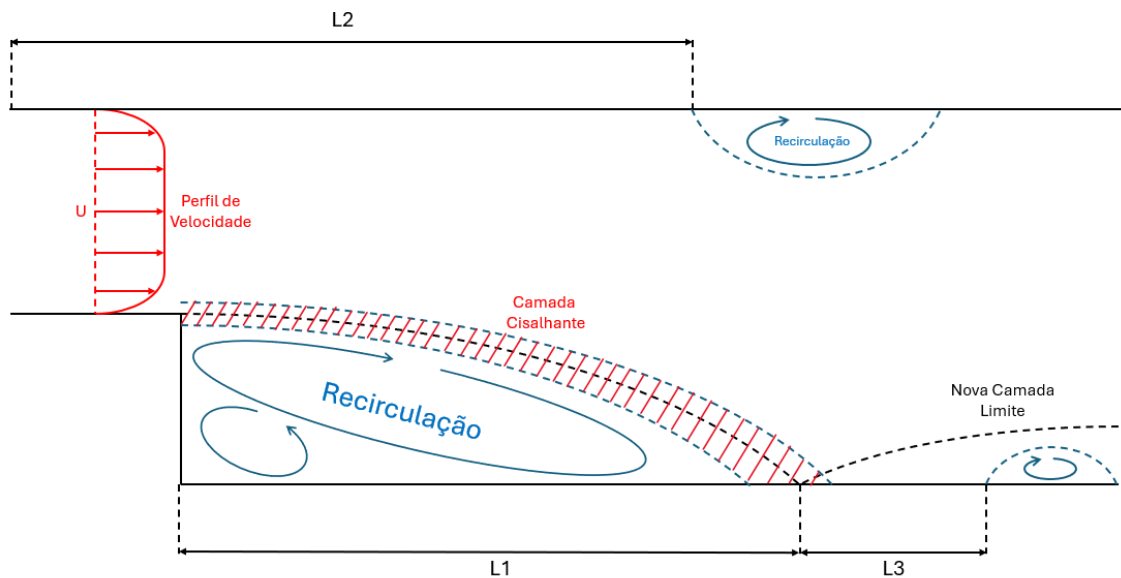


Figura 3.4: Principais medições realizadas

Este relatório focará na dimensão de 3 fenômenos do escoamento:

- L1: Comprimento da Zona de Recirculação imediatamente após o degrau
- L2: Distância entre a entrada do escoamento da tubulação até a Zona de Recirculação na parede superior da tubulação
- L3: Distância entre as Zonas de Recirculação na parede inferior da tubulação

Capítulo 4

Análise de Regressão

Regressão é um conjunto de técnicas que visam estimar qual é a relação entre 2 ou mais variáveis independentes. O objetivo é criar um modelo estatístico capaz de prever a evolução de variáveis para dados inexistentes. No escopo deste estudo, as variáveis independentes são o número de Reynolds e o comprimento da Zona de Recirculação.

4.1 Construção do Modelo de Regressão

A Regressão é realizada a partir do banco de dados pré-definido. Os dados passam por um processo iterativo, onde a função final que relaciona as variáveis independentes é refinada, visando reduzir o erro das previsões.

Primeiro define-se qual o tipo de regressão que será utilizada, ou seja, deve-se decidir qual o tipo de relação esperada entre as variáveis independentes. Por exemplo, em uma regressão polinomial de ordem n a relação é:

$$\hat{L} = w_1 \cdot Re^n + w_2 \cdot Re^{n-1} + \dots + w_{n-1} \cdot Re^1 + w_n \quad (4.1)$$

Ou pode-se impor uma regressão logarítmica:

$$\hat{L} = w_1 \cdot \log(Re) + w_2 \quad (4.2)$$

Como partimos do princípio de que o comportamento da Zona de Recirculação é desconhecido, é necessário criar e testar os vários tipos de Regressão, até que se

obtenha uma equação capaz de prever o comportamento da Zona de Recirculação com o mínimo de erro possível.

$$\hat{L} = w_1 + w_2 Re + w_3 Re^2 + \dots \quad (4.3)$$

$$s.t. \quad \min \sum_{j=1}^m |\hat{L}_j - L_j| \quad (4.4)$$

Onde m é o número de dados à análise de Regressão, \hat{L}_j é a previsão da curva de Regressão e L_j é o Resultado real, extraído da análise numérica. As Equações 4.3 e 4.4 compõem a chamada *Loss Function*, responsável por calcular o erro da Regressão ao fim de cada iteração. Os valores de w_n são então corrigidos, visando reduzir o erro.

4.2 Validação da Modelo de Regressão

Idealmente, o erro entre as previsões de uma Análise de Regressão e os resultados numéricos deve ser igual a zero. Em prática, contudo, utilizam-se métodos estatísticos para quantificar a variância das previsões. Uma solução comum é o Coeficiente de Determinação, ou R^2 . O cálculo do R^2 é dado pela Equação (4.5), onde \hat{L}_j é a previsão da Regressão, L_j é a medição extraída da análise numérica e \bar{L} é a média dos resultados numéricos.

$$R^2 = 1 - \frac{\sum_{j=1}^m (\hat{L}_j - L_j)^2}{\sum_{j=1}^m ((\hat{L}_j - \bar{L})^2)}, \quad R^2 \in [0, 1] \quad (4.5)$$

Um $R^2 = 1$ indica que o modelo de Regressão, através de suas previsões, é capaz de representar 100% da variância dos resultados numéricos, enquanto que um $R^2 = 0$ indica que o modelo representa 0% da variância. Deseja-se que a análise de Regressão possua um R^2 elevado, idealmente igual ou maior que 0.8.

Criticamente, é necessário que o modelo de Regressão seja testado tanto com Dados Brutos utilizados em seu desenvolvimento, como também com dados completamente novos, no interesse de evitar o fenômeno do Sobreajuste. Descrito por RAO *et al.* (2018), Sobreajuste ocorre quando utiliza-se um modelo de Regressão extremamente complexo e com múltiplos coeficientes w_n , como, por exemplo, uma Regressão

polinomial de ordem $n = 4$. Curvas de Regressão apresentando Sobreajuste possuem usabilidade limitada, pois o modelo produzido é impreciso para previsões fora do escopo de treinamento. Este comportamento é observável graficamente, como mostra a Figura 4.1.

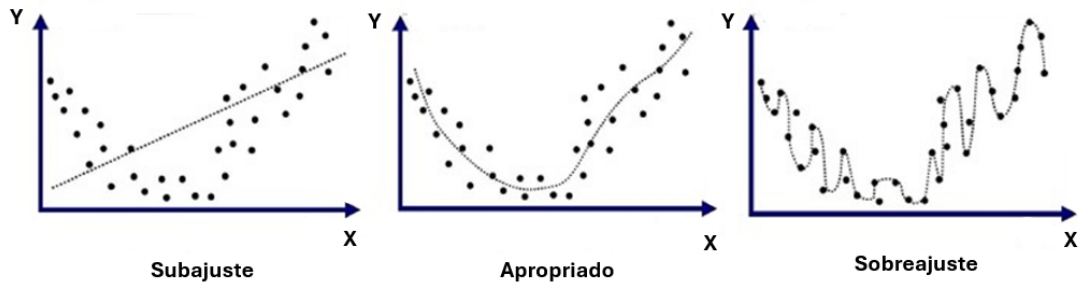


Figura 4.1: Exemplo de Sobreajuste

Para compensar a ocorrência de Sobreajuste, alterações serão necessárias no cálculo do erro. Detalhes sobre essas mudanças estão disponíveis no capítulo de Implementação Computacional.

Capítulo 5

Implementação Computacional

A implementação do Método de Elementos Finitos foi realizada utilizando a linguagem e programação *Python*. Esta escolha deve-se não só à simplicidade do *script*, visto que *Python* possui uma sintaxe relativamente intuitiva, mas também devido às vastas bibliotecas e pacotes disponíveis abertamente para análise de fluidos, tais como *Numpy*, *Scipy*, *Matplotlib*, *Gmsh* e *FeniCS*.

A base de código produzida neste estudo pode ser dividida em 2 grupos, a partir de sua funcionalidade:

- Código gerado para a definição e resolução do problema de Equações diferenciais Parciais,
- Código gerado para desenvolvimento da Análise de Regressão, seguida da avaliação de sua performance e erro de previsão.

5.1 Dinâmica de Fluidos Computacional

A abordagem computacional para a resolução das Equações de Navier-Stokes é intuitiva, pois a solução numérica do problema pode ser representada como um conjunto de matrizes cujos elementos contêm os valores de velocidade e pressão do fluido para um ponto discretizado no espaço-tempo. A implementação computacional do método de elementos finitos utiliza o módulo *Python FeniCS* e deriva dos tutoriais apresentados por LANGTANGEN e LOGG (2017) e GOCKENBACH (2006).

5.1.1 Discretização do Espaço

A estrutura de dados que armazenará os resultados é construída a partir de uma rede de nós denominada *Mesh*, onde cada nó representa um ponto no domínio, neste caso, uma tubulação.

A *Mesh* é construída utilizando a biblioteca *Python Gmsh*, que permite modelar o espaço 2D do problema como um conjunto de pontos e linhas. O modelo então pode ser discretizado para a análise numérica. A *Mesh* é então exportada no formato de arquivo *.MSH* para pós-processamento. Exemplos dessas malhas podem ser observados nas Figuras 5.1 e 5.2.

