

UNIVERSIDADE DE SÃO PAULO ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

SÉRGIO DE PAULA PELLEGRINI

**Aplicação dos Grafos de Ligação a escoamentos unidimensionais
incompressíveis em dutos**

São Paulo

11 de dezembro de 2009

SÉRGIO DE PAULA PELLEGRINI

**Aplicação dos Grafos de Ligação a escoamentos unidimensionais
incompressíveis em dutos**

Área de Concentração: Eng. Mecânica de
Energia e Ambiente

Orientador: Prof. Dr. Jorge Luis Baliño

São Paulo

11 de dezembro de 2009

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

FICHA CATALOGRÁFICA

Pellegrini, Sérgio de Paula

Aplicação de metodologia baseada nos grafos de ligação a problemas incompressíveis de dinâmica dos fluídos computacional / S.P. Pellegrini. – São Paulo, 2009.

117 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.

1. Simulação (Modelagem) 2. Mecânica dos fluídos computacional 3. Escoamento I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II. t.

AGRADECIMENTOS

Agradeço aos professores que contribuíram direta ou indiretamente para a realização deste trabalho, em especial aos professores Alexandre Kawano, Euryale de Jesus Zerbini, Fábio Saltara e Marcos de Mattos Pimenta, listados em ordem alfabética.

À FAPESP, por bolsa de auxílio a iniciação científica.

Agradecimento mais do que especial ao professor Jorge Luis Baliño, por suas enormes paciência e dedicação.

Finalmente, agradeço também à própria USP, por ter proporcionado minha formação até o momento presente.

RESUMO

O trabalho de formatura que segue se propõe a desenvolver o equacionamento para escoamentos incompressíveis unidimensionais em um duto de paredes rígidas utilizando uma metodologia nova, que envolve Dinâmica dos Fluidos Computacional e Grafos de Ligação (*Bond Graphs*).

Primeiramente, são introduzidas as teorias de grafos de ligação, dinâmica dos fluidos computacional e, finalmente, a teoria de BG-CFD – que integra as duas anteriores.

A seguir, o equacionamento do tipo de escoamento em estudo é obtido a partir dos grafos de ligação. Diferentes condições de contorno são exploradas. Também, apresenta-se a forma do código implementado computacionalmente, além do código em si.

Finalmente, são discutidos exemplos de aplicação. Para cada aplicação é realizado um estudo de convergência de malha antes da apresentação dos resultados.

Palavras-chave: *Grafos de Ligação, Dinâmica dos Fluidos Computacional, Escoamento incompressível*

ABSTRACT

The present monograph's intention is to develop and implement a model for one-dimensional incompressible duct flow using a new methodology, which unites techniques of Computational Fluid Dynamics and Bond Graphs.

Initially, the theories of Bond Graphs, Computational Fluid Dynamics and BG-CFD – combining the two previous ones – are presented.

Next, the equations that will be used to model the phenomenon are obtained, from the bond graph. Different boundary conditions are explored. With those equations, a computational code was developed and is also presented.

Finally, two study cases are developed, with a mesh convergence study. The numerical results obtained are presented.

Keywords: *Bond Graphs, Computational Fluid Dynamics, Incompressible Flow*

	Junção de esforço comum	6
	Junção de fluxo comum	7
2.1.3	Obtenção do grafo	7
2.1.4	Obtenção das equações de estado	7
2.1.4.1	Causalidade sem os grafos de ligação	8
2.1.4.2	Causalidade nos grafos de ligação	8
2.1.4.3	Desenvolvimento ao espaço de estados	10
2.1.5	Generalização dos elementos básicos	10
2.1.5.1	<i>Multibonds</i>	11
2.1.5.2	Transformadores e giradores multiportos	11
2.1.5.3	Campos armazenadores de energia	11
	Campo C	13
	Campo I	13
	Campo IC	13
2.1.5.4	Campo resistivo	13
2.1.5.5	Estruturas de junção	14
2.1.6	Sistemas termodinâmicos fechados	14
2.1.6.1	Transformadas de Legendre	14
2.1.6.2	O modelo em grafos de ligação	15
2.2	Dinâmica dos fluidos computacional	15
2.2.1	Classificação dos problemas	16
2.2.2	Noção local <i>versus</i> noção global	16
2.2.3	Os métodos de resíduos	17
2.2.3.1	Erro absoluto	17
2.2.3.2	Colocação	17
2.2.3.3	Subdomínio	18

2.2.3.4	Ortogonalidade	18
2.2.4	Funções de teste	18
2.2.4.1	Exemplos de funções de teste	19
2.2.4.2	Escolha das funções de teste	21
2.2.5	Condições de contorno	22
2.3	Metodologia BG-CFD para escoamentos incompressíveis	23
2.3.1	O escoamento incompressível	24
2.3.2	Equações de balanço	24
2.3.3	Discretização	26
2.3.4	Equacionamento matemático	27
2.3.5	Modelo em grafos de ligação	28
2.4	Metodologia BG-CFD para escoamentos incompressíveis unidimensionais em dutos	29
2.4.1	O escoamento incompressível unidimensional em um duto	29
2.4.2	Equações de balanço	31
2.4.3	Discretização	31
2.4.4	Equacionamento matemático	32
2.4.5	Modelo em grafos de ligação	33
3	Desenvolvimento do equacionamento	35
3.1	Escolha das funções de teste	35
3.1.1	As funções de teste escolhidas	36
3.1.1.1	O esquema de <i>upwind</i>	37
3.2	Desenvolvimento dos termos no interior do domínio	38
3.2.1	Porto inercial	38
3.2.1.1	Pressão gravitacional	38
3.2.1.2	Pressão convectiva	39

3.2.1.3	Atrito viscoso	40
3.2.1.4	Inércia hidráulica	40
3.2.2	Porto capacitivo	40
3.2.2.1	Calor trocado com as paredes	40
	Com calor trocado definido	41
	Com temperatura da parede definida	41
3.2.2.2	Calor trocado axialmente	42
3.2.2.3	Calor trocado por efeitos convectivos	43
3.2.2.4	Calor resultante de geração interna de energia	44
3.2.2.5	Calor trocado pela não-homogeneidade nos elementos	45
3.2.2.6	Dissipação viscosa	45
3.2.2.7	Matriz de temperatura	46
3.3	Desenvolvimento dos termos nos contornos	47
3.3.1	Para a junção 1	47
3.3.1.1	Condição integral	47
3.3.1.2	Condição derivativa	48
3.3.2	Para a junção 0	48
3.3.2.1	Condição integral completa	48
	Condução	49
	Convecção	49
3.3.2.2	Condições mistas	50
3.4	Evolução temporal	50
3.4.1	O método preditor-corretor	50
3.4.2	O equacionamento para o caso em estudo	51
3.4.2.1	Inicialização	52
3.4.2.2	Iteração	52

3.4.2.3	Critério de parada	53
3.4.2.4	Finalização	53
4	Implementação numérica	54
4.1	Funções válidas para qualquer aplicação	55
4.1.1	Funções interpoladoras	55
4.1.2	Inversão de matriz	55
4.1.3	Determinação das propriedades do fluido	55
4.1.4	Fator de atrito	56
4.1.5	Cálculo do número de Nusselt	57
4.1.6	Determinação dos parâmetros de <i>upwind</i>	59
4.1.7	Função para teste de convergência de malha	59
4.2	Funções a serem alteradas conforme aplicação	60
4.2.1	Função principal	60
4.2.2	Diâmetros do duto	62
4.2.3	Condições de fronteira	62
4.2.4	Predição	62
4.3	Dados de entrada	63
5	Estudo de caso: escoamento de água em regime de aquecimento	65
5.1	O problema a ser estudado	65
5.1.1	O modelo físico	65
5.1.1.1	Hipóteses	65
5.1.2	Tabela de propriedades do fluido	67
5.1.3	Sobre o método numérico	67
5.2	Comparação com solução analítica	68
5.3	Pressão variável na entrada	69

6	Estudo de caso: escoamento de óleo em regime de aquecimento.....	73
6.1	O problema a ser estudado	73
6.1.1	O modelo físico	73
6.1.1.1	Hipóteses	73
6.1.2	Tabela de propriedades do fluido	75
6.1.3	Sobre o método numérico	75
6.2	Verificação da conservação de energia	76
6.3	Com temperatura de entrada constante	77
6.4	Com temperatura de entrada variável	78
7	Conclusões.....	81
	Referências	82
	Apêndice A – Código no MatLab para funções gerais	84
A.1	Interpolações lineares	84
A.1.1	“ <i>lin_interp</i> ”	84
A.1.2	“ <i>v_lin_interp</i> ”	84
A.1.3	“ <i>m_lin_interp</i> ”	85
A.2	Inversão de matriz	85
A.3	Determinação das propriedades do fluido	86
A.4	Fator de atrito	86
A.5	Cálculo do número de Nusselt	87
A.6	Determinação dos parâmetros de <i>upwind</i>	89
A.7	Teste de convergência de malha	90
	Apêndice B – Código no Matlab para funções específicas.....	93
B.1	Primeiro caso estudado – Água em aquecimento.....	93

B.1.1	Tabelas com dados auxiliares	93
B.1.2	Diâmetros do duto	95
B.1.3	Condições de fronteira	95
B.1.4	Arquivo principal	95
B.1.5	Predição	102
B.2	Segundo caso estudado – Óleo em aquecimento	105
B.2.1	Tabelas com dados auxiliares	105
B.2.2	Diâmetros do duto	107
B.2.3	Condições de fronteira	107
B.2.4	Arquivo principal	107
B.2.5	Predição	114
B.2.6	Temperatura de entrada	117

LISTA DE FIGURAS

Figura - 2.1	Notação para um <i>multibond</i>	11
Figura - 2.2	Exemplo de campo capacitivo denotado nas formas (a) implícita e (b) explícita	12
Figura - 2.3	Representação de um sistema termodinâmico fechado com diferentes causalidades, descrito através (a) da energia livre (b) da entalpia (c) da energia livre de Helmholtz e (d) energia livre de Gibbs	15
Figura - 2.4	Funções de teste unidimensionais lineares, plotadas para um único elemento	19
Figura - 2.5	Funções de teste unidimensionais quadráticas, plotadas para um único elemento	20
Figura - 2.6	Funções de teste unidimensionais cúbicas de Hermite, plotadas para um único elemento	20
Figura - 2.7	Funções de teste bidimensionais bi-lineares, plotadas para um único elemento (adaptado de Hunter e Pullan (2003))	21
Figura - 2.8	Funções de teste bidimensionais quadrático-lineares, escritas para um único elemento de seis nós (adaptado de Hunter e Pullan (2003))	22
Figura - 2.9	Elemento triangular	22
Figura - 2.10	Função de teste unidimensional com esquema de <i>upwind</i>	23
Figura - 2.11	Exemplo de discretização ruim	23
Figura - 2.12	Grafo de ligação para escoamentos incompressíveis	29
Figura - 2.13	Grafo de ligação para escoamentos incompressíveis unidimensionais em dutos	34
Figura - 3.1	Funções de forma (em tracejado) e de peso (em linha contínua) escolhidas para a entropia	36
Figura - 3.2	Comparação entre função exata e função estimadora para <i>upwind</i> , para todos os valores de β	38
Figura - 3.3	Comparação entre função exata e função estimadora para <i>upwind</i> , para $-20 <$	

$Pe < 20$	39
Figura - 3.4 Porto de inércia do modelo para escoamento incompressível em duto com causalidade integral atribuída	47
Figura - 3.5 Porto de inércia do modelo para escoamento incompressível em duto com as duas possibilidades de causalidades derivativas	48
Figura - 3.6 Porto de inércia do modelo para escoamento incompressível em duto com causalidade integral atribuída	49
Figura - 5.1 Modelo físico do primeiro problema a ser estudado	65
Figura - 5.2 Comparação entre soluções analítica e numérica	69
Figura - 5.3 Gráficos para o caso em estudo, para os tempos: (a) $t = 0,00$ s (b) $t = 0,56$ s (c) $t = 1,45$ s (d) $t = 2,30$ s (e) $t = 2,87$ s (f) $t = 3,45$ s (g) $t = 3,91$ s (h) $t = 4,35$ s	71
Figura - 5.4 Gráficos para o caso em estudo, para os tempos: (i) $t = 11,04$ s (j) $t = 18,00$ s	72
Figura - 6.1 Máximo resíduo no processo de refinamento de malha <i>versus</i> número de nós	76
Figura - 6.2 Balanço de energia e custo computacional com o refinamento de malha ...	77
Figura - 6.3 Perfil de temperaturas no regime permanente para escoamento de óleo em aquecimento	78
Figura - 6.4 Vazão volumétrica <i>versus</i> tempo, para escoamento de óleo com temperatura de entrada constante	79
Figura - 6.5 Evolução da temperatura de entrada ao longo do tempo, para a aplicação em especial	80
Figura - 6.6 Evolução da vazão volumétrica ao longo do tempo, para escoamento de óleo com temperatura de entrada variante ao longo do tempo	80

LISTA DE TABELAS

Tabela 2.1	Elementos de um porto, suas representações e equações características		4
Tabela 2.2	Elementos de dois portos, suas representações e equações características	...	5
Tabela 2.3	Elementos de junção, suas representações e equações características		7
Tabela 2.4	Transformadores e giradores multiportos, suas representações e equações características		11
Tabela 2.5	Campos armazenadores de energia, com suas representação explícita, energia armazenada e relações de Maxwell		12

1 INTRODUÇÃO

1.1 Os grafos de ligação e a dinâmica dos fluidos

Dentro da mecânica dos fluidos, o estudo de escoamentos incompressíveis viscosos com trocas térmicas é um tema relevante, visto que há diversos fenômenos físicos que podem ser modelados através dessas hipóteses.

Alguns exemplos são escoamentos em que há interesse apenas na conservação de energia mecânica — como oleodutos e problemas da hidráulica, de forma geral — ou escoamentos em que as trocas térmicas também são objeto de análise — como a circulação sanguínea ou os trocadores de calor, sendo esses últimos objetos fundamentais da engenharia térmica.

Mais do que apenas relevante, esse campo de estudo tem se mostrado em contínuo desenvolvimento com relação à construção do conhecimento científico. A não obtenção de soluções analíticas para as equações de conservações de momento linear e de energia para condições não-restritivas, aliada ao atual desenvolvimento vertiginoso da capacidade de processamento dos computadores digitais, tem motivado o desenvolvimento da Dinâmica dos Fluidos Computacional.

No entanto, a Dinâmica dos Fluidos Computacional ainda enfrenta algumas dificuldades, relacionadas essencialmente com convergência e com nodalização.

Não obstante, o presente trabalho também abre campo para outras aplicações em engenharia, como as máquinas de fluxo, problema no qual ainda teria que ser levado em consideração as interações fluido-estrutura.

Se, por um lado, essa última aplicação foge do escopo deste texto, a metodologia empregada pode facilitar o desenvolvimento de uma teoria que abranja também esse problema. Isso ocorre pois os grafos de ligação – *bond graphs*, em inglês – permitem uma representação simples, mas sem perda de generalidade, de sistemas dinâmicos que englobam diferentes domínios de energia.

Além disso, a teoria dos grafos de ligação permite, através de um procedimento ordenado, a atribuição de condições de contorno de uma forma unívoca, proporcionando a obtenção de um conjunto de equações que sempre representará um problema matematicamente bem posto, pois sempre lidará coerentemente com as condições de contorno.

Assim, uma teoria que incorpore os grafos de ligação à dinâmica dos fluidos computacional mostra-se de grande interesse. O primeiro trabalho na área, Baliño (2005), apresenta a teoria que será usada como principal ferramenta neste trabalho.

É interessante ressaltar que a teoria dos grafos de ligação é tipicamente usada para sistemas de parâmetros concentrados. A aplicação a sistemas com parâmetros distribuídos ainda é vista com ressalvas pelos usuários dos grafos.

Nesse sentido, este trabalho se propõe a desenvolver simulações de escoamentos incompressíveis em dutos e apresentar seus resultados com uma metodologia baseada nos grafos de ligação.

1.2 Estrutura do texto

Com esses objetivos, o texto que segue é estruturado da seguinte forma:

O primeiro capítulo realiza uma breve introdução teórica aos temas de grafos de ligação, dinâmica dos fluidos computacional e a uma terceira teoria, que integra a modelagem do sistema com o uso de grafos de ligação e a discretização espacial com o uso da dinâmica dos fluidos computacional.

Esta última teoria é apresentada com dois modelos, um para escoamento incompressível tridimensional e outro para escoamento incompressível unidimensional em um duto.

O capítulo seguinte apresenta o equacionamento, partindo do modelo em grafos de ligação e desenvolvendo até um ponto em que as equações estão prontas para serem implementadas em um computador.

Posteriormente, outro capítulo expõe a forma com que foi realizada a implementação numérica, descrevendo cada função implementada.

Finalmente, os dois últimos capítulos apresentam simulações realizadas com o código gerado. Cada um desses dois capítulos apresenta um ponto de verificação do código e resultados.

2 FERRAMENTAL TEÓRICO

A principal ferramenta teórica para o trabalho é uma teoria que incorpora os grafos de ligação à dinâmica dos fluidos computacional, chamada BG-CFD.

Mas antes de abordar esse ponto teórico central, é necessário primeiro discorrer sobre seus alicerces, os grafos de ligação e a dinâmica dos fluidos computacional.

2.1 Grafos de ligação

As considerações teóricas sobre o formalismo dos grafos de ligação estão fundamentadas principalmente no texto de Karnopp, Margolis e Rosenberg (2000). Algumas considerações também estão baseadas em Borutzky (2004) e em Cellier (2008).

2.1.1 Definições iniciais

De forma a modelar sistemas de diferentes domínios energéticos da mesma forma, são estabelecidas quatro variáveis generalizadas — esforço e , fluxo f , momento p e deslocamento q .

Essas variáveis são definidas de forma que o produto da variável esforço pela variável fluxo sempre gere uma potência, isto é, $ef = P$. Ainda, a integral no tempo da variável esforço é a variável momento $p = \int e dt$ e a integral no tempo da variável fluxo é a variável deslocamento $q = \int f dt$.

Para modelar sistemas dinâmicos, são usados elementos com diferentes equações características e as ligações (ou *bonds*). A cada ligação estão associados um esforço e um fluxo — e, portanto, uma potência. A ligação é representada com uma semi-flecha.

O sentido da ligação representa o sentido da transmissão de potência no sistema. Cada lado da semi-flecha é conectado a um elemento através de um porto.

2.1.2 Elementos básicos

2.1.2.1 Elementos de um porto

A tab. 2.1 resume o que será apresentado nessa seção, mostrando a notação dos elementos e suas possíveis equações características.

Tabela 2.1: Elementos de um porto, suas representações e equações características

Resistência	$\frac{e}{f} \searrow R$	$e = \phi_R(f)$ $f = \phi_R^{-1}(e)$	ou
Capacitância	$\frac{e}{\dot{q}} \searrow C$	$q = \phi_C(e)$ $e = \phi_C^{-1}(q)$	ou
Inércia	$\frac{\dot{p}}{f} \searrow I$	$p = \phi_I(f)$ $f = \phi_I^{-1}(p)$	ou
Fonte ou sumidouro de esforço	$\frac{e}{f} \searrow S_e$ ou $\swarrow \frac{e}{f} S_e$	$e = cte$	
Fonte ou sumidouro de fluxo	$\frac{e}{f} \searrow S_f$ ou $\swarrow \frac{e}{f} S_f$	$f = cte$	

Resistência

A equação característica deste elemento relaciona as variáveis de esforço e fluxo através de uma função estática, propriedade do elemento. A resistência tem símbolo R e dissipa energia.

Apesar de, rigorosamente, não existir dissipação de energia – uma vez que energia não pode ser destruída –, esta afirmação pode ser correta, dependendo dos objetivos da modelagem do sistema. É preciso ter em mente, no entanto, que a energia dissipada pela resistência é, na verdade, transformada em energia térmica de forma irreversível.

Assim, caso não sejam modelados o comportamento termodinâmico do sistema, a representação com o elemento R será suficiente. Caso haja interesse no comportamento termodinâmico, deve-se utilizar o elemento RS , que será introduzido posteriormente.

Capacitância

A capacitância tem uma equação característica que relaciona esforços com deslocamentos (ou vice-versa). Este elemento tem por símbolo C e pode armazenar energia.

Inércia

A equação característica da inércia relaciona momento e fluxo através de uma função estática. Este elemento tem por símbolo I e também pode armazenar energia.

Fonte de esforço

Fornece qualquer quantidade de potência ao sistema, impondo um valor de esforço. Em casos particulares pode tirar potência do sistema, impondo um valor de esforço. Nesses casos o elemento será chamado de sumidouro. Este elemento tem por símbolo S_e .

Vale ressaltar que, assim como para a resistência, a afirmação de “criar” ou “consumir” energia pode ser feita, dependendo do objetivo da modelagem. Cabe comentar também que este é um elemento ideal, já que, em sistemas reais não é possível garantir que o esforço será mantido em um valor pré-estabelecido para grandes potências transmitidas.

Fonte de fluxo

É um elemento análogo à fonte de esforço, mas impondo fluxo. Seu símbolo é S_f .

2.1.2.2 Elementos de dois portos

A tab. 2.2 resume o que será apresentado nessa seção, mostrando a notação dos elementos e suas possíveis equações características.

Tabela 2.2: Elementos de dois portos, suas representações e equações características

Transformador	$\frac{e}{f} \searrow \overset{m}{\text{TF}} \frac{e}{f} \swarrow$	$\begin{cases} e_1 = m e_2 \\ m f_1 = f_2 \end{cases}$	
Girador	$\frac{e}{f} \searrow \overset{r}{\text{GY}} \frac{e}{f} \swarrow$	$\begin{cases} e_1 = r f_2 \\ r f_1 = e_2 \end{cases}$	
Elementos modulados	$\frac{e}{f} \searrow \overset{\downarrow m}{\text{MTF}} \frac{e}{f} \swarrow$	$\frac{e}{f} \searrow \overset{\downarrow}{\text{MRS}} \frac{e}{f} \swarrow$	$\frac{e}{f} \searrow \overset{\downarrow r}{\text{MGY}} \frac{e}{f} \swarrow$

Transformador

O transformador tem por símbolo TF e conserva energia, isto é, a cada instante de tempo, a potência que entra é igual a potência que sai desse elemento. O fluxo da ligação que retira potência do elemento é proporcional ao fluxo da ligação que insere potência no elemento.

O parâmetro que determina essa proporcionalidade é o módulo de transformação, m , propriedade do elemento.

Frequentemente, encontra-se na literatura afirmações de que o elemento RS , citado anteriormente, poderia ser enxergado como um transformador que só permite a transmissão de potência em uma direção, devido ao à irreversibilidade na transdução de potência de algum domínio energético para o domínio térmico.

No entanto, a atribuição de causalidade para o elemento RS não é similar ao elemento TF , de forma que a analogia não é total. De fato, o elemento RS é melhor descrito como um campo resistivo que não dissipa energia. A energia é conservada, mas é degradada, gerando entropia.

Girador

O girador tem por símbolo GY e conserva energia, isto é, a cada instante de tempo, a potência que entra é igual a potência que sai desse elemento. O esforço da ligação que retira potência do elemento é proporcional ao fluxo da ligação que insere potência no elemento.

O parâmetro que determina essa proporcionalidade é o módulo de giração, r , propriedade do elemento.

Transformador modulado, Elemento RS modulado e Girador modulado

Suas características são idênticas às do transformador, elemento RS e girador, respectivamente, com uma única exceção. Nos elementos modulados, os módulos de m e r são variáveis. Seus símbolos são, respectivamente, MTF , MRS e MGY .

2.1.2.3 Elementos de junção

A tab. 2.3 resume o que será apresentado nessa seção, mostrando a notação dos elementos e suas possíveis equações características.

Junção de esforço comum

De símbolo 0 é, justamente por isso, também chamada de junção 0 . Este elemento, assim como os elementos de dois portos, conserva energia. No entanto, essa junção pode ter vários portos. Todas as ligações conectadas a esse elemento compartilham o mesmo valor para o esforço.

Tabela 2.3: Elementos de junção, suas representações e equações características

Junção de esforço comum	
	$\begin{cases} e_1 = e_2 = e_3 = \dots = e_n \\ \sum_{i=1}^n f_i = 0 \end{cases}$
Junção de fluxo comum	
	$\begin{cases} f_1 = f_2 = f_3 = \dots = f_n \\ \sum_{i=1}^n e_i = 0 \end{cases}$

Junção de fluxo comum

De símbolo 1 é, justamente por isso, também chamada de junção 1. Este elemento, assim como os elementos de dois portos, conserva energia. No entanto, essa junção pode ter vários portos. Todas as ligações conectadas a esse elemento compartilham o mesmo valor para o fluxo.

2.1.3 Obtenção do grafo

Composições com esses elementos geram grafos que podem representar coerentemente modelos físicos, respeitando os sentidos de transferência de potência e as propriedades de cada elemento.

No entanto, é importante realçar que, apesar de permitir uma simples interpretação do modelo físico gerado, o grafo não modelará obrigatoriamente o sistema físico real, mas sim o modelo físico mentalizado pelo “modelador”.

2.1.4 Obtenção das equações de estado

Com objetivo final de se obter as equações de estado, inicialmente deve-se compor uma coleção de equações que relacione todos os valores relevantes para o sistema. O conjunto encontrado tem um número de equações maior do que o dobro do número de elementos que compõem o sistema.

Após obter esse conjunto de equações que representam as propriedades de cada elemento, pode ser bastante trabalhoso organizá-las de forma a obter um conjunto linearmente independente que represente o comportamento dinâmico do sistema modelado. Diversos algoritmos se prestam a essa função e citaremos agora alguns deles.

2.1.4.1 Causalidade sem os grafos de ligação

Em primeiro lugar, é necessário verificar quais são as variáveis independentes e as variáveis dependentes. Há algoritmos de ordenação horizontal e orientação vertical, que permitem — supondo inicialmente conhecidas as variáveis de estado — ordenar as equações de forma a obter um caminho de único para substituição de variáveis visando obter uma representação na forma de espaço de estados. Cellier (2008) apresenta o algoritmo de Tarjan, um desses algoritmos.

No entanto, podem ser encontrados dois tipos de problemas: laços algébricos e singularidades estruturais.

Sobre os laços algébricos, o algoritmo de Tarjan, em especial, também garante que os laços algébricos encontrados são mínimos. No entanto, ainda é necessário romper esses laços, com a seleção de uma variável de rompimento para cada laço, o que nem sempre é evidente. Se o sistema for não-linear, será requerido o uso de métodos iterativos.

Cellier (2008) também coloca que as singularidades estruturais podem ser resolvidas através da aplicação (diversas vezes, se necessário) do algoritmo de Pantelides, que adiciona novas variáveis e equações e resulta na criação de laços algébricos.

2.1.4.2 Causalidade nos grafos de ligação

Uma solução alternativa a todos esses algoritmos é o uso dos grafos de ligação, para os quais um procedimento gráfico de regras simples atribui causalidades coerentes e unívocas para todo o conjunto de equações, facilitando a posterior obtenção da representação em espaço de estados.

O procedimento consiste basicamente em incluir uma barra vertical (|) no porto que impõe o valor de fluxo ao elemento conectado. No entanto, há algumas propriedades que devem ser observadas.

Nem todas as causalidades são possíveis para alguns elementos. Os elementos que conservam energia (TF , GY , RS , 0, 1), por exemplo, transmitem causalidade de uma maneira especial, coerente com suas equações características.

Além disso, existem elementos que têm uma causalidade preferencial (impondo esforço ou fluxo). Também, certos elementos têm prioridade sobre outros.

De forma a respeitar esse conjunto de restrições, foi desenvolvido o seguinte procedi-

mento de atribuição de causalidade. O procedimento é direcional e é realizado até que não haja mais ligações sem causalidade atribuída.

1. Atribuir a causalidade preferencial para uma fonte ($\neg S_f$ ou $\vdash S_e$)
 - Propagá-la através dos elementos que conservam energia, até onde for possível
 - Repetir desde o passo 1 até que não haja mais fontes sem causalidade atribuída
2. Atribuir causalidade integral a um elemento armazenador de energia ($\vdash C$ ou $\neg I$)
 - Propagá-la através dos elementos que conservam energia, até onde for possível
 - Repetir desde o passo 2 até que não haja elementos I ou C sem causalidade atribuída
3. Atribuir uma causalidade arbitrária para uma resistência
 - Propagá-la através dos elementos que conservam energia, até onde for possível
 - Repetir desde o passo 3 até que não haja resistências sem causalidade atribuída
4. Atribuir uma causalidade arbitrária para uma ligação qualquer
 - Propagá-la através dos elementos que conservam energia, até onde for possível
 - Repetir desde o passo 4 até que não haja nenhuma ligação sem causalidade atribuída

As causalidades mencionadas no item 2 tem esse nome pois elas fazem com que as equações características da capacitância e da inércia sejam, respectivamente, $e = \phi_C^{-1}(\int f dt) = \phi_C^{-1}(q)$ e $f = \phi_I^{-1}(\int e dt) = \phi_I^{-1}(p)$.

Embora a preferência dos elementos armazenadores de energia possa parecer um mero detalhe, cada vez que ela não é respeitada é dado origem a uma singularidade estrutural. Em outras palavras, as singularidades estruturais ocorrem quando há conflito de causalidade — quando são atribuídas causalidades derivativas aos elementos I ou C .

Já os laços algébricos ocorrem quando há a necessidade de atribuir uma causalidade arbitrária a uma resistência.

Um aparte importante é que o elemento RS tem causalidade fixa no porto S , já que a dissipação de energia é sempre modelada como um gerador de entropia, e nunca de temperatura.

2.1.4.3 Desenvolvimento ao espaço de estados

Uma vez determinadas as causalidades de todo o grafo, as variáveis de estado serão as variáveis p e q , respectivamente, das inércias e capacitâncias que tiverem sido assinaladas com causalidade integral. Elementos armazenadores de energia com causalidade derivativa não são dinamicamente independentes e, portanto, não geram variáveis de estado.

De posse das variáveis de estado — que irão compor um vetor X — e das variáveis de entrada — e dos elementos S_e e f dos elementos S_f , compondo um vetor U — há subsídios suficientes para se determinar o comportamento matemático do modelo do sistema. Este comportamento pode ser traduzido por uma equação na forma $\dot{X} = A \cdot X + B \cdot U$, para o caso linear, ou $\dot{X} = \phi(X, U)$, no caso geral.

Não há um procedimento estabelecido na bibliografia pesquisada, mas se recomenda listar todas as equações dos elementos, observando as causalidades e os sentidos de transmissão de potência das ligações, que determinarão sinais nas equações.

Com a prática, o usuário desta técnica poderá, tendo como ponto de partida as equações características das inércias e capacitâncias em causalidade integral, ir substituindo rapidamente as equações até obter a representação em espaço de estados.

2.1.5 Generalização dos elementos básicos

Os elementos apresentados até aqui, apesar de permitirem a modelagem de um vasto número de sistemas físicos, ainda não são suficientes para o trabalho que será apresentado posteriormente. Assim, serão apresentadas generalizações dos elementos já apresentados a elementos e estruturas com vários portos.

Assim, primeiramente serão mostrados os multibonds e os transformadores e giradores multiportos. Posteriormente, serão apresentados os campos: inicialmente serão apresentadas propriedades comuns a todos os campos, seguido de propriedades comuns aos campos armazenadores de energia, de um estudo de cada campo armazenador de energia (C , I e IC) e do campo resistivo. Finalmente, será mostrada a estrutura de junção.

Para cada novo elemento serão estudadas características da matriz que, no caso linear, relaciona as variáveis de entrada e de saída. Algumas características dessas matrizes serão relacionadas a propriedades de cada elemento.

2.1.5.1 Multibonds

Os *multibonds* são uma forma compacta de representar certo conjunto de ligações, assim como um vetor comporta um grupo de escalares. Assim, pode-se representar n ligações simples por um *multibond* de ordem n . Sua notação é mostrada na fig. 2.1.

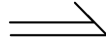


Figura 2.1: Notação para um *multibond*

2.1.5.2 Transformadores e giradores multiportos

Através desses *multibonds* serão conectados elementos generalizados, de vários portos. Um possível elemento generalizado é o transformador multiporto.

Esse elemento é a generalização do TF e tem a mesma notação. Entretanto, visando relacionar o vetor de variáveis e_1 , do porto de entrada, com o vetor e_2 , do porto de saída, e f_1 com f_2 , é necessário que o módulo de transformação seja uma matriz. De um modo análogo, pode ser definido o girador multiporto, GY .

Assim, as notações e as equações características do transformador e do girador multiportos ficam conforme mostrado na tab. 2.4.

Tabela 2.4: Transformadores e giradores multiportos, suas representações e equações características

Transformador multiporto	
$\Rightarrow TF \Rightarrow$	$\begin{bmatrix} e_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{T}^T \\ \mathbf{T} & 0 \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ e_2 \end{bmatrix}$
Girador multiporto	
$\Rightarrow GY \Rightarrow$	$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{G}^T \\ \mathbf{G} & 0 \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$

2.1.5.3 Campos armazenadores de energia

Serão apresentadas, inicialmente, propriedades gerais, comuns aos campos armazenadores de energia. Há duas notações para qualquer tipo de campo: a implícita, onde o campo é descrito apenas através de elementos simples, e a explícita, que descreve o campo através de um único elemento multiporto.

A fig. 2.2 mostra um exemplo de campo capacitivo, na forma implícita e na forma explícita.

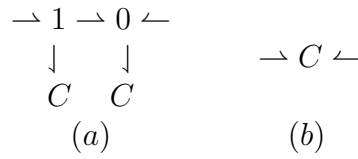


Figura 2.2: Exemplo de campo capacitivo denotado nas formas (a) implícita e (b) explícita

Analogamente aos elementos simples, em que nem todas as causalidades são possíveis, os campos também não obrigatoriamente permitem qualquer causalidade.

Saber quais são as causalidades possíveis pode não ser uma tarefa simples ao representar o grafo com um campo explícito. Por outro lado, a obtenção de um campo implícito — em que a atribuição de causalidade é mais clara — a partir de um campo explícito pode também não ser evidente.

Para campos armazenadores de energia (C , I e IC) há restrições para as leis constitutivas, uma vez que o elemento não pode fornecer energia que não tenha sido previamente armazenada. Essas relações são conhecidas como relações recíprocas de Maxwell.

Como consequência dessas relações, para o caso linear e causalidades completas, as matrizes que relacionam o vetor composto pelas variáveis de entrada com o vetor composto pelas variáveis de saída, tanto para campos C , I e IC , resultam simétricas. No caso de causalidade mista há também termos anti-simétricos.

A tab. 2.5 resume o que será apresentado nessa seção, mostrando a notação, a energia armazenada e as relações de Maxwell para cada um dos campos armazenadores de energia.

Tabela 2.5: Campos armazenadores de energia, com suas representação explícita, energia armazenada e relações de Maxwell

Campo C

$$\begin{array}{ccc}
 \begin{array}{c} \text{Diagram of field C: A central node 'C' with incoming arrows from 'q1', 'q2', 'q3' and outgoing arrows to 'q1', 'q2', 'q3'. Labels e1, e2, e3 are on the outgoing arrows. } \end{array} & E = \int \sum_{i=1}^n (e_i f_i) dt = \int \sum_{i=1}^n e_i(\mathbf{q}) dq_i = \int \mathbf{e}(\mathbf{q}) \cdot d\mathbf{q} & \frac{\partial e_i}{\partial q_j} = \frac{\partial^2 E}{\partial q_i \partial q_j} = \frac{\partial e_j}{\partial q_i} \\
 & & i, j = 1, 2, 3, \dots, n
 \end{array}$$

Campo I

$$\begin{array}{ccc}
 \begin{array}{c} \text{Diagram of field I: A central node 'IC' with incoming arrows from 'p1', 'p2', 'pn' and outgoing arrows to 'q1', 'q2', 'qn'. Labels f1, f2, fn are on the incoming arrows. } \end{array} & E = \int \sum_{i=1}^n (e_i f_i) dt = \int \sum_{i=1}^n f_i(\mathbf{p}) dp_i = \int \mathbf{f}(\mathbf{p}) \cdot d\mathbf{p} & \frac{\partial f_i}{\partial p_j} = \frac{\partial^2 E}{\partial p_i \partial p_j} = \frac{\partial f_j}{\partial p_i} \\
 & & i, j = 1, 2, 3, \dots, n
 \end{array}$$

Campo IC

$$\begin{array}{ccc}
 \begin{array}{c} \text{Diagram of field IC: A central node 'IC' with incoming arrows from 'p1', 'p2', 'pn' and outgoing arrows to 'q1', 'q2', 'qn'. Labels f1, f2, fn are on the incoming arrows. } \end{array} & E = \int \sum_{i=1}^n (e_i f_i) dt = \int \sum_{i=1}^j \mathbf{f}_i \cdot d\mathbf{p}_i + \int \sum_{k=j+1}^n \mathbf{e}_k \cdot d\mathbf{q}_k & \begin{array}{l} \frac{\partial f_i}{\partial q_k} = \frac{\partial^2 E}{\partial q_k \partial p_i} = \frac{\partial e_k}{\partial p_i} \quad 1 \leq i, l \leq j \\ \frac{\partial f_i}{\partial p_l} = \frac{\partial^2 E}{\partial p_i \partial p_l} = \frac{\partial f_l}{\partial p_i} \quad j+1 \leq k \\ \frac{\partial e_k}{\partial q_m} = \frac{\partial^2 E}{\partial q_k \partial q_m} = \frac{\partial e_m}{\partial q_k} \quad m \leq n \end{array}
 \end{array}$$

Campo C

Quando o campo C assume causalidade integrativa completa, isto é, são atribuídas causalidades integrais a todas as suas ligações, diz-se que o campo está na forma de rigidez. Por outro lado, quando o campo assume causalidade derivativa completa, ele está na forma de flexibilidade. Também são possíveis diferentes causalidades mistas, em que parte das ligações tem causalidade integral e parte tem causalidade derivativa.

Campo I

Para o campo I , quando em causalidade integrativa completa, diz-se que o campo está na forma de indutância. Já em causalidade derivativa completa, ele está na forma de massa. Também são possíveis diferentes causalidades mistas.

Campo IC

No campo IC , parte dos portos tem características de inércias e parte tem características de capacitâncias. De forma semelhante aos campos apresentados anteriormente, diversas causalidades podem ser possíveis para o campo IC .

2.1.5.4 Campo resistivo

Sobre o campo resistivo, além das observações sobre notação implícitas e explícitas, já realizadas, pode-se afirmar que a matriz que relaciona e e f é definida positiva e uma das suas propriedades é que todos os termos de sua diagonal principal são positivos. Essa propriedade da matriz decorre diretamente do fato de um campo R dissipar energia.

O elemento RS , citado anteriormente, é melhor descrito como um campo resistivo que não dissipa energia. A energia, nesse caso, seria transformada ao domínio térmico, como forma de calor — degradando a energia e gerando entropia. Para esses casos, a matriz que relaciona e e f será semi-definida positiva.

Sobre causalidade, afirma-se que, para causalidades completas, o campo pode ser escrito na forma de resistência, em que as equações características indicam esforços em função dos fluxos, ou na forma de condutância, em que todas as variáveis f estão escritas em função de e .

Ainda, afirma-se que se o campo implícito for escrito sem o uso de nenhum elemento

GY , as matrizes obedecerão as formas de Onsager ou Casimir, dependendo das causalidades associadas ao campo. O significado dessas formas foge do escopo deste trabalho. Todavia, as matrizes na forma de Onsager serão simétricas e as matrizes na forma de Casimir terão alguns termos simétricos e outros anti-simétricos.

Para campos R descritos na forma implícita sem o uso de nenhum girador, as matrizes que os caracterizam terão a forma de Onsager, se tiverem causalidade completa, ou a forma de Casimir, se tiverem causalidade mista. Contudo, um campo R genérico não segue nem a forma de Onsager nem a de Casimir.

2.1.5.5 Estruturas de junção

A estrutura de junção pode ser definida como o conjunto de todos os elementos que conservam energia em um grafo. Também, ela pode ser prevista como uma generalização do campo R em que não se dissipa energia. Logo, algumas observações realizadas para o campo resistivo também são válidas para estruturas de junção.

Contudo, para o caso linear, a matriz que associa o vetor de variáveis de entrada com o de variáveis de saída tem propriedades próprias, entre elas o fato de ser anti-simétrica e de que todos os elementos da diagonal principal serem iguais a zero.

2.1.6 Sistemas termodinâmicos fechados

Com o objetivo de modelar sistemas termodinâmicos fechados, as transformadas de Legendre são entendidas como formas de alterar as causalidades atribuídas.

2.1.6.1 Transformadas de Legendre

A transformada de Legendre, basicamente, busca expressar uma função de um conjunto de entradas inicial como função de um novo conjunto de entradas, idêntico ao anterior com a exceção de uma única variável — que dará lugar à derivada parcial da função inicial com relação à própria variável substituída.

Em linguagem matemática, $y_i = y_i(x_1, x_2, \dots, x_n)$ será transformado a $\phi_i = \phi_i(x_1, x_2, \dots, x_{i-1}, p_i, x_{i+1}, \dots, x_n)$, onde $p_i = \frac{\partial y}{\partial x_i}$. A condição matemática para que isso possa ocorrer é que a segunda derivada parcial da função inicial com relação à variável a ser substituída seja diferente de zero, isto é, $\frac{\partial^2 y}{\partial x_i^2} \neq 0$.

Para a termodinâmica, essa condição matemática representa uma condição de equilíbrio. Respeitada essa condição de equilíbrio, a transformada de Legendre permite a passagem de um potencial termodinâmico a outro.

Como cada potencial termodinâmico é função de duas diferentes variáveis independentes, a transformada de Legendre permitirá a utilização de diferentes potenciais e consequentemente, dentro do formalismo BG, descrever sistemas termodinâmicos com causalidades diversas.

2.1.6.2 O modelo em grafos de ligação

Sistemas termodinâmicos fechados podem ser descritos como campos C de dois portos. As variáveis e e f representarão, respectivamente, a temperatura T e a variação de entropia específica $\dot{s}$ num porto e a pressão p e a variação de volume específico $\dot{v}$ na outra. Seguindo essa definição de variáveis de esforço e fluxo, o formalismo dos grafos de ligação utiliza como potencial termodinâmico a energia interna.

Assim, as direções de transmissão de potência serão sempre analisadas tendo em vista a energia interna como potencial termodinâmico. Mais ainda, outros potenciais termodinâmicos serão utilizados apenas para a obtenção da equação característica dos elementos, quando necessário. A fig. 2.3 apresenta campos C com diferentes causalidades e seus potenciais termodinâmicos associados.

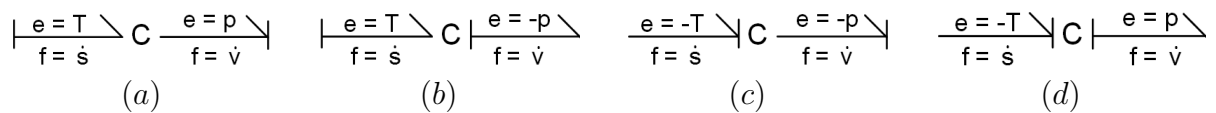


Figura 2.3: Representação de um sistema termodinâmico fechado com diferentes causalidades, descrito através (a) da energia livre (b) da entalpia (c) da energia livre de Helmholtz e (d) energia livre de Gibbs

2.2 Dinâmica dos fluidos computacional

Para o estudo em técnicas de discretização espacial será dada uma maior importância ao estudo do Método de Elementos Finitos, devido a sua maior generalidade e complexidade teórica quando comparado a outros métodos de CFD. Também, a metodologia BG-CFD incorpora aspectos desse método em especial. Assim, seguirá o resultado de um estudo dirigido especificamente ao MEF.

Maliska (2004) comenta que o objetivo dos métodos de discretização espacial é obter

uma aproximação numérica discreta para um sistema de equações diferenciais aplicadas em um meio contínuo. Para que seja garantida a convergência da solução numérica para a solução real do sistema é necessário e suficiente garantir que a solução numérica é consistente — as equações discretizadas tendem às equações diferenciais conforme sejam tomados cada vez maior de pontos — e estável — a solução numérica obtida é a solução exata das equações discretizadas.

De acordo com Norrie e Vries (1973), o Método de Elementos Finitos pode ser analisado sob a perspectiva do cálculo variacional ou através de um método de resíduos. Neste relatório será desenvolvida a abordagem através do método de resíduos.

Para todos os efeitos, serão considerados, nesta seção, dois conjuntos de variáveis: o vetor de variáveis independentes, x , e o conjunto de variáveis dependentes, u .

2.2.1 Classificação dos problemas

Antes de discutir características do MEF, serão discutidas algumas características gerais, próprias de problemas físicos. De acordo com Maliska (2004), modelos matemáticos podem ser classificados em três tipos: elípticos, parabólicos ou hiperbólicos.

Problemas elípticos são aqueles em que informações físicas são transmitidas a todos os pontos do domínio de solução. Norrie e Vries (1973) afirmam também que as condições de contorno são prescritas para toda a fronteira.

Problemas parabólicos e hiperbólicos permitem o procedimento de marcha, uma vez que para determinar a solução em um ponto, é necessário conhecer apenas as condições de contorno a montante dele. Tipicamente, problemas desses tipos são casos de propagação.

2.2.2 Noção local *versus* noção global

Uma noção fundamental do MEF é o uso de elementos e nós. Para a aplicação do método, o domínio de cálculo será dividido em pequenos pedaços que serão chamados de elementos. Cada elemento tem pré-estabelecidos uma forma geométrica e certa quantidade de nós — pontos nos quais serão calculados valores aproximados para as variáveis dependentes.

Dividindo o domínio em milhares de elementos e nós, gera-se um número enorme de cálculos para serem executados. Entretanto, esses elementos não são quaisquer: eles podem ser divididos em grupos de elementos com a mesma forma geométrica e número de nós.

Cada grupo, cada tipo de elemento, é caracterizado matematicamente por um conjunto de equações. As equações, quando escritas em um sistema de coordenadas próprio a cada

elemento — os sistema de coordenadas locais —, serão escritas do mesmo modo para cada tipo de elemento.

Assim, são poupados esforços para caracterizar todos os elementos da malha. No entanto, uma vez que há fatores como o tamanho dos lados e dos ângulos que diferenciam elementos de um mesmo tipo entre si, é necessário transpor todas as equações dos sistemas locais a um único sistema, que será chamado de global.

Para essa transposição, faz-se necessário não apenas o uso do jacobiano — que, essencialmente, altera a escala na transformação de coordenadas —, mas também uma forma especial de montagem no sistema global, uma vez que cada elemento ocupa uma posição específica no domínio.

2.2.3 Os métodos de resíduos

O MEF, em nenhum caso, obterá um vetor de soluções exato, u , mas sim um vetor de soluções aproximadas, $\tilde{u}$. A diferença entre esses dois vetores multiplicada por uma matriz de operadores diferenciais, A , é definida como o resíduo, R . Assim,

$$A \cdot u - A \cdot \tilde{u} = R \quad (2.1)$$

A idéia de todos os métodos de resíduo é forçar que esse resíduo se aproxime de zero, forçando que a solução aproximada $\tilde{u}$ convirja para a solução exata u . De acordo com Norrie e Vries (1973), há diferentes formas de cumprir essa tarefa. Serão apresentadas algumas delas a seguir:

2.2.3.1 Erro absoluto

Nesse caso, busca-se assegurar que o maior valor, em número absoluto, de resíduo em todo o domínio de cálculo seja menor que um valor pré-estabelecido. Em forma matemática,

$$\max |R(x_j)| \leq \xi \quad \text{para } x_j \in D \quad (2.2)$$

2.2.3.2 Colocação

Para a colocação, o resíduo é imposto em zero para alguns pontos, escolhidos de uma forma razoavelmente uniforme em algum padrão simples, pertencentes ao domínio de solução.

2.2.3.3 Subdomínio

Aqui, o domínio de solução D é subdividido em n subdomínios D_i e a integral do resíduo em cada subdomínio é igualada a zero. Em forma matemática,

$$\int_{D_i} R dD_i = 0; \quad i = 1, 2, \dots, n \quad (2.3)$$

2.2.3.4 Ortogonalidade

Para o método da ortogonalidade, cria-se conjunto linearmente independente de n funções de teste ϕ_i definidas no domínio D . Posteriormente, força-se que o vetor de resíduos R seja ortogonal a esse conjunto de funções, assegurando que integrais do produto dos resíduos com o conjunto de funções sejam nulas. Em linguagem matemática,

$$\int_D R \cdot \phi_i dD = 0; \quad i = 1, 2, \dots, n \quad (2.4)$$

É importante ressaltar que frequentemente não se conhece a solução exata, u . Logo, pode não se conhecer o vetor de resíduos R de uma forma direta. Isso e outros fatores motivaram a maior utilização do método de Galerkin, um caso particular da ortogonalidade, como método de minimização dos resíduos.

Entretanto, o método de Galerkin, por si só, ainda não viabiliza o desenvolvimento de soluções que permitam o estudo de efeitos advectivos. Isso só será possível através do método de Petrov-Galerkin, em que funções de peso são utilizadas, dando maior importância para alguns valores nodais, em detrimento de outros.

2.2.4 Funções de teste

Como afirmado anteriormente, um conjunto linearmente independente de funções de teste — representadas pela letra ϕ — será usado, por exemplo, no método de Petrov-Galerkin. Há, matematicamente, três condições para as funções de teste, citadas por Sobrinho (2006).

A condição que garante que o conjunto de funções é linearmente independente é descrita pela eq. (2.5).

$$\phi_i = \begin{cases} 1; & \text{para o nó } i \\ 0; & \text{para qualquer outro nó} \end{cases} \quad (2.5)$$

A segunda condição garante a normalização da base formada por esse conjunto de funções e a uniformidade na representação do domínio, de forma que a soma de todas as funções de teste em um ponto qualquer resulta igual a 1. Isso pode ser descrito matematicamente pela eq. (2.6).

$$\sum_{i=1}^n \phi_i(r) = 1; \quad \text{onde } r \text{ é um vetor posição qualquer} \quad (2.6)$$

Há ainda uma terceira condição, que determina que a primeira derivada das funções de teste deve ser integrável no domínio de cálculo.

A nomenclatura genérica “funções de teste” abrange dois tipos de função: as funções de forma e as funções de peso. Ambas aproximam os valores de u em um ponto qualquer do domínio. Entretanto, as funções de forma são invariantes com o tempo, isto é, dependem unicamente da posição, e as funções de peso podem depender de parâmetros calculados para cada instante de tempo.

As funções de peso permitem que uma maior importância relativa seja atribuída a valores nodais particulares. Será apresentado posteriormente um esquema de *upwind*, em que uma maior importância é dada para valores nodais a montante do escoamento.

A seguir serão mostrados alguns exemplos de funções de teste para espaços contínuos unidimensionais e bidimensionais, extraídos de Hunter e Pullan (2003).

2.2.4.1 Exemplos de funções de teste

A função unidimensional mais simples é a linear. Nesta, para cada elemento são definidas duas funções, cada uma relacionada a um nó. As funções são mostradas e plotadas na fig. 2.4.

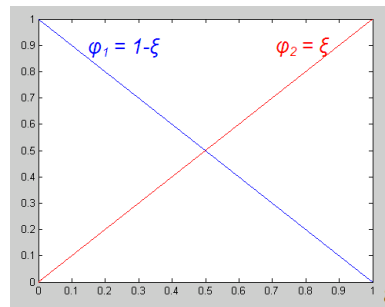


Figura 2.4: Funções de teste unidimensionais lineares, plotadas para um único elemento

Um exemplo de funções de teste um pouco mais complexas são as funções quadráticas,

plotadas na fig. 2.5.

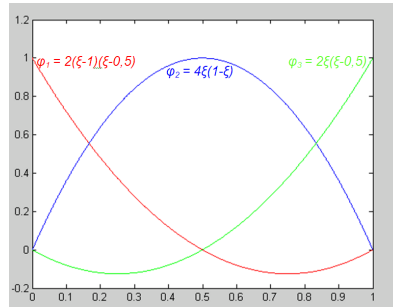


Figura 2.5: Funções de teste unidimensionais quadráticas, plotadas para um único elemento

Um último exemplo de funções de teste unidimensionais, mostrado na fig. 2.6 serão as funções de teste cúbicas de Hermite, que aproximam não apenas o valor da função no elemento, mas também o valor de sua primeira derivada. Para a utilização deste conjunto de funções, é preciso dispor dos valores da função e de sua primeira variável em cada nó.

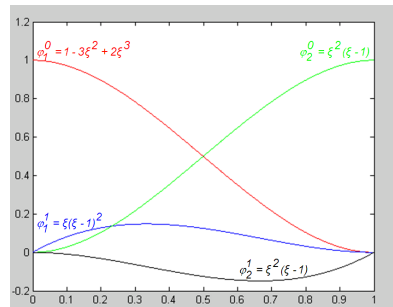


Figura 2.6: Funções de teste unidimensionais cúbicas de Hermite, plotadas para um único elemento

Para domínios bidimensionais, podem ser criados elementos quadriláteros ou triangulares. Elementos quadriláteros serão reduzidos, quando escritos em função das coordenadas locais, a quadrados.

Devido à ortogonalidade dos eixos coordenados, basta multiplicar funções de teste unidimensionais para obter equações de funções de teste de elementos quadrados. Funções de teste bi-lineares são mostradas na fig. 2.7.

Cabe salientar que não é obrigatório que o elemento seja composto por duas funções unidimensionais do mesmo tipo. Para descrever o comportamento de um fenômeno que acontece de forma mais importante em uma única direção pode ser utilizado um elemento como o da fig. 2.8, por exemplo.

Em relação às funções de teste bidimensionais bi-cúbicas de Hermite, necessita-se dos valores nodais não apenas da função, u , mas também suas duas primeiras derivadas direcionais,

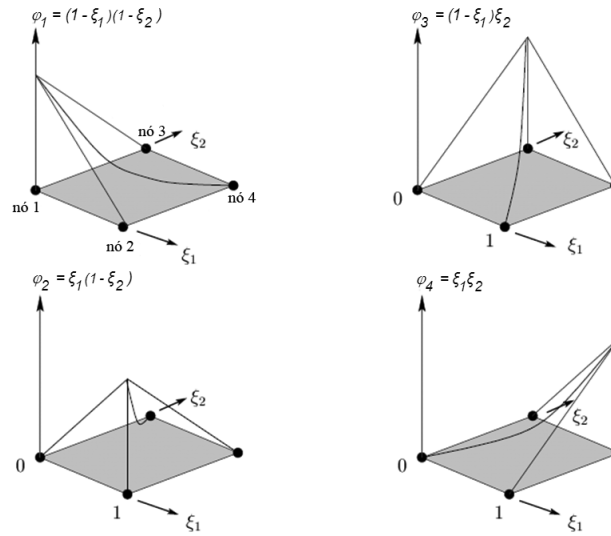


Figura 2.7: Funções de teste bidimensionais bi-lineares, plotadas para um único elemento (adaptado de Hunter e Pullan (2003))

$\frac{\partial u}{\partial \xi_1}$ e $\frac{\partial u}{\partial \xi_2}$, e sua segunda derivada cruzada, $\frac{\partial^2 u}{\partial \xi_1 \partial \xi_2}$.

Para elementos triangulares, as funções de teste são definidas através de uma razão de áreas, de forma que $\phi_1 = \frac{Area(P23)}{Area(123)}$, $\phi_2 = \frac{Area(1P3)}{Area(123)}$ e $\phi_3 = \frac{Area(12P)}{Area(123)}$, sendo P um ponto genérico pertencente ao elemento, como indica a fig. 2.9.

Para o elemento triangular mostrado, ressalta-se que $\phi_1 + \phi_2 + \phi_3 = 1$, respeitando uma das condições necessárias. Também, essas funções de teste resultam lineares. Entretanto também podem ser obtidas funções de teste não lineares para elementos triangulares.

Um último exemplo de função de forma, apresentado na fig. 2.10, é tipicamente usada em problemas unidimensionais como função de peso para um esquema de upwind. Esta função de forma é baseada na função de teste linear, apresentada anteriormente.

No entanto, um parâmetro β , que varia em função do número de Péclet, desloca cada reta na direção vertical, de forma a dar maior importância a um nó e menos a outro para as interpolações dos valores de variáveis dependentes.

2.2.4.2 Escolha das funções de teste

É desejável que as funções de teste sejam escolhidas de forma coerente com o comportamento do modelo físico a ser discretizado, de forma a obter melhores resultados com um menor número de incógnitas.

Um exemplo de escolha ruim de funções de teste para aproximar uma função é ilustrado pela fig. 2.11. Para evitar uma este problema há duas soluções possíveis: refinar a malha,

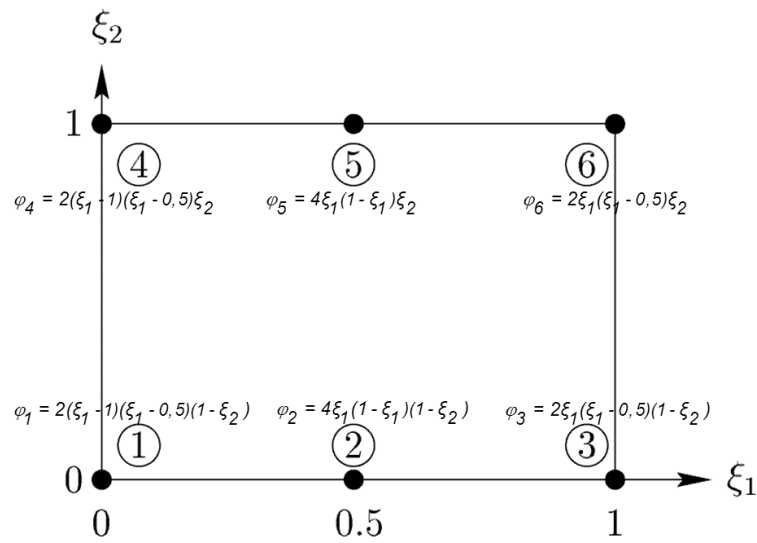


Figura 2.8: Funções de teste bidimensionais quadrático-lineares, escritas para um único elemento de seis nós (adaptado de Hunter e Pullan (2003))

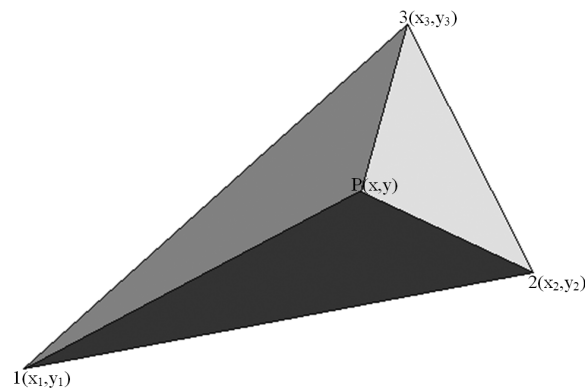


Figura 2.9: Elemento triangular

mantendo a escolha de funções de teste, ou escolher uma função de teste que aproxime melhor a função original, como funções de teste quadráticas, por exemplo.

2.2.5 Condições de contorno

Sobrinho (2006) cita três tipos básicos de condições de contorno.

Para a condição de contorno de Dirichlet, o valor da variável dependente u é prescrito na fronteira. Em linguagem matemática,

$$u = g \quad (2.7)$$

em que g é uma função conhecida.

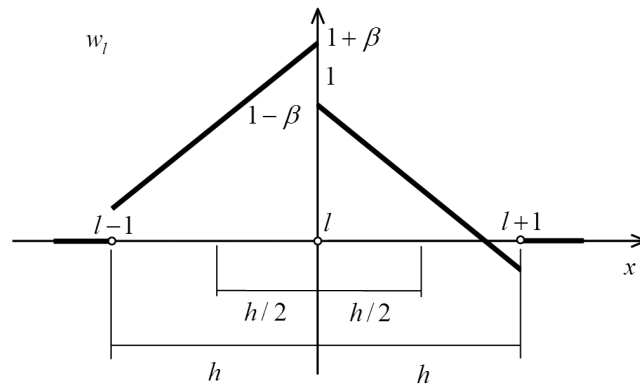


Figura 2.10: Função de teste unidimensional com esquema de *upwind*

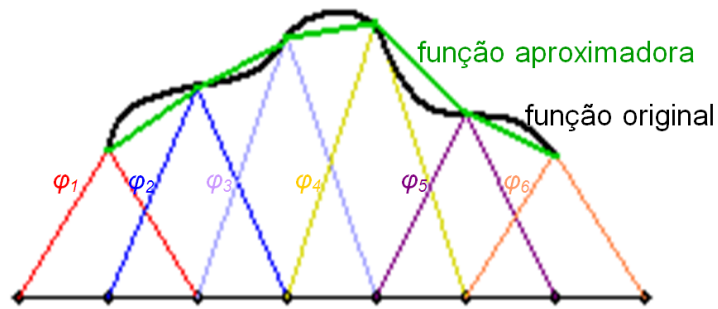


Figura 2.11: Exemplo de discretização ruim

Para a condição de contorno de Neumann, uma combinação linear do valor das derivadas da variável dependente u nas direções normal e tangencial assume um valor prescrito. Em linguagem matemática,

$$\alpha \frac{\partial u}{\partial n} + \beta \frac{\partial u}{\partial s} = g \quad (2.8)$$

A bibliografia também cita uma terceira condição de contorno, em que uma combinação linear da variável u e suas derivadas nas direções normal e tangencial é prescrita. Em linguagem matemática,

$$\alpha \frac{\partial u}{\partial n} + \beta \frac{\partial u}{\partial s} + \gamma u = g \quad (2.9)$$

2.3 Metodologia BG-CFD para escoamentos incompressíveis

Com as ferramentas teóricas apresentadas anteriormente, é possível agora discorrer sobre a metodologia BG-CFD.

Esta seção é fundamentada principalmente em Baliño (2009) e apresentará um modelo em grafos de ligação para a resolução numérica de escoamentos incompressíveis.

2.3.1 O escoamento incompressível

Um escoamento incompressível pode ser conceituado como aquele em que a massa específica de cada partícula fluida é uma constante.

O fato da derivada da massa específica ser nula gera uma simplificação na equação da continuidade. Dessa equação resulta a condição comumente usado para definir matematicamente o escoamento, em que o divergente da velocidade vetorial é nulo. Em forma matemática,

$$\nabla \cdot \mathbf{V} = 0 \quad (2.10)$$

Outra condição frequentemente usada para definir escoamentos incompressíveis é um baixo número de Mach. White (2002) mostra essa condição como consequência da definição anterior. O autor também afirma que o limite usual é $Ma < 0,3$.

Mais do que isso, a pressão deixa de ser uma variável termodinâmica e o campo de pressão se ajusta para satisfazer a condição imposta pela equação da continuidade.

Do ponto de vista do tipo de problema a ser resolvido numericamente, é importante ressaltar, que como qualquer perturbação se propaga com velocidade infinita, qualquer problema será do tipo elíptico no espaço, para cada instante temporal. Para a análise temporal, no entanto, o problema será parabólico.