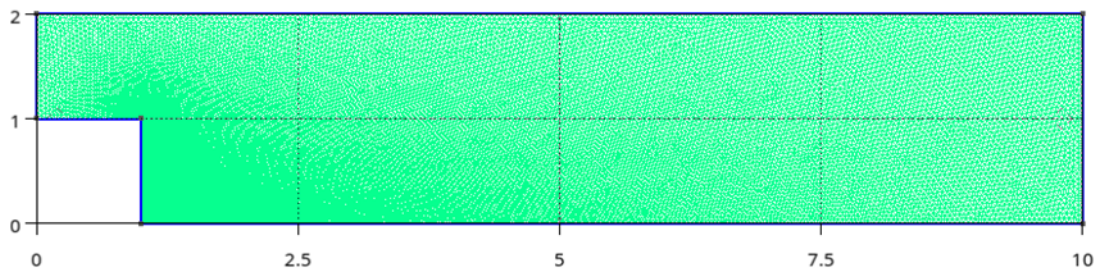


Figura 5.1: Malha discretizando o Espaço (1)

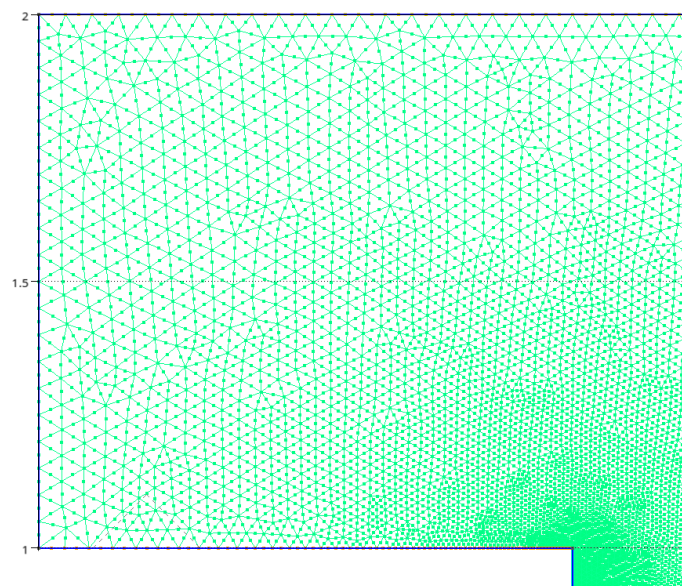


Figura 5.2: Malha discretizando o Espaço (2)

Após múltiplos testes, e considerando a magnitude dos números de Reynolds modelados, foi escolhida uma malha irregular de ordem 2 com faces triangulares, totalizando 42166 faces e 85097 nós, com uma distância mínima de 0.001 mm entre os nós. Como explicado anteriormente, a escolha de uma malha de ordem 2 aumenta a precisão dos resultados numéricos com um custo computacional relativamente pequeno se comparado a um aumento no número de faces. A Figura 5.3 mostra a diferença de elementos de ordem 1 e 2. O elemento triangular de ordem 2 dobra o número de nós por face, introduzindo-os nos centros de cada aresta.

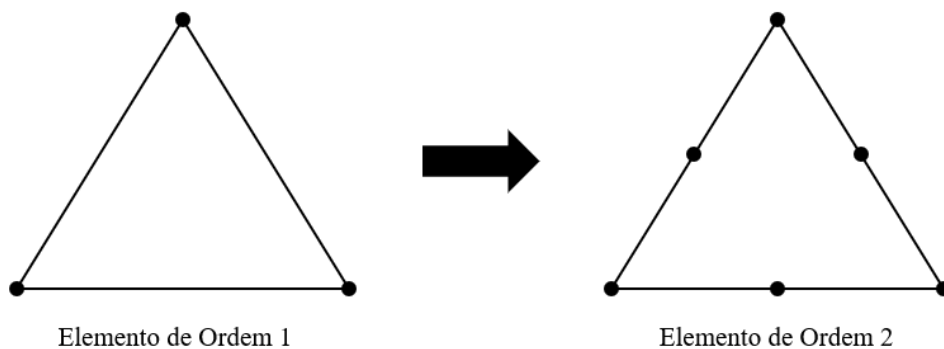


Figura 5.3: Exemplo de Elemento de Malha de Ordem 1 e 2

Como este estudo utiliza a biblioteca *Dolphin* para a realização do método de Elementos Finitos, o arquivo *.MSH* precisa ser convertido para o formato *.XML*. A biblioteca *Dolphin* fornece ferramentas para automatizar esta conversão (especificamente o comando de terminal *dolphin-convert*). Vale mencionar que na versão mais atualizada do projeto *Dolphin*, também denominado *DolphinX*, o suporte para *.XML* foi abandonado, em favor do formato *.XDMF*. A conversão do arquivo *.MSH* para *.XDMF* pode ser feita através da biblioteca *Mashio*.

5.1.2 Definindo o Passo de Tempo

A discretização do tempo é um passo importante no Método de Elementos Finitos, e deve ser realizada com cuidado para que a análise computacional ocorra sem erros. Assim como na discretização do espaço, o tempo é dividido em múltiplos instantes, separados por um intervalo de tempo denominado Passo de Tempo. Ao contrário da Malha, onde o espaço pode ser discretizado de maneira irregular com nós separados a distâncias diversas, o Passo de Tempo precisa ser constante entre todos os instantes da análise.

A escolha do Passo de Tempo é um balanço entre velocidade de análise e convergência do método numérico. Se o Passo de Tempo for um valor muito pequeno, a análise ocorrerá normalmente, porém o número total de instantes analisados aumenta proporcionalmente. Por exemplo, considere que deseja-se estudar o comportamento de um escoamento em um intervalo de tempo de 10 segundos, e seja escolhido um Passo de Tempo de 0,001 segundos. Isso significa que o Método de Elementos Finitos deverá ser aplicado 10.000 vezes para todos os elementos da Malha para que se possa descrever corretamente todos os instantes no domínio do tempo.

Contudo, o uso de um Passo de Tempo muito grande impede que o Método de Elementos Finitos convirja corretamente devido ao fenômeno de Singularidade. No escopo de dinâmica de fluidos, a Singularidade ocorre quando o escoamento apresenta gradientes elevados de velocidade ou quando o fluido atravessa um número grande de elementos da malha entre dois instantes de tempo. A Singularidade desestabiliza o cálculo numérico da velocidade, normalmente levando ao crescimento descontrolado de sua magnitude e inevitavelmente levando ao cancelamento automático da análise computacional devido ao *Overflow* na memória RAM. Logo, uso de um critério é recomendado para a otimização do Passo de Tempo. No caso deste estudo, será utilizada a Condição de Courant-Friedrichs-Lewy, mostrada abaixo.

$$dt = dx \times \frac{C}{V_{max}} \quad (5.1)$$

Onde dt é o Passo de Tempo, dx é a menor distância entre 2 elementos na malha, V_{max} é a maior velocidade esperada para o fluido durante a simulação e C é o *Número de Courant*. Como o método numérico utilizado neste estudo é explícito, o Número de Courant $C = 1$ pode ser adotado. Devido às forças externas, como a gravidade,

às quais o escoamento está submetido, espera-se que a velocidade do escoamento seja maior no interior da tubulação do que na sua entrada. Logo, recomenda-se utilizar a Condição de Courant-Friedrichs-Lewy como um limite máximo para o Passo de Tempo, e sempre testar com valores menores até que se escolha um valor que consiga convergir adequadamente rápido, sem ocorrência de Singularidades RANOGA *et al.* (2021).

5.1.3 Escrevendo o Código *Python*

Com o domínio da análise construído, partimos para a implementação das equações de Navier-Stokes em código *Python*. Para isso, utilizaremos a biblioteca *FeniCS*, uma plataforma *Open-Source* especializada para a formulação e resolução de sistemas de Equações Diferenciais Parciais.

A primeira vantagem de métodos numéricos computacionais é a parametrização do problema, fornecendo rápido acesso às principais variáveis do problema, tais como as propriedades do fluido sendo analisado, as dimensões da tubulação, o tempo de análise e passo de tempo. Abaixo está apresentado um exemplo do código onde a análise é feita utilizando um fluido com as propriedades físicas d'água.

```

def modelGen():
    #=====Inputs=====
    #Propriedades do Fluido
    mu = 0.001 # Viscosidade Cinematica
    rho = 1 # Densidade
    maxVelocity = 1.5 # Velocidade Maxima na Entrada

    #Dominio no Tempo
    T = 30.0/(maxVelocity) # Tempo final
    dt = 0.001 * 1/(maxVelocity) # Condicao Courant-Friedrichs-Lewy
    numSteps = int(T/dt) + 1 # Numero de Passos de Tempo

    #Definindo Pressao de Saída do Escoamento
    pOutflow = 0

    #Definindo Altura do Degrau
    height = 1
    #=====

```

FeniCS utiliza um paradigma Orientado a Objetos em sua base de códigos. As EDP são destiladas em um conjunto de instâncias de Classes, contendo em si as propriedades do problema. Por exemplo, as Condições de Contorno são objetos que agrupam não só os valores de Velocidade e Pressão, mas também as coordenadas dos pontos na fronteira do domínio.

5.1.4 Extração de Resultados

Os resultados são armazenados em arquivos *.XDMF* e *.H5*, ideais para visualização e pós-processamento. Neste estudo, a aplicação utilizada para visualização da evolução do fluido é o *Paraview*.

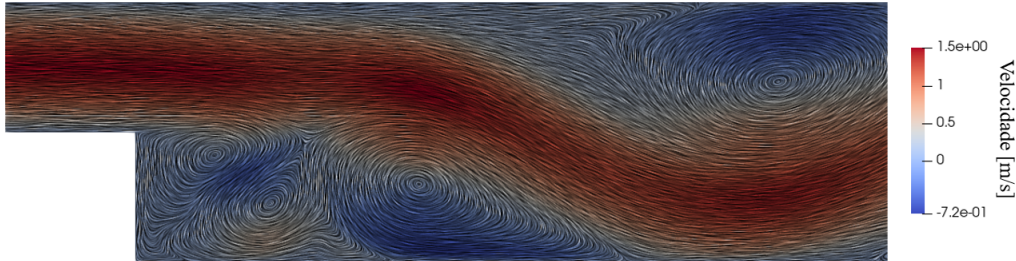


Figura 5.4: Exemplo de Perfil de Velocidade do Fluido Obtido por *FeniCS*

A Figura 5.4 mostra uma visualização, usando a ferramenta *Paraview*, dos resultados numéricos de velocidade obtidos pelo *FeniCS*. Os resultados foram pós-processados, aplicando-se *Convolução de Integral de Linha* para que o campo de vetores possa ser facilmente discernido. Pode-se observar a ocorrência do fenômeno de Recirculação após a expansão do diâmetro da tubulação.

Essas visualizações são fundamentais para extrair as medições relevantes para este estudo. Particularmente, deseja-se medir as dimensões nas regiões do domínio onde espera-se a ocorrência de uma Zona de Recirculação, como descreve a Figura 5.5.