As outras quantidades conservadas no escoamento podem ser modeladas através das eqs. (2.11), para a quantidade de movimento, e (2.12), para a energia térmica.

$$\rho_0 \frac{\partial \mathbf{V}}{\partial t} = -\nabla t_v^* + \rho_0 \mathbf{V} \times (\nabla \times \mathbf{V}) - \nabla P + \rho_0 \mathbf{G} + \nabla \cdot \underline{\underline{\tau}} \quad (2.11)$$

$$\frac{\partial u_v}{\partial t} = -\nabla \cdot \mathbf{q} - \nabla u_v \cdot \mathbf{V} + \nabla \mathbf{V} : \underline{\underline{\tau}} + \rho_0 \Phi \quad (2.12)$$

2.3.2 Equações de balanço

As duas últimas equações apresentadas descrevem o comportamento do sistema físico. No entanto, para descrevê-las com o formalismo dos grafos de ligação é ainda necessário

desenvolvê-las um pouco mais, de forma a obter as variáveis de esforço e fluxo.

Vale ressaltar novamente que essas equações, tendo em vista o formalismo dos grafos de ligação, evidenciarão as contribuições parciais de cada fenômeno no fluxo de potências no sistema. Um termo em especial realiza um acoplamento entre os dois portos — modelando a perda de pressão de estagnação.

Para o porto de momento linear, é preciso, em primeiro lugar, obter o produto escalar da eq. (2.11) por V . No desenvolvimento posterior ainda são usadas a equação da continuidade, (2.10), e as seguintes identidades:

$$[V \times (\nabla \times V)] \cdot V = 0 \quad (2.13)$$

$$(\nabla \cdot \underline{\underline{\tau}}) \cdot V = \nabla \cdot (\underline{\underline{\tau}} \cdot V) - \nabla V : \underline{\underline{\tau}} \quad (2.14)$$

A primeira equação de balanço, então, resulta como:

$$p_v \cdot \frac{\partial V}{\partial t} = \nabla \cdot [(-P \underline{\underline{I}} + \underline{\underline{\tau}}) \cdot V] - \nabla t_v^* \cdot V + \rho_0 G \cdot V - \nabla V : \underline{\underline{\tau}} \quad (2.15)$$

Para o porto de energia térmica, deve-se usar a condição de continuidade, (2.10), e a eq. (2.16), que define o potencial de energia interna. É importante ressaltar que a equação a seguir também explicita que, para um fluido incompressível, a energia interna é função unicamente da temperatura.

$$\theta = \frac{d u_v}{d s_v} \quad (2.16)$$

Assim, obtêm-se:

$$\theta \frac{\partial s_v}{\partial t} = -\nabla \cdot q + \nabla \cdot (\theta s_v V) - s_v \nabla \theta \cdot V + \rho_0 \Phi + \nabla V : \underline{\underline{\tau}} \quad (2.17)$$

Os termos das equações de balanço podem ser classificados em quatro tipos: transientes, de divergência, de acoplamento e de fonte.

Os termos transientes estão relacionados com a evolução temporal do comportamento do sistema.

Os de divergência transmitem as condições de contorno através do domínio do fenômeno.

O de acoplamento, de uma forma mais geral, modela a transformação de energia. É interessante notar que, dado que a transformação em estudo é uma degradação de energia, esse termo é não-negativo, de forma que a transformação de energia térmica ser convertida em energia mecânica não é possível.

Finalmente, os termos de fonte modelam a adição de energia ao sistema. Cabe lembrar que, apesar de não ser possível “criar” energia, a afirmação cabe, dependendo dos objetivos de modelagem.

2.3.3 Discretização

Tendo dito tudo isso, é preciso afirmar que frequentemente não há solução analítica para essas equações contínuas. Passam a ser importantes, então, soluções numéricas que só podem ser obtidas após uma discretização das equações.

De forma similar ao MEF, as funções definidas no meio contínuo serão discretizadas com o uso de funções de forma e valores nodais, de forma que:

$$V(r, t) = \sum_{m=1}^{n_V} V_m(t) \varphi_{V_m}(r) = \underline{V}^T \cdot \underline{\varphi}_V \quad (2.18)$$

$$s_v(r, t) = \sum_{l=1}^{n_S} s_{vl}(t) \varphi_{sl}(r) = \underline{s}_v^T \cdot \underline{\varphi}_S \quad (2.19)$$

$$P(r, t) = \sum_{k=1}^{n_P} P_k(t) \varphi_{Pk}(r) = \underline{P}^T \cdot \underline{\varphi}_P \quad (2.20)$$

Para essas equações, $\underline{V}$, $\underline{s}_v$ e $\underline{P}$ são vetores de valores nodais e $\underline{\varphi}_V$, $\underline{\varphi}_S$ e $\underline{\varphi}_P$ são vetores que armazenam as funções de forma, respectivamente, da velocidade, entropia e pressão de cada nó.

Posteriormente, será analisada a evolução temporal do vetor nodal de entropia, $\underline{S}$, que pode ser relacionado com o vetor de valores nodais de entropia específica, $\underline{s}_v$, como:

$$\underline{S} = \underline{\Omega}_S \cdot \underline{s}_v \quad (2.21)$$

Onde $\underline{\underline{\Omega_S}}$ é a matriz diagonal de volume, cujos elementos são definidos por:

$$\Omega_{Sl} = \int_{\Omega} \varphi_{Sl} d\Omega \quad (2.22)$$

Para a descrição da entropia serão utilizadas ainda funções de peso, $w_{Sl}(r, t)$, de forma análoga ao formalismo de Petrov-Galerkin, com o objetivo de respeitar a energia trocada pelo sistema em suas fronteiras, além de ponderar a importância entre nós vizinhos em diversos termos de equações de balanço de energia. Essas funções de peso permitirão, por exemplo, a introdução de esquemas de *upwind*.

2.3.4 Equacionamento matemático

Ao integrar as equações de balanço no domínio Ω , tendo antes multiplicado a equação pela função de teste respectiva, e após aplicar o Teorema de Gauss quando necessário, obtêm-se, para o momento linear, a seguinte equação:

$$\underline{\underline{M}} \cdot \dot{\underline{V}} = \underline{F_V^{(\Gamma)}} + \underline{F_P^{(\Gamma)}} + \underline{F_G} + \underline{F_K} + \underline{F_R} + \underline{F_P} - \underline{F_V} \quad (2.23)$$

onde:

- $\underline{F_V^{(\Gamma)}} = \int_{\Gamma} (\underline{\tau} \cdot \underline{\check{n}}) \underline{\varphi_V} d\Gamma$ é o termo de acoplamento e representa a força exercida pelo atrito viscoso na fronteira do sistema
- $\underline{F_P^{(\Gamma)}} = - \int_{\Gamma} P \underline{\varphi_V} \underline{\check{n}} d\Gamma$ está relacionado com a pressão aplicada na fronteira do sistema
- $\underline{F_G} = \rho_0 \int_{\Omega} G \underline{\varphi_V} d\Omega$ é a força exercida pelo campo gravitacional
- $\underline{F_K} = - \int_{\Omega} \nabla t_v^* \underline{\varphi_V} d\Omega$ está relacionado com gradientes da coenergia cinética
- $\underline{F_R} = \rho_0 \int_{\Omega} \underline{V} \times (\nabla \times \underline{V}) \underline{\varphi_V} d\Omega$ leva em consideração efeitos rotacionais
- $\underline{F_P} = \int_{\Omega} P \nabla \underline{\varphi_V} d\Omega$ está relacionado com a pressão no interior do domínio de cálculo
- $\underline{F_V} = \int_{\Omega} \underline{\tau} \cdot \nabla \underline{\varphi_V} d\Omega$ é o termo de acoplamento e representa a força exercida pelo atrito viscoso no interior do domínio de cálculo, gerando entropia de forma irreversível

Já para o balanço de energia térmica, tem-se que:

$$\underline{\dot{S}} = \underline{\dot{S}}_Q^{(\Gamma)} + \underline{\dot{S}}_C^{(\Gamma)} + \underline{\dot{S}}_Q + \underline{\dot{S}}_C + \underline{\dot{S}}_F + \underline{\dot{S}}_A + \underline{\dot{S}}_V \quad (2.24)$$

onde:

- $\underline{\dot{S}}_Q^{(\Gamma)} = -\underline{\underline{\Theta}}^{-1} \cdot \left(\int_{\Gamma} \underline{w}_S \underline{q} \cdot \underline{\check{n}} d\Gamma \right)$ é o termo devido à condução de calor na fronteira do domínio de cálculo. $\underline{\check{n}}$ é um vetor normal à fronteira, apontando para fora do domínio de cálculo. Para a obtenção desse termo foi aplicado o teorema de Gauss
- $\underline{\dot{S}}_C^{(\Gamma)} = \underline{\underline{\Theta}}^{-1} \cdot \left(\int_{\Gamma} \underline{w}_S \underline{\theta} s_v \underline{V} \cdot \underline{\check{n}} d\Gamma \right)$ é um termo devido à convecção na fronteira do domínio de cálculo. Para a obtenção desse termo foi aplicado o teorema de Gauss
- $\underline{\dot{S}}_Q = \underline{\underline{\Theta}}^{-1} \cdot \left(\int_{\Omega} \underline{q} \cdot \underline{\nabla w}_S d\Omega \right)$ representa o fluxo de entropia relacionado com o calor trocado no interior do domínio de cálculo. Para a obtenção desse termo, foi aplicado o teorema de Gauss
- $\underline{\dot{S}}_C = -\underline{\underline{\Theta}}^{-1} \cdot \left(\int_{\Omega} \underline{\theta} s_v \underline{V} \cdot \underline{\nabla w}_S d\Omega \right)$ representa o fluxo de entropia devido à convecção no interior do domínio de cálculo. Para a obtenção desse termo, foi aplicado o teorema de Gauss
- $\underline{\dot{S}}_F = \underline{\underline{\Theta}}^{-1} \cdot \left(\rho_0 \int_{\Omega} \underline{w}_S \Phi d\Omega \right)$ é a entropia gerada por uma fonte interna de calor
- $\underline{\dot{S}}_A = -\underline{\underline{\Theta}}^{-1} \cdot \left(\int_{\Omega} \underline{w}_S s_v \underline{V} \cdot \underline{\nabla \theta} d\Omega \right)$ representa a geração de entropia devido ao gradiente de temperatura interno ao elemento. Em outras palavras, esse termo é devido à não-homogeneidade dentro de cada elemento
- $\underline{\dot{S}}_V = \underline{\underline{\Theta}}^{-1} \cdot \left[\int_{\Omega} \underline{w}_S (\underline{\nabla V} : \underline{\tau}) d\Omega \right]$ é o termo de acoplamento, que representa o fluxo de entropia gerado pelo atrito

2.3.5 Modelo em grafos de ligação

O equacionamento mostrado na seção anterior pode ser condensado a e completamente representado pelo grafo de ligação mostrado na fig. 2.12.

A equação de balanço de momento linear está representada na junção 1 e a equação de balanço de energia térmica, na junção 0. Há um termo de acoplamento, representado através de um elemento MRS, que degrada energia.

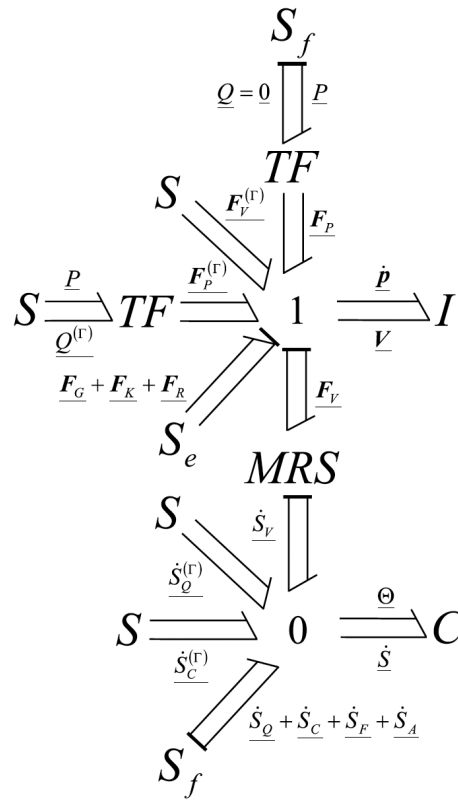


Figura 2.12: Grafo de ligação para escoamentos incompressíveis

2.4 Metodologia BG-CFD para escoamentos incompressíveis unidimensionais em dutos

Finalmente, esta seção apresentará o problema de interesse deste trabalho de formatura: o escoamento incompressível unidimensional em um duto. A referência principal é Baliño (2006).

2.4.1 O escoamento incompressível unidimensional em um duto

Já foi afirmado o que se entende por escoamento incompressível. Resta agora conceituar um escoamento unidimensional e um duto.

De acordo com Pimenta (1981), um escoamento unidimensional é tal que todas as propriedades do fluido são funções unicamente do tempo e de uma coordenada espacial. Pela própria definição de função, pode-se entender que, para cada seção transversal a essa coordenada, há um único valor, constante, para cada propriedade.

Já um duto é compreendido por Munson, Young e Okiishi (2004) como um conduto fechado de seção transversal qualquer. Dessa forma, tem-se como hipótese que o fluido ocupa

as seções por inteiro, isto é, que o duto está sempre completamente preenchido.

Ainda, será modelado apenas o caso de paredes rígidas, isto é, paredes que não sofrem deformação.

Esse tipo de escoamento é, portanto, um caso particular do escoamento apresentado na seção anterior. As equações que modelam esse fenômeno são, então, diferentes.

Considerando unitário o fator de peso dos fluxos de momento linear ou energia cinética através da área de passagem e desprezando tensões viscosas normais, obtêm-se as equações de conservação de momento linear e de energia térmica mostradas a seguir:

$$\frac{\rho_0}{A} \dot{Q} = -\frac{\partial P}{\partial x} + \rho_0 G_x - \frac{\rho_0 Q^2}{A} \frac{d}{dx} \left(\frac{1}{A} \right) - \tau_{wx} \frac{\mathcal{P}_w}{A} \quad (2.25)$$

$$\theta \frac{\partial s_v}{\partial t} = -q_w \frac{\mathcal{P}_h}{A} - \frac{1}{A} \frac{\partial}{\partial x} (q_x A) - \frac{Q}{A} \frac{\partial}{\partial x} (\theta s_v) + \rho_0 \Phi + \frac{Q}{A} s_v \frac{\partial \theta}{\partial x} + \tau_{wx} Q \frac{\mathcal{P}_w}{A^2} \quad (2.26)$$

Nessas equações, P é a pressão, τ_{wx} a tensão de cisalhamento na parede devido à viscosidade. Essa tensão pode ser modelada como

$$\tau_{wx} = \frac{1}{8} f \rho_0 \frac{Q |Q|}{A^2} \quad (2.27)$$

Onde f é o fator de atrito de Darcy-Weisbach.

$\mathcal{P}_w$ e $\mathcal{P}_h$ são os perímetros molhado e aquecido, respectivamente. G_x é uma força externa por unidade de massa na direção do duto e no sentido de x positivo. Normalmente essa força será uma componente da força gravitacional.

Ainda, Φ é uma fonte de calor interna por unidade de massa, q_w e q_x são os fluxos de calor na parede e axial. O calor trocado com a parede pode ser escrito como função da temperatura da parede e do coeficiente de convecção H :

$$q_w = -H (\theta_w - \theta) \quad (2.28)$$

Já o calor trocado axialmente pode ser modelado pela lei de Fourier, com λ representando a condutividade térmica:

$$q_x = -\lambda \frac{\partial \theta}{\partial x} = -\frac{\lambda \theta}{\rho_0 c_v} \frac{\partial s_v}{\partial x} \quad (2.29)$$

2.4.2 Equações de balanço

Pelos mesmos argumentos apresentados na seção 2.3.2, aqui se faz necessário multiplicar a eq. (2.25) pela velocidade de fluido. Assim, a equação de balanço de momento linear será desenvolvida a:

$$p_v \dot{Q} = -\frac{Q}{A} \frac{\partial P}{\partial x} + \rho_0 G_x \frac{Q}{A} - \frac{\rho_0 Q^3}{A^2} \frac{d}{dx} \left(\frac{1}{A} \right) - \tau_{wx} Q \frac{\mathcal{P}_w}{A^2} \quad (2.30)$$

A segunda equação de balanço, para a energia térmica, é exatamente a apresentada em (2.26).

2.4.3 Discretização

De forma análoga à discretização definida na seção anterior, será definida a discretização de variáveis. No entanto, tendo em vista a continuidade no duto, a vazão é única para cada instante temporal. A velocidade do fluido será determinada a partir da eq. (2.31).

$$V(x, t) = \frac{Q(t)}{A(x)} \quad (2.31)$$

Dessa forma, a única variável que necessita de discretização é a entropia. Assim, para a entropia por unidade de volume,

$$s_v(x, t) = \sum_{l=1}^n s_{vl}(t) \varphi_{Sl}(x) = \underline{s}_v^T \cdot \underline{\varphi}_S \quad (2.32)$$

A relação entre o vetor nodal de entropia e o vetor nodal de entropia por unidade de volume apresentada na seção 2.3.3 é válida aqui também.

Além disso, também serão utilizadas funções de peso para a entropia, $w_{Sl}(x, t)$, de forma similar ao formalismo de Petrov-Galerkin para o MEF.

2.4.4 Equacionamento matemático

Integrando a eq. (2.30) no domínio, isto é, para $x = 0$ até $x = L$, obtêm-se

$$I \frac{dQ}{dt} = \Delta P + \Delta P_G + \Delta P_B - \Delta P_{VW} \quad (2.33)$$

onde:

- $\Delta P = P_0 - P_L$ é a diferença de pressão na entrada e na saída do duto
- $\Delta P_G = \rho_0 G_x L$ é a pressão gerada pela componente do campo gravitacional na direção do duto. Uma formulação alternativa será deduzida posteriormente, e resulta em $\Delta P_G = -\rho_0 g Z$
- $\Delta P_B = \frac{1}{2} \rho_0 Q^2 \left(\frac{1}{A_0^2} - \frac{1}{A_L^2} \right)$ é a variação de pressão convectiva (por efeito Bernoulli), gerada pela diferença de áreas no duto
- $\Delta P_{VW} = \int_0^L \tau_{wx} \frac{P_w}{A} dx$ é o termo de acoplamento, que representa a perda de pressão devida ao atrito, gerando entropia de forma irreversível
- $I = \rho_0 \int_0^L \frac{dx}{A}$ é a inércia hidráulica

Já para o balanço de energia térmica, a eq. (2.26) é multiplicada pelas funções de peso $w_{sl}(x, t)$. Após integração e aplicação do Teorema de Gauss, obtêm-se a formulação mostrada em Baliño (2006). Uma pequena alteração foi realizada, fazendo com que os termos parciais da soma no lado direito da equação sejam potências, gerando:

$$\underline{\Theta} \cdot \dot{\underline{S}} = -\underline{W_Q^{(\Gamma)}} - \underline{W_C^{(\Gamma)}} - \underline{W_{QW}} + \underline{W_Q} + \underline{W_C} + \underline{W_F} + \underline{W_A} + \underline{W_{VW}} \quad (2.34)$$

onde:

- $\underline{W_Q^{(\Gamma)}} = (q_{xL} A_L \delta_{ln} - q_{x0} A_0 \delta_{l1})$ é o termo devido à condução de calor axialmente, na fronteira do escoamento, isto é, na entrada e na saída do duto. Para a obtenção desse termo, foi aplicado o teorema de Gauss
- $\underline{W_C^{(\Gamma)}} = Q (\theta_L s_{vL} \delta_{ln} - \theta_0 s_{v0} \delta_{l1})$ é um termo devido à convecção na fronteira do escoamento. Para a obtenção desse termo, foi aplicado o teorema de Gauss

- $\underline{W}_{QW} = \int_0^L q_w \mathcal{P}_h \underline{w}_S dx$ representa a potência específica relacionada com o calor trocado com as paredes internas do duto
- $\underline{W}_Q = \int_0^L A q_x \frac{\partial w_S}{\partial x} dx$ representa a potência específica relacionada com o calor trocado axialmente no interior do domínio de cálculo. Para a obtenção desse termo, foi aplicado o teorema de Gauss
- $\underline{W}_C = Q \int_0^L \theta s_v \frac{\partial w_S}{\partial x} dx$ representa a potência específica devida à convecção no interior do domínio de cálculo. Para a obtenção desse termo, foi aplicado o teorema de Gauss
- $\underline{W}_F = \rho_0 \int_0^L \Phi A \underline{w}_S dx$ é a a potência específica gerada por uma possível fonte interna de calor
- $\underline{W}_A = Q \int_0^L s_v \frac{\partial \theta}{\partial x} \underline{w}_S dx$ descreve a geração de entropia devido ao gradiente de temperatura interno a cada elemento. Em outras palavras, esse termo é devido à não-homogeneidade dentro de cada elemento
- $\underline{W}_{VW} = Q \int_0^L \tau_{wx} \frac{\mathcal{P}_w}{A} \underline{w}_S dx$ é o termo de acoplamento, que representa a potência específica gerada pelo atrito
- $\underline{\Theta}$ é a matriz de temperatura e seus termos são definidos como $(\underline{\Theta})_{lj} = \frac{1}{\Omega_{Sj}} \int_0^L A \theta w_{Sl} \varphi_{Sj} dx$

Vale lembrar que para esse tipo de escoamento a equação de conservação de massa se reduz à mostrada pela eq. (2.35).

$$Q_0 = Q_L = Q(t) \quad (2.35)$$

2.4.5 Modelo em grafos de ligação

De forma similar à seção anterior, será apresentado na fig. 2.13 o modelo usando grafos de ligação.

É interessante notar que a equação da continuidade está descrita pelo fato de haver apenas uma vazão para todo o sistema, isto é, apenas um fluxo para todos os grafos ligados na junção 1.

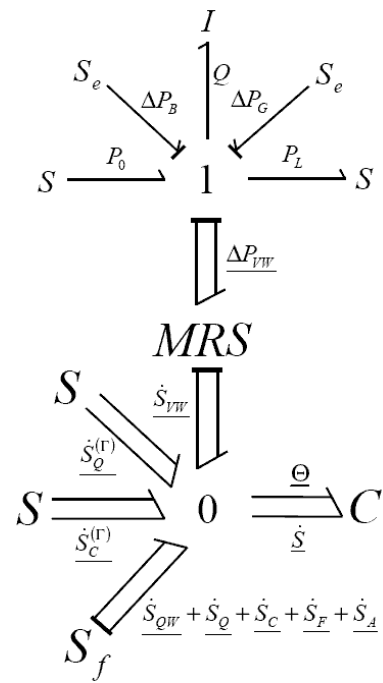


Figura 2.13: Grafo de ligação para escoamentos incompressíveis unidimensionais em dutos

3 DESENVOLVIMENTO DO EQUACIONAMENTO

Neste capítulo será apresentado o desenvolvimento numérico das equações que modelam o escoamento incompressível unidimensional em um duto, apresentadas na seção anterior.

3.1 Escolha das funções de teste

Para a escolha do conjunto de funções de teste da metodologia BG-CFD basta que as duas primeiras condições das funções de teste do MEF, descritas nas eqs. (2.5) e (2.6), sejam satisfeitas, sem a necessidade de assegurar a terceira condição.

Isso implica que podem ser usadas funções contínuas por partes ou até distribuições para o conjunto de funções de teste.

Há, no entanto, um porém importante: Cordaro e Kawano (2002), por exemplo, reforçam que o produto de distribuições não é definido. O produto entre uma distribuição e uma função, no entanto, é definido. Dessa forma, duas possíveis soluções podem ser vislumbradas.

A primeira delas seria escolher funções de teste de forma a não obter nenhum termo com produtos de distribuições.

A segunda possível solução é suavizar uma das funções com o uso de um *mollifier* (também conhecido por aproximação da identidade). Um *mollifier* é uma função suave de suporte compacto — isto é, o subconjunto do domínio para o qual a função é não nula é fechado e limitado — que respeita as condições mostradas na eq. (3.1), onde δ representa um delta de Dirac.

$$\begin{aligned} \int_{\mathbb{R}^n} \psi(x) dx &= 1 \\ \lim_{\epsilon \rightarrow 0} \psi_\epsilon(x) &= \lim_{\epsilon \rightarrow 0} \epsilon^n \psi(\epsilon x) = \delta(x) \end{aligned} \tag{3.1}$$

Pode-se então definir uma função $\tilde{\varphi}_{S,\epsilon}$, função suavizada de φ_S , tal que

$$\tilde{\varphi}_{S,\epsilon} = \varphi_S * \psi_\epsilon \quad (3.2)$$

Como no limite com $\epsilon \rightarrow 0$, ψ_ϵ tende a δ — elemento neutro da convolução, $\tilde{\varphi}_{S,\epsilon}$ tende a φ_S , tomando o mesmo limite.

Um *mollifier* usado tipicamente é uma gaussiana com desvio padrão de ϵ .

Tudo isso, claro, foi respeitado no trabalho aqui desenvolvido. No entanto, as equações expostas aqui serão mostradas com esse formalismo, a fim de não sobrecarregar a notação.

3.1.1 As funções de teste escolhidas

Para o problema em estudo, duas funções de teste devem ser selecionadas: uma função de forma e uma função de peso, ambas para a análise discreta da entropia.

Foi escolhida a função de forma constante para a entropia, pelo fato de ser a mais simples possível.

Já a função de peso tem um parâmetro de *upwind* que desloca verticalmente por partes uma função inicialmente linear.

A fig. 3.1 mostra as funções de teste plotadas para um elemento l . Em tracejado está a função de forma e em linha contínua, a função de peso.

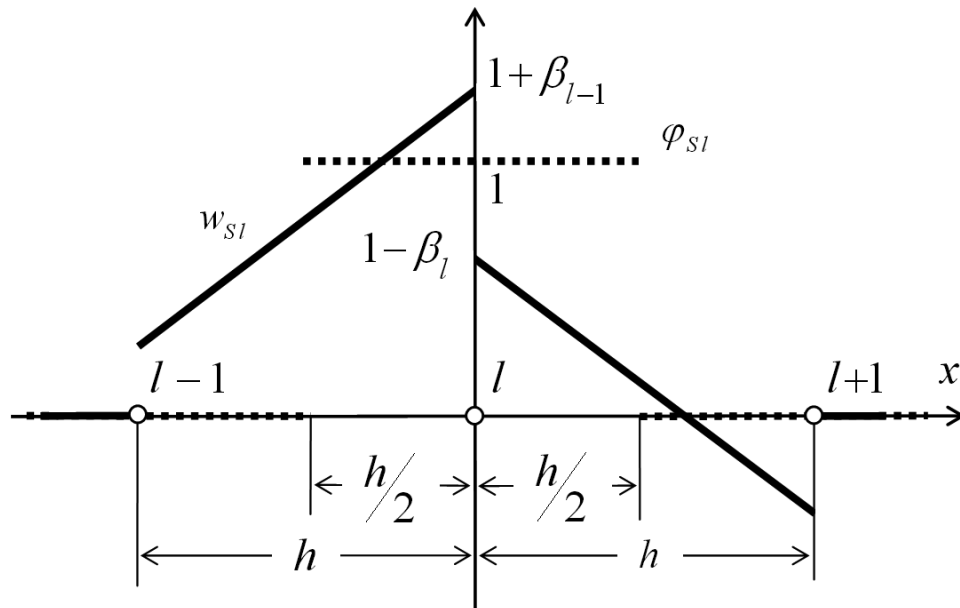


Figura 3.1: Funções de forma (em tracejado) e de peso (em linha contínua) escolhidas para a entropia

3.1.1.1 O esquema de *upwind*

Esquemas de *upwind* — avaliação a montante, no português — são usados para garantir a condição de *boundedness*, isto é, que as propriedades variem de forma suave.

Para escoamentos com advecção predominante sobre a difusão, as soluções numéricas obtidas sem o uso de avaliação a montante não representarão corretamente um sistema físico, pois será gerada uma solução numérica com instabilidade numérica, ou *wiggles*.

Raso (2006) apresenta um esquema de *upwind* que é exato para o estudo de escoamentos de advecção-difusão unidimensionais em regime permanente.

Para esse esquema, uma função linear é deslocada de β , função do número de Péclet de acordo com a eq. (3.3), a seguir.

$$\beta = \frac{e^{Pe} + 1}{2(e^{Pe} - 1)} - \frac{1}{Pe} \quad (3.3)$$

No entanto, frequentemente é colocado na bibliografia que o uso de exponenciais é custoso computacionalmente. Além disso, a função descrita na eq. (3.3) apresentará erros numéricos, para valores de Pe muito próximos de zero, e impossibilidade de cálculo, para valores altos de Pe .

Assim como em Maliska (2004), buscou-se uma função estimadora para o esquema exponencial. No entanto, como o esquema de *upwind* apresentado na referência não é exatamente o da eq. (3.3), foi necessário encontrar uma outra função estimadora.

Para isso, criou-se um código que, para um vetor com valores de Pe positivos e negativos, com módulo variando em milésimos de décadas logarítmicas, de 10^{-6} a 10^6 , acha valores de a e b , parâmetros da função estimadora, que minimizam a variância do sinal discreto determinado por $(\beta(\underline{Pe}) - \hat{\beta}(\underline{Pe}))$. A função estimadora de beta, $\hat{\beta}$, tem sua forma mostrada na eq. (3.4).

$$\hat{\beta} = \sigma(Pe) \frac{Pe^2}{a + b|Pe| + 2Pe^2} \quad (3.4)$$

Com esse algoritmo, obtêm-se uma variância de $2,0318 \cdot 10^{-5}$ para a função apresentada na eq. (3.5), que é a que será efetivamente utilizada no desenvolvimento do programa.

$$\hat{\beta} = \sigma(Pe) \frac{Pe^2}{6,226 + 4,878|Pe| + 2Pe^2} \quad (3.5)$$

As figuras 3.2 e 3.3 mostram gráficos com as funções exata e estimadora para o esquema de *upwind*.

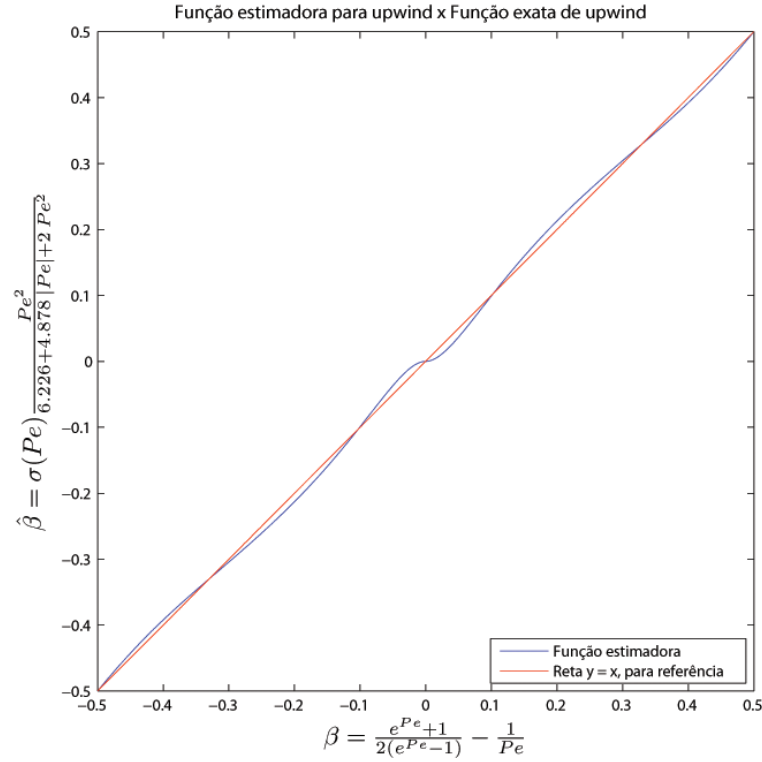


Figura 3.2: Comparação entre função exata e função estimadora para *upwind*, para todos os valores de β

3.2 Desenvolvimento dos termos no interior do domínio

3.2.1 Porto inercial

3.2.1.1 Pressão gravitacional

A perda ou ganho de pressão devido à gravidade pode ser calculado por

$$\Delta P_G = \int_0^L \rho_0 G_x dx \quad (3.6)$$

onde $G_x = -g \sen\theta$. Dessa forma,

$$\Delta P_G = -\rho_0 \int_0^L g \sen\theta dx = -\rho_0 g (z_L - z_0) \quad (3.7)$$

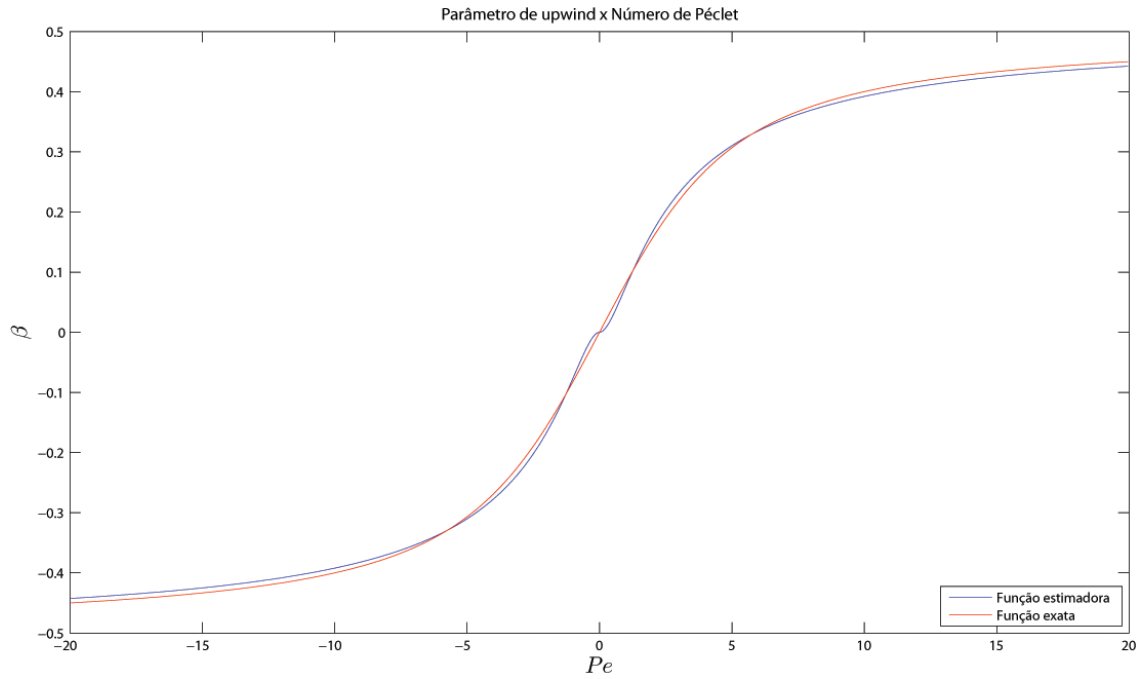


Figura 3.3: Comparação entre função exata e função estimadora para *upwind*, para $-20 < Pe < 20$

Finalmente, resulta que

$$\Delta P_G = -\rho_0 g Z \quad (3.8)$$

onde $Z = z_L - z_0$

3.2.1.2 Pressão convectiva

$$\Delta P_B = \frac{1}{2} \rho_0 Q^2 \left(\frac{1}{A_0^2} - \frac{1}{A_L^2} \right) \quad (3.9)$$

A área de passagem A pode ser escrita em função do diâmetro hidráulico D , de forma a obter-se

$$\Delta P_B = \frac{8}{\pi^2} \rho_0 Q^2 \left(\frac{1}{D_0^4} - \frac{1}{D_L^4} \right) \quad (3.10)$$

3.2.1.3 Atrito viscoso

Esse termo acopla os campos energéticos de momento linear e térmico. Sua formulação inicial é descrita por

$$\Delta P_{VW} = \int_0^L \tau_{wx} \frac{P_w}{A} dx \quad (3.11)$$

A tensão de cisalhamento na parede pode ser desenvolvida e a área e o perímetro molhado podem ser escritos em função do diâmetro hidráulico, resultando em

$$\Delta P_{VW} = \int_0^L \frac{1}{8} f \rho_0 \frac{Q |Q|}{A^2} \frac{P_w}{A} dx = \int_0^L \frac{1}{8} f \rho_0 Q |Q| \frac{64}{\pi^2 D^5} dx \quad (3.12)$$

Finalmente, os termos que se mantêm constantes podem ser isolados, obtendo-se

$$\Delta P_{VW} = \frac{8}{\pi^2} \rho_0 Q |Q| \int_0^L \frac{f}{D^5} dx \quad (3.13)$$

3.2.1.4 Inércia hidráulica

$$I = \rho_0 \int_0^L \frac{1}{A} dx \quad (3.14)$$

Novamente, escrevendo a área de passagem em função do diâmetro hidráulico, será obtido

$$I = \frac{4 \rho_0}{\pi} \int_0^L \frac{1}{D^2} dx \quad (3.15)$$

3.2.2 Porto capacitivo

3.2.2.1 Calor trocado com as paredes

$$\underline{W_{QW}} = \int_0^L q_w P_h \underline{\omega_s} dx \quad (3.16)$$

Há duas diferentes condições que podem ser impostas para a troca de calor na parede: com a determinação da temperatura da parede determinada ou do próprio calor trocado. Ambas condições de contorno, no entanto, resultam na mesma causalidade atribuída para o sistema de equações, já que a troca de calor sempre gera entropia, determinando seu valor. A eq. (3.16), no entanto será desenvolvida de forma diferente.

Com calor trocado definido

O termo $\underline{W_{QW}}$ será, então, desenvolvido a:

$$\underline{W_{QW}} = \begin{bmatrix} W_{QW1} \\ \vdots \\ W_{QWl} \\ \vdots \\ W_{QWn} \end{bmatrix} \quad (3.17)$$

$$\text{onde } W_{QW1} = J_1 \left[\left(\frac{3}{8} - \frac{\beta_1}{2} \right) q_{w1} P_{h1} + \left(\frac{1}{8} - \frac{\beta_1}{2} \right) q_{w2} P_{h2} \right],$$

$$\begin{aligned} W_{QWl} = J_l & \left[\left(\frac{1}{8} + \frac{\beta_{l-1}}{2} \right) q_{wl-1} P_{hl-1} + \left(\frac{3}{4} + \frac{\beta_{l-1} - \beta_l}{2} \right) q_{wl} P_{hl} + \right. \\ & \left. + \left(\frac{1}{8} - \frac{\beta_l}{2} \right) q_{wl+1} P_{hl+1} \right] \end{aligned}$$

$$\text{e } W_{QWn} = J_n \left[\left(\frac{1}{8} + \frac{\beta_{n-1}}{2} \right) q_{wn-1} P_{hn-1} + \left(\frac{3}{8} + \frac{\beta_{n-1}}{2} \right) q_{wn} P_{hn} \right].$$

J_l é o jacobiano de transformação da coordenada local para a coordenada global do problema, para o nó l .

Com temperatura da parede definida

O termo $\underline{W_{QW}}$ será, então, reescrito como:

$$\underline{W_{QW}} = \int_0^L -H (\theta_w - \theta) P_h \underline{\omega_s} dx \quad (3.18)$$

Desenvolvendo a equação anterior, obtêm-se:

$$\underline{W_{QW}} = \begin{bmatrix} W_{QW1} \\ \vdots \\ W_{QWl} \\ \vdots \\ W_{QWn} \end{bmatrix} \quad (3.19)$$

onde

$$W_{QW_1} = J_1 \left[\left(\frac{3}{8} - \frac{\beta_1}{2} \right) H_1 P_{h1} (\theta_1 - \theta_{w1}) + \left(\frac{1}{8} - \frac{\beta_1}{2} \right) H_2 P_{h2} (\theta_2 - \theta_{w2}) \right],$$

$$W_{QW_l} = J_l \left[\left(\frac{1}{8} + \frac{\beta_{l-1}}{2} \right) H_{l-1} P_{hl-1} (\theta_{l-1} - \theta_{wl-1}) + \left(\frac{3}{4} + \frac{\beta_{l-1} - \beta_l}{2} \right) H_l P_{hl} (\theta_l - \theta_{wl}) + \left(\frac{1}{8} - \frac{\beta_l}{2} \right) H_{l+1} P_{hl+1} (\theta_{l+1} - \theta_{wl+1}) \right] e$$

$$W_{QW_n} = J_n \left[\left(\frac{1}{8} + \frac{\beta_{n-1}}{2} \right) H_{n-1} P_{hn-1} (\theta_{n-1} - \theta_{wn-1}) + \left(\frac{3}{8} + \frac{\beta_{n-1}}{2} \right) H_n P_{hn} (\theta_n - \theta_{wn}) \right].$$

3.2.2.2 Calor trocado axialmente

$$\underline{W_Q} = \int_0^L A q_x \frac{\partial \omega_s}{\partial x} dx \quad (3.20)$$

Desenvolvendo a área como função do diâmetro hidráulico e o calor trocado axialmente, obtêm-se

$$\underline{W_Q} = \int_0^L \frac{\pi D^2}{4} \left(-\frac{\lambda \theta}{\rho_0 c_v} \frac{\partial s_v}{\partial x} \right) \frac{\partial \omega_s}{\partial x} dx \quad (3.21)$$

Isolando os termos constantes, resulta que

$$\underline{W_Q} = \frac{\pi}{4} \frac{1}{\rho_0} \int_0^L \left(-\frac{\lambda D^2 \theta}{c_v} \right) \frac{\partial s_v}{\partial x} \frac{\partial \omega_s}{\partial x} dx \quad (3.22)$$

Resolvendo a equação para as funções de teste apresentadas e colocando em forma

vetorial, $\underline{W}_Q$ pode ser descrito como

$$\underline{W}_Q = \frac{\pi}{4} \frac{1}{\rho_0} \begin{bmatrix} W_{Q11} & W_{Q12} & 0 & \dots & 0 \\ & \ddots & & & \vdots \\ & & W_{Ql(l-1)} & W_{Qu} & W_{Ql(l+1)} \\ \vdots & & & & \\ 0 & \dots & 0 & W_{Qn(n-1)} & W_{Qnn} \end{bmatrix} \cdot \underline{s}_v \quad (3.23)$$

$$\text{onde } W_{Q11} = - \frac{\lambda_{1+} D_{1+}^2 \theta_{1+}}{c_{v1+}},$$

$$W_{Q12} = \frac{\lambda_{1+} D_{1+}^2 \theta_{1+}}{c_{v1+}},$$

$$W_{Ql(l-1)} = \frac{\lambda_{l-} D_{l-}^2 \theta_{l-}}{c_{vl-}},$$

$$W_{Qu} = - \left(\frac{\lambda_{l-} D_{l-}^2 \theta_{l-}}{c_{vl-}} + \frac{\lambda_{l+} D_{l+}^2 \theta_{l+}}{c_{vl+}} \right),$$

$$W_{Ql(l+1)} = \frac{\lambda_{l+} D_{l+}^2 \theta_{l+}}{c_{vl+}},$$

$$W_{Qn(n-1)} = \frac{\lambda_{n-} D_{n-}^2 \theta_{n-}}{c_{vn-}} e$$

$$W_{Qnn} = - \frac{\lambda_{n-} D_{n-}^2 \theta_{n-}}{c_{vn-}}.$$

onde os termos com índices “-” ou “+” representam as interfaces anterior e posterior, respectivamente, de cada elemento. Dessa forma, as entropias nas interfaces são calculadas como a média aritmética dos valores nodais vizinhos e as temperaturas são calculadas para esse mesmo valor, na forma referida na eq. (3.25), isto é:

$$s_{vl}^- = \frac{s_{vl-1} + s_{vl}}{2} \quad e \quad s_{vl}^+ = \frac{s_{vl} + s_{vl+1}}{2} \quad (3.24)$$

$$\theta_l^- = \theta_R \exp \left(\frac{s_{vl}^-}{\rho_0 c_v} \right) \quad e \quad \theta_l^+ = \theta_R \exp \left(\frac{s_{vl}^+}{\rho_0 c_v} \right) \quad (3.25)$$

As propriedades do fluido são calculadas a partir da temperatura nas interfaces. O diâmetro também é calculado diretamente em cada interface.

3.2.2.3 Calor trocado por efeitos convectivos

$$\underline{W}_C = Q \int_0^L \theta s_v \frac{\partial \omega_s}{\partial x} dx \quad (3.26)$$

Desenvolvendo o termo em forma matricial,

$$\underline{W}_C = Q \begin{bmatrix} W_{C11} & W_{C12} & 0 & \dots & 0 \\ & \ddots & & & \vdots \\ & & W_{Cl(l-1)} & W_{Cll} & W_{Cl(l+1)} \\ \vdots & & & \ddots & \\ 0 & \dots & 0 & W_{Cn(n-1)} & W_{Cnn} \end{bmatrix} \underline{s}_v \quad (3.27)$$

$$\text{onde } W_{C11} = \left(-\beta_1 - \frac{1}{2}\right) \theta_1,$$

$$W_{C12} = \left(\beta_1 - \frac{1}{2}\right) \theta_2,$$

$$W_{Cl(l-1)} = \left(\beta_{l-1} + \frac{1}{2}\right) \theta_{l-1},$$

$$W_{Cll} = \left(-\beta_{l-1} - \beta_l\right) \theta_l,$$

$$W_{Cl(l+1)} = \left(\beta_l - \frac{1}{2}\right) \theta_{l+1},$$

$$W_{Cn(n-1)} = \left(\beta_{n-1} + \frac{1}{2}\right) \theta_{n-1} \text{ e}$$

$$W_{Cnn} = \left(-\beta_{n-1} + \frac{1}{2}\right) \theta_n.$$

3.2.2.4 Calor resultante de geração interna de energia

$$\underline{W}_F = \rho_0 \int_0^L \Phi A \underline{\omega}_s dx \quad (3.28)$$

Reescrevendo a área em função do diâmetro hidráulico, resulta que

$$\underline{W}_F = \frac{\rho_0 \pi}{4} \int_0^L \Phi D^2 \underline{\omega}_s dx \quad (3.29)$$

Desenvolvendo o formalismo de discretização em forma vetorial, obtêm-se

$$\underline{W}_F = \frac{\rho_0 \pi}{4} \begin{bmatrix} W_{F1} \\ \vdots \\ W_{Fl} \\ \vdots \\ W_{Fn} \end{bmatrix} \quad (3.30)$$

$$\text{onde } W_{F1} = J_1 \left[\left(\frac{3}{8} - \frac{\beta_1}{2}\right) \Phi_1 D_1^2 + \left(\frac{1}{8} - \frac{\beta_1}{2}\right) \Phi_2 D_2^2 \right],$$

$$W_{Fl} = J_l \left[\left(\frac{1}{8} + \frac{\beta_{l-1}}{2}\right) \Phi_{l-1} D_{l-1}^2 + \left(\frac{3}{4} + \frac{\beta_{l-1} - \beta_l}{2}\right) \Phi_l D_l^2 + \left(\frac{1}{8} - \frac{\beta_l}{2}\right) \Phi_{l+1} D_{l+1}^2 \right],$$

$$W_{Fn} = J_n \left[\left(\frac{1}{8} + \frac{\beta_{n-1}}{2} \right) \Phi_{n-1} D_{n-1}^2 + \left(\frac{3}{8} + \frac{\beta_{n-1}}{2} \right) \Phi_n D_n^2 \right].$$

3.2.2.5 Calor trocado pela não-homogeneidade nos elementos

$$\underline{W_A} = Q \int_0^L s_v \frac{\partial \theta}{\partial x} \underline{\omega_s} dx \quad (3.31)$$

Desenvolvendo a derivada da temperatura:

$$\underline{W_A} = \frac{Q}{\rho_0} \int_0^L \frac{\theta s_v}{c_v} \frac{\partial s_v}{\partial x} \underline{\omega_s} dx \quad (3.32)$$

Finalmente, o termo $\underline{W_A}$ pode, então, ser colocado em forma matricial

$$\underline{W_A} = \frac{Q}{\rho_0} \begin{bmatrix} W_{A11} & W_{A12} & 0 & \dots & 0 \\ & \ddots & & & \vdots \\ & & W_{Al(l-1)} & W_{All} & W_{Al(l+1)} \\ \vdots & & & & \\ 0 & \dots & 0 & W_{An(n-1)} & W_{Ann} \end{bmatrix} \cdot \underline{s_v} \quad (3.33)$$

$$\text{onde } W_{A11} = -J_1 \left[\left(\frac{1}{2} - \beta_1 \right) \frac{\theta_{1+} s_{v1+}}{c_{v1+}} \right],$$

$$W_{A12} = J_1 \left[\left(\frac{1}{2} - \beta_1 \right) \frac{\theta_{1+} s_{v1+}}{c_{v1+}} \right],$$

$$W_{Al(l-1)} = -J_l \left[\left(\frac{1}{2} + \beta_{l-1} \right) \frac{\theta_{l-} s_{vl-}}{c_{vl-}} \right],$$

$$W_{All} = J_l \left[\left(\frac{1}{2} + \beta_{l-1} \right) \frac{\theta_{l-} s_{vl-}}{c_{vl-}} - \left(\frac{1}{2} - \beta_l \right) \frac{\theta_{l+} s_{vl+}}{c_{vl+}} \right],$$

$$W_{Al(l+1)} = J_l \left[\left(\frac{1}{2} - \beta_l \right) \frac{\theta_{l+} s_{vl+}}{c_{vl+}} \right],$$

$$W_{An(n-1)} = -J_n \left[\left(\frac{1}{2} + \beta_{n-1} \right) \frac{\theta_{n-} s_{vn-}}{c_{vn-}} \right] \text{ e}$$

$$W_{Ann} = J_n \left[\left(\frac{1}{2} + \beta_{n-1} \right) \frac{\theta_{n-} s_{vn-}}{c_{vn-}} \right].$$

3.2.2.6 Dissipação viscosa

$$\underline{W_{VW}} = Q \int_0^L \tau_{wx} \frac{P_w}{A} \underline{\omega_s} dx \quad (3.34)$$

A tensão de cisalhamento na parede pode ser desenvolvida e a área e o perímetro molhado podem ser escritos em função do diâmetro hidráulico, resultando em

$$\underline{W_{VW}} = Q \int_0^L \frac{1}{8} f \rho_0 \frac{Q |Q|}{A^2} \frac{4}{\pi D} \underline{\omega_s} dx \quad (3.35)$$

Os termos que se mantêm constantes ainda podem ser isolados, obtendo-se

$$\underline{W_{VW}} = \frac{8 \rho_0}{\pi^2} Q^2 |Q| \int_0^L \frac{f}{D^5} \underline{\omega_s} dx \quad (3.36)$$

Finalmente, esse termo pode ser desenvolvido através da notação matricial, obtendo-se

$$\underline{W_{VW}} = \frac{8 \rho_0}{\pi^2} Q^2 |Q| \begin{bmatrix} W_{VW1} \\ \vdots \\ W_{VWl} \\ \vdots \\ W_{VWn} \end{bmatrix} \quad (3.37)$$

$$\text{onde } W_{VW1} = J_1 \left[\left(\frac{3}{8} - \frac{\beta_1}{2} \right) \frac{f_1}{D_1^5} + \left(\frac{1}{8} - \frac{\beta_1}{2} \right) \frac{f_2}{D_2^5} \right],$$

$$W_{VWl} = J_l \left[\left(\frac{1}{8} + \frac{\beta_{l-1}}{2} \right) \frac{f_{l-1}}{D_{l-1}^5} + \left(\frac{3}{4} + \frac{\beta_{l-1}-\beta_l}{2} \right) \frac{f_l}{D_l^5} + \left(\frac{1}{8} - \frac{\beta_l}{2} \right) \frac{f_{l+1}}{D_{l+1}^5} \right] \text{ e}$$

$$W_{VWn} = J_n \left[\left(\frac{1}{8} + \frac{\beta_{n-1}}{2} \right) \frac{f_{n-1}}{D_{n-1}^5} + \left(\frac{3}{8} + \frac{\beta_{n-1}}{2} \right) \frac{f_n}{D_n^5} \right].$$

3.2.2.7 Matriz de temperatura

A matriz de temperatura é tal que cada um de seus elementos é descrito pela eq. (3.38).

$$\Theta_{lj} = \frac{1}{\Omega_{Sj}} \int_0^L A \theta \omega_{sl} \varphi_{sj} dx \quad (3.38)$$

Onde Ω é a matriz de volume. Esta matriz é diagonal e é descrita por:

$$\Omega_{Sll} = \int_0^L A \varphi_{sl} dx \quad (3.39)$$

$$\underline{\underline{\Theta}} = \frac{\pi}{4} \begin{bmatrix} \Theta_{11} & \Theta_{12} & 0 & \dots & 0 \\ & \ddots & & & \vdots \\ & & \Theta_{l(l-1)} & \Theta_{ll} & \Theta_{l(l+1)} \\ \vdots & & & & \\ 0 & \dots & 0 & \Theta_{n(n-1)} & \Theta_{nn} \end{bmatrix} \quad (3.40)$$

$$\begin{aligned}
&\text{onde } \Theta_{11} = J_1 \left[\left(\frac{3}{8} - \frac{\beta_1}{2} \right) \frac{D_1^2 \theta_1}{\Omega_{s1}} \right], \\
\Theta_{12} &= J_2 \left[\left(\frac{1}{8} - \frac{\beta_2}{2} \right) \frac{D_2^2 \theta_2}{\Omega_{s2}} \right], \\
\Theta_{l(l-1)} &= J_{l-1} \left[\left(\frac{1}{8} + \frac{\beta_{l-1}}{2} \right) \frac{D_{l-1}^2 \theta_{l-1}}{\Omega_{sl-1}} \right], \\
\Theta_{ll} &= J_l \left[\left(\frac{3}{4} + \frac{\beta_{l-1}-\beta_l}{2} \right) \frac{D_l^2 \theta_l}{\Omega_{sl}} \right], \\
\Theta_{l(l+1)} &= J_{l+1} \left[\left(\frac{1}{8} - \frac{\beta_l}{2} \right) \frac{D_{l+1}^2 \theta_{l+1}}{\Omega_{sl+1}} \right], \\
\Theta_{n(n-1)} &= J_{n-1} \left[\left(\frac{1}{8} + \frac{\beta_{n-1}}{2} \right) \frac{D_{n-1}^2 \theta_{n-1}}{\Omega_{sn-1}} \right] e \\
\Theta_{nn} &= J_n \left[\left(\frac{3}{8} + \frac{\beta_{n-1}}{2} \right) \frac{D_n^2 \theta_n}{\Omega_{sn}} \right].
\end{aligned}$$

3.3 Desenvolvimento dos termos nos contornos

Aqui se faz necessário analisar cada caso em separado.

3.3.1 Para a junção 1

3.3.1.1 Condição integral

Para a condição integral, as condições de contorno impõem pressões para o duto. O grafo de ligação parcial mostrado pela fig. 3.4 mostra a causalidade na junção 1.

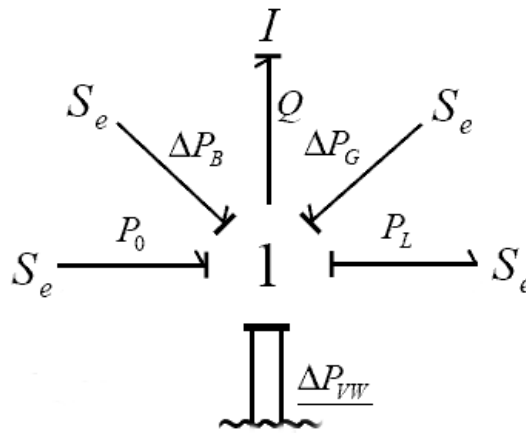


Figura 3.4: Porto de inércia do modelo para escoamento incompressível em duto com causalidade integral atribuída

Dessa forma, P_0 e P_L são dados, e a equação obtida por essa parte do grafo é a apresentada em (3.41). É interessante reparar que essa equação é, essencialmente, idêntica a (2.33).

$$\frac{dQ}{dt} = \frac{P_0 - P_L + \Delta P_G + \Delta P_B - \Delta P_{VW}}{I} \quad (3.41)$$

A vazão, portanto, é configurada por um comportamento dinâmico — que depende de valores em outros instantes temporais.

3.3.1.2 Condição derivativa

Para a condição derivativa, serão impostos a vazão em um contorno e uma pressão de referência em outro. A pressão no contorno que impõe a vazão resultará de uma igualdade algébrica, e portanto será determinada pelo escoamento.

Há duas possibilidades de causalidade derivativa, impondo Q . Para completar as condições de contorno, uma das pressões, P_0 ou P_L , deve ser imposta. Ambas alternativas de atribuição de causalidade nessa parte do grafo são apresentadas na fig. 3.5.

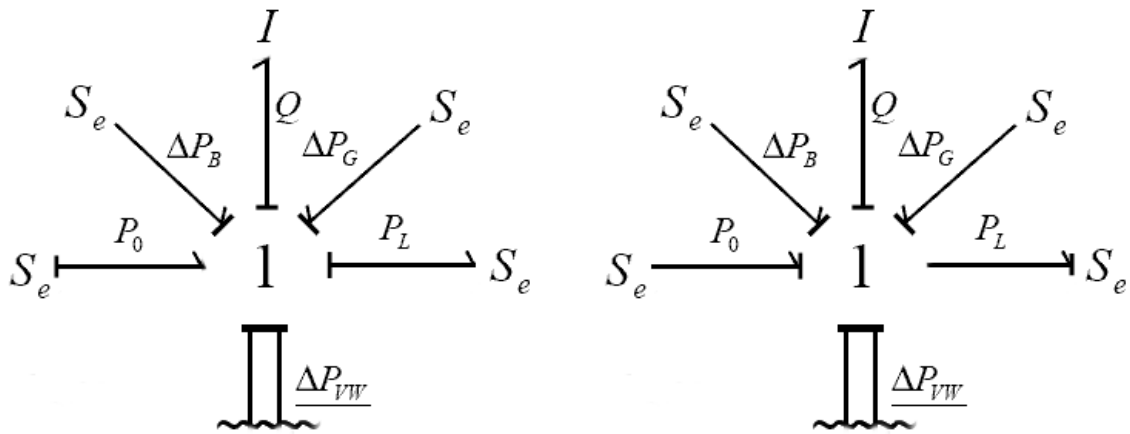


Figura 3.5: Porto de inércia do modelo para escoamento incompressível em duto com as duas possibilidades de causalidades derivativas

A vazão é determinada em um contorno e o porto inercial é, portanto, configurado pelo comportamento de um sistema instantâneo. Em outras palavras, a pressão no contorno em que a vazão é imposta é determinada por uma relação algébrica válida tempo a tempo.

3.3.2 Para a junção 0

3.3.2.1 Condição integral completa

O porto capacitivo em causalidade integral completa tem grafo ilustrado pela fig. 3.6.

Em cada nó, valerá uma equação dinâmica, como apresentado na eq. (2.34).

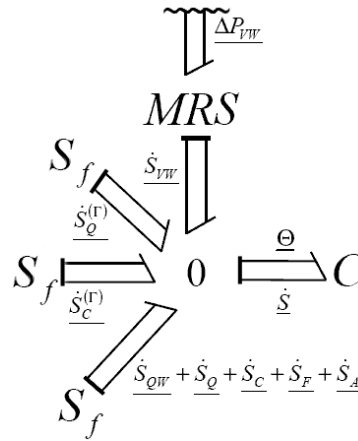


Figura 3.6: Porto de inércia do modelo para escoamento incompressível em duto com causalidade integral atribuída

É essencial, agora, desenvolver mais os termos não nulos no contorno.

Condução

$$\underline{W}_Q^{(\Gamma)} = (q_{xL} \underline{A}_{xL} \underline{\delta}_{ln} - q_{x0} \underline{A}_{x0} \underline{\delta}_{l1}) \quad (3.42)$$

A equação acima pode ser desenvolvida a

$$\underline{W}_Q^{(\Gamma)} = \frac{\pi}{4} (q_{xL} D_L^2 \underline{\delta}_{ln} - q_{x0} D_0^2 \underline{\delta}_{l1}) \quad (3.43)$$

Finalmente, escrevendo a área em função do diâmetro hidráulico e isolando o vetor $\underline{s}_v$, obtêm-se

$$\underline{W}_Q^{(\Gamma)} = \frac{\pi}{4} \begin{bmatrix} W_{Q1}^{(\Gamma)} \\ 0 \\ \vdots \\ W_{Qn}^{(\Gamma)} \end{bmatrix} \cdot \underline{s}_v \quad (3.44)$$

onde $W_{Q1}^{(\Gamma)} = -q_{x0} D_0^2$ e

$W_{Qn}^{(\Gamma)} = q_{xL} D_L^2$. Aqui, q_{x0} e q_{xL} representam a condução axial, a ser determinada para cada caso.

Convecção

$$\underline{W}_C^{(\Gamma)} = Q (\theta_L s_{vL} \underline{\delta}_{ln} - \theta_0 s_{v0} \underline{\delta}_{l1}) \quad (3.45)$$

A equação acima pode ser escrita em forma matricial, isolando $\underline{s}_v$

$$\underline{W}_C^{(\Gamma)} = Q \begin{bmatrix} W_{C11}^{(\Gamma)} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & W_{Cnn}^{(\Gamma)} \end{bmatrix} \cdot \underline{s}_v \quad (3.46)$$

onde $W_{C11}^{(\Gamma)} = -\theta_1$ e $W_{Cnn}^{(\Gamma)} = \theta_n$.

3.3.2.2 Condições mistas

Para as condições mistas no caso em estudo — que tem dois contornos —, um dos nós resulta em causalidade integral e outro em causalidade derivativa.

O nó em causalidade integral será equacionado da forma apresentada anteriormente.

O nó em causalidade derivativa tem um comportamento análogo ao apresentado pelo porto inercial quando em causalidade derivativa, isto é, ele passa a ser um sistema instantâneo.

Faz-se necessário, então, para esse nó, anular a dinâmica do sistema linear, forçando que $\dot{s}_v$ seja nulo e impondo o novo valor de s_v . Isso será apresentado mais claramente nos capítulos 4 e 5.

3.4 Evolução temporal

Para o avanço temporal decidiu-se usar o método preditor-corretor. A seguir será apresentada uma breve introdução teórica ao método.

3.4.1 O método preditor-corretor

Vamos supor, por simplicidade, que temos uma equação da forma:

$$\frac{df}{dt} = g(f) \quad (3.47)$$

Deseja-se obter o valor de uma função f em um tempo $t + \Delta t$, partindo dos valores de $f(t)$ e $g(f)$. O método preditor-corretor é um método de integração numérica em que são efetuadas algumas iterações para obter o valor de $f(t + \Delta t)$ com alguma precisão desejada, resultando finalmente um esquema implícito.

O termo preditor é calculado com valores representativos iterados das variáveis no intervalo de integração, sendo que na primeira iteração é utilizado o valor no tempo t , resultando em um cálculo explícito.

Já o termo corretor depende de um fator de sub-relaxação, ζ , que é um número maior que zero e menor igual à unidade.