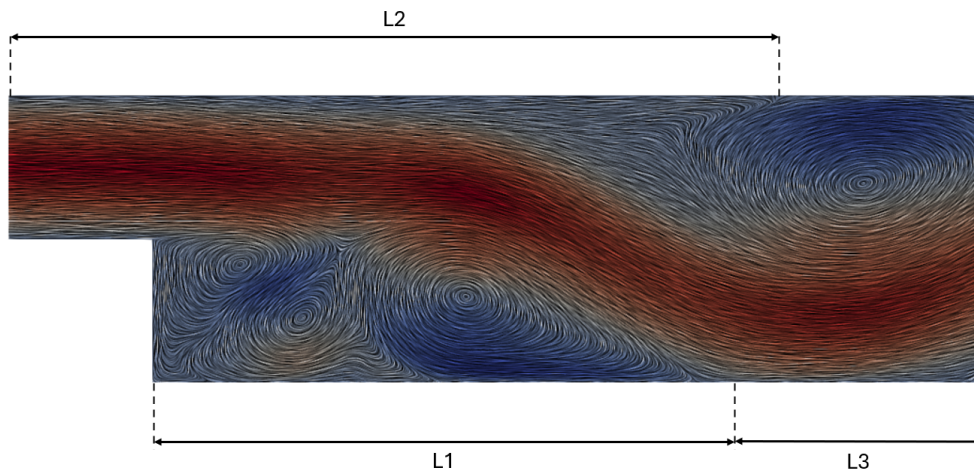


Figura 5.5: Medições da Recirculação

Para que as medições sejam precisas, é necessário que as seções do domínio

onde elas se localizam sejam mais refinadas, ou seja, que a distância entre nós seja reduzida. A justificativa para este imperativo é devido ao grande gradiente de velocidade que se espera nessas regiões. Elevadas flutuações de velocidade no fluido fazem com que a Zona de Recirculação evolua rapidamente de tamanho. Com as dimensões dos elementos da Malha reduzidas, pequenas alterações no campo de velocidade transiente do fluido podem ser facilmente observadas.

A precisão das medições é crítica para a Análise de Regressão, que busca encontrar uma correlação entre a evolução da Zona de Recirculação e o número de Reynolds do fluido. Caso as variações no Re não sejam perceptíveis nas medições, a análise fornecerá uma curva final com previsibilidade reduzida. As medições são coletadas e armazenadas numa planilha que será fornecida para a Análise de Regressão.

Re	L_3 [mm]	L_2 [mm]	L_1 [mm]
666.66	1880	5350	1706
1000	1800	5350	1730
1333.33	1790	5350	1690
1666.66	1835	5350	1623
2000	1798	5350	1572
2333.33	1880	5300	1504
2666.66	1843	5300	1494
3000	1907	5300	1457
3333.33	1915	5300	1475
3666.66	1961	5300	1457

Tabela 5.1: Relação das Medições com o Número de Reynolds do Fluido

Na Tabela 5.2 pode-se observar como as diferentes regiões com Recirculação variam de tamanho em função do número de Reynolds. L_1 é o comprimento da Zona de Recirculação imediatamente após o degrau na parede inferior da tubulação, até o ponto na parede da tubulação onde a velocidade do fluido na parede retorna à sua condição de não deslizamento (velocidade = 0). L_2 é a distância entre a entrada da tubulação e a Zona de Recirculação na parede superior da tubulação. Por fim, L_3 é a distância entre as Zonas de Recirculação que ocorrem na parede inferior. As

medições são salvas em um arquivo *.CSV* chamado *ML_Data.csv*, e compõem a base de dados para Regressão.

5.1.5 Verificação do Método Numérico

Antes de avançarmos com a geração de análises numéricas, é essencial que se verifique a qualidade dos resultados do código, comparando-os com outros estudos na literatura disponível. Isso garante que a Regressão seja feita com dados que refletem adequadamente a realidade. Serão usados como referência os resultados demonstrados por BISWAS *et al.* (2004), para um Número de Reynolds $Re = 400$, para a mesma geometria de tubulação utilizada neste relatório.

	L_1 [mm]
Referência	≈ 7700
Resultados Numéricos	7020

Tabela 5.2: Verificação da Análise Numérica

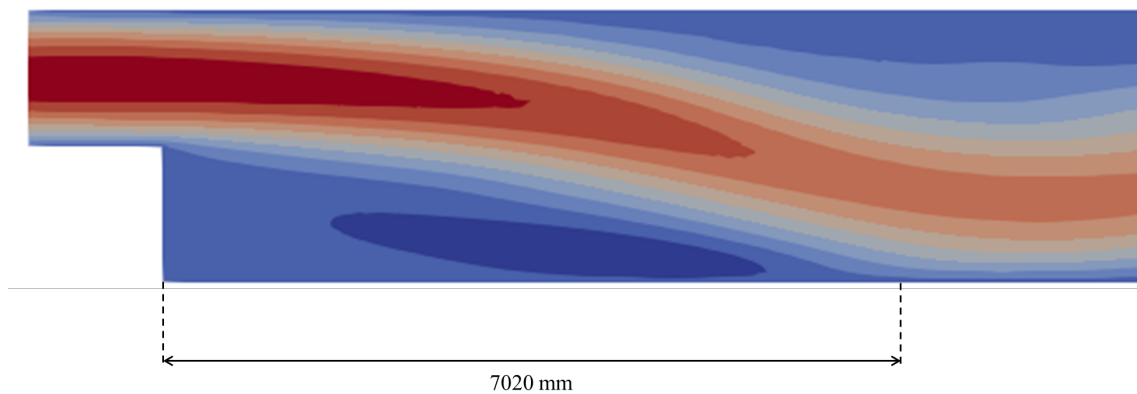


Figura 5.6: Comprimento da Zona de Recirculação para $Re = 400$ (Resultado Numérico)

O resultado numérico extraído do código apresentou um erro de 8.83% em relação aos dados apresentados na literatura, um valor relativamente pequeno que legitima a qualidade do *Output* do código *FeniCS*.

5.2 Regressão

Para a análise de regressão será utilizada a biblioteca *Python Open Source scikit-learn*, que oferece um vasto conjunto de algoritmos estimadores capazes de relacionar o número de Reynolds com a dimensão da Zona de Recirculação.

Como explicado na seção anterior, as medições extraídas dos resultados numéricos estão armazenadas no formato *.CSV* e são lidas pelo código *Python* utilizando a biblioteca *Pandas*. O próximo passo é pós-processar a informação coletada para que se encaixe no modelo algébrico $f(Re) = L$ sendo testado, que relaciona o Número de Reynolds com o tamanho da Recirculação.

Por exemplo, se for utilizada a hipótese de que a zona é quadraticamente proporcional ao Número de Reynolds, é necessário calcular o quadrado do número de Reynolds para todas as medições realizadas. A biblioteca *Sklearn* já fornece as funções necessárias para o processamento dos dados.

Com os dados processados, a última etapa antes da Análise de Regressão é a escolha do tipo de Curva que melhor se encaixa aos resultados numéricos. Como discutido no capítulo anterior, uma *Loss Function* modificada é utilizada para combater os possíveis efeitos de Sobreajuste. No escopo deste estudo, os efeitos de Sobreajuste são compensados na avaliação modificando o método de mínimos quadrados utilizado na Regressão.

$$\min_{w_i} \frac{1}{2n} \sum (\hat{L} - L)^2 + \alpha \sum (w_i)^2 \quad (5.2)$$

É possível notar algumas diferenças quando comparado ao método de mínimos quadrados tradicional 4.3. Aqui, n é o número de amostras utilizadas na análise, ou seja, o número de modelos numéricos gerados e α é um parâmetro de ajuste, entre 0 e 1. A função do $\frac{1}{2n}$ é normalizar o erro esperado para uma base de dados muito extensa, enquanto a expressão $\alpha \sum (w_i)^2$ visa combater os efeitos de *Overfitting* observáveis em Regressões muito complexas, como, por exemplo, um polinômio de grau 4 ou maior.

Nesta análise será utilizado um peso $\alpha = 0.3$. Recomenda-se testar a regressão com vários valores de α , especialmente com bases de dados maiores, até que se obtenha um resultado consistente da Regressão.

Com isso, a etapa de regressão está pronta para começar. No *Scikit-learn* a

análise é abstraída na forma de um Objeto. Especificamente, é uma instância da Classe *Ridge* (devido à *Loss Function* utilizada). A construção de um Objeto *Ridge* requer alguns argumentos de entrada, entre eles:

- O peso α ;
- Os resultados do Método de Elementos Finitos para Análise de Regressão. Neste estudo o número e iterações máximo é 2000;
- A porcentagem h dos Resultados numéricos que serão usados na apenas na validação da Curva de Regressão, e não em sua construção. Neste estudo 30% dos Resultados numéricos serão utilizados apenas para validação.

O Objeto gerado representa a Curva de Regressão. A validação desta Curva é feita a partir de funções da Classe *Ridge*, que retornam o Erro Absoluto Médio e o resultado final da *Loss Function*.

Com a Curva de Regressão definida, basta então armazená-la em um formato conveniente para rápida reutilização, seja para fazer novas previsões ou para continuar seu treinamento. A biblioteca *Python Pickle* facilita isso, salvando o Objeto *Ridge* no arquivo de formato *.PKL*. Esse arquivo pode então ser armazenado localmente ou mesmo compartilhado com outros usuários e computadores. O modelo de Regressão final é propriamente portátil, um dos objetivos deste estudo.

Capítulo 6

Discussão de Resultados

Com os modelos gerados, a análise de Regressão é realizada. Como mencionado no capítulo anterior, uma fração dos modelos gerados é utilizada exclusivamente para construção da Regressão, enquanto os demais são usados para uma fase de testes, a fim de eliminar os efeitos de Sobreajuste. No escopo deste projeto, consideramos uma razão 70% modelos treinamento / 30% modelos teste.

Os dados de L_1 , extraídos da Tabela 5.2, serão utilizados para a Análise de Regressão. Na Figura 6.1, pode-se observar um gráfico, descrevendo a evolução da distância L_1 em função do Número de Reynolds nas simulações computacionais.

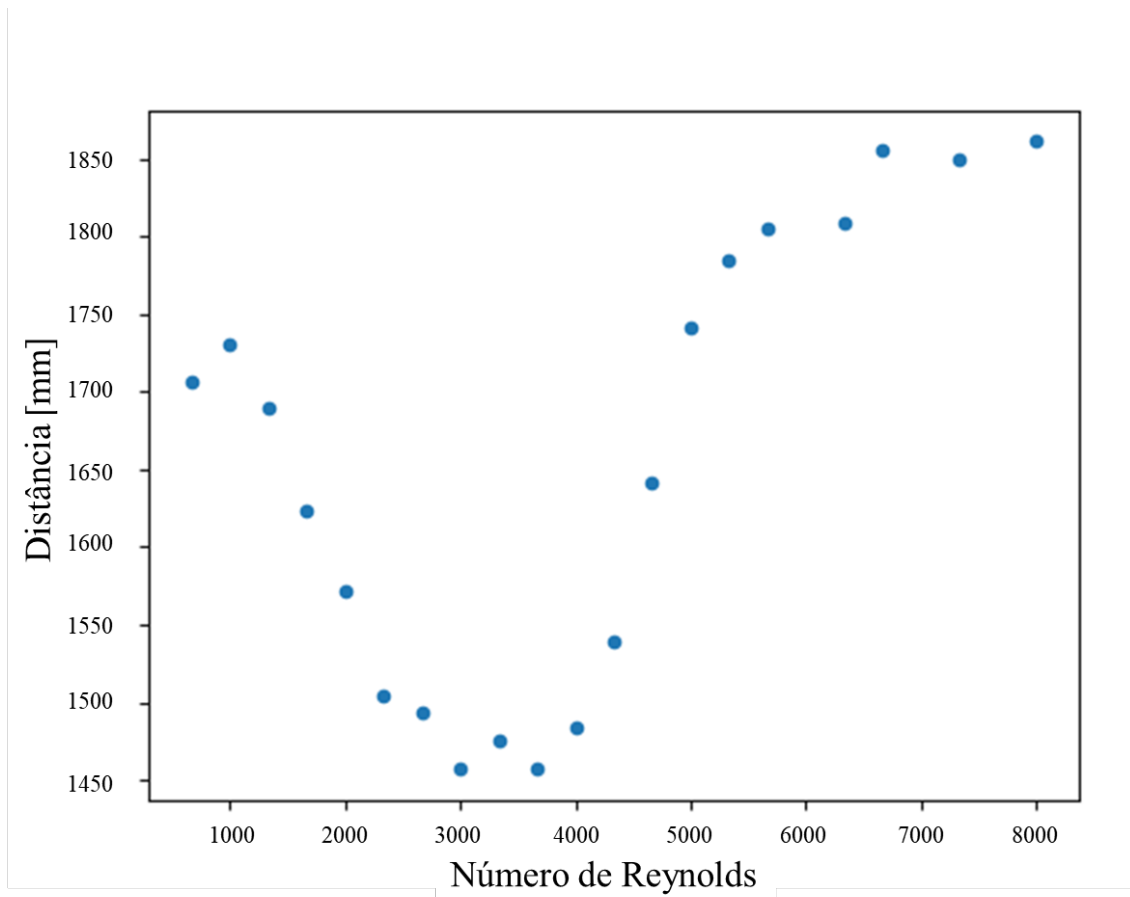


Figura 6.1: Resultados das Medições do Comprimento da Zona de Recirculação em função do Número de Reynolds

Como pode-se observar, a dimensão da zona de recirculação claramente não é linear em função do Número de Reynolds; logo, precisamos testar múltiplas hipóteses sobre qual será essa relação e realizar a Regressão adequadamente.

6.1 Hipótese 1: Relação Polinomial de Ordem 2

Partimos do princípio de que o desprendimento segue a função

$$\hat{L} = w_0 + w_1 \cdot Re + w_2 \cdot Re^2 \quad (6.1)$$

A Regressão aqui deve resultar nos 3 coeficientes w_0 , w_1 e w_2 que retornem o menor resultado da *Loss Function*. Após feita a Regressão, obtemos os seguintes valores apresentados na Tabela 6.1. A partir desses resultados, a equação descrevendo a evolução da recirculação pode ser desenhada.

Coeficiente	Valor
w_0	1745.55
w_1	-1.00E-01
w_2	1.61E-05

Tabela 6.1: Coeficientes Polinomiais de Ordem 2 Determinados pela Regressão

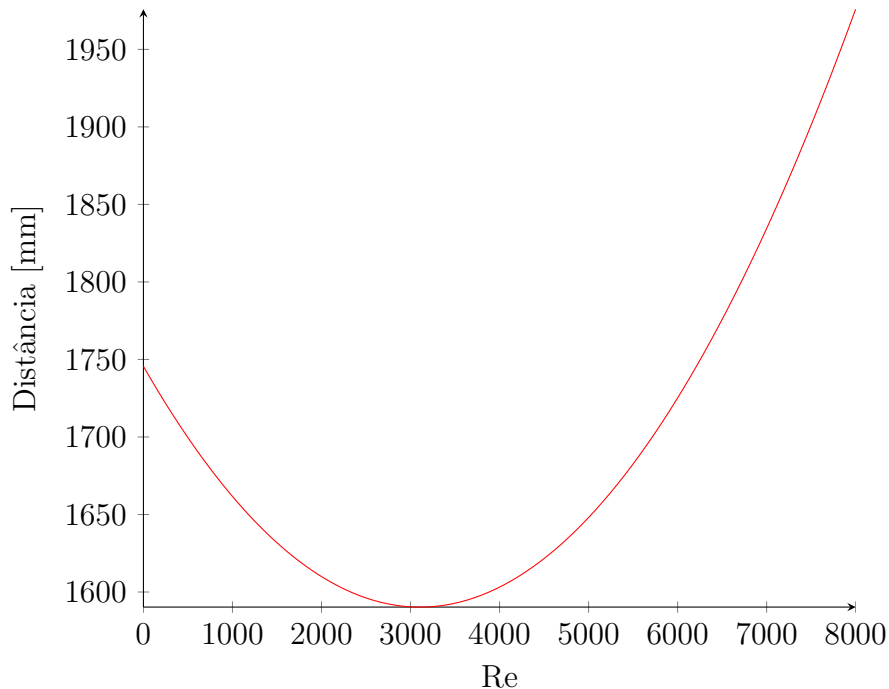


Figura 6.2: Resultados Obtidos pela Regressão para Comprimento da Zona de Recirculação em função do Número de Reynolds - Polinômio de Ordem 2

Observa-se que a curva resultante descreve aproximadamente as medições apresentadas na Figura 6.1. Contudo, com exceção da concavidade da curva na região de

$2500 < Re < 3500$, um polinômio de Ordem 2 é incapaz de descrever o comprimento da zona para $Re < 1000$ e $Re > 6000$. Isso torna-se mais evidente após a validação da análise.

Na validação da análise de Regressão, comparamos os resultados extraídos das simulações computacionais com as previsões obtidas pela Regressão, para todos os modelos gerados. Como mencionado anteriormente, a validação deve ser realizada tanto com os modelos utilizados para a análise de Regressão, como também para modelos novos, utilizados exclusivamente para este teste.

Para facilitar a visualização dos resultados, foi gerada a Figura 6.3. Demonstrado por NADDA *et al.* (2023), cada ponto representa um caso sendo testado, ou seja, um número de Reynolds sendo estudado. Já os eixos indicam qual o comprimento da zona obtido, sendo no eixo das abscissas o valor retirado diretamente das simulações, enquanto o eixo das ordenadas representa a previsão fornecida pela Curva de Regressão. Quanto mais próximos esses pontos estiverem da linha diagonal preta, menor será o erro entre as medições extraídas de simulações numéricas e as previsões feitas pela Regressão. Quanto mais afastados esses pontos estiverem da linha diagonal, maior será a discrepância entre as duas. Observa-se na Figura 6.3 que um modelo polinomial quadrático é impreciso para a maioria dos casos testados, uma vez que a maioria dos pontos está afastada da linha diagonal preta.

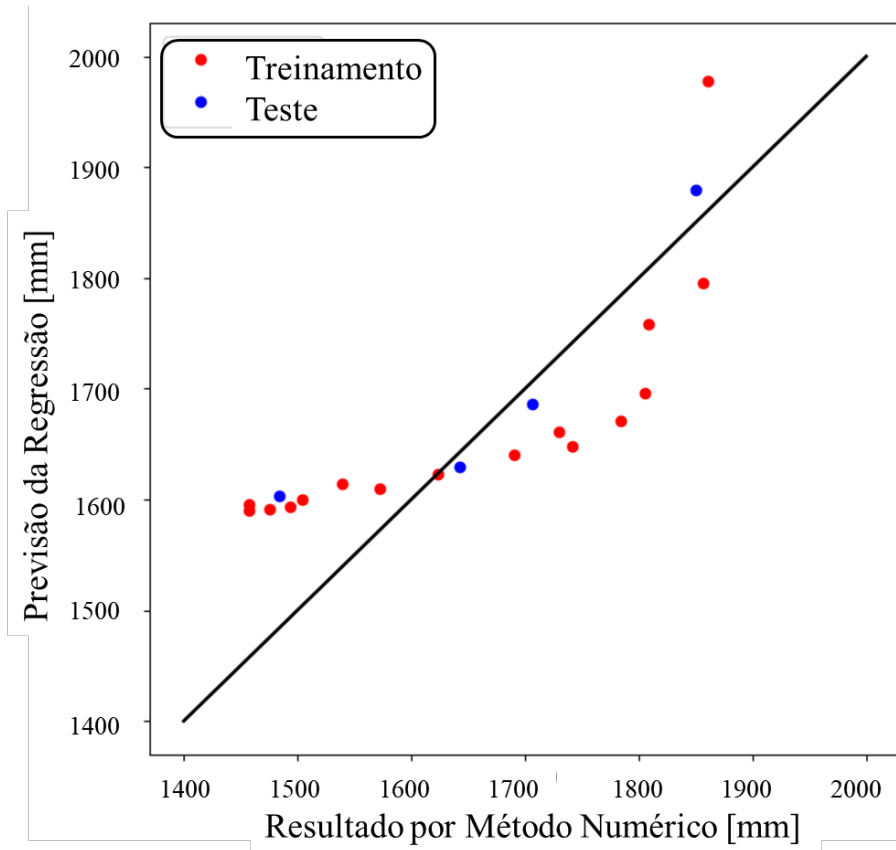


Figura 6.3: Comparação entre as Previsões da Regressão Polinomial e o Valores Reais

Como discutido no capítulo de Regressão, a acurácia da curva é determinada através do Coeficiente de Determinação, ou R^2 , que para esta hipótese é aproximadamente 63.42%

6.2 Hipótese 2: Relação Polinomial de Ordem 3

Partimos do princípio de que o desprendimento segue a função

$$\hat{L} = w_0 + w_1 \cdot Re + w_2 \cdot Re^2 + w_3 \cdot Re^3 \quad (6.2)$$

Assim como na Hipótese 1, a Regressão aqui deve fornecer os coeficientes, agora w_0 , w_1 , w_2 e w_3 , que retornem o menor resultado da *Loss Function*. Após feita a Regressão, obtemos os valores na Tabela 6.2:

Coeficiente	Valor
w_0	2248
w_1	-6.13E-01
w_2	1.48E-04
w_3	-9.76E-09

Tabela 6.2: Coeficientes Polinomiais de Ordem 3 Determinados pela Regressão

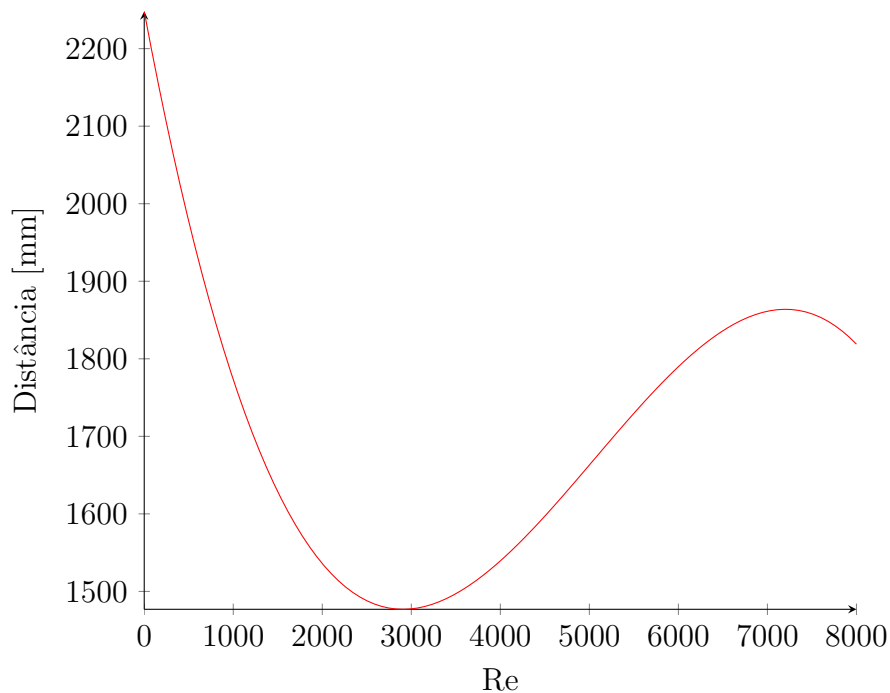


Figura 6.4: Resultados obtidos pela Análise de Regressão para o Comprimento da Zona de Recirculação em função do Número de Reynolds - Polinômio de Ordem 3

Comparada à hipótese anterior, um polinômio de ordem 3 descreve mais precisamente a dimensão da Zona de Recirculação para números de Reynolds elevados.

Como mencionado anteriormente, a validação deve ser realizada tanto com modelos utilizados para a Análise de Regressão, como também para modelos novos, utilizados exclusivamente para este teste.

A precisão da Regressão é ilustrada na Figura 6.5. Comparado aos resultados na Figura 6.3, a hipótese polinomial de ordem 3 possui uma precisão consideravelmente maior, uma vez que a distância entre os pontos até a linha preta no gráfico é reduzida consideravelmente. Após cálculos, o R^2 obtido é aproximadamente 82.96%.

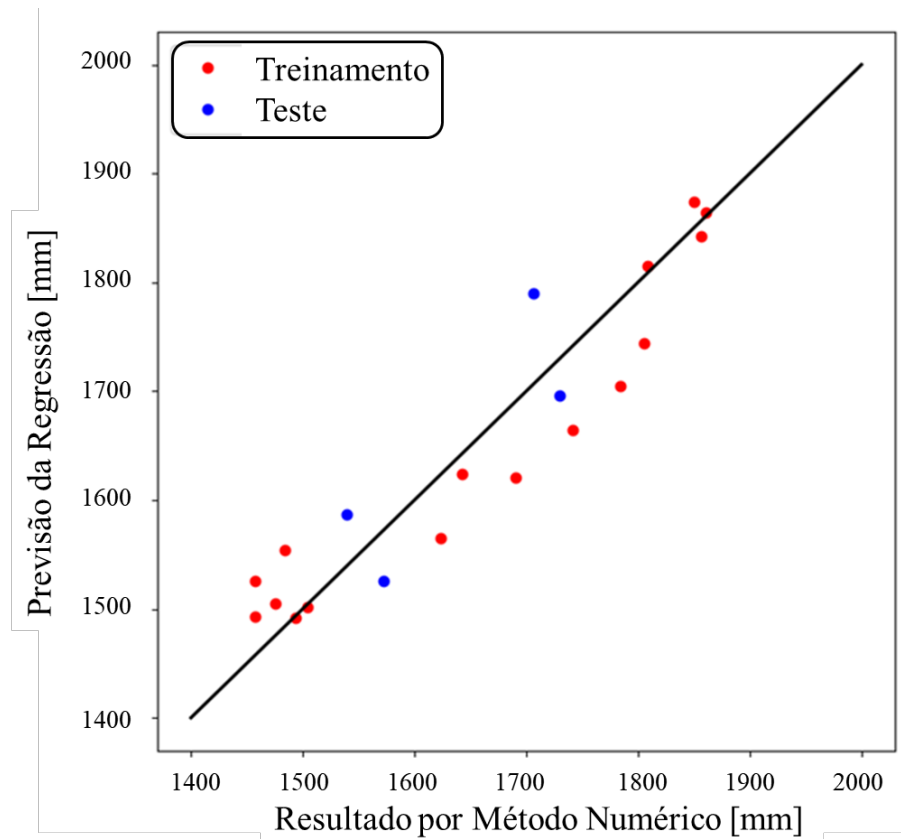


Figura 6.5: Comparação entre as Previsões da Regressão Polinomial e o Valores Reais

6.3 Hipótese 3: Relação Polinomial de Ordem 4

Partimos do princípio de que o desprendimento segue a função

$$\hat{L} = w_0 + w_1 \cdot Re + w_2 \cdot Re^2 + w_3 \cdot Re^3 + w_4 \cdot Re^4 \quad (6.3)$$

Assim como na Hipótese 1, a Regressão deve fornecer os coeficientes, agora w_0 , w_1 , w_2 , w_3 e w_4 , que levem ao menor resultado da *Loss Function*. Após feita a Regressão, obtemos os seguintes valores:

Coeficiente	Valor
w_0	2305.03
w_1	-6.98E-01
w_2	1.81E-04
w_3	-1.43E-08
w_4	2.05E-13

Tabela 6.3: Coeficientes Polinomiais de Ordem 4 Determinados pela Curva de Regressão

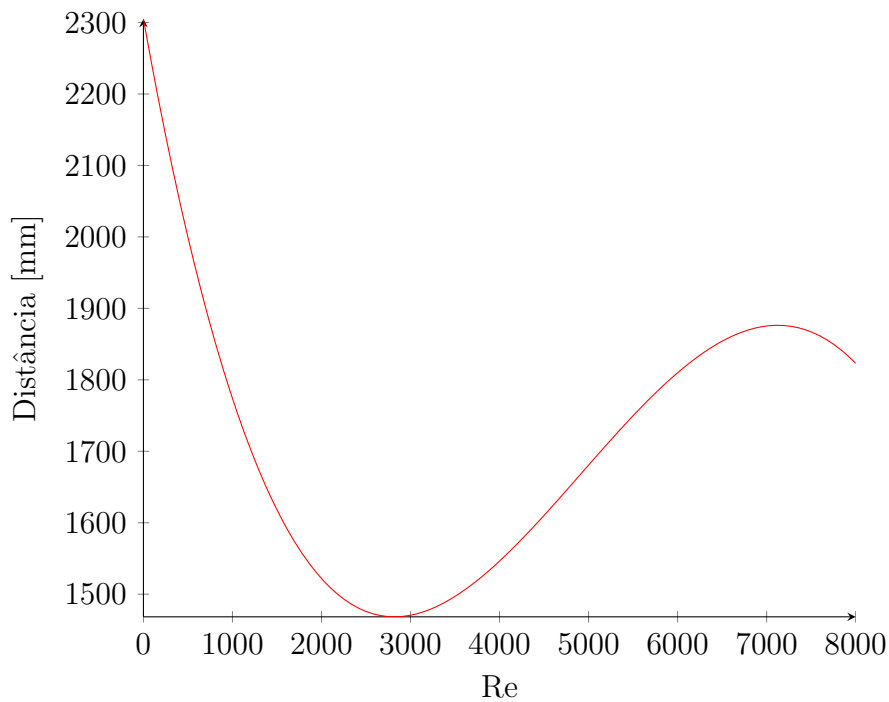


Figura 6.6: Resultados obtidos pela Análise de Regressão para Comprimento da Zona de Recirculação em função do Número de Reynolds - Polinômio de Ordem 4

Comparando os polinômios de ordem 3 e 4, observa-se uma mudança pequena na curva obtida, devido principalmente à pequena magnitude do coeficiente w_4 . Podemos concluir que o uso de Regressões Polinomiais de ordens maiores que 4 produzirá retornos cada vez maiores na precisão.

Como mencionado anteriormente, a validação deve ser realizada tanto com modelos utilizados para a análise de Regressão, como também por modelos novos, utilizados exclusivamente para este teste.

A precisão da curva de Regressão é ilustrada na Figura 6.7. Comparada à Figura 6.5, a distância dos pontos até a linha diagonal é similar, indicando que não houve aumento perceptível na precisão da Regressão ao utilizar um polinômio mais complexo como hipótese. Após cálculos, o R^2 obtido é aproximadamente 81.60%.

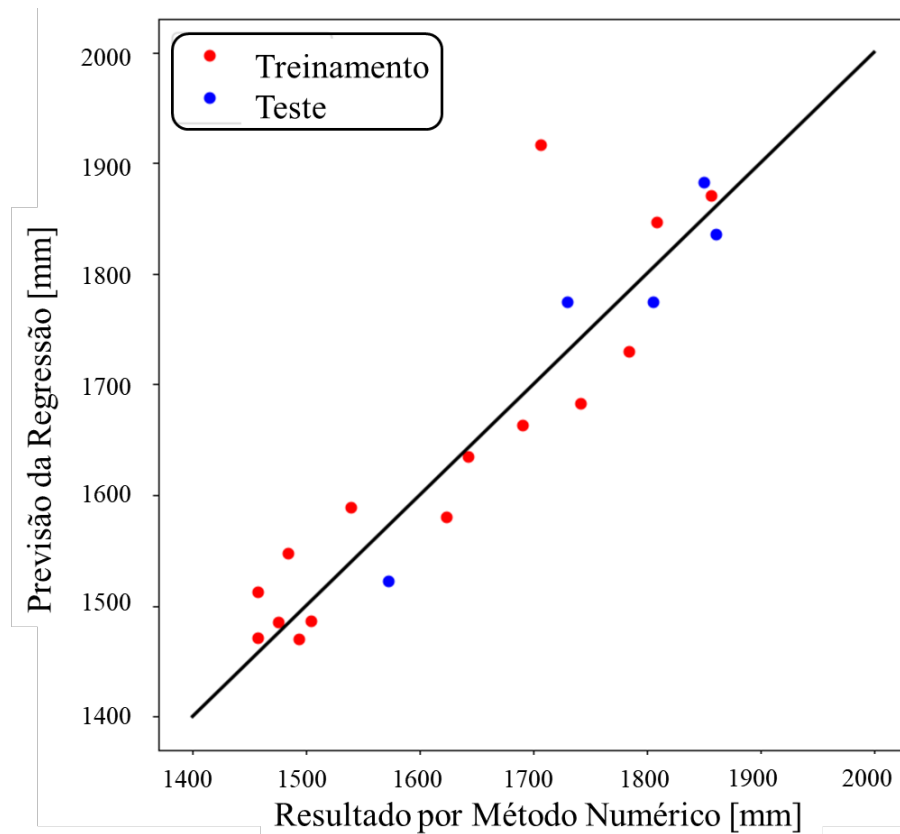


Figura 6.7: Comparação entre as Previsões da Regressão Polinomial e o Valores Reais