Para cada iteração i , o termo preditor pode ser descrito matematicamente como:

$$f_p^i = f(t) + g(f_c^{i-1}) \Delta t \quad (3.48)$$

E o termo corretor, na forma:

$$f_c^i = f_c^{i-1} + \zeta (f_p^i - f_c^{i-1}) \quad (3.49)$$

A inicialização dos termos deve ser executada conforme a eq. (3.50)

$$\begin{aligned} f_p^0 &= f(t) + g(f(t)) \Delta t \\ f_c^0 &= f(t) + \zeta (f_p^0 - f(t)) \end{aligned} \quad (3.50)$$

Deve-se continuar o processo iterativo até que o critério de parada, mostrado na eq. (3.51) seja satisfeito. ε é o erro máximo desejado.

$$\left| \frac{f_p^{i+1} - f_c^i}{f_p^{i+1}} \right| < \varepsilon \quad (3.51)$$

A função no próximo tempo será dita numericamente igual ao último valor calculado para o termo preditor, isto é, assim que o critério de parada for satisfeito,

$$f(t + \Delta t) = f_p^{i+1} \quad (3.52)$$

O método preditor-corretor é útil, em especial, quando a função g é não linear.

3.4.2 O equacionamento para o caso em estudo

Para a porta de pressão, retomando a eq. (2.33), pode-se afirmar que

$$\frac{dQ}{dt} = \frac{\Delta P + \Delta P_G + \Delta P_B - \Delta P_{vw}}{I} = \alpha_1(Q) \quad (3.53)$$

Com uma aproximação linear da derivada, é possível escrever

$$Q^{t+\Delta t} = Q^t + \alpha_1 \Delta t \quad (3.54)$$

Analogamente, para a porta de entropia, têm-se que

$$\underline{S}^{t+\Delta t} = \underline{S}^t + \underline{\alpha}_2 \Delta t \quad (3.55)$$

Para a implementação numérica será utilizado o método preditor-corretor. As equações mostradas a seguir são causais, seguindo a lógica usual de programação. Isto evitará o uso de índices denotando “anterior” ou “próximo”.

Sobre a implementação, cabe ressaltar que os cálculos em ambas as portas devem ser feitos em conjunto, dado o acoplamento bidirecional possível para esse tipo de escoamento. Assim, cada iteração calculará valores para Q e S .

3.4.2.1 Inicialização

A inicialização do método pode, então, ser dada por

$$\begin{aligned} Q_C &= Q^t \\ \underline{S}_C &= \underline{S}^t \\ Q_P &= Q^t + \alpha_1(Q_C) \Delta t \\ \underline{S}_P &= \underline{S}^t + \underline{\alpha}_2(\underline{S}_C) \Delta t \end{aligned} \quad (3.56)$$

3.4.2.2 Iteração

Cada iteração será composta por quatro equações, referentes tanto à vazão quanto à entropia.

$$\begin{aligned} Q_C &= Q_C + \zeta (Q_P - Q_C) \\ \underline{S}_C &= \underline{S}_C + \zeta (\underline{S}_P - \underline{S}_C) \\ Q_P &= Q^t + \alpha_1(Q_C) \Delta t \\ \underline{S}_P &= \underline{S}^t + \underline{\alpha}_2(\underline{S}_C) \Delta t \end{aligned} \quad (3.57)$$

Aqui, ζ é o fator de sub-relaxação.

3.4.2.3 Critério de parada

O critério de parada verifica a flutuação dos valores na atual iteração. Para a vazão, vale

$$\left| \frac{Q_P - Q_C}{Q_P} \right| < \epsilon \quad (3.58)$$

Já para a entropia, o critério deve checar todos os nós l , de forma a respeitar a equação a seguir

$$\left| \frac{S_P^l - S_C^l}{S_P^l} \right| < \epsilon \quad (3.59)$$

3.4.2.4 Finalização

Quando o critério de parada for satisfeito, pode-se atribuir o valor de Q_P a $Q^{t+\Delta t}$ e de S_P a $S^{t+\Delta t}$.

4 IMPLEMENTAÇÃO NUMÉRICA

Para a implementação numérica, inicialmente havia sido escolhido o desenvolvimento com o uso de linguagem C. No entanto, após desenvolvimento parcial, preferiu-se dar sequência ao projeto com o uso do MatLab.

Isso pois o código em MatLab, sendo bem mais compacto e limpo, permite melhores visualização e modularização do código, com o uso de funções. Esses recursos simplificam um pouco o trabalho em um arquivo do porte exigido.

O tratamento vetorial permitido pelo MatLab se aproxima mais da forma humana de pensar. As variáveis também foram agrupadas com o uso de estruturas, de forma a manter juntas variáveis de mesma natureza. Tudo isso facilita a compreensão e manipulação posterior por terceiros do código implementado, abrindo campo para o desenvolvimento do tema.

Não que o MatLab só apresente vantagens. O uso de *loops*, por exemplo, exige um tempo de processamento bem maior do que em outras linguagens. Mesmo tendo em vista que há situações em que *loops* não podem ser evitados, buscou-se evitar o uso de comandos como “*while*” ou “*for*” sempre que possível.

Os métodos numéricos usados foram implementados, apesar de existir a possibilidade de usar comandos prontos. Para a resolução de sistema, por exemplo, foi implementado o algoritmo de Thomas. Tudo isso será apresentado neste capítulo.

Para facilitar a posterior análise de problemas, as funções serão separadas em duas categorias: a de função escritas da mesma forma para qualquer problema, apresentadas na seção 4.1 e as funções que devem ser alteradas de acordo cada problema em especial, introduzidas brevemente na seção 4.2.

4.1 Funções válidas para qualquer aplicação

4.1.1 Funções interpoladoras

Funções auxiliares que fazem interpolações lineares em função de um vetor ou de uma tabela.

Estas funções são usadas para interpolar valores de propriedades do fluido em função da temperatura, as temperaturas da parede e a geração de calor por unidade de massa em função da posição e do tempo, além de condições iniciais em função da posição.

Três funções diferentes foram desenvolvidas, com os nomes “*lin_interp*”, “*v_lin_interp*” e “*m_lin_interp*”.

A primeira interpola um valor em um vetor, resultando em apenas um valor interpolado. A segunda também interpola valores em um vetor, mas tem como saída um vetor de valores interpolados. A terceira delas gera um vetor de valores interpolados em uma tabela.

É interessante notar que, apesar das duas primeiras funções serem casos particulares da última, optou-se por criar três diferentes funções, diminuindo o custo computacional total.

Todas essas funções são apresentadas no apêndice A.

4.1.2 Inversão de matriz

Poderia ter sido optado por usar os comandos prontos de inversão de matrizes do Matlab, mas preferiu-se o desenvolvimento de uma função própria.

Devido às funções de peso selecionadas, as matrizes que compõem os sistemas lineares a serem resolvidos resultam tridiagonais. Dessa forma, pode-se usar um algoritmo com menor custo computacional.

Fletcher (1991) apresenta o algoritmo de Thomas, adequado para o caso em estudo. Esse foi o algoritmo implementado e segue integralmente no apêndice, na seção A.2.

4.1.3 Determinação das propriedades do fluido

Uma função estabelece, a partir das temperaturas nos nós e nas interfaces dos elementos e de uma tabela de propriedades do fluido, as propriedades de interesse para o código.

As propriedades são interpoladas na tabela, de acordo com a temperatura de cada

ponto.

A rotina segue na seção A.3.

4.1.4 Fator de atrito

Nesta função será calculado o fator de atrito de Darcy-Weisbach. O fator de atrito é uma forma de introduzir a dissipação de energia cinética do escoamento no balanço de energia no escoamento. Essa dissipação ocorre devido à viscosidade do fluido, que resulta na condição de não escorregamento.

A descrição precisa desse efeito para o escoamento em dutos deve ser realizada com o uso de coordenadas bidimensionais. No entanto, os valores utilizados para o fator de atrito introduzem a dissipação no cálculo unidimensional de forma exata para um perfil de velocidades completamente desenvolvido.

A região em que o perfil de velocidades ainda não é completamente desenvolvido é chamada de região de entrada hidrodinâmica e, apesar de ter valores maiores para f – uma vez que a acomodação do escoamento gera maiores perdas irreversíveis –, esta não será modelada.

O fator de atrito é definido de diferentes formas, dependendo do número de Reynolds relativo ao diâmetro. De forma que é importante desenvolver sobre cada diferente faixa estabelecida para o cálculo.

De acordo com Munson, Young e Okiishi (2004), o escoamento em um duto pode ser considerado laminar para $Re < 2100$ e turbulento a partir de $Re = 4000$.

Para escoamento turbulento, foi implementado um algoritmo de cálculo iterativo com o método de Newton-Raphson. De forma básica, o método de Newton-Raphson, para cada iteração, lineariza a equação a ser aproximada em torno do valor atual de cálculo e acha o próximo valor de cálculo como a raiz dessa função aproximadora.

A função a ser aproximada, no caso, é a equação de Colebrook-White, mostrada em (4.1).

$$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{2,51}{Re \sqrt{f}} + \frac{\epsilon}{3,71 D} \right) \quad (4.1)$$

Para o escoamento laminar, o fator de atrito foi determinado pela eq. (4.2).

$$f = \frac{64}{Re} \quad (4.2)$$

Uma terceira faixa de divisão no cálculo de f terá de ser estabelecida para baixos valores de Re . É importante lembrar que, como a vazão pode ser nula, é preciso criar uma restrição de valor limite para o fator de atrito. Assim, para $Re \leq 640$, f será fixado em 0, 1.

Finalmente, uma última faixa de cálculo é para o escoamento transicional. Aqui foi efetuada uma interpolação linear em Re entre os valores limites para os escoamentos laminar e turbulento.

O código segue na seção A.4.

4.1.5 Cálculo do número de Nusselt

O número de Nusselt, assim como o fator de atrito, introduz efeitos de gradientes internos à cada seção do duto nos balanços unidimensionais de energia. Aqui, no entanto, não é o gradiente de velocidade que é de interesse, mas sim o gradiente de temperatura e o balanço de energia em estudo é o de energia térmica.

Em essência, tanto a viscosidade quanto a condutividade térmica estão ligadas à hipótese de meio contínuo, sendo resultado das interações moleculares em escala microscópica.

De forma semelhante ao fator de atrito, a região em que o perfil de temperaturas não está plenamente desenvolvido é chamada de região de entrada térmica. O comprimento da região de entrada térmica é determinado em função do comprimento da região de entrada cinemática e do número de Prandtl, razão entre a viscosidade cinemática e a difusividade térmica.

Para problemas reais, normalmente as camadas limites hidrodinâmica e térmica se desenvolvem de forma combinada, isto é, as regiões de entrada se sobrepõem. Se $Pr > 1$, o comprimento da região de entrada térmica é maior que o comprimento da região de entrada cinemática. Se $Pr < 1$, a camada limite térmica se desenvolverá mais rapidamente.

Ainda da mesma forma que para o fator de atrito, na região de entrada as trocas de energia ocorrem de forma mais intensa do que na região de perfil de temperaturas plenamente desenvolvido, acarretando maiores valores locais para o número de Nusselt. Novamente, optou-se por não levar em consideração os comprimentos de entrada, nem o hidrodinâmico nem o térmico.

Correlações teóricas, apresentadas por Kays e Crawford (1993), serão utilizadas para

o cálculo do número de Nusselt. Cada equação calculará o número de Nusselt (Nu) como função dos números de Reynolds (Re) e de Prandtl (Pr). As correlações implementadas são para o caso de temperatura da parede constante.

Para o caso de escoamento laminar, isto é, com $Re \leq 2100$, serão utilizadas correlações que contabilizam a condução axial. As equações são apresentadas em (4.3).

$$Nu = \begin{cases} 4,180654 - 0,183460 Pe & Pe < 1,5 \\ 3,656794 + \frac{4,487}{Pe^2} & Pe > 5 \end{cases} \quad (4.3)$$

Ainda para o caso de escoamento laminar, para a situação em que $1,5 < Pe < 5$ será realizada uma interpolação linear em Pe entre os valores limites das equações acima.

Buscando uma correlação para o escoamento turbulento, foi encontrada uma equação válida não apenas para o escoamento turbulento, como mostrado na eq. (4.4). Esta equação cobre números de Prandtl mais altos.

$$Nu = \frac{(Re-1000) Pr^{\frac{f}{8}}}{1,0+12,7 \sqrt{\frac{f}{8}} (Pr^{\frac{2}{3}}-1,0)} \quad com \quad \begin{matrix} 1 \leq Pr \leq 2000 \\ 2300 \leq Re \leq 5 \cdot 10^6 \end{matrix} \quad (4.4)$$

Já para menores valores de Prandtl e escoamento turbulento, é válida a eq. (4.5).

$$Nu = 4,8 + 0,0156 Re^{0,85} Pr^{0,93} \quad com \quad 0,001 < Pr < 0,1 \quad (4.5)$$

Finalmente, ainda para o escoamento turbulento, foi realizada uma interpolação linear em Pr para o trecho com $0,1 < Pr < 1$.

Para os casos com $2100 < Re < 2300$ seria necessário interpolar valores, entre diferentes equações, para $Pe \leq 1,5$, para $1,5 < Pe < 5$, para $Pe \geq 5$ e também para $Pr \leq 0,1$, para $0,1 < Pr < 1$ e $Pr \geq 1$. O que, em um olhar inicial, sugere que seriam nove regiões com diferentes equações aproximadoras. No entanto, resultam quatro regiões, já que Pe , Re e Pr se relacionam de acordo com a eq. (4.6).

$$Pe = Pr Re \quad (4.6)$$

Essa equação faz com que as regiões de $Pe \leq 1,5$ e $Pr \leq 0,1$, $Pe \leq 1,5$

e $0,1 < Pr < 1$, $Pe \leq 1,5$ e $Pr \geq 1$, $1,5 < Pe < 5$ e $0,1 < Pr < 1$, $1,5 < Pe < 5$ e $Pr \geq 1$ não tenham significado físico.

Dessa forma, foram realizadas interpolações lineares em Re para a região de $Pe \geq 5$ e $Pr \leq 0.1$ e para a região $Pe \geq 5$ e $Pr \geq 1$. Na região de $1.5 < Pe < 5$ e $Pr \leq 0.1$ foi implementada uma aproximação bi-linear em Re e Pe . Para a última região, de $Pe \geq 5$ e $0.1 < Pr < 1$, foi gerada uma aproximação bi-linear em Re e Pr .

A função implementada e mostrada na seção A.5 ainda imprime na tela um aviso, no caso de valores de Re e Pr para os quais as correlações não são consideradas válidas.

4.1.6 Determinação dos parâmetros de *upwind*

Esta função calcula os parâmetros de *upwind* de acordo com a eq. (3.5) e é apresentada na seção A.6.

É interessante ressaltar a importância do uso de um esquema de *upwind*. Cálculos efetuados mostraram que, para o primeiro caso estudado, seria necessário um número de elementos da ordem de centenas de milhões para que se obtivesse estabilidade em um esquema de diferenças centradas.

4.1.7 Função para teste de convergência de malha

Cada problema pode ser resolvido chamando apenas a função principal, apresentada posteriormente. No entanto é necessário estabelecer um conjunto de nós para os quais a equação será discretizada.

A prática comum é fazer um estudo de convergência por refinamento de malha, de forma a estabelecer a malha para a qual a solução será calculada. A resolução correta para a malha é um ponto importante, pois o uma malha pouco refinada pode gerar resultados que não representem a situação física real e uma malha muito refinada pode introduzir erros numéricos.

Com o objetivo de gerar uma boa malha para cada caso – isto é, que gere resultados que independam da malha em si –, foi gerado um algoritmo que refina a malha de forma adaptativa. Esse algoritmo é apresentado na seção A.7.

O algoritmo determina uma malha inicial a partir dos dados de entrada e chama a função principal, que calcula os valores de vazão e temperaturas até o tempo final definido nas planilhas dos dados de entrada. É importante que o tempo final seja grande o suficiente para se obter a solução em regime permanente.

Então, o código dobra o número de elementos e recalcula os valores de interesse, agora para a nova malha.

O próximo passo é testar, para a temperatura, qual a variação relativa de valores obtidos da malha anterior para a nova. É interessante notar que essa comparação não pode ser feita diretamente, pois o número de elementos não é igual. Optou-se por realizar uma interpolação linear para valores obtidos com a malha anterior, mantendo-se os nós da malha nova.

Aqui, o programa entra em um *loop*, do qual ele sai apenas quando a variação relativa para cada elemento é menor que um valor determinado.

Dentro desse *loop*, o programa compara a variação relativa para o perfil de temperaturas em regime permanente com um valor de precisão determinado inicialmente. Optou-se por refinar a malha dobrando o número de pontos e mantendo espaçamento constante na malha.

Essa opção se deu principalmente por dois motivos. Em primeiro lugar, o jacobiano da transformação de coordenadas locais para as globais seria não-linear. Em segundo lugar, o tipo de escoamento em estudo não seria tão fortemente beneficiado por um refinamento de malha com espaçamento não constante – com exceção do caso de um tubo muito longo, em que o gradiente de temperatura ao longo do duto variasse mais.

Dessa forma, apesar de ter sido implementado um equacionamento geral, como desenvolvido no capítulo 3, as malhas utilizadas serão de espaçamento constante entre nós.

4.2 Funções a serem alteradas conforme aplicação

Aqui são apresentadas as estruturas principais das funções que terão de ser modificadas de acordo com cada aplicação. Isso pois, por mais que as funções seja alteradas de aplicação para aplicação, a forma geral é a mesma.

No entanto, as funções serão apresentadas integralmente oportunamente, de forma a incluir as particularidades.

4.2.1 Função principal

Este é o arquivo ponto de partida da implementação. Funções, explicadas nas seções subsequentes serão chamadas pelo código contido nesse arquivo. Apesar do nome inicialmente atribuído, “main”, esta função deve ser chamada por outra, que fornece o conjunto de pontos sobre os quais será calculada a solução do sistema de equações.

A estrutura deste arquivo é composta por:

- Inicialização
 - Inicialização específica para cada problema **
 - Acesso a arquivos externos
 - Inicialização de variáveis gerais
 - Determinação do tamanho do primeiro passo de tempo
- Para cada tempo
 - Inicialização de variáveis temporais
 - Inicialização do método preditor-corretor
 - Enquanto o critério de parada não for satisfeito
 - ◇ Iteração do método preditor-corretor
 - ◇ Visualização do resíduo máximo *
 - Finalização do método preditor-corretor
 - Visualização do resíduo máximo *
 - Atualização temporal de variáveis específicas ao problema **
 - Correção do passo de tempo
- Para o tempo final
 - Inicialização de variáveis temporais
 - Inicialização do método preditor-corretor
 - Enquanto o critério de parada não for satisfeito
 - ◇ Iteração do método preditor-corretor
 - Finalização do método preditor-corretor
 - Atualização temporal de variáveis específicas ao problema **
- Plotagem de gráficos *
 - Vídeo do perfil de temperaturas, ao longo do tempo *

Os itens marcados com um asterisco, *, são opcionais. Já os marcados com dois asteriscos, **, são próprios ao problema a ser resolvido.

Aqui cabe dissertar sobre a escolha do passo temporal. O passo de tempo foi escolhido com o uso do número de Courant-Friedrichs-Lewy, definido pela eq. (4.7). Optou-se fixar o número adimensional em 0,5. Vale lembrar que a condição para estabilidade em um método explícito é de $C < 1$.

$$C = \frac{u \Delta t}{h} \quad (4.7)$$

No código gerado, o próximo passo de tempo é calculado após a determinação da vazão no passo atual.

O passo de tempo inicial é determinado da mesma forma se a vazão é não-nula. Caso contrário, é fixado um valor arbitrário para o passo de tempo inicial.

4.2.2 Diâmetros do duto

Chamada de “*duct_shape*”, esta função recebe um vetor de posições e calcula, para cada uma delas, o diâmetro do duto.

Essa função é chamada pela função principal, na inicialização de variáveis gerais, para calcular o diâmetro nos nós e nas interfaces. O código implementado vai variar para cada problema.

4.2.3 Condições de fronteira

Nomeada “*boundary_1*”, esta função calcula condições de fronteira para o porto de inércia do sistema.

4.2.4 Predição

Nesta função são calculadas as propriedades do fluido e do escoamento e finalmente são efetuados cálculos para predição das variáveis. Esta função devolve os valores preditos para o programa base, onde serão efetuados os passos do método preditor-corretor.

A estrutura deste trecho de código é mostrada a seguir:

- Inicialização

- Cálculo de propriedades
 - Volume dos elementos
 - Temperaturas
 - Propriedades do fluido: condutividade, calores específicos, viscosidade e número de Prandtl
 - Números de Reynolds e de Péclet
 - Fator de atrito de Darcy-Weisbach
 - Coeficiente de transferência por convecção
- Cálculos das funções de peso
- Avaliação dos termos no interior do domínio
- Avaliação dos termos nas fronteiras - causalidades integrais
- Montagem das matrizes, do sistema linear
- Implementação de causalidades derivativas **
- Passo preditivo

Uma possível alternativa é, em vez de calcular o número de Prandtl, incluir sua obtenção na função que determina as propriedades do fluido, referida na seção 4.1.3. Essa alternativa, no entanto, está condicionada à inserção na tabela de propriedades do fluido de uma nova coluna, com valores do número de Prandtl.

Novamente, o item marcado com dois asteriscos, **, será escrito em formas diferentes, de acordo com a aplicação.

4.3 Dados de entrada

Um arquivo separado alimenta o programa com os dados de interesse. Optou-se por usar um arquivo no formato “xls”, apenas para facilitar a visualização e operabilidade para outras pessoas que, eventualmente, se interessem pelo tema.

Este arquivo deve ser estabelecido para cada problema em especial, introduzindo condições geométricas e térmicas, constantes gerais e outros escalares relacionados à discretização

espacial, ao método de cálculo de evolução temporal, aos limites de tempo para a resposta a ser obtida. Além disso, uma tabela com propriedades do fluido em questão também é adicionada.

Finalmente, também são colocados dados que definem as condições iniciais, as condições da parede e as condições de geração interna de calor, para qualquer posição e espaço de tempo. Essas condições são chamadas, de forma geral, de condições auxiliares. A alimentação contínua desses dados é uma das coisas que garante que o problema é bem posto matematicamente.

O uso de grafos de ligação garante que cada parte do contorno tenha uma variável imposta, de forma a não resultar em um problema com condições de contorno sub- ou sobredefinidas.

Em adição, para se obter um problema bem posto, a solução deve existir e ser única. De acordo com Fletcher (1991), a existência da solução não costuma ser um problema.

Sobre a unicidade, Fletcher afirma que uma razão comum para a não-unicidade é o fracasso ao combinar as condições auxiliares com o tipo de equação diferencial parcial governante.

A unicidade da solução é usualmente de difícil verificação, tendo em vista que as equações são não lineares. Mais do que isso, algumas escoamentos podem ter mais de uma solução fisicamente possível. De acordo com Fletcher, uma situação em que isso ocorre é o escoamento transicional. O autor afirma que cabe, então, à pessoa que cria o modelo gerar uma formulação matemática que tenha uma única solução.

Essas planilhas serão apresentadas para cada aplicação, nos próximos capítulos.

5 ESTUDO DE CASO: ESCOAMENTO DE ÁGUA EM REGIME DE AQUECIMENTO

Este capítulo apresentará um primeiro caso estudado. A solução em regime permanente será comparada com a solução analítica. Finalmente, um problema de esvaziamento de um tanque é apresentado.

5.1 O problema a ser estudado

O primeiro caso estudado é de um escoamento de água em regime de aquecimento.

5.1.1 O modelo físico

Para descrever matematicamente o fenômeno é necessário ter um modelo físico. O fenômeno a ser estudado é modelado de acordo com a fig. 5.1, descrita nesta seção.

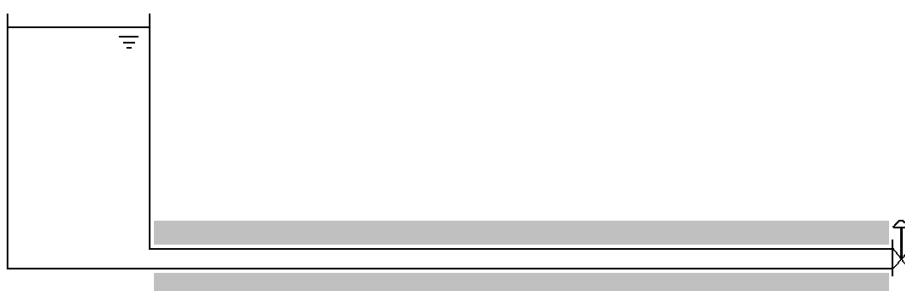


Figura 5.1: Modelo físico do primeiro problema a ser estudado

Um tanque armazena água. A sua jusante encontra-se um duto de paredes rígidas e com uma temperatura fixa. Ainda mais a jusante, há uma válvula e, finalmente, uma abertura à atmosfera.

5.1.1.1 Hipóteses

As hipóteses aplicadas são:

- Pressões impostas em ambas extremidades do duto, a serem calculadas de acordo com a geometria
- Diâmetro do duto constante
- Variação de cota nula para o duto
- Temperatura do tanque imposta e constante, em 25° C
- O fluxo de calor por condução axial na saída do duto é nulo
- Temperatura da parede do duto é imposta e constante, em 80° C
- Taxa de geração de calor por unidade de volume mantida nula

A função que mantém o diâmetro do duto constante é determinada pela função nomeada “duct_shape”, apresentada na seção B.1.2 do apêndice B.

Dessas hipóteses segue que a pressão no contorno onde o fluido entra é calculada pela eq. (5.1).

$$P_0 = P_{atm} + \rho g H - \frac{\rho}{2} \left(\frac{4Q}{\pi D_1^2} \right)^2 - K_e \frac{\rho}{2} \left(\frac{4Q}{\pi D_1^2} \right)^2 \quad (5.1)$$

Para o contorno em que o fluido sai, a pressão é calculada pela eq. (5.2).

$$P_L = P_{atm} + K_v \frac{\rho}{2} \left(\frac{4Q}{\pi D_1^2} \right)^2 \quad (5.2)$$

onde:

- à pressão atmosférica é atribuído o valor de $1,013 \cdot 10^5 Pa$. Esse valor foi extraído de Munson, Young e Okiishi (2004)
- a aceleração da gravidade é fixada em $9,807 \frac{m}{s^2}$
- o coeficiente K_e , que modela perda de pressão de estagnação devido à mudança de seção no escoamento na entrada, é fixado em 0,5, valor válido para estricção com cantos vivos, de acordo com Munson, Young e Okiishi (2004)
- o coeficiente K_v , modela a perda de pressão de estagnação na válvula na saída do duto. Para este caso, modelou-se uma válvula esfera totalmente aberta, que resulta em $K_v = 0,05$, de acordo com a mesma referência

- o duto tem diâmetro de 50 mm , comprimento de 20 m e rugosidade equivalente de $0,26\text{ mm}$, que de acordo com Munson, Young e Okiishi (2004), equivale a um tubo novo de ferro fundido
- o tanque a montante do escoamento tem uma altura inicial de 5 m de água e um diâmetro constante de 1 m

5.1.2 Tabela de propriedades do fluido

As propriedades termofísicas do fluido de trabalho, água, foram retiradas de Lide (2007).

É preciso dizer que os valores de calor específico a volume constante foram replicados a partir do calor específico a pressão constante. Isso pois, para um fluido idealmente incompressível – modelo que aproxima bem o comportamento da água –, ambos os calores específicos são idênticos.

Além disso, é preciso justificar a decisão por não usar um único calor específico, já que o outro seria igual para o caso de fluido incompressível. Procurou manter-se a generalidade do código, uma vez que o modelo de escoamento incompressível não exige que o fluido seja incompressível.

5.1.3 Sobre o método numérico

Alguns valores atribuídos são relativos ao método numérico aplicado. A precisão dos cálculos em geral é fixada em 10^{-6} . A exceção é para a precisão imposta para o processo convergência por refinamento de malha, que foi fixada em 10^{-3} .

O fator de sub-relaxação para o método preditor corretor é de $\zeta = 0,5$.

A simulação será realizada a partir de um instante chamado de $t = 0\text{ s}$ – em que a condição de válvula aberta e o perfil de temperatura são impostos de forma equivalente a excitar o sistema com entradas do tipo degrau – até outro momento, com $t = 18\text{ s}$ – em que já é obtida uma resposta em regime permanente.

Como o caso é de vazão inicial nula, será necessário fixar um valor para o passo de tempo inicial, como explicado na seção 4.2.1. Assim o primeiro passo temporal é fixado em $0,2\text{ s}$.

Um outro dado numérico de entrada que é relevante é a temperatura de referência,

usada para o cálculo da entropia específica. Essa será fixada a **273, 15 K**.

Esse conjunto de dados auxiliares, que determinam o modelo físico, é informado ao código através de um conjunto de tabelas, mostradas na seção B.1.1.

Para essas condições, começando com uma nodalização de $n = 51$ elementos, o programa determina que devem ser usados no mínimo 201 elementos para obter a precisão desejada para a convergência de malhas. Com esse número de nós, o código implementado gera os resultados que seguem nas próximas seções.

5.2 Comparação com solução analítica

Para efeitos de verificação do método numérico implementado, será efetuada uma comparação com o resultado analítico para o caso em estudo. O resultado analítico que será utilizado é o apresentado por Incorpora e DeWitt (2003).

É importante ressaltar que a obtenção do resultado analítico é condicionada à imposição de mais três hipóteses: de condução de calor axial nula entre os elementos, de calor específico constante ao longo do escoamento e de vazão mássica constante.

Para a comparação ser coerente, essas hipóteses foram também impostas no programa numérico. O calor específico foi obtido para a temperatura que caracteriza o escoamento, determinada pela média logarítmica da diferença de temperaturas, isto é, para a temperatura dada pela eq. (5.3).

$$T = T_{parede} - \frac{(T_{parede} - T_L) - (T_{parede} - T_0)}{\ln \left(\frac{T_{parede} - T_L}{T_{parede} - T_0} \right)} \quad (5.3)$$

Na verdade, essas duas hipóteses não causam praticamente nenhuma alteração no resultado final do perfil de temperatura numérico em regime permanente para o caso em estudo. Isso ocorre visto que o escoamento apresenta um alto número de Péclet – ou seja, a convecção é predominante – e uma pequena variação nos valores de calor específico – variação máxima de 0, 43 % na faixa de temperaturas.

O perfil de temperaturas obtido analiticamente pode ser descrito conforme a eq. (5.4).

$$T(x) = T_{parede} + (T_{tanque} - T_{parede}) \exp \left(-\frac{P_h \bar{h}_x}{\rho Q c_p} x \right) \quad (5.4)$$

Os dois perfis de temperatura para o regime permanente, o analítico e o obtido numericamente, podem ser comparados na fig. 5.2.