6.4 Hipótese 4: Relação Logarítmica

Partimos do princípio de que o desprendimento segue a função

$$\hat{L} = w_0 + w_1 \cdot \ln(Re) \quad (6.4)$$

Assim como nas demais hipóteses, a Regressão Logarítmica aqui deve retornar os coeficientes, agora w_0 e w_1 , que retornem o menor resultado da *Loss Function*. Após construída a Regressão, obtemos os seguintes valores:

Coefficiente	Valor
w_0	1272.91
w_1	49.77

Tabela 6.4: Coeficientes Logarítmicos Determinados pela Análise de Regressão

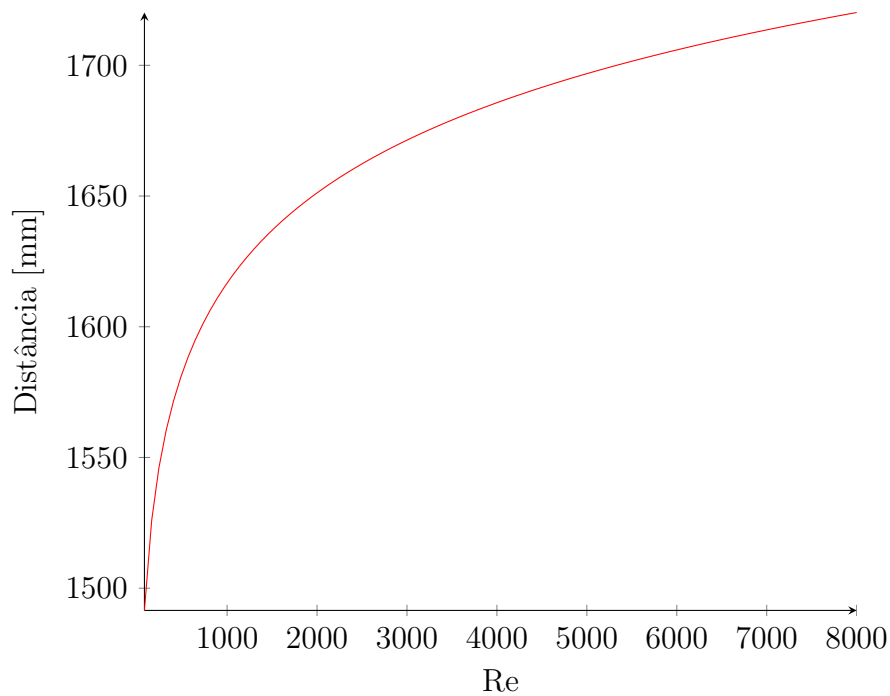


Figura 6.8: Resultados obtidos pela Regressão Logarítmica para comprimento da Zona de Recirculação em função do Número de Reynolds

Claramente, esta curva não representa adequadamente a evolução da zona, um fato facilmente observado na Figura 6.9. A Regressão Logarítmica forneceu a pior precisão de todas as hipóteses testadas, com a maior divergência entre os resultados numéricos e a Curva de Regressão. Após cálculos, o R^2 obtido é aproximadamente 6.79%.

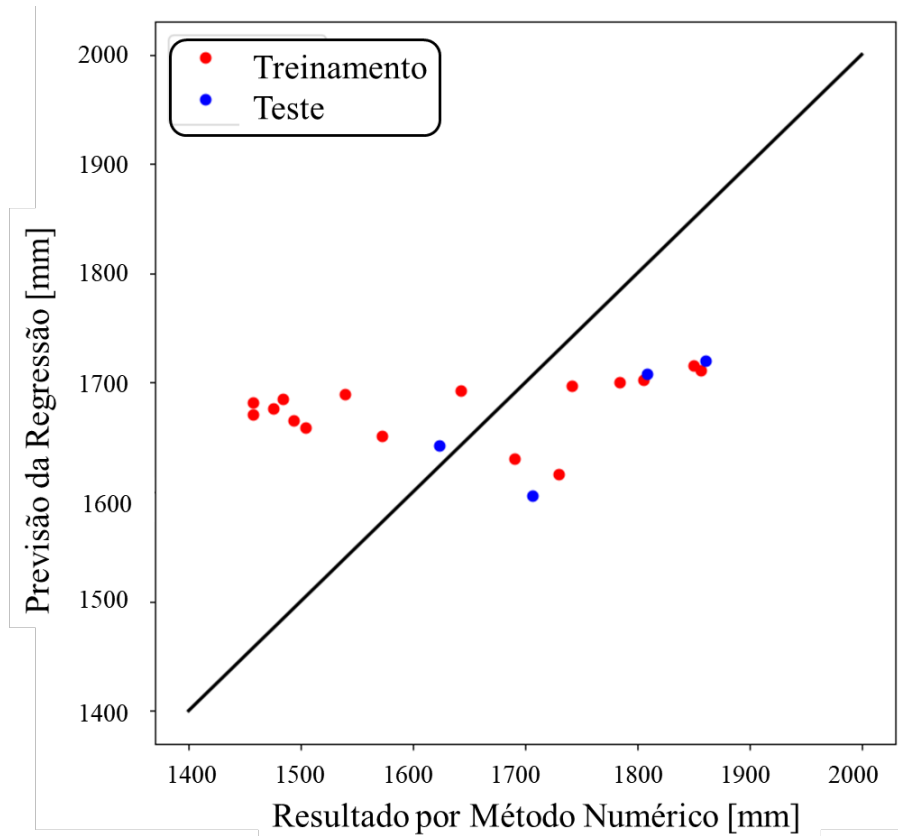


Figura 6.9: Comparação entre as Previsões da Regressão Logarítmica e o Valores Reais

Capítulo 7

Conclusão

A partir dos resultados, podemos concluir que, dentre as propostas testadas, a Regressão Polinomial de ordem 3 fornece a curva mais precisa relacionando o Número de Reynolds com o comprimento da Zona de Recirculação. Quando comparada às demais hipóteses, a abordagem polinomial de 3^o grau apresentou a maior acurácia nas previsões, com o Coeficiente de Determinação R^2 adequadamente próximo de 1, e, ao mesmo tempo, sem apresentar sinais de Sobreajuste devido ao excesso de coeficientes. Observa-se que o uso de Regressões Polinomiais de ordens superiores fornece aumentos decrescentes na precisão da ferramenta, com um aumento no risco de Sobreajuste. Portanto, recomenda-se que testem hipóteses diferentes que descrevam a relação entre o Número de Reynolds e a Zona de Recirculação.

A principal vantagem da Regressão está no baixo custo computacional e no tempo de resposta. Uma vez criada, a curva consegue facilmente prever o comportamento da Recirculação em função do número de Reynolds. No mesmo intervalo de tempo que uma análise numérica é realizada computacionalmente, dezenas de milhares de casos podem ser iterados.

7.1 Retorno de Experiência

Um dos objetivos do trabalho descrito neste relatório foi desenvolver ferramentas em *Python* para a resolução das Equações de Navier-Stokes usando o Método de Elementos Finitos (FEM) e o uso de Regressão para prever acuradamente a magnitude de fenômenos hidrodinâmicos. No fim, o projeto resultou na criação de

múltiplas ferramentas (disponíveis nos anexos) e no domínio de outras ferramentas *Open-Source*:

- Domínio da ferramenta *Gmsh* para a criação da malha relativa a um modelo com múltiplos elementos e diferentes condições de contorno,
- Domínio da ferramenta *FeniCS* para a resolução das Equações Diferenciais Parciais que descrevem o comportamento do escoamento ao longo da tubulação,
- Criação de um código capaz de solucionar as Equações de Navier-Stokes para múltiplos números de Reynolds automaticamente,
- Domínio da ferramenta *Sklearn* para a construção e validação de um Modelo de Regressão capaz de ser facilmente compartimentalizado e reutilizado em uma gama ampla de dispositivos.

Além dos pontos levantados acima, o estudo forneceu um melhor entendimento sobre a história do *Software* comercial para resolução de Equações diferenciais parciais, os principais fatores limitantes na precisão de métodos numéricos computacionais e a aplicabilidade de Regressão para contorná-los.

7.2 Limitações e Sugestões para Projetos Futuros

O aprimoramento da Análise de Regressão depende principalmente da qualidade dos dados fornecidos durante o seu treinamento. Isso significa que a etapa de geração de modelos numéricos é onde se concentram as principais recomendações para projetos futuros que desejem continuar com o estudo apresentado neste relatório. Porém, deve-se ter em mente que quaisquer alterações no método numérico computacional estão sujeitas às limitações de *Hardware* presentes.

O uso de um número maior de elementos para o Método de Elementos Finitos proporciona medições que melhor ilustram a dimensão da Zona de Recirculação e, crucialmente, permite que pequenas variações no Número de Reynolds sejam mensuráveis nos modelos numéricos e, conseqüentemente, registradas na curva final. O mesmo pode ser dito para a ordem dos elementos. Contudo, vale sempre considerar

que o refinamento da malha espacial inevitavelmente aumenta o tempo de processamento necessário para a resolução. O refinamento de malha também requer a adoção de um Passo de Tempo menor, a fim de evitar a ocorrência de singularidades durante a resolução do Método de Elementos Finitos. Essas alterações inevitavelmente requerem o uso de uma solução de armazenamento grande o bastante para conter todos os arquivos *.XDMF* e *.H5* resultantes da análise computacional.

Todos os resultados descritos neste relatório são intrinsecamente ligados à geometria utilizada para a tubulação. As previsões da curva construída aqui não descrevem a Zona de Recirculação para outro curso de escoamento. Contudo, é teoricamente possível realizar uma análise capaz de produzir previsões dependendo não só do Número de Reynolds, mas também das dimensões do espaço, tais como diâmetro do tubo, altura do degrau e comprimento da tubulação. A influência que esses fatores têm sobre a magnitude da Zona de Recirculação deverá ser averiguada, gerando múltiplos modelos numéricos para a Regressão e testando hipóteses sobre qual é a possível função matemática entre a geometria do tubo e a Zona de Recirculação.