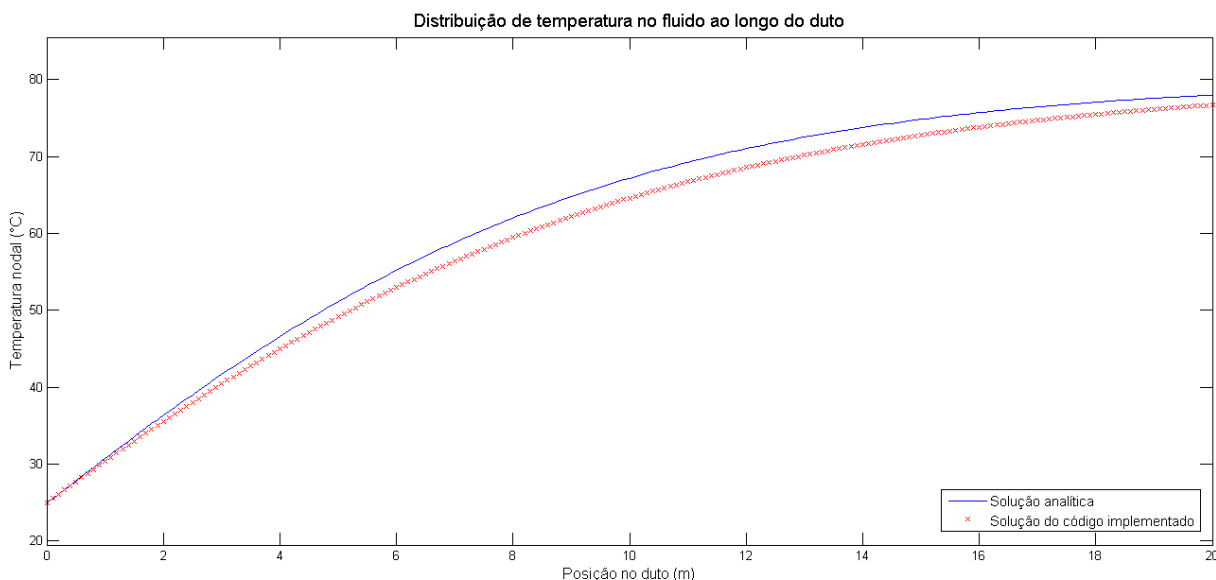


Figura 5.2: Comparação entre soluções analítica e numérica

5.3 Pressão variável na entrada

O mesmo caso será agora desenvolvido para o caso de pressão variável na entrada, na forma de esvaziamento do tanque.

A simulação permite também a análise do regime transiente, permitindo a visualização do avanço de uma frente de onda fria ao longo do duto, com o passar do tempo. A fig. 5.3 ilustra exatamente esse avanço.

Para cada tempo, três gráficos são plotados. Descrevendo-os, partindo do maior em sentido anti-horário: o perfil de temperaturas *versus* a posição no duto, a vazão volumétrica *versus* o tempo de simulação e a altura no tanque *versus* o tempo de simulação.

É interessante também notar na fig. 5.4, que um máximo para a vazão volumétrica é obtido por volta de $t = 11s$. Isso também pode ser observado com a quantidade de iterações que o programa tem de realizar para determinar o valor das variáveis, nulo – isto é, o código não chega a entrar no *loop* que calcula um passo de correção e outro de predição para cada iteração.

O código implementado também gera um vídeo, com frames dos gráficos apresentados para cada tempo. Uma alteração importante de ser executada antes de gerar o vídeo é mudar o

critério para o passo temporal, que não deve mais depender do número de Courant, mas sim ser constante, de forma a gerar um vídeo em que o tempo avance de forma constante.

O vídeo gerado é apresentado em um CD, em anexo. Cada 2,5 segundos de vídeo equivalem a aproximadamente um segundo para o problema real.

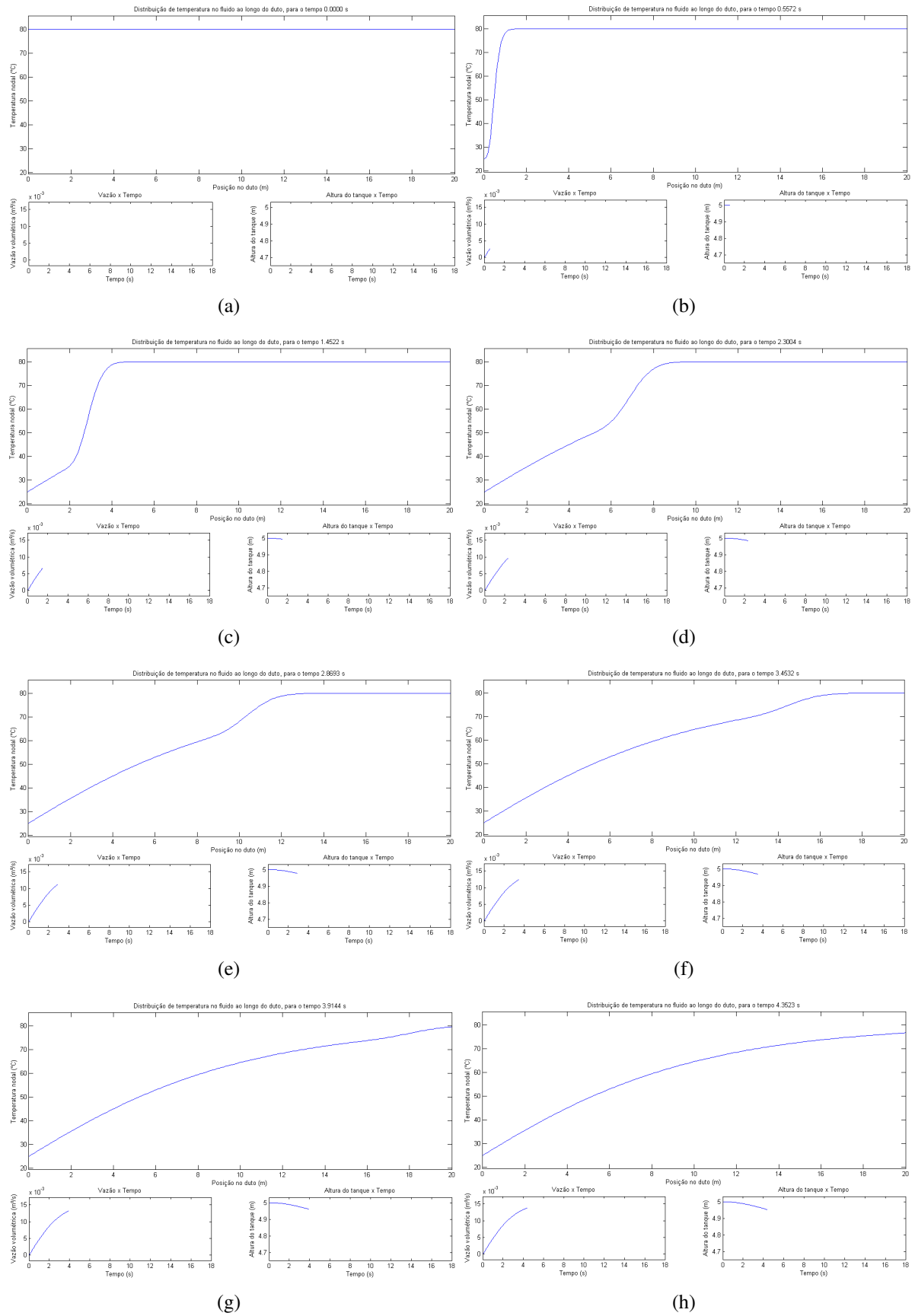
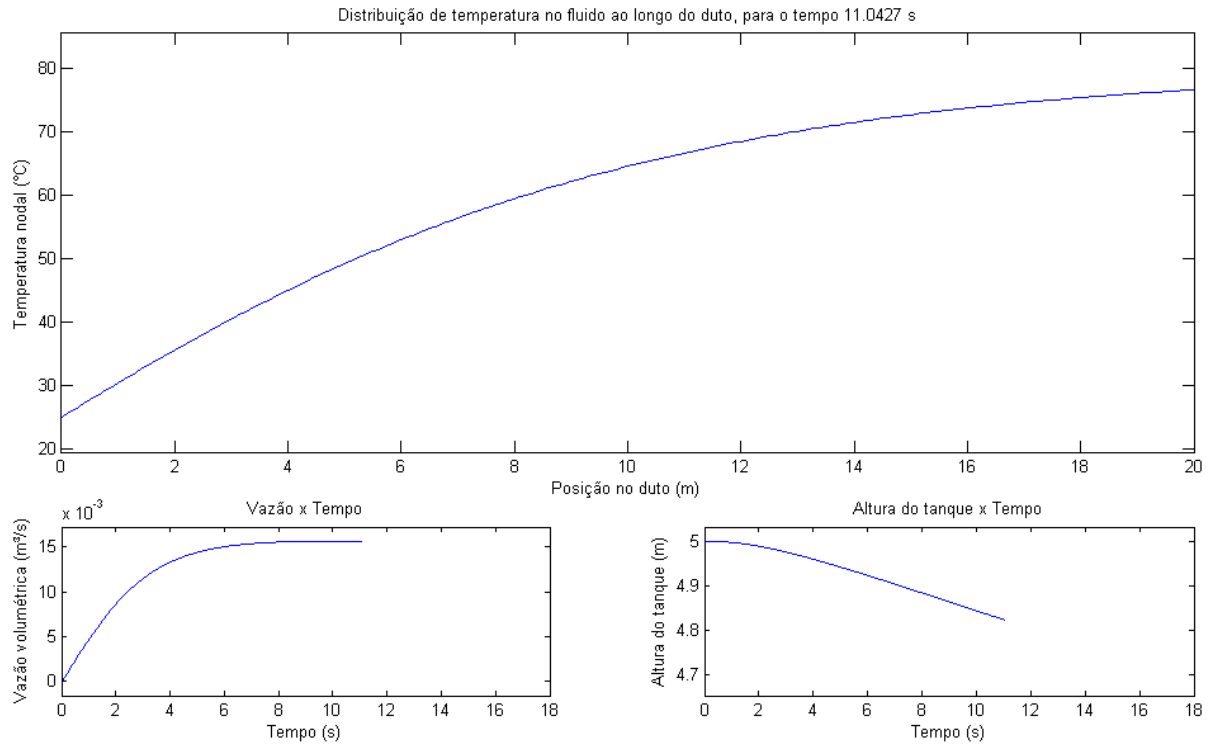
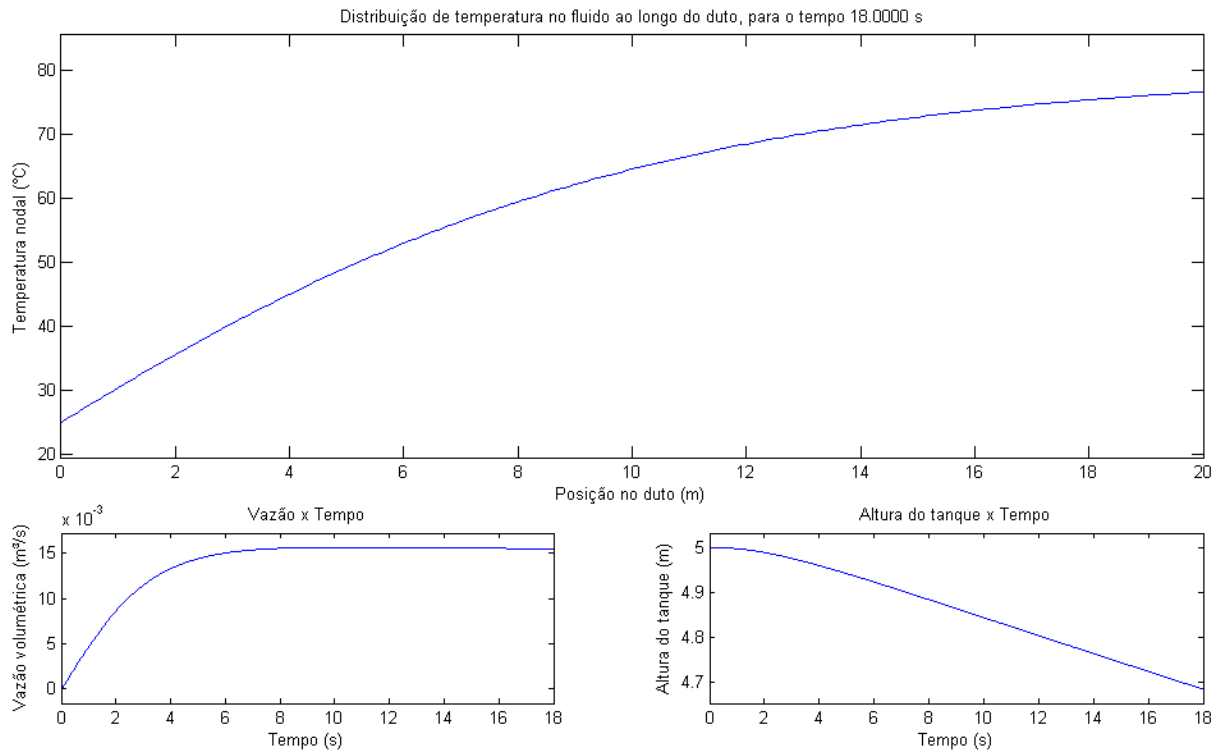


Figura 5.3: Gráficos para o caso em estudo, para os tempos: (a) $t = 0,00$ s (b) $t = 0,56$ s (c) $t = 1,45$ s (d) $t = 2,30$ s (e) $t = 2,87$ s (f) $t = 3,45$ s (g) $t = 3,91$ s (h) $t = 4,35$ s



(a)



(b)

Figura 5.4: Gráficos para o caso em estudo, para os tempos: (i) $t = 11,04$ s (j) $t = 18,00$ s

6 ESTUDO DE CASO: ESCOAMENTO DE ÓLEO EM REGIME DE AQUECIMENTO

Este capítulo apresentará o estudo de escoamento de óleo para motor em um conduto. A intenção é mostrar a importância que pode ter um método numérico que considere a variação da viscosidade dinâmica com a temperatura.

6.1 O problema a ser estudado

O caso a ser estudado nesse capítulo é de óleo de motor em regime de aquecimento.

6.1.1 O modelo físico

O modelo físico é bastante similar ao modelo físico criado para o caso estudado no capítulo anterior. É interessante ressaltar que poderia-se o modelo apresentado anteriormente resulta completamente análogo a um modelo que fixe as pressões na entrada e na saída do duto.

Para esse capítulo, no entanto, há algumas modificações. Em primeiro lugar, o fluido de trabalho é um óleo virgem para motor. À jusante do duto há um reservatório, com pressão atmosférica. Há também modificações de valores para a geometria e para as condições térmicas.

É importante notar que, apesar de o modelo sugerir uma situação que não é obtida com frequência na engenharia, o equacionamento obtido descreve perfeitamente um outro modelo com dois reservatórios, um a montante e outro a jusante, com pressões definidas. Isso será melhor desenvolvido quando forem atribuídos valores numéricos ao modelo.

6.1.1.1 Hipóteses

As hipóteses aplicadas são:

- Pressões impostas em ambas extremidades do duto, a serem calculadas de acordo com a

geometria

- Diâmetro do duto constante
- Variação de cota nula para o duto
- Temperatura do tanque imposta
- O fluxo de calor por condução axial na saída do duto é nulo
- Temperatura da parede do duto é imposta e constante, em 140°C
- Taxa de geração de calor por unidade de volume mantida nula

Dessas hipóteses segue que as pressões na entrada e na saída do duto serão calculadas de forma análoga à apresentada no capítulo anterior.

Alguns valores substituídos são diferentes. Assim:

onde:

- a aceleração da gravidade e a pressão atmosférica são as mesmas apresentadas no capítulo 5, isto é, $9,807 \frac{\text{m}}{\text{s}^2}$ e $1,013 \cdot 10^5 \text{ Pa}$, respectivamente
- o coeficiente K_e também tem o mesmo valor do caso estudado anteriormente, de 0,5
- na saída do duto, o coeficiente K_v , modela a perda de pressão de estagnação na saída do duto. Para este caso, modelou-se a descarga em um tanque, com cantos vivos, resultando em $K_v = 1,0$, de acordo com Munson, Young e Okiishi (2004)
- o duto tem diâmetro de 50 mm , comprimento de 10 m e rugosidade equivalente de $0,26 \text{ mm}$, que de acordo com Munson, Young e Okiishi (2004), equivale a um tubo novo de ferro fundido
- o diâmetro do tanque é de 1 m . Essa informação, no entanto, não será utilizada, uma vez que não será estudado o esvaziamento do tanque
- o tanque a montante do escoamento tem uma altura inicial de $0,5 \text{ m}$ de óleo

A altura de $0,5 \text{ m}$ de óleo equivale a uma pressão na entrada do duto, ignorando momentaneamente os efeitos dinâmicos, $4,2 \cdot 10^3 \text{ Pa}$ maior do que na saída. Como qualquer referência pode ser atribuída para a pressão, o caso simulado poderia também representar um sistema pressurizado, com uma pressão na entrada de aproximadamente $1,056 \cdot 10^5 \text{ Pa}$ e de $1,014 \cdot 10^5 \text{ Pa}$ na saída.

6.1.2 Tabela de propriedades do fluido

As propriedades termofísicas do fluido de trabalho, óleo virgem para motor, foram retiradas de Incropera e DeWitt (2003). Não foram encontradas tabelas de propriedades termofísicas para óleos com a nomenclatura especificada.

De forma análoga ao capítulo anterior e pelos mesmos motivos, os valores de calor específico a volume constante foram replicados a partir do calor específico a pressão constante.

6.1.3 Sobre o método numérico

A precisão dos cálculos em geral é fixada em 10^{-6} . A exceção é para a precisão imposta para o processo convergência por refinamento de malha, fixada em 10^{-4} .

O fator de sub-relaxação para o método preditor corretor é de $\zeta = 0,3$. O valor de sub-relaxação é diminuído com relação ao caso anterior, já que as não-linearidades são mais acentuadas – a viscosidade do óleo varia em ordens de grandeza na faixa de temperatura em estudo.

De forma análoga ao caso anterior, a simulação será realizada a partir de um instante chamado de $t = 0\text{ s}$ – em que todas as condições são impostas de forma equivalente a excitar o sistema com entradas do tipo degrau – até outro momento, com $t = 150\text{ s}$ – em que já é obtida uma resposta em regime permanente.

Também de forma similar ao capítulo anterior, uma vez que o sistema tem vazão inicial nula, será necessário fixar um valor para o passo de tempo inicial. Portanto, o primeiro passo temporal é fixado em $0,6\text{ s}$.

A temperatura de referência, usada para o cálculo da entropia específica, será fixada a $293,15\text{ K}$.

Esse conjunto de dados auxiliares, que determinam o modelo físico, é informado ao código através de um conjunto de tabelas, mostradas na seção B.2.1.

Com essas condições, e partindo de uma nodalização inicial de $n = 6$ elementos, o programa determina que devem ser usados no mínimo 41 elementos para obter a precisão determinada, de 10^{-4} . Com essa quantidade de elementos, são gerados os resultados que serão apresentados nas próximas seções.

A fig. 6.1 mostra um gráfico com os resíduos entre uma malha e outra, com o dobro de elementos. É interessante notar que, conforme o número de nós é aumentado, é cada vez mais

difícil obter menores resíduos.

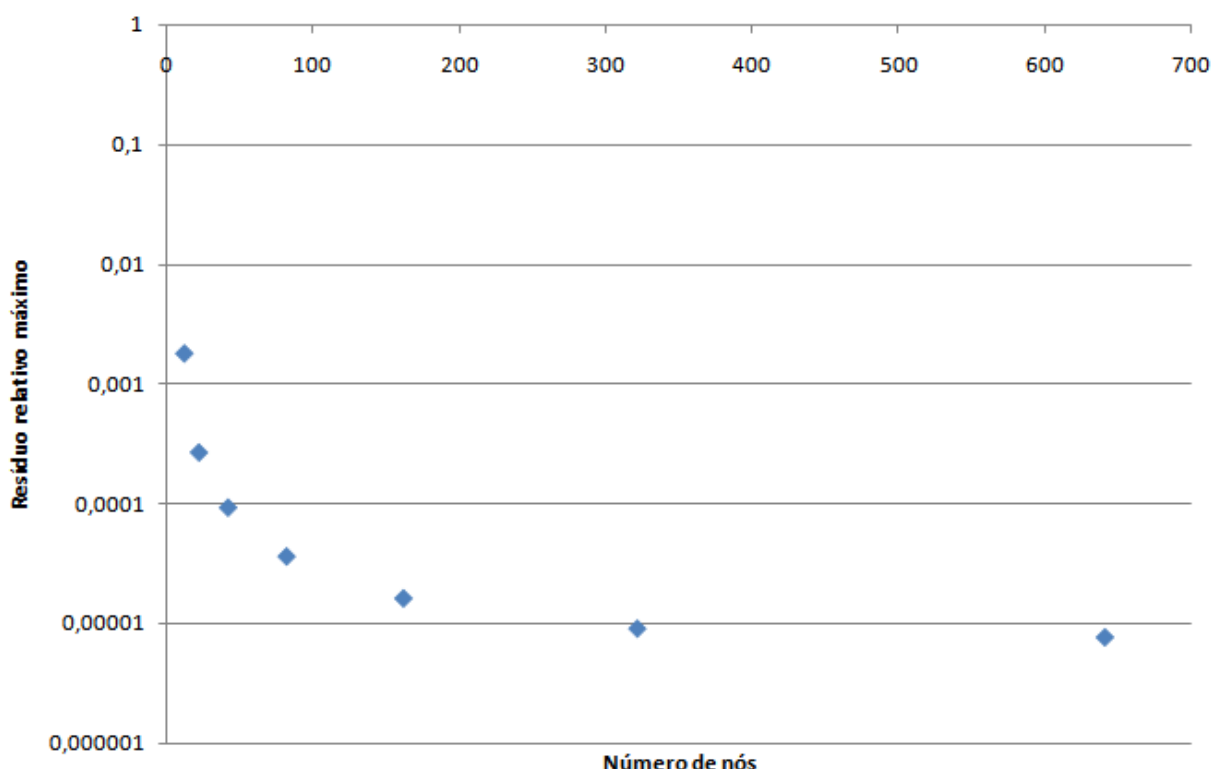


Figura 6.1: Máximo resíduo no processo de refinamento de malha *versus* número de nós

6.2 Verificação da conservação de energia

Antes de proceder com a exposição de resultados, será apresentado outro ponto de verificação do código gerado. Uma vez que o método dos elementos finitos não garante a conservação de energia por si só, é importante checar se o balanço de energia é respeitado.

Na verdade, o método dos elementos finitos assegura o balanço de energia térmica de forma fraca, isto é, que o balanço de energia será melhor respeitado quanto maior for o número de nós.

Um dos inconvenientes é que, com o aumento da nodalização, passa a ser necessário o uso de muito mais recursos computacionais. A fig. 6.2 ilustra isso, mostrando, em azul, a verificação do balanço de energia no regime permanente e, em vermelho, uma indicação do custo computacional.

O custo computacional descrito na figura é o tempo decorrido para efetuar os cálculos, em segundos. A abscissa do gráfico mostra o logaritmo do número de nós na base 2. Isso mostra-se conveniente, uma vez que a quantidade de nós é dobrada a cada iteração.

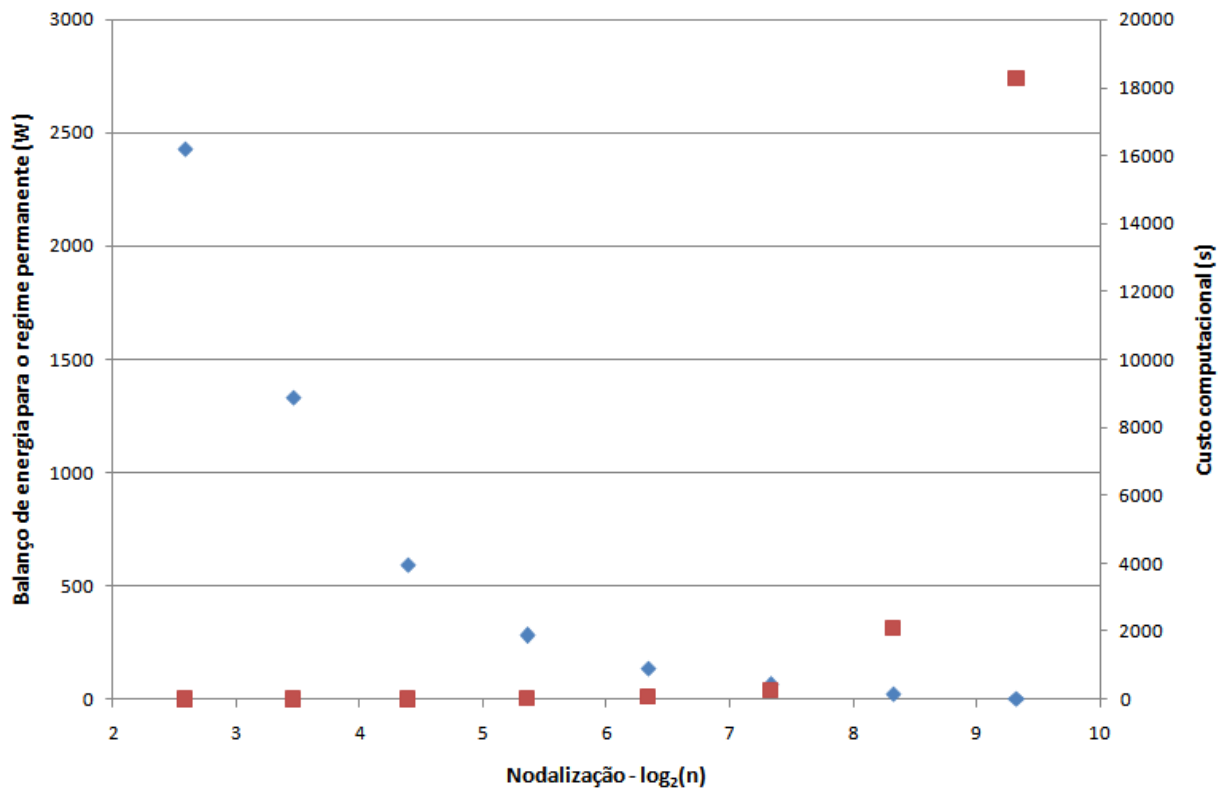


Figura 6.2: Balanço de energia e custo computacional com o refinamento de malha

O balanço de energia foi realizado em regime permanente, e a fig. 6.1 ilustra os valores absolutos em módulo. O balanço de energia é realizado conforme a eq. (6.1). Como a análise é realizada em regime permanente, o termo sobre a variação da energia armazenada é nulo.

$$W_{entra} + W_{gerada} - W_{sai} \quad (6.1)$$

Para o sistema físico, no meio contínuo, o resultado da eq. (6.1) deve ser nulo. Para o método numérico implementado, tendo em vista que a potência gerada é da ordem de $17,5 \text{ kW}$, o uso de 41 nós gera um erro de menos de 2 %.

6.3 Com temperatura de entrada constante

Inicialmente serão apresentados resultados de uma simulação que, de forma similar ao caso anterior, tem uma temperatura de entrada constante, agora fixada em 353 K .

O perfil de temperaturas para o regime permanente é mostrado na fig. 6.3.

Uma característica central do escoamento em estudo é descrita pelos baixos valores para o número de Reynolds – variando de 0, no início, ao máximo de 2678, no regime perma-

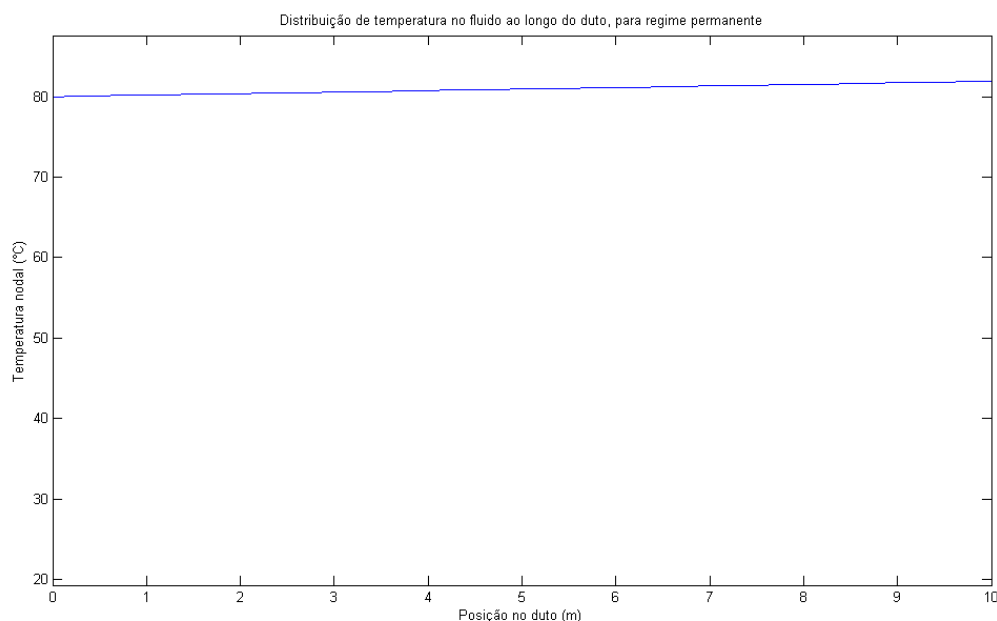


Figura 6.3: Perfil de temperaturas no regime permanente para escoamento de óleo em aquecimento

nente –, que caracterizam o escoamento em regime permanente como transicional.

Apesar de o escoamento ser considerado transicional – podendo, portanto ter diferentes soluções possíveis –, Kays e Crawford (1993) sugere uma correlação para a obtenção do número de Nusselt que abrange a faixa de Reynolds que caracteriza o escoamento em estudo no regime permanente.

Com o uso dessa correlação, conforme explicitado na seção 4.1.5, obtém-se um coeficiente de convecção máximo de $186,9 \frac{W}{m^2 K}$, que acarreta em uma menor troca de calor com as paredes do duto.

Ainda para esse caso, a evolução temporal da vazão é mostrada na fig. 6.4 e tem um comportamento similar ao do sistema do capítulo anterior, com água.

6.4 Com temperatura de entrada variável

Tendo em vista que o fluido em estudo tem uma forte dependência da viscosidade dinâmica com a temperatura, passa a ser de interesse a análise sobre como a variação da temperatura média do escoamento influi no desenvolvimento da vazão para o sistema.

Para efetuar essa análise, foi desenvolvida outra rotina, esta com o objetivo de permitir a variação temporal da condição de contorno térmica na entrada do duto. Esta função é

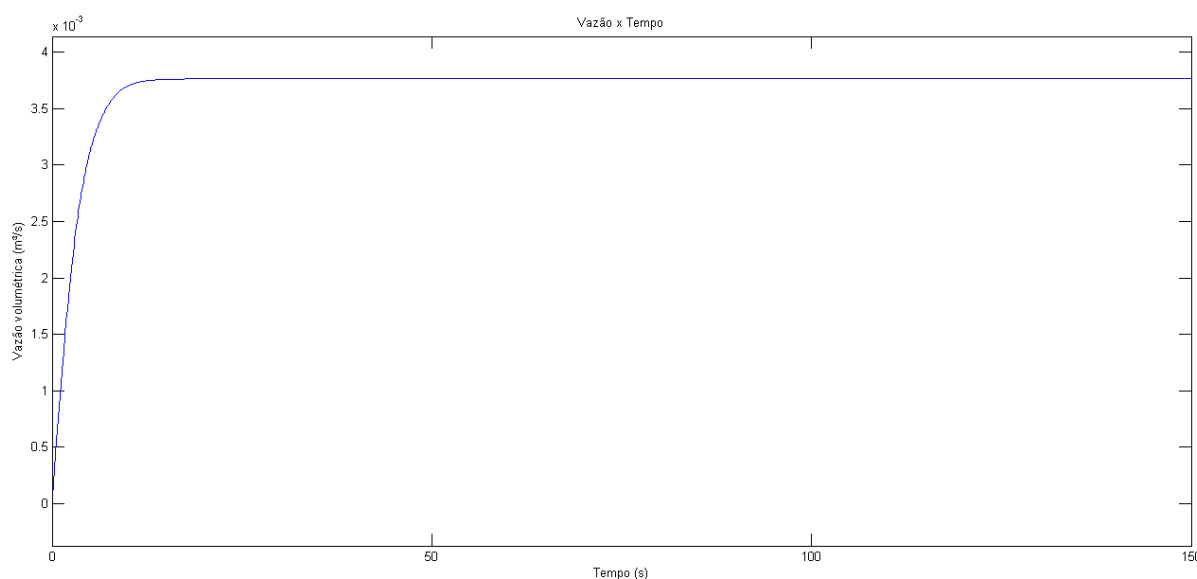


Figura 6.4: Vazão volumétrica *versus* tempo, para escoamento de óleo com temperatura de entrada constante

apresentada na seção B.2.6.