Contudo, caso deseje-se expandir o escopo da análise, vale considerar uma total substituição da Análise de Regressão por um modelo Neural. Métodos de Regressão possuem aplicabilidade limitada no estudo de grandes *Datasets*, especialmente quando a relação entre as variáveis independentes não é linear ou polinomial. Redes Neurais, em comparação, são capazes de descrever comportamentos mais complexos e não requerem que o pesquisador assuma uma relação entre as variáveis.

Referências Bibliográficas

- HEATH, M. T. *Scientific Computing - An Introductory Survey*, v. 1. MC GRAW HILL, 2002. doi: 10.1007/s42967-021-00159-w.
- LEWIS, R. W., NITHIARASU, P., SEETHARAMU, K. N. *Fundamentals of the finite element method for heat and fluid flow*. John Wiley Sons, 2004.
- ANJOS, G. R. “COMPUTACAO CIENTIFICA PARA ENGENHEIROS”, 2024.
- CHORIN, A. J. “Numerical Solution of the Navier-Stokes Equations”, *Mathematics of Computation*, v. 22, n. 104, pp. 745–762, 1968. ISSN: 00255718, 10886842. Disponível em: <<http://www.jstor.org/stable/2004575>>.
- RAO, D. M., PRASAD, V. V. R., TEJA, P. S. R. “A Survey on Prevention of Overfitting in Convolution Neural Networks Using Machine Learning Techniques”, *Miscellaneous*, v. 7, n. 2.32, pp. 177–177, 2018. doi: 10.14419/ijet.v7i2.32.15399.
- JOHAN, HUGHES, T. J., MATHUR, K. K., et al. “A data parallel finite element method for computational fluid dynamics on the Connection Machine system”, *Computer Methods in Applied Mechanics and Engineering*, v. 99, n. 1, pp. 113–134, 1992.
- CODINA, R. *A finite element model for incompressible flow problems*. Universitat Politecnica de Catalunya, 1992.
- PERAIRE, J., MORGAN, K., PEIRO, J. “Application of the TRANAIR rectangular grid approach to the aerodynamic analysis of complex configurations: UNSTRUCTURED FINITE ELEMENT MESH GENERATION AND ADAPTIVE PROCEDURES FOR CFD”, v. 21, pp. 18, 1989.

- SEIBERT, W., FRITZ, W., LEICHER, S. “Application of the TRANAIR rectangular grid approach to the aerodynamic analysis of complex configurations: ON THE WAY TO AN INTEGRATED MESH GENERATION SYSTEM FOR INDUSTRIAL APPLICATIONS”, v. 21, pp. 14, 1989.
- GOUTINES, M., HAH, C., KARADIMAS, G. “Application of the TRANAIR rectangular grid approach to the aerodynamic analysis of complex configurations: MESH GENERATION FOR FLOW COMPUTATION IN TURBOMACHINE”, v. 21, pp. 6, 1989.
- BISWAS, G., BREUER, M., DURST, F. “Backward-Facing Step Flows for Various Expansion Ratios at Low and Moderate Reynolds Numbers”, *Journal of Fluids Engineering-Transactions of the Asme*, v. 126, pp. 362–374, 05 2004. doi: 10.1115/1.1760532.
- THABIT, H., SENAN, T., YASER, A. J. “CFD Analysis of a Backward Facing Step Flows”, *Miscellaneous*, v. 2, n. 3, pp. 10–16, 2018. doi: 10.30939/ijastech..447973.
- TIAN, W., MAO, Z., ZHAO, F., et al. “Layout Optimization of Two Autonomous Underwater Vehicles for Drag Reduction with a Combined CFD and Neural Network Method”, *Complexity*, , n. 1, pp. 5769794, 2017. doi: <https://doi.org/10.1155/2017/5769794>. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1155/2017/5769794>>.
- HU, Q., ZHAI, X., LI, Z. “Multi-Objective Optimization of Deep-Sea Mining Pump Based on CFD, GABP Neural Network and NSGA-III Algorithm”, *Journal of Marine Science and Engineering*, v. 10, n. 8, 2022. ISSN: 2077-1312. doi: 10.3390/jmse10081063. Disponível em: <<https://www.mdpi.com/2077-1312/10/8/1063>>.
- ZHANG, C., JANEWAY, M. “Optimization of Turbine Blade Aerodynamic Designs Using CFD and Neural Network Models”, *International Journal of Turbomachinery, Propulsion and Power*, v. 7, n. 3, 2022. ISSN: 2504-186X. doi: 10.3390/ijtp7030020. Disponível em: <<https://www.mdpi.com/2504-186X/7/3/20>>.
- SAFIKHANI, H., KHAKHALI, A., FARAJPOOR, M. “Pareto Based Multi-Objective Optimization of Centrifugal Pumps Using CFD, Neural Networks and

- Genetic Algorithms”, *Engineering Applications of Computational Fluid Mechanics*, v. 5, n. 1, pp. 37–48, 2011. doi: 10.1080/19942060.2011.11015351. Disponível em: <<https://doi.org/10.1080/19942060.2011.11015351>>.
- GAUTIER, N., AIDER, J.-L. “Upstream open loop control of the recirculation area downstream of a backward-facing step”, *Comptes Rendus Mecanique*, v. 342, n. 6-7, pp. 382–388, 2014. doi: 10.1016/j.crme.2014.05.004.
- JOVIC, S., DRIVER, D. M. *Backward-facing step measurements at low Reynolds number, Re (sub h)= 5000*. Relatório técnico, ”NASA”, 1994.
- HENNING WESSELS, CHRISTIAN WEIBENFELS, P. W. “The neural particle method – An updated Lagrangian physics informed neural network for computational fluid dynamics”, *Elsevier ScienceDirect Journals*, v. 368, 2020. doi: 10.1016/j.cma.2020.113127.
- PIRES, P. B., SANTOS, J. D., PEREIRA, I. V. “Artificial Neural Networks: History and State of the Art”, *Computer Methods in Applied Mechanics and Engineering*, 2025. doi: <https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-6684-7366-5.ch037>.
- LI, W. K., LI, S., PARK, H. S. “Eighty Years of the Finite Element Method: Birth, Evolution, and Future”, *Archives of Computational Methods in Engineering*, 2022. doi: <https://doi.org/10.1007/s11831-022-09740-9>.
- AMEUR, H., MENNI, Y. “Non-Newtonian fluid flows through backward-facing steps”, *SN Applied Sciences*, 2019. doi: <https://doi.org/10.1007/s42452-019-1792-6>.
- RANOCHA, H., DALCIN, L., PARSANI, M., et al. “Optimized Runge-Kutta Methods with Automatic Step Size Control for Compressible Computational Fluid Dynamics”, *Miscellaneous*, v. 4, n. 4, pp. 1191–1228, 2021. doi: 10.1007/s42967-021-00159-w.
- ANDERSON, D., TANNEHILL, J. C., PLETCHER, R. H., et al. *Computational fluid mechanics and heat transfer*. CRC press, 2020.

- NADDA, M., SHAH, S. K., ROY, S., et al. “CFD-based deep neural networks (DNN) model for predicting the hydrodynamics of fluidized beds”, *Miscellaneous*, v. 8, pp. 100113–100113, 2023. doi: 10.1016/j.dche.2023.100113.
- ZHOU, Q., OOKA, R. “Implementation of a coupled simulation framework with neural network and Modelica for fast building energy simulation considering non-uniform indoor environment”, *Elsevier ScienceDirect Journals*, v. 211, pp. 108740–108740, 2021. doi: 10.1016/j.buildenv.2021.108740.
- LANGTANGEN, H., LOGG, A. *Solving PDEs in Python: The FEniCS Tutorial I*. Simula SpringerBriefs on Computing. Springer International Publishing, 2017. ISBN: 9783319524610. Disponível em: <<https://books.google.com.br/books?id=tP71MAAACAAJ>>.
- GOCKENBACH, M. *Understanding and Implementing the Finite Element Method*. 01 2006. ISBN: 978-0-89871-614-6. doi: 10.1137/1.9780898717846.
- ABDOLLAHPOUR, M., GUALTIERI, P., VETSCH, D. F., et al. “Numerical Study of Flow Downstream a Step with a Cylinder Part 1: Validation of the Numerical Simulations”, *Fluids*, v. 8, n. 2, 2023. ISSN: 2311-5521. doi: 10.3390/fluids8020055. Disponível em: <<https://www.mdpi.com/2311-5521/8/2/55>>.
- JEHAD, D., HASHIM, G., KADHIM ZARZOR, A., et al. “Numerical Study of Turbulent Flow over Backward-Facing Step with Different Turbulence Models”, *Journal of Advanced Research Design*, v. 4, pp. 2289–7984, 01 2015.
- PIOCH, F., HARMENING, J. H., MÜLLER, A. M., et al. “Turbulence Modeling for Physics-Informed Neural Networks: Comparison of Different RANS Models for the Backward-Facing Step Flow”, *Fluids*, v. 8, n. 2, 2023. ISSN: 2311-5521. doi: 10.3390/fluids8020043. Disponível em: <<https://www.mdpi.com/2311-5521/8/2/43>>.
- MORTENSEN, M., LANGTANGEN, H. P., WELLS, G. N. “A FEniCS-based programming framework for modeling turbulent flow by the Reynolds-averaged Navier–Stokes equations”, *Advances in Water Resources*, v. 34, n. 9, pp. 1082–1101, 2011. ISSN: 0309-1708. doi: <https://doi.org/10.1016/j.advwatres.2011>.

02.013. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S030917081100039X>>. New Computational Methods and Software Tools.

RAJABI, E., AND, M. R. K. “Intelligent Prediction of Turbulent Flow over Backward-Facing Step using Direct Numerical Simulation Data”, *Engineering Applications of Computational Fluid Mechanics*, v. 6, n. 4, pp. 490–503, 2012. doi: 10.1080/19942060.2012.11015437. Disponível em: <<https://doi.org/10.1080/19942060.2012.11015437>>.

GNEDIN, N. Y., SEMENOV, V. A., KRAVTSOV, A. V. “Enforcing the Courant–Friedrichs–Lewy condition in explicitly conservative local time stepping schemes”, *Journal of Computational Physics*, v. 359, pp. 93–105, 2018. ISSN: 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2018.01.008>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0021999118300184>>.

MONTAZER, E., YARMAND, H., SALAMI, E., et al. “A brief review study of flow phenomena over a backward-facing step and its optimization”, *Renewable and Sustainable Energy Reviews*, v. 82, pp. 994–1005, 2018. ISSN: 1364-0321. doi: <https://doi.org/10.1016/j.rser.2017.09.104>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032117313667>>.

PICHI, F., BALLARIN, F., ROZZA, G., et al. “An artificial neural network approach to bifurcating phenomena in computational fluid dynamics”, *Computers Fluids*, v. 254, pp. 105813, 2023. ISSN: 0045-7930. doi: <https://doi.org/10.1016/j.compfluid.2023.105813>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0045793023000385>>.

PEHERSTORFER, B., WILLCOX, K. “Dynamic data-driven reduced-order models”, *Computer Methods in Applied Mechanics and Engineering*, v. 291, pp. 21–41, 2015. ISSN: 0045-7825. doi: <https://doi.org/10.1016/j.cma.2015.03.018>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0045782515001280>>.

Anexo A - Código *Python* para Aplicação do Método de Elementos Finitos

```
from fenics import *
from mshr import *
import numpy as np
import matplotlib as mpl
import csv

def modelGen(maxVelocity):
    #=====Inputs=====
    #Fluid Properties
    mu = 0.001 # (Viscosidade Cinematica) #Water Default = 0.001
    rho = 1 # density #Water Default = 1
    #maxVelocity = 1.5 #Default = 1.5

    #Time domain
    T = 30.0/(maxVelocity) # final time
    dt = 0.001 * 1/(maxVelocity) # Courant-Friedrichs-Lewy condition
    num_steps = int(T/dt) + 1 # number of time steps
    #num_steps = 20000

    #Pressure properties
    P_outflow = 0
```

```

#Defining the channel dimensions
height = 1
#=====

#Defining Inflow Reynolds Number
mean_vel = 2/3 * maxVelocity
Re = (mean_vel*(2*height)) / mu

#file paths
externalDrive = "/Folder/to/save/results/"
file_path = externalDrive + "Re_%s/" %(str(Re))

# Create mesh
mesh = Mesh("StepLong/StepLong.xml")
fd = MeshFunction('size_t', mesh, "StepLong/StepLong_facet_region.xml")

# Define function spaces
V = VectorFunctionSpace(mesh, 'P', 2)
Q = FunctionSpace(mesh, 'P', 1)

# Define boundaries
# Define inflow profile
inflow_profile = (
    '4.0*%s*(x[1] - %s)*(2 * %s - x[1]) / pow(%s, 2)' %(
        str(maxVelocity), str(height), str(height), str(height)),
    '0')

# Define boundary conditions
bcu_inflow = DirichletBC(
    V,

```

```

        Expression(inflow_profile, degree=2),
        fd,
    7)

bcu_walls = DirichletBC(
    V,
    Constant((0, 0)),
    fd,
    9)

bcp_outflow = DirichletBC(
    Q,
    Constant(P_outflow),
    fd,
    8)

bcu = [bcu_inflow, bcu_walls]
bcp = [bcp_outflow]

# Define trial and test functions
u = TrialFunction(V)
v = TestFunction(V)
p = TrialFunction(Q)
q = TestFunction(Q)

# Define functions for solutions at previous and current time steps
u_n = Function(V)
u_ = Function(V)
p_n = Function(Q)
p_ = Function(Q)

```

```

# Define expressions used in variational forms
U = 0.5*(u_n + u)
n = FacetNormal(mesh)
f = Constant((0, 0))
k = Constant(dt)
mu = Constant(mu)

# Define symmetric gradient
def epsilon(u):
    return sym(nabla_grad(u))

# Define stress tensor
def sigma(u, p):
    return 2*mu*epsilon(u) - p*Identity(len(u))

# Define variational problem for step 1
F1 = rho*dot((u - u_n) / k, v)*dx \
    + rho*dot(dot(u_n, nabla_grad(u_n)), v)*dx \
    + inner(sigma(U, p_n), epsilon(v))*dx \
    + dot(p_n*n, v)*ds - dot(mu*nabla_grad(U)*n, v)*ds \
    - dot(f, v)*dx

a1 = lhs(F1)
L1 = rhs(F1)

# Define variational problem for step 2
a2 = dot(nabla_grad(p), nabla_grad(q))*dx
L2 = dot(nabla_grad(p_n), nabla_grad(q))*dx - (1/k)*div(u_)*q*dx

# Define variational problem for step 3
a3 = dot(u, v)*dx
L3 = dot(u_, v)*dx - k*dot(nabla_grad(p_ - p_n), v)*dx

```

```

# Assemble matrices
A1 = assemble(a1)
A2 = assemble(a2)
A3 = assemble(a3)

# Apply boundary conditions to matrices
[bc.apply(A1) for bc in bcu]
[bc.apply(A2) for bc in bcp]

# Create XDMF files for visualization output
xdmffile_u = XDMFFile(file_path + 'velocity.xdmf')
xdmffile_p = XDMFFile(file_path + 'pressure.xdmf')

# Create time series (for use in reaction_system.py)
timeseries_u = TimeSeries(file_path + 'velocity_series')
timeseries_p = TimeSeries(file_path + 'pressure_series')

# Save mesh to file (for use in reaction_system.py)
File(file_path + 'step.xml.gz') << mesh

# Create progress bar
progress = Progress('Time-stepping')
set_log_level(LogLevel.PROGRESS)

# Time-stepping
t = 0

for n in range(num_steps):

    # Step 1: Tentative velocity step

```

```

b1 = assemble(L1)
[bc.apply(b1) for bc in bcu]
solve(A1, u_.vector(), b1, 'bicgstab', 'hypre_amg')

# Step 2: Pressure correction step
b2 = assemble(L2)
[bc.apply(b2) for bc in bcp]
solve(A2, p_.vector(), b2, 'bicgstab', 'hypre_amg')

# Step 3: Velocity correction step
b3 = assemble(L3)
solve(A3, u_.vector(), b3, 'cg', 'sor')

# Plot solution
plot(u_, title='Velocity')
plot(p_, title='Pressure')

# Save solution to file (XDMF/HDF5)
xdmffile_u.write(u_, t)
xdmffile_p.write(p_, t)

# Save nodal values to file
timeseries_u.store(u_.vector(), t)
timeseries_p.store(p_.vector(), t)

# Update previous solution
u_n.assign(u_)
p_n.assign(p_)

# Update progress bar
#progress.update(t / T)
#print('u max:', u_.vector().array().max())

```

```

print('u max:', np.max(np.array(u_.vector())))
print('Sim Time:', t)

# Update current time
t += dt

#Saving properties as csv
with open(file_path + '/properties.csv', 'w', encoding='UTF8') as f:
    writer = csv.writer(f)

    # write the data
    writer.writerow(["Reynolds", Re])
    writer.writerow(["Pressure Outflow", P_outflow])
    writer.writerow(["Max Speed", maxVelocity])
    writer.writerow(["Density", rho])
    writer.writerow(["Dynamic Viscosity", mu])

```

Anexo B - Código para Iterar o Problema de Navier-Stokes com Múltiplos Números de Reynolds

```
import NavierStokes_step_import

VelocityCases = [1.0,1.5,2,2.5,3.0,3.5] #m/s

for v in VelocityCases:
    NavierStokes_step_import.modelGen(v)
```

Anexo C - Código para Análise de Regressão

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression, Ridge
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error as MAE
from sklearn.metrics import r2_score as R2
import pickle as pkl
import matplotlib.pyplot as plt

ML_Data = pd.read_csv('ML_Data.csv')

Re = ML_Data.loc[:, ['Re']]
L1 = ML_Data.loc[:, ['L1']]
L2 = ML_Data.loc[:, ['L2']]
L3 = ML_Data.loc[:, ['L3']]

STUDIED_VAR = L1
VAR_NAME = 'L1'
PLOT_DATA = True
PLOT_MODEL = True
PLOT_COEF = True
```

```

#Usar Regressão original
LOAD_OLD_NEURAL = False

#Ordem de hipótese de Polinômio
degree = 3

p = preprocessing.PolynomialFeatures(degree).fit(Re)
print(p.get_feature_names_out(Re.columns))

Re = preprocessing.PolynomialFeatures(degree).fit_transform(Re)

x_train, x_test, y_train, y_test = train_test_split(Re,
                                                    STUDIED_VAR,
                                                    test_size=0.2)

model_1 = Ridge

#Loading old Neural Network
if LOAD_OLD_NEURAL:
    with open("NeuralModel.pkl", 'rb') as f:
        model_1 = pickle.load(f)
else:
    #Creating new Neural Network
    #model_1 = LinearRegression().fit(x_train,y_train)
    model_1 = Ridge(alpha=0.0, max_iter=2000).fit(x_train,y_train)

    #Saving model:
    with open("NeuralModel.pkl", "wb") as f:
        pickle.dump(model_1, f, protocol=5)

y_model_train = model_1.predict(x_train)
y_model_test = model_1.predict(x_test)

```

```

print(MAE(y_train, y_model_train)) #calculate mean absolute error on training set
print(MAE(y_test, y_model_test)) #calculate mean absolute error on testing set
print(R2(y_train, y_model_train)) #calculate R^2 on training set
print(R2(y_test, y_model_test)) #calculate R^2 on testing set
print(model_1.coef_[0])

```

#Plot Data

```

if PLOT_DATA:
    ML_Data.plot.scatter(x='Re', y=VAR_NAME)
    plt.show()

```

#Plot Prediction Model

```

if PLOT_MODEL:
    plt.plot(y_train, y_model_train, 'ro')
    plt.plot(y_test, y_model_test, 'bo')

```

#Plotando uma linha preta de guia

```

plt.plot([1400,2000], [1400,2000], 'k', linewidth=2)

```

```

plt.xlabel('Simulacao Computacional', fontsize=20)

```

```

plt.ylabel('Previsao Neural', fontsize=20)

```

```

plt.xticks(fontsize=16)

```

```

plt.yticks(fontsize=16)

```

```

plt.legend(['train', 'test'], fontsize=16)

```

```

fig = plt.gcf()

```

```

fig.set_size_inches (8,8)

```

```

plt.show()

```

#Plot Coefficients

```
if PLOT_COEF:
    #plt.yscale('log')

    if degree == 3:
        plt.bar(["1", "Re", "Re^2", "Re^3"], model_1.coef_[0])
    elif degree == 2:
        plt.bar(["1", "Re", "Re^2"], model_1.coef_[0])
    elif degree == 1:
        plt.bar(["1", "Re"], model_1.coef_[0])

plt.ylabel('Coefficient', fontsize=16)
plt.xlabel('Feature', fontsize=16)

plt.show()
```