A função deve atribuir uma forma para a variação da temperatura de entrada. Na verdade, para que a análise de interesse pudesse ser realizada, bastaria que a temperatura de entrada continuasse a variar após o tempo necessário para que a vazão volumétrica atinja seu valor de regime permanente. Mais do que isso, a forma atribuída pode ser qualquer e independe do modelo desenvolvido, para análise do escoamento no duto.

Decidiu-se implementar o comportamento de um sistema de primeira ordem. Para isso, a função interpola valores em uma tabela, com valores que caracterizam a resposta de um sistema de primeira ordem a uma entrada do tipo degrau, com uma constante de tempo $k = 20$ s. As temperaturas de entrada resultantes são mostradas na fig. 6.5.

O gráfico que representa a evolução temporal da vazão volumétrica é mostrado na fig. 6.6.

Para essa evolução temporal da temperatura de entrada, a vazão volumétrica continuará sendo alterada, até que o perfil de temperaturas atinja o regime permanente. Isso é uma consequência direta do fato de o fenômeno de dissipação viscosa ser acoplado pela viscosidade dinâmica, que é modulada pela temperatura.

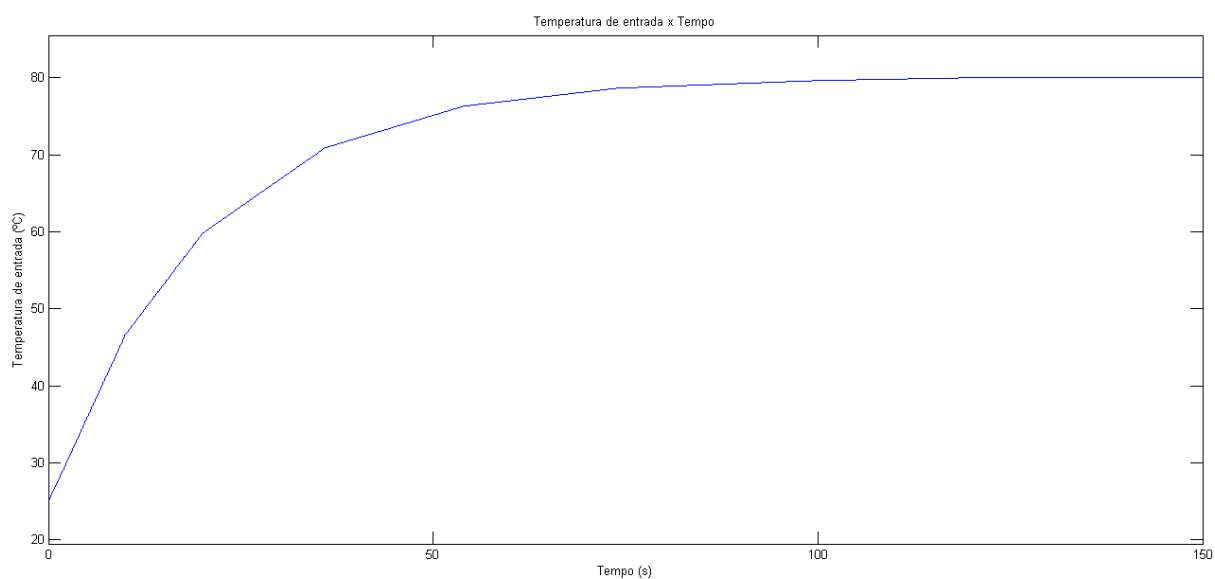


Figura 6.5: Evolução da temperatura de entrada ao longo do tempo, para a aplicação em especial

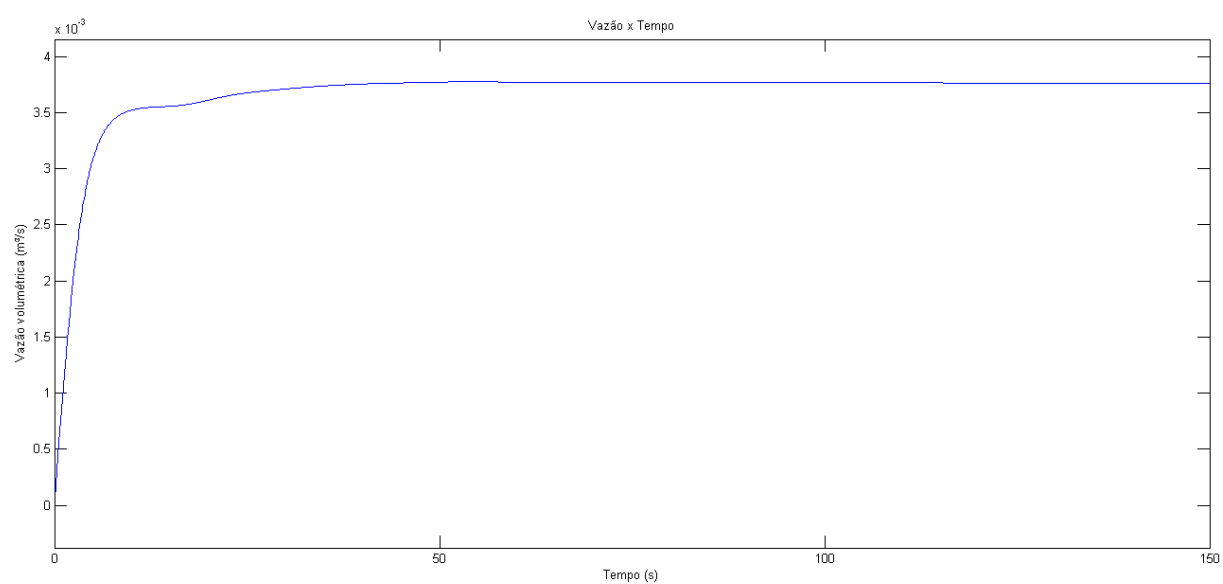


Figura 6.6: Evolução da vazão volumétrica ao longo do tempo, para escoamento de óleo com temperatura de entrada variante ao longo do tempo

7 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um código numérico que simula o comportamento do escoamento de um fluido incompressível em um duto de paredes rígidas, desde uma leitura de seu ferramental teórico – recente – até a aplicação em casos particulares.

Estudos de convergência foram apresentados e os resultados gerados são coerentes com a realidade. Dessa forma, acredita-se que o objetivo inicial principal foi atingido.

Um desenvolvimento interessante para o trabalho pode ser na direção de otimizar o código numérico implementado. Imagina-se que a implementação do equacionamento com o uso de uma linguagem orientada a objetos, usando cada elemento do método de elementos finitos como um objeto, possa gerar um código ainda mais rápido e versátil.

Espera-se que o trabalho motive o desenvolvimento de outros modelos para sistemas de parâmetros distribuídos, especialmente modelos que considerem fenômenos acoplados, como por exemplo interações fluido-estrutura ou escoamentos magneto-hidrodinâmicos.

REFERÊNCIAS

- BALIÑO, J. L. Bond graph modeling of incompressible flows. In: *17th. IMACS World Congress, Scientific Computation, Applied Mathematics and Simulation (IMACS 2005)*. Paris: [s.n.], 2005. p. 8.
- BALIÑO, J. L. Modeling one-dimensional incompressible duct flows. In: *European Conference on Modelling and Simulation, 20th*. Sankt-Augustin: Fachhochschule Bonn-Rhein-Sieg, 2006. p. 657–662.
- BALIÑO, J. L. Galerkin finite element method for incompressible thermofluid flows framed within the bond graph theory. *Simulation Modelling Practice and Theory*, v. 17, n. 1, p. 35–49, Janeiro 2009.
- BORUTZKY, W. *Bond Graphs: A methodology for modelling multidisciplinary dynamic systems*. Erlangen: SCS, 2004. 440 p.
- CELLIER, F. E. *Modelado Orientado a Objetos de Sistemas Físicos*. 2008. Internet. Disponível em: <http://www.inf.ethz.ch/personal/fcellier/Lect/MMPS/Ppt/mmeps_ppt_cast.html>.
- CORDARO, P. D.; KAWANO, A. *O Delta de Dirac: uma introdução à teoria das distribuições para a Engenharia*. São Paulo: Livraria da Física, 2002. 73 p.
- FLETCHER, C. A. J. *Computational Techniques for Fluid Dynamics 1: Fundamental and General Techniques*. 2nd. ed. Berlin: Springer-Verlag, 1991. 401 p.
- HUNTER, P.; PULLAN, A. *FEM/BEM Notes*. 2003. Internet. Disponível em: <<http://www.cs.rutgers.edu/~suejung/fembemnotes.pdf>>.
- INCORPERA, F. P.; DEWITT, D. P. *Fundamentos de Transferência de Calor e de Massa*. 5ª. ed. Rio de Janeiro: LTC, 2003. 698 p.
- KARNOPP, C. R.; MARGOLIS, D. L.; ROSENBERG, R. C. *System Dynamics: Modeling and Simulation of Mechatronic Systems*. 3rd. ed. New York: Wiley-Interscience, 2000. 507 p.
- KAYS, W. M.; CRAWFORD, M. E. *Convective Heat and Mass Transfer*. 3rd. ed. New York: McGraw-Hill, 1993. 601 p.
- LIDE, D. R. (Ed.). *CRC Handbook of Chemistry and Physics*. 87th. ed. [S.l.]: CRC, 2007. Internet version.
- MALISKA, C. R. *Transferência de calor e mecânica dos fluidos computacional*. 2ª. ed. Rio de Janeiro: LTC, 2004. 453 p.
- MUNSON, B. R.; YOUNG, D. F.; OKIISHI, T. H. *Fundamentos da Mecânica dos Fluidos*. São Paulo: Edgard Blücher, 2004. 572 p.

NORRIE, D. H.; VRIES, G. D. *The Finite Element Method: fundamentals and applications*. New York: Academic Press, 1973. 322 p.

PIMENTA, M. M. *Introdução à Dinâmica dos Gases: PMC-480 Termodinâmica do Escoamento Compressível*. 1981.

RASO, E. F. G. *Desarrollo de una metodología basada en los bond graphs para problemas de fluidodinámica computacional*. Tese (Doutorado) — Universidad Nacional de Cuyo, abril 2006.

SOBRINHO, A. S. C. *Introdução ao Método dos Elementos Finitos*. Rio de Janeiro: Ciência Moderna, 2006. 403 p.

WHITE, F. M. *Mecânica dos fluidos*. 4^a. ed. Rio de Janeiro: McGraw-Hill, 2002. 570 p.

APÊNDICE A – CÓDIGO NO MATLAB PARA FUNÇÕES GERAIS

Neste apêndice apresentam-se o código das funções que são iguais para todos os problemas estudados. Será seguida a ordem apresentada no capítulo 4.

A.1 Interpolações lineares

A.1.1 “*lin_interp*”

```
function y = lin_interp ( t, table )

aux = sum ( t >= table(1, :) );

y = ( (table(1, aux+1) - t) * table(2:end, aux) + ...
      (t - table(1, aux)) * table(2:end, aux+1) ) ...
    / (table(1, aux+1) - table(1, aux));
```

A.1.2 “*v_lin_interp*”

```
function y = v_lin_interp ( t, table )

y = zeros(size(t,1), size(table,2)-1);

for i = 1:size(t,1)
    aux = sum ( t(i) >= table(:, 1) );

    y(i,:) = ( (table(aux+1, 1) - t(i)) * table(aux, 2:end) + ...
               (t(i) - table(aux, 1)) * table(aux+1, 2:end) ) ...
              / (table(aux+1, 1) - table(aux, 1));
end
```

A.1.3 “*m_lin_interp*”

```
function y = m_lin_interp ( x, t, table )

y1 = zeros(size(x));
y2 = zeros(size(x));

aux_t = sum ( t >= table(1, 2:end) ) + 1;

for i = 1:size(x,1)
    aux_x = sum ( x(i) >= table(2:end, 1) ) + 1;

    y1(i) = ((table(aux_x+1,1) - x(i))*table(aux_x,aux_t+1) + ...
             (x(i) - table(aux_x,1))*table(aux_x+1,aux_t+1)) ...
            / (table(aux_x+1,1)-table(aux_x,1));
    y2(i) = ((table(aux_x+1,1) - x(i))*table(aux_x,aux_t) + ...
             (x(i) - table(aux_x,1))*table(aux_x+1,aux_t)) ...
            / (table(aux_x+1,1)-table(aux_x,1));
end

y = (y1*(t-table(1,aux_t)) + y2*(table(1,aux_t+1)-t)) ...
    / (table(1,aux_t+1)-table(1,aux_t));
```

A.2 Inversão de matriz

```
function y = thomas ( Ab )

Ab(1,:) = Ab(1,:) / Ab(1,1);

for i = 2:size(Ab,1)
    Ab(i,end) = Ab(i,end) - Ab(i,i-1) * Ab(i-1,end);
    Ab(i,:) = Ab(i,:) / ( Ab(i,i) - Ab(i,i-1)*Ab(i-1,i) );
end

y = Ab(:,end);

for i = (size(Ab,1)-1):-1:1
    y(i) = y(i) - y(i+1) * Ab(i,i+1);
end
```

A.3 Determinação das propriedades do fluido

```
function [ lambda_n, c_v_n, c_p_n, mu_n, lambda_intf, c_v_intf, ...
         c_p_intf ] = fluid_properties ( theta_node, theta_interface, ...
         properties )

aux1 = v_lin_interp ( theta_node, [properties(:,1) properties(:,3:6)] );
% aux1 = v_lin_interp ( theta_node, [properties(:,1) properties(:,3:7)] );
aux2 = v_lin_interp ( theta_interface, [properties(:,1) properties(:,3:5)]);

lambda_n = aux1 (:, 1);
c_v_n = aux1 (:, 2);
c_p_n = aux1 (:, 3);
mu_n = aux1 (:, 4);
% Pr = aux1(:,5);

lambda_intf = aux2 (:, 1);
c_v_intf = aux2 (:, 2);
c_p_intf = aux2 (:, 3);
```

A.4 Fator de atrito

```
function f = friction_factor ( abs_rug, D, Re, eps )
%% Correção, para o caso numérico de Q = Re = 0 e Q < 0
Re(isinf(1./Re)) = 1e-9;
Re = abs(Re);

%% "Switches" lógicos
a = Re < 640;
b = Re <= 2100;
c = Re >= 4000;
d = ~b&~c;

%% Variável geral
aux = abs_rug ./ (3.7 .* D);

%% Fator de atrito para escoamento transicional
% Interpola linearmente f entre valores numéricos limites de comportamento
% laminar e turbulento
a_4000 = 6.742 * ones(size(D)); % Chute inicial padrão: f = 0.022
```

```

func = d;
while sum( abs(func) < eps ) ~= size(D)
    func = a_4000 + 2 .* d .* log10 (aux + 2.51 .* a_4000 ./ 4000);
    delfunc = 1 + 5.02 * d ./
        ( 4000 .* log(10) .* (aux + 2.51.*a_4000 ./ 4000 ) );

    a_prox = a_4000 - func ./ delfunc;
    a_4000 = a_prox;
end

f_4000 = 1./ a_4000.^2;
f_4000(isinf(f_4000)) = 0;

%% Fator de atrito para escoamento turbulento
% Equação de Colebrook-White, válida apenas para Re >~ 4000
a_ant = 6.742 * c; % Chute inicial padrão: f = 0.022

func = c;
while sum( abs(func) < eps ) ~= size(D)
    func = a_ant + 2 * c .* log10 (aux + 2.51 .* a_ant ./ Re);
    delfunc = 1 + 5.02 * c ./
        ( Re .* log(10) .* (aux + 2.51 .* a_ant ./ Re ) );

    a_prox = a_ant - func ./ delfunc;
    a_ant = a_prox;
end

f = 1./ a_ant.^2;
f(isinf(f)) = 0;

%% Definição híbrida para f
f = 0.1*a + 64./Re.*(b&~a) +
    d .* ((Re - 2100)./(1900) .* (f_4000 - 64/2100) + 64/2100) + f;

```

A.5 Cálculo do número de Nusselt

```

function Nu = Nu ( Re, Pr, Pe, f)
Nu = zeros(size(Re));

%% Correção, para Re e Pe < 0
Re = abs(Re);

```

```

Pe = abs(Pe);

%% Switches lógicos
aux_Re1 = Re <= 2100;
aux_Re2 = Re >= 2300;
aux_Re3 = Re >= 5e6;

aux_Pe1 = Pe <= 1.5;
aux_Pe2 = Pe >= 5;

aux_Pr1 = Pr <= 0.1;
aux_Pr2 = Pr >= 1;
aux_Pr3 = Pr >= 2000;
aux_Pr4 = Pr <= 1e-3;

%% Avisos críticos ao usuário
if sum(aux_Pr3&aux_Re2) ~= 0
    disp('Não tenho correlação válida para este alto número de Pr')
end
if sum(aux_Pr4) ~= 0
    disp('Não tenho correlação válida para este baixo número de Pr')
end
if sum(aux_Re3) ~= 0
    disp('Não tenho correlação válida para este alto número de Re')
end

%% Todas as definições parciais, válidas p/ temperatura de parede constante

% Laminar, Pe <= 1.5
aux1 = aux_Re1&aux_Pe1;
Nu(aux1) = 4.180654 - 0.183460*Pe(aux1);

% Laminar, Pe >= 5
aux2 = aux_Re1&aux_Pe2;
Nu(aux2) = 3.656794 + 4.487./Pe(aux2).^2;

% Laminar, 1.5 < Pe < 5, aproximação linear em Pe
aux3 = aux_Re1&~(aux_Pe1|aux_Pe2);
Nu(aux3) = ( (5-Pe(aux3))*3.905464 + (Pe(aux3)-1.5)*3.836274 ) ./ 3.5;

% Turbulento, Pr <= 0.1
aux4 = aux_Re2&aux_Pr1;
Nu(aux4) = 4.8 + 0.0156 * Re(aux4).^0.85 .* Pr(aux4).^0.93;

```

```

% Turbulento,  $Re < 5 \cdot 10^6$  e  $Pr \geq 1$ 
aux5 = aux_Re2&aux_Pr2;
Nu(aux5) = ((Re(aux5)-1000) .* Pr(aux5) .* f(aux5)/8) ./
            (1 + 12.7 * (f(aux5)/8).^0.5 .* (Pr(aux5).^(2/3)-1));

% Turbulento,  $Re < 5 \cdot 10^6$  e  $0.1 < Pr < 1$ , aproximação linear em Pr
aux6 = aux_Re2&~(aux_Pr1|aux_Pr2);
Nu(aux6) = ( (0.5-Pr(aux6)).*(4.8+0.00183284*Re(aux6).^0.85) +
            (Pr(aux6)-0.1).*((Re(aux6)-1000) .* 0.5 .* f(aux6)/8) ./
            (1 + 12.7 * (f(aux6)/8).^0.5 .* (-0.370039)) ) ./ 0.4;

% Transicional,  $1.5 < Pe < 5$  e  $Pr \leq 0.1$ , aproximação bi-linear em Re e Pe
aux7 = ~(aux_Re1|aux_Re2)&~aux_Pe2;
Nu(aux7) = ( (2300-Re(aux7)).*
            ( (5-Pe(aux7))*3.905464 + (Pe(aux7)-1.5)*3.836274 ) ./ 3.5 +
            (Re(aux7)-2100).*((1300*Pr(aux7).*f(aux7)/8) ./
            (1+12.7*(f(aux7)/8).^0.5 .* (Pr(aux7).^(2/3)-1))) ) / 200;

% Transicional,  $Pe \geq 5$  e  $Pr \leq 0.1$ , aproximação linear em Re
aux8 = ~(aux_Re1|aux_Re2)&aux_Pe2&aux_Pr1;
Nu(aux8) = ( (2300-Re(aux8)).*(3.656794 + 4.487./Pe(aux8).^2) +
            (Re(aux8)-2100).*(4.8 + 11.235522 * Pr(aux8).^0.93) ) / 200;

% Transicional,  $Pe \geq 5$  e  $0.1 < Pr < 1$ , aproximação bi-linear em Re e Pr
aux9 = ~(aux_Re1|aux_Re2)&aux_Pe2&~(aux_Pr1|aux_Pr2);
Nu(aux9) = ( (2300-Re(aux9)).*(3.656794 + 4.487./Pe(aux9).^2) +
            (Re(aux9)-2100).*(( (1-Pr(aux9)).*(6.120059) + (Pr(aux9)-0.1) .*
            ( 1300 .* f(aux9)/8 ) ) ) ./ 0.9 ) ) / 200;

% Transicional,  $Pe \geq 5$  e  $Pr \geq 1$ , aproximação linear em Re
aux10 = ~(aux_Re1|aux_Re2)&aux_Pe2&aux_Pr2;
Nu(aux10) = ( (2300-Re(aux10)).*(3.656794 + 4.487./Pe(aux10).^2) +
            (Re(aux10)-2100).*(( 1300 .* Pr(aux10) .* f(aux10)/8) ./
            (1 + 12.7 * (f(aux10)/8).^0.5 .* (Pr(aux10).^(2/3)-1)) ) ) / 200;

```

A.6 Determinação dos parâmetros de *upwind*

```
function beta = upwind (Pe)
```

```
beta = sign(Pe) .* Pe.^2 ./ ( 6.226 + 4.878*abs(Pe) + 2*Pe.^2 );
```

A.7 Teste de convergência de malha

```

clear all
close all
clc

xlsconstants = xlsread('Duto.xls', 'Constants', 'C3:C11');
xlsduct = xlsread('Duto.xls', 'Duct');

eps = xlsconstants(6);

%% Faz cálculos com a malha imposta no arquivo de entrada
L = xlsduct(1);
n = xlsconstants(1);

disp(['Estou começando os cálculos com n = ', num2str(n)])

J = L / (n-1) * ones(n,1);

nodes = zeros(n,1);
for i = 2:n
    nodes(i,1) = nodes(i-1,1) + J(i-1,1);
end

interf = nodes(1:n-1,1) + J(1:n-1,1)/2;

[THETA_ant, T_ant, Q_ant, H_ant] = main (n, J, nodes, interf);

nodes_ant = nodes;

%% Dobra o número de pontos e refaz os cálculos
n = 2*n-1;

disp(' ')
disp(['Estou começando os cálculos com n = ', num2str(n)])

J = L / (n-1) * ones(n,1);

nodes = zeros(n,1);
for i = 2:n
    nodes(i,1) = nodes(i-1,1) + J(i-1,1);

```

```

end

interf = nodes(1:n-1,1) + J(1:n-1,1)/2;

[THETA_p, T_p, Q_p, H_p] = main (n, J, nodes, interf);

cor_THETA_ant = v_lin_interp (nodes, ...
    [nodes_ant THETA_ant(:,end); nodes_ant(end,1)+1 THETA_ant(end,end)]);
test_mc = abs((cor_THETA_ant(:,end)-THETA_p(:,end))./THETA_p(:,end));

nodes_ant = nodes;

disp([' O máximo resíduo entre essa malha e a anterior é de ', ...
    num2str(max(test_mc))])

res(1) = (log10(max(test_mc)/eps));
j = 1;
figure(1)
plot(j, res(j),'x')

%% Refina a malha e recalcula valores
while(sum(test_mc > eps)~= 0)
    n = 2*n-1;

    disp(' ')
    disp(['Estou começando os cálculos com n = ', num2str(n)])

    J = L / (n-1) * ones(n,1);

    THETA_ant = THETA_p;

    nodes = zeros(n,1);
    for i = 2:n
        nodes(i,1) = nodes(i-1,1) + J(i-1,1);
    end

    interf = nodes(1:n-1,1) + J(1:n-1,1)/2;

    [THETA_p, T_p, Q_p, H_p] = main (n, J, nodes, interf);

    cor_THETA_ant = v_lin_interp (nodes, [nodes_ant THETA_ant(:,end); ...

```

```

        nodes_ant(end,1)+1 THETA_ant(end,end)]);
test_mc = abs((cor_THETA_ant(:,end)-THETA_p(:,end)) ./
    THETA_p(:,end));

disp([' O máximo resíduo entre essa malha e a anterior é de ',
    num2str(max(test_mc))])

nodes_ant = nodes;

j = j+1;
res(j) = (log10(max(test_mc)/eps));
plot(j, res(j),'x')
end

disp(['Devem ser usados ', num2str(n), ' elementos'])

figure('color',[1 1 1])
plot(1:j, res(1:j),'ro')
title('Teste de convergência por refinamento de malha')
xlabel ('Iteração')
ylabel ('log10( |resíduo| / eps )')

```

APÊNDICE B – CÓDIGO NO MATLAB PARA FUNÇÕES ESPECÍFICAS

B.1 Primeiro caso estudado – Água em aquecimento

Nesta seção serão apresentadas as tabelas de dados auxiliares e as funções implementadas para o primeiro caso estudado, apresentado no capítulo 5.

B.1.1 Tabelas com dados auxiliares

Tabela “Duct”

L	20	m
Z	0	m
rug. eq.	0,00026	m

Tabela “Specific”

K_{valv}	0,05	-
K_{estr}	0,5	-
Nível inicial	5	m
Diâmetro do tanque	1	m
Temperatura do tanque	298	K

Tabela “Initial”

Posição (m) - apenas para θ	θ (K)	$Q \left(\frac{m^3}{s} \right)$
	0	353
	20	353
	20,2	353

Tabela “Constants”

Discretização	n	51	-
Constantes gerais	g	9,807	$\frac{m}{s^2}$
	P_{atm}	1,01E+05	Pa
	θ_{ref}	273,15	K
	precisão	1,00E-06	-
	precisão converg. de malha	1,00E-03	-
Tempo	começo	0	s
	fim	18	s
Preditor-Corretor	ζ	0,5	-

Tabela “Fluid_properties”

Temperatura (K)	$\rho \left(\frac{kg}{m^3} \right)$	$\lambda \left(\frac{W}{K m} \right)$	$c_v \left(\frac{J}{kg K} \right)$	$c_p \left(\frac{J}{kg K} \right)$	$\mu (Pa s)$
273	999,84	0,5610	4217,6	4217,6	0,0017930
278	999,97	0,5705	4204,9	4204,9	0,0015500
283	999,70	0,5800	4192,1	4192,1	0,0013070
293	998,21	0,5984	4181,8	4181,8	0,0010020
303	995,65	0,6154	4178,4	4178,4	0,0007977
313	992,22	0,6305	4178,5	4178,5	0,0006532
323	988,03	0,6435	4180,6	4180,6	0,0005470
333	983,20	0,6543	4184,3	4184,3	0,0004665
343	977,78	0,6631	4189,5	4189,5	0,0004040
353	971,82	0,6700	4196,3	4196,3	0,0003544
363	965,35	0,6753	4205,0	4205,0	0,0003145
373	958,40	0,6791	4215,9	4215,9	0,0002818

Tabela “Internal_heat”

Tempo (s)	0	18	18,18
Posição (m)			
0	0	0	0
20	0	0	0
20,2	0	0	0

Tabela “Wall”

Tempo (s)	0	18	18,18
Posição (m)			
0	353	353	353
20	353	353	353
20,2	353	353	353

B.1.2 Diâmetros do duto

```
function D = duct_shape (x)
% D deve resultar em metros
D = 0.050 + 0*x;
```

B.1.3 Condições de fronteira

```
function [P_0 P_L] = boundary_1 (rho, g, Patm, Q, D, H, K_valv, K_estr)

P_0 = Patm + rho * g * H - 0.5 * rho * (4 * Q / (pi * D(1)^2))^2 ...
      - 0.5 * K_estr * rho * (4 * Q / (pi * D(1)^2))^2;
P_L = Patm + 0.5 * K_valv * rho * (4 * Q / (pi * D(2)^2))^2;
```

B.1.4 Arquivo principal

```
function [THETA, T, Q, H] = main ( aux_n, aux_J, aux_nodes, aux_interf )

tic,
clearvars -except aux_n aux_J aux_nodes aux_interf

%% Inicialização específica para cada problema
xlsspecific = xlsread('Duto.xls', 'Specific');

specific = struct ('K_valv', xlsspecific(1), 'K_estr', xlsspecific(2), ...
                  'H', xlsspecific(3), 'A_res', pi*xlsspecific(4)^2 / 4, ...
                  'theta_res', xlsspecific(5));

%% Acesso a arquivos externos
xlsconstants = xlsread('Duto.xls', 'Constants', 'C3:C11');
xlsduct = xlsread('Duto.xls', 'Duct');
xlsinitial = xlsread('Duto.xls', 'Initial');
xlsfluid = xlsread('Duto.xls', 'Fluid_properties');
```

```

xlsinternalheat = xlsread ('Duto.xls', 'Internal_heat');
xlswall = xlsread ('Duto.xls', 'Wall');

disp('Já peguei os dados das planilhas')

%% Inicialização das variáveis gerais
duct = struct ('L', xlsduct(1), 'Z', xlsduct(2), ...
    'abs_rugosity', xlsduct(3), 'D', duct_shape( aux_nodes ), ...
    'D_punctual_interf', duct_shape ( aux_interf ));

discrete = struct ('n', aux_n, 'J', aux_J );

constant = struct ('g', xlsconstants(2), 'Patm', xlsconstants(3), ...
    'theta_ref', xlsconstants(4), 'eps', xlsconstants(5));

time = struct ('first', xlsconstants(7), 'last', xlsconstants(8), ...
    'step', 0, 'current', 0);

predcorr = struct ('zeta', xlsconstants(9) );

initial = struct ('Q', xlsinitial(1,3), ...
    'theta', v_lin_interp(aux_nodes, xlsinitial(:,1:2)) );

% As the flow is incompressible, rho is calculated as one for the whole code
rho = lin_interp ( 323, xlsfluid(:,1:2)) );

% It is important to initialize P_0 - P_L that would allow the Q_0 imposed
boundary = struct ('P_0', 1, 'P_L', 1);

wall = struct ('theta', m_lin_interp ( aux_nodes, time.current, xlswall )...
    , 'P_h', pi*duct.D);

fluid = struct ( 'c_v', ...
    v_lin_interp ( initial.theta, [xlsfluid(:,1) xlsfluid(:,4)] ), ...
    'c_v_interf', v_lin_interp ( (initial.theta(1:end-1)+ ...
    initial.theta(2:end))/2, [xlsfluid(:,1) xlsfluid(:,4)] ), ...
    'phi', m_lin_interp ( aux_nodes, time.current, xlsinternalheat ) );

H = specific.H;

Q = initial.Q;
THETA = initial.theta;
T = time.first;

```

```

sv = rho * fluid.c_v .* log (initial.theta / constant.theta_ref );

i = 1;

%% Determinação do tamanho do primeiro passo de tempo
if initial.Q == 0
    time.step = 0.2;
else
    time.step = discrete.J * pi * min(duct.D)^2 / ( 2 * 4 * initial.Q );
end

time.current = time.first + time.step;

disp('Terminei toda a parte de inicialização geral')

%% Para cada tempo
while time.current < time.last
    % Inicialização de variáveis temporais
    fluid.phi = m_lin_interp ( aux_nodes, time.current, xlsinternalheat );

    wall.theta = m_lin_interp ( aux_nodes, time.current, xlswall );

    % Inicialização do método preditor-corretor
    Q_C = Q(1, i);
    sv_C = sv(:, i);

    [boundary.P_0, boundary.P_L] = boundary_f (rho, constant.g, ...
        constant.Patm, Q_C, [duct.D(1) duct.D(end)], specific.H, ...
        specific.K_valv, specific.K_estr);

    [Q_P, sv_P, fluid.c_v, fluid.c_v_interf] = predict ( xlsfluid, Q_C, ...
        sv_C, Q(1,i), sv(:,i), discrete.J, duct.Z, duct.abs_rugosity, ...
        duct.D, duct.D_punctual_interf, constant.g, constant.theta_ref, ...
        constant.eps, boundary.P_0, boundary.P_L, wall.theta, wall.P_h, ...
        fluid.phi, rho, fluid.c_v, fluid.c_v_interf, specific.theta_res, ...
        time.step );

    j = 0;

    % Enquanto o critério de parada não for satisfeito
    while sum( abs([Q_P; sv_P] - [Q_C; sv_C])./[Q_P; sv_P]) < ...
        constant.eps ) ~= (discrete.n + 1)
        % Iteração do método preditor-corretor

```

```

Q_C = Q_C + predcorr.zeta * (Q_P - Q_C);
sv_C = sv_C + predcorr.zeta * (sv_P - sv_C);

[boundary.P_0, boundary.P_L] = boundary_f (rho, constant.g, ...
    constant.Patm, Q_C, [duct.D(1) duct.D(end)], specific.H, ...
    specific.K_valv, specific.K_estr);

[Q_P, sv_P, fluid.c_v, fluid.c_v_interf] = predict ( xlsfluid, ...
    Q_C, sv_C, Q(1,i), sv(:,i), discrete.J, duct.Z, ...
    duct.abs_rugosity, duct.D, duct.D_punctual_interf, ...
    constant.g, constant.theta_ref, constant.eps, boundary.P_0, ...
    boundary.P_L, wall.theta, wall.P_h, fluid.phi, rho, ...
    fluid.c_v, fluid.c_v_interf, specific.theta_res, time.step );

j = j + 1;

% Visualização do resíduo máximo
%     if max(abs([Q_P; sv_P] - [Q_C; sv_C])./[Q_P; sv_P])) == ...
%         max([Q_P; sv_P] - [Q_C; sv_C])./[Q_P; sv_P])
%         plot(j, log10(max(abs([Q_P; sv_P] - [Q_C; sv_C])./ ...
%             [Q_P; sv_P]))/constant.eps),'bo')
%     else
%         plot(j, log10(max(abs([Q_P; sv_P] - [Q_C; sv_C])./ ...
%             [Q_P; sv_P]))/constant.eps),'rx')
%     end
%     hold on
%     if j > 24
%         disp(['Em ',num2str(time.current),' s, o sistema divergiu!'])
%         break
%     end
end

% Visualização do resíduo máximo
%     hold off
%     legend ('em azul, resíduo positivo      em vermelho, resíduo negativo')
%     title(['Tempo = ',num2str(time.current), ' s'])
%     xlabel ('iteração')
%     ylabel ('log_1_0( |resíduo| / eps )')
%     pause
%     if j ~= 25
%         disp('O sistema divergiu!')
%     else
%         fprintf

```

```

        ('      Em %7.3f s, o sistema convergiu em %2.0f iterações\n'...
        , time.current, j)

%      end

% Finalização do método preditor-corretor
i = i + 1;
Q(i) = Q_P;
sv(:, i) = sv_P;

THETA(:, i) = constant.theta_ref*exp( sv(:, i) ./ (rho*fluid.c_v) );
T(i) = time.current;

% Específico do problema - atualização do nível do reservatório
% specific.H = specific.H - (Q(i)+Q(i+1))/2 * time.step /specific.A_res;
H(i+1) = specific.H;

% Correção do passo de tempo
time.step = min(discrete.J .* duct.D.^2) * pi / ( 2 * 4 * Q(i) );
% time.step = 0.02;

time.current = time.current + time.step;
end

%% Para o tempo final
time.current = time.last;
% Inicialização de variáveis temporais
fluid.phi = m_lin_interp ( aux_nodes, time.current, xlsinternalheat );

wall.theta = m_lin_interp ( aux_nodes, time.current, xlswall );

% Inicialização do método preditor-corretor
Q_C = Q(1, i);
sv_C = sv(:, i);

[boundary.P_0, boundary.P_L] = boundary_f (rho, constant.g, ...
      constant.Patm, Q_C, [duct.D(1) duct.D(end)], specific.H, ...
      specific.K_valv, specific.K_estr);

[Q_P, sv_P, fluid.c_v, fluid.c_v_interf] = predict ( xlsfluid, Q_C, sv_C,...
      Q(1,i), sv(:,i), discrete.J, duct.Z, duct.abs_rugosity, duct.D, ...
      duct.D_punctual_interf, constant.g, constant.theta_ref, constant.eps,...
      boundary.P_0, boundary.P_L, wall.theta, wall.P_h, fluid.phi, rho, ...

```

```

    fluid.c_v, fluid.c_v_intf, specific.theta_res, time.step );

j = 0;

% Enquanto o critério de parada não for satisfeito
while sum( abs([Q_P; sv_P] - [Q_C; sv_C])./[Q_P; sv_P]) < constant.eps )...
    ~= (discrete.n + 1)
    % Iteração do método preditor-corretor
    Q_C = Q_C + predcorr.zeta * (Q_P - Q_C);
    sv_C = sv_C + predcorr.zeta * (sv_P - sv_C);

    [boundary.P_0, boundary.P_L] = boundary_f (rho, constant.g, ...
        constant.Patm, Q_C, [duct.D(1) duct.D(end)], specific.H, ...
        specific.K_valv, specific.K_estr);

    [Q_P, sv_P, fluid.c_v, fluid.c_v_intf] = predict ( xlsfluid, Q_C, ...
        sv_C, Q(1,i), sv(:,i), discrete.J, duct.Z, duct.abs_rugosity, ...
        duct.D, duct.D_punctual_intf, constant.g, constant.theta_ref, ...
        constant.eps, boundary.P_0, boundary.P_L, wall.theta, wall.P_h, ...
        fluid.phi, rho, fluid.c_v, fluid.c_v_intf, specific.theta_res, ...
        time.step );

    j = j + 1;
end

fprintf('      Em %7.3f s, o sistema convergiu em %3.0f iterações\n', ...
    time.current, j)

% Finalização do método preditor-corretor
i = i+1;
Q(i) = Q_P;
sv(:, i) = sv_P;

THETA(:, i) = constant.theta_ref*exp( sv(:, i) ./ (rho*fluid.c_v) );
T(i) = time.current;

% Específico do problema - atualização do nível do reservatório
H(i) = specific.H - (Q(i-1)+Q(i))/2 * time.step /specific.A_res;

% %% Solução analítica
% h_m = h/2;
% for i = 2:discrete.n-1
%     h_m(i) = ((2*i-3)*h_m(i-1) + 2*h(i)) / (2*i-1);

```

```

% end
% h_m(end) = ((2*discrete.n-3)*h_m(discrete.n-1) + h(end))/(2*discrete.n-2);
%
% THETA_analytical = wall.theta + (specific.theta_res - wall.theta) .* ...
%     exp ( - wall.P_h .* h_m .* aux_nodes ./ (rho * Q(end) * fluid.c_v) );
%
% aux_comp = THETA_analytical - THETA(:,end);
% disp([min(aux_comp) mean(aux_comp) median(aux_comp) max(aux_comp)])
% disp(['A variação máxima percentual é de ', num2str(max(abs(aux_comp) ./ ...
%     THETA_analytical))*100), ' %'])

%% Plotagem de gráficos; pós-tratamento
% aux_scale = 0.1;
% scale_TH = [ 0 L                                     ...
%     ((1+aux_scale)*min(min(THETA)) - aux_scale*max(max(THETA)) - 273) ...
%     ((1+aux_scale)*max(max(THETA)) - aux_scale*min(min(THETA)) - 273) ];
% figure('units','normalized','position',[.05 .1 .9 .8], 'color',[1 1 1]);
% plot(0:L/(n-1):L, TH_analitico-273)
% hold on
% plot(0:L/(n-1):L, THETA(:,end)-273, 'rx')
% hold on
% axis on
% axis(scale_TH)
% title ('Distribuição de temperatura no fluido ao longo do duto')
% xlabel('Posição no duto (m)')
% ylabel('Temperatura nodal (°C)')
% legend('Solução analítica', 'Solução do código implementado', ...
%     'Location','Southeast')

aux_scale = 0.1;
scale_TH = [ 0 duct.L                                     ...
%     ((1+aux_scale)*min(min(THETA)) - aux_scale*max(max(THETA)) - 273) ...
%     ((1+aux_scale)*max(max(THETA)) - aux_scale*min(min(THETA)) - 273) ];
scale_Q = [ T(1) T(end) ((1+aux_scale)*min(Q) - aux_scale*max(Q)) ...
%     ((1+aux_scale)*max(Q) - aux_scale*min(Q)) ];
scale_H = [ T(1) T(end) ((1+aux_scale)*min(H) - aux_scale*max(H)) ...
%     ((1+aux_scale)*max(H) - aux_scale*min(H)) ];
% scale_H = [T(1) T(end) 4.8 5.2];
fig = figure('units','normalized','position',[.05 .1 .9 .8],'name','Video');
for i = 1:size(T,2)
    subplot(3,2,[1 2 3 4])

```

```

plot(0:duct.L/(discrete.n-1):duct.L, THETA(:,i)-273)
axis on
axis(scale_TH)
title(['Distribuição de temperatura no fluido ao longo do duto,', ...
      ' para o tempo ', num2str(T(i),'\%6.4f'), ' s'])
xlabel('Posição no duto (m)')
ylabel('Temperatura nodal (°C)')

subplot(3,2,5)
plot(T(1:i),Q(1:i))
axis on
axis(scale_Q)
title('Vazão x Tempo')
xlabel('Tempo (s)')
ylabel('Vazão volumétrica (m^3/s)')

subplot(3,2,6)
plot(T(1:i),H(1:i))
axis on
axis(scale_H)
title('Altura do tanque x Tempo')
xlabel('Tempo (s)')
ylabel('Altura do tanque (m)')

Video(i) = getframe(fig);
end

movie2avi(Video,'agua.avi','FPS',20,'COMPRESSION','None');

clearvars -except THETA T Q H

```

B.1.5 Predição

```

function [Q_P, sv_P, c_v_n, c_v_interf] = predict ( xlsfluid, Q, sv, Q_t,...
    sv_t, J, Z, abs_rugosity, D, D_punctual_interf, g, theta_ref, eps, ...
    P_0, P_L, theta_wall, P_h, phi, rho, c_v_n, c_v_interf, theta_res, ...
    delta_t )

n = size (sv,1);

% Cálculo de temperaturas, propriedades do fluido e do escoamento
omega = pi/4 .* ([0.5; ones(n-2, 1); 0.5] .* (J .* D.^2));

```

```

theta_nodes = theta_ref * exp( sv ./ c_v_n / rho);

theta_interfaces = theta_ref * exp( ( sv (1:end-1) + sv (2:end) ) ...
    ./ (2 * rho * c_v_intf) );

[ lambda_n, c_v_n, c_p_n, mu_n, lambda_intf, c_v_intf, c_p_intf ] ...
    = fluid_properties ( theta_nodes, theta_interfaces, xlsfluid );

Pr = mu_n .* c_p_n ./ lambda_n;

Re = 4 * abs(Q) * rho / pi ./ mu_n ./ D;

Pe = 4 * Q * rho / pi * 0.5 * (J(1:end-1)+J(2:end)) .* c_p_intf ./ ...
    lambda_intf ./ D_punctual_intf .^2;
Pe_n = 4 * Q * rho * J .* c_p_n / pi ./ lambda_n ./ D .^2;

f = friction_factor ( abs_rugosity, D, Re, eps );
H = lambda_n .* Nu (Re, Pr, Pe_n, f) ./ D;

%% Cálculos das funções de peso
flow_beta = upwind (Pe);

weight = diag( [( 3/8 - flow_beta(1)/2 ); ...
    ( 3/4 + (flow_beta(1:end-1) - flow_beta(2:end))/2 ); ...
    (3/8 + flow_beta(end)/2)] ) + diag( 1/8 - flow_beta/2, 1 ) ...
    + diag( 1/8 + flow_beta/2, -1 );

derv_weight = diag( [ -0.5 - flow_beta(1); ...
    - flow_beta(1:end-1) - flow_beta(2:end); 0.5 - flow_beta(end) ] ) ...
    + diag( flow_beta - 0.5, 1 ) + diag( flow_beta + 0.5, -1 );

%% Avaliação dos termos no interior do domínio
press_convective = 8 / pi^2 * rho * Q * (D(1)^(-4) - D(end)^(-4) );
press_dissipation = 8 / pi^2 * rho * abs(Q) * ...
    [0.5 ones(1,n-2) 0.5] * (J .* f) / (D' * D.^4);

press_gravity = - rho * g * Z;

aux_conduction = lambda_intf .* D_punctual_intf .^2 .* ...
    theta_interfaces ./ c_v_intf;
pow_conduction = pi / (4*rho) * ( - diag([aux_conduction; 0]) + ...
    diag(aux_conduction, 1) - diag([0; aux_conduction]) + ...
    diag(aux_conduction, -1) );

```

```

pow_convection = Q * derv_weight .* ( diag( theta_nodes ) + ...
    diag( theta_nodes(2:end), 1) + diag( theta_nodes(1:end-1), -1) );

aux_nonhomog = theta_interfaces .* ( sv(1:end-1) + sv(2:end) ) / 2 ...
    ./ c_v_intf;
pow_nonhomog = Q / rho * ...
    ( - diag (J .* [ ( 0.5-flow_beta) .* aux_nonhomog] ; 0]) ...
    + diag (J .* [ 0; ( 0.5+flow_beta) .* aux_nonhomog]) ...
    + diag ( J(1:end-1) .* (0.5-flow_beta) .* aux_nonhomog, 1) ...
    - diag ( J(2:end) .* (0.5+flow_beta) .* aux_nonhomog, -1) );

pow_wall = weight * ( J .* H .* P_h .* ( theta_nodes - theta_wall ));
pow_generation = pi * rho / 4 * weight * ( J .* phi .* D.^2 );
pow_dissipation = 8 / pi^2 * rho * Q^2 * abs(Q) * weight * ...
    ( J .* f ./ D.^5 );

inertia = 4 * rho / pi * [0.5 ones(1,n-2) 0.5] * (J .* D.^(-2));

matrix_theta = pi / 4 * weight .* ( ...
    diag ( J .* D.^2 .* theta_nodes ./ omega ) + ...
    diag ( J(2:end) .* D(2:end).^2 .* theta_nodes(2:end) ./ ...
    omega(2:end), 1 ) + diag ( J(1:end-1) .* D(1:end-1).^2 .* ...
    theta_nodes(1:end-1) ./ omega(1:end-1), -1 ) );

%% Avaliação dos termos nas fronteiras - causalidades integrais
press_boundary = P_0 - P_L;

% O valor de q_x^0 não importa para esse problema em específico
pow_boundcond = pi / 4 * [ 1 * D(1)^2; zeros(n-2, 1); 0 * D(end)^2 ];

pow_boundconv = Q * diag([-theta_nodes(1); zeros(n-2,1); theta_nodes(end)]);

%% Montagem das matrizes do sistema linear
A1 = ( press_convective - press_dissipation ) / inertia;
b1 = ( press_gravity + press_boundary ) / inertia;

A1 = 1 - A1 * delta_t;
b1 = Q_t + b1 * delta_t;

```

```

A2 = pow_conduction + pow_convection + pow_nonhomog - pow_boundconv;
b2 = pow_generation + pow_dissipation - pow_wall - pow_boundcond;

A2 = matrix_theta * diag(omega) - A2 * delta_t;
b2 = matrix_theta * diag(omega) * sv_t + b2 * delta_t;

%% Implementação de causalidades derivativas
A2(1,:) = [1; zeros(n-1,1)]';
b2(1,1) = rho * c_v_n(1) * log(theta_res / theta_ref);

%% Passo preditivo
Q_P = b1 / A1;

sv_P = thomas ([A2 b2]);

```

B.2 Segundo caso estudado – Óleo em aquecimento

Nesta seção serão apresentadas as tabelas de dados auxiliares e as funções implementadas para o segundo caso estudado, apresentado no capítulo 6.

B.2.1 Tabelas com dados auxiliares

Tabela “Duct”

L	20	m
Z	0	m
rug. eq.	0,00026	m

Tabela “Specific”

K_{saida}	1,0	-
$K_{entrada}$	0,5	-
Nível inicial	0,5	m
Diâmetro do tanque	1	m
Temperatura do tanque	353	K

Tabela “Fluid_properties”

Temperatura (K)	$\rho \left(\frac{kg}{m^3} \right)$	$\lambda \left(\frac{W}{K m} \right)$	$c_v \left(\frac{J}{kg K} \right)$	$c_p \left(\frac{J}{kg K} \right)$	$\mu (Pa s)$	Pr (-)
273	899,1	0,147	1796	1796	3,85	47000
280	895,3	0,144	1827	1827	2,17	27500
290	890,0	0,145	1868	1868	0,999	12900
300	884,1	0,145	1909	1909	0,486	6400
310	877,9	0,145	1951	1951	0,253	3400
320	871,8	0,143	1993	1993	0,141	1965
330	865,8	0,141	2035	2035	0,0836	1205
340	859,9	0,139	2076	2076	0,0531	793
350	853,9	0,138	2118	2118	0,0356	546
360	847,8	0,138	2161	2161	0,0252	395
370	841,8	0,137	2206	2206	0,0186	300
380	836,0	0,136	2250	2250	0,0141	233
390	830,6	0,135	2294	2294	0,0110	187
400	825,1	0,134	2337	2337	0,00874	152
410	818,9	0,133	2381	2381	0,00698	125
420	812,1	0,133	2427	2427	0,00564	103
430	806,5	0,132	2471	2471	0,00470	88

Tabela “Wall”

Tempo (s)	0	150	151,5
Posição (m)			
0	413	413	413
20	413	413	413
20,2	413	413	413

Tabela “Internal_heat”

Tempo (s)	0	150	151,5
Posição (m)			
0	0	0	0
20	0	0	0
20,2	0	0	0

Tabela “Initial”

Posição (m) - apenas para θ	θ (K)	Q ($\frac{m^3}{s}$)
0	298	0
20	298	
20,2	298	

Tabela “Constants”

Discretização	n	6	-
Constantes gerais	g	9,807	$\frac{m}{s^2}$
	P_{atm}	1,01E+05	Pa
	θ_{ref}	293,15	K
	precisão	1,00E-06	-
	precisão converg. de malha	1,00E-03	-
Tempo	começo	0	s
	fim	150	s
Preditor-Corretor	ζ	0,3	-

B.2.2 Diâmetros do duto

```
function D = duct_shape (x)
% D deve resultar em metros
D = 0.050 + 0*x;
```

B.2.3 Condições de fronteira

```
function [P_0 P_L] = boundary_1 (rho, Pentr, Kentr, Q, D, Psai, Ksai)

P_0 = Pentr - 0.5 * rho * (4 * Q / (pi * D(1)^2))^2 ...
      - 0.5 * Kentr * rho * (4 * Q / (pi * D(1)^2))^2;
P_L = Psai + 0.5 * Ksai * rho * (4 * Q / (pi * D(2)^2))^2;
```

B.2.4 Arquivo principal

```
function [THETA, T, Q, H] = main ( aux_n, aux_J, aux_nodes, aux_interf )

tic,
clearvars -except aux_n aux_J aux_nodes aux_interf
```

```

%% Inicialização específica para cada problema
xlsspecific = xlsread('Duto.xls', 'Specific');

specific = struct ('K_valv', xlsspecific(1), 'K_estr', xlsspecific(2), ...
    'H', xlsspecific(3), 'A_res', pi*xlsspecific(4)^2 / 4, ...
    'theta0_initial', xlsspecific(5), 'theta0', xlsspecific(5));

%% Acesso a arquivos externos
xlsconstants = xlsread('Duto.xls', 'Constants', 'C3:C11');
xlsduct = xlsread('Duto.xls', 'Duct');
xlsinitial = xlsread('Duto.xls', 'Initial');
xlsfluid = xlsread('Duto.xls', 'Fluid_properties');
xlsinternalheat = xlsread('Duto.xls', 'Internal_heat');
xlswall = xlsread('Duto.xls', 'Wall');

disp('Já peguei os dados das planilhas')

%% Inicialização das variáveis gerais
duct = struct ('L', xlsduct(1), 'Z', xlsduct(2), ...
    'abs_rugosity', xlsduct(3), 'D', duct_shape( aux_nodes ), ...
    'D_punctual_interf', duct_shape ( aux_interf ));

discrete = struct ('n', aux_n, 'J', aux_J );

constant = struct ('g', xlsconstants(2), 'Patm', xlsconstants(3), ...
    'theta_ref', xlsconstants(4), 'eps', xlsconstants(5));

time = struct ('first', xlsconstants(7), 'last', xlsconstants(8), ...
    'step', 0, 'current', 0);

predcorr = struct ('zeta', xlsconstants(9) );

initial = struct ('Q', xlsinitial(1,3), ...
    'theta', v_lin_interp(aux_nodes, xlsinitial(:,1:2)) );

% As the flow is incompressible, rho is calculated as one for the whole code
rho = lin_interp ( 353, xlsfluid(:,1:2)' );

% It is important to initialize P_0 - P_L that would allow the Q_0 imposed
boundary = struct ('P_0', 1, 'P_L', 1);

wall = struct ('theta', m_lin_interp ( aux_nodes, time.current, xlswall )...

```

```

, 'P_h', pi*duct.D);

fluid = struct ( 'c_v', ...
    v_lin_interp ( initial.theta, [xlsfluid(:,1) xlsfluid(:,4)] ), ...
    'c_v_interf', v_lin_interp ( (initial.theta(1:end-1)+
    initial.theta(2:end))/2, [xlsfluid(:,1) xlsfluid(:,4)] ), ...
    'phi', m_lin_interp ( aux_nodes, time.current, xlsinternalheat ) );

H = specific.H;

Q = initial.Q;
THETA = initial.theta;
T = time.first;
sv = rho * fluid.c_v .* log (initial.theta / constant.theta_ref );

i = 1;

%% Determinação do tamanho do primeiro passo de tempo
if initial.Q == 0
    time.step = 3;
else
    time.step = discrete.J * pi * min(duct.D)^2 / ( 2 * 4 * initial.Q );
end

time.current = time.first + time.step;

disp('Terminei toda a parte de inicialização geral')

%% Para cada tempo
while time.current < time.last
    % Inicialização de variáveis temporais
    fluid.phi = m_lin_interp ( aux_nodes, time.current, xlsinternalheat );

    wall.theta = m_lin_interp ( aux_nodes, time.current, xlswall );

    specific.theta0 = TH_entrada ( initial.theta(1), ...
        specific.theta0_initial, time.first, time.last, time.current);

    % Inicialização do método preditor-corretor
    Q_C = Q(1, i);
    sv_C = sv(:, i);

    [boundary.P_0, boundary.P_L] = boundary_f (rho, constant.g, ...

```

```

constant.Patm, Q_C, [duct.D(1) duct.D(end)], specific.H, ...
specific.K_valv, specific.K_estr);

[Q_P, sv_P, fluid.c_v, fluid.c_v_interf] = predict ( xlsfluid, Q_C, ...
sv_C, Q(1,i), sv(:,i), discrete.J, duct.Z, duct.abs_rugosity, ...
duct.D, duct.D_punctual_interf, constant.g, constant.theta_ref, ...
constant.eps, boundary.P_0, boundary.P_L, wall.theta, wall.P_h, ...
fluid.phi, rho, fluid.c_v, fluid.c_v_interf, specific.theta0, ...
time.step );

j = 0;

% Enquanto o critério de parada não for satisfeito
while sum( abs([Q_P; sv_P] - [Q_C; sv_C])./[Q_P; sv_P]) < ...
    constant.eps ) ~= (discrete.n + 1)
    % Iteração do método preditor-corretor
    Q_C = Q_C + predcorr.zeta * (Q_P - Q_C);
    sv_C = sv_C + predcorr.zeta * (sv_P - sv_C);

    [boundary.P_0, boundary.P_L] = boundary_f (rho, constant.g, ...
        constant.Patm, Q_C, [duct.D(1) duct.D(end)], specific.H, ...
        specific.K_valv, specific.K_estr);

    [Q_P, sv_P, fluid.c_v, fluid.c_v_interf] = predict ( xlsfluid, ...
        Q_C, sv_C, Q(1,i), sv(:,i), discrete.J, duct.Z, ...
        duct.abs_rugosity, duct.D, duct.D_punctual_interf, ...
        constant.g, constant.theta_ref, constant.eps, boundary.P_0, ...
        boundary.P_L, wall.theta, wall.P_h, fluid.phi, rho, ...
        fluid.c_v, fluid.c_v_interf, specific.theta0, time.step );

    j = j + 1;

% Visualização do resíduo máximo
% if max(abs([Q_P; sv_P] - [Q_C; sv_C])./[Q_P; sv_P])) == ...
%     max([Q_P; sv_P] - [Q_C; sv_C])./[Q_P; sv_P])
%     plot(j, log10(max(abs([Q_P; sv_P] - [Q_C; sv_C])./ ...
%         [Q_P; sv_P]))/constant.eps), 'bo')
% else
%     plot(j, log10(max(abs([Q_P; sv_P] - [Q_C; sv_C])./ ...
%         [Q_P; sv_P]))/constant.eps), 'rx')
% end
% hold on
% if j > 24

```

```

%             disp(['Em ', num2str(time.current), ' s, o sistema divergiu!'])
%             break
%         end
    end

    % Visualização do resíduo máximo
    %     hold off
    %     legend ('em azul, resíduo positivo      em vermelho, resíduo negativo')
    %     title(['Tempo = ', num2str(time.current), ' s'])
    %     xlabel ('iteração')
    %     ylabel ('log10( |resíduo| / eps )')
    %     pause
    %     if j ~= 25
    %         disp('O sistema divergiu!')
    %     else
    %         fprintf
    %             ...
    %             ('      Em %7.3f s, o sistema convergiu em %2.0f iterações\n' ...
    %             , time.current, j)
    %     end

    % Finalização do método preditor-corretor
    i = i + 1;
    Q(i) = Q_P;
    sv(:, i) = sv_P;

    THETA(:, i) = constant.theta_ref*exp( sv(:, i) ./ (rho*fluid.c_v) );
    T(i) = time.current;

    % Específico do problema - atualização do nível do reservatório
    % specific.H = specific.H - (Q(i)+Q(i+1))/2 * time.step /specific.A_res;
    H(i+1) = specific.H;

    %     if (j == 0)
    %         break
    %     else
    %         Correção do passo de tempo
    time.step = min(discrete.J .* duct.D.^2) * pi / ( 2 * 4 * Q(i) );
    % time.step = 0.02;

    time.current = time.current + time.step;
    %     end
end

```

```

%% Para o tempo final
time.current = time.last;
% Inicialização de variáveis temporais
fluid.phi = m_lin_interp ( aux_nodes, time.current, xlsinternalheat );

wall.theta = m_lin_interp ( aux_nodes, time.current, xlswall );

specific.theta0 = TH_entrada ( initial.theta(1), specific.theta0_initial,...
    time.first, time.last, time.current);

% Inicialização do método preditor-corretor
Q_C = Q(1, i);
sv_C = sv(:, i);

[boundary.P_0, boundary.P_L] = boundary_f (rho, constant.g, ...
    constant.Patm, Q_C, [duct.D(1) duct.D(end)], specific.H, ...
    specific.K_valv, specific.K_estr);

[Q_P, sv_P, fluid.c_v, fluid.c_v_interf] = predict ( xlsfluid, Q_C, sv_C,...
    Q(1,i), sv(:,i), discrete.J, duct.Z, duct.abs_rugosity, duct.D, ...
    duct.D_punctual_interf, constant.g, constant.theta_ref, constant.eps,...
    boundary.P_0, boundary.P_L, wall.theta, wall.P_h, fluid.phi, rho, ...
    fluid.c_v, fluid.c_v_interf, specific.theta0, time.step );

j = 0;

% Enquanto o critério de parada não for satisfeito
while sum( abs([Q_P; sv_P] - [Q_C; sv_C])./[Q_P; sv_P]) < constant.eps )...
    ~= (discrete.n + 1)
    % Iteração do método preditor-corretor
    Q_C = Q_C + predcorr.zeta * (Q_P - Q_C);
    sv_C = sv_C + predcorr.zeta * (sv_P - sv_C);

    [boundary.P_0, boundary.P_L] = boundary_f (rho, constant.g, ...
        constant.Patm, Q_C, [duct.D(1) duct.D(end)], specific.H, ...
        specific.K_valv, specific.K_estr);

    [Q_P, sv_P, fluid.c_v, fluid.c_v_interf] = predict ( xlsfluid, Q_C, ...
        sv_C, Q(1,i), sv(:,i), discrete.J, duct.Z, duct.abs_rugosity, ...
        duct.D, duct.D_punctual_interf, constant.g, constant.theta_ref, ...
        constant.eps, boundary.P_0, boundary.P_L, wall.theta, wall.P_h, ...
        fluid.phi, rho, fluid.c_v, fluid.c_v_interf, specific.theta0, ...

```

```

        time.step );

    j = j + 1;
end

fprintf('      Em %7.3f s, o sistema convergiu em %3.0f iterações\n',    ...
        time.current, j)

% Finalização do método preditor-corretor
i = i+1;
Q(i) = Q_P;
sv(:, i) = sv_P;

THETA(:, i) = constant.theta_ref*exp( sv(:, i) ./ (rho*fluid.c_v) );
T(i) = time.current;

% Específico do problema - atualização do nível do reservatório
H(i) = specific.H; %- (Q(i-1)+Q(i))/2 * time.step /specific.A_res;

%% Plotagem de gráficos; pós-tratamento
aux_scale = 0.1;
scale_TH = [ 0 L ((1+aux_scale)*min(min(THETA)) - aux_scale*    ...
        max(max(THETA)) - 273) ((1+aux_scale)*max(max(THETA)) -    ...
        aux_scale*min(min(THETA)) - 273) ];
scale_Q = [ T(1) T(end) ((1+aux_scale)*min(Q) - aux_scale*max(Q))    ...
        ((1+aux_scale)*max(Q) - aux_scale*min(Q)) ];
% scale_TH0=[T(1) T(end) (1-aux_scale)*THETA(1,1) (1+aux_scale)*THETA(1,1)];
scale_TH0 = [ T(1) T(end) ((1+aux_scale)*min(THETA(1,:))-273) - aux_scale*...
        max(THETA(1,:)-273)) ((1+aux_scale)*max(THETA(1,:)-273) - aux_scale* ...
        min(THETA(1,:)-273)) ];
fig = figure('units','normalized','position',[.05 .1 .9 .8],'name','Video');
for i = 1:size(T,2)
    subplot(3,2,[1 2 3 4])
    plot(nodes, THETA(:,i)-273)
    axis on
    axis(scale_TH)
    title (['Distribuição de temperatura no fluido ao longo do duto,',    ...
        ' para o tempo ', num2str(T(i),'%6.4f'), ' s'])
    xlabel('Posição no duto (m)')
    ylabel('Temperatura nodal (°C)')

```

```

subplot(3,2,5)
plot(T(1:i),Q(1:i))
axis on
axis(scale_Q)
title('Vazão x Tempo')
xlabel('Tempo (s)')
ylabel('Vazão volumétrica (m^3/s)')

subplot(3,2,6)
plot(T(1:i),THETA(1,1:i)-273)
axis on
axis(scale_TH0)
title('Temperatura de entrada x Tempo')
xlabel('Tempo (s)')
ylabel('Temperatura de entrada (°C)')

Video(i) = getframe(fig);
end

% movie2avi(Video,'agua.avi','FPS',30,'COMPRESSION','None');

clearvars -except THETA T Q H

```

B.2.5 Predição

```

function [Q_P, sv_P, c_v_n, c_v_interf] = predict ( xlsfluid, Q, sv, Q_t,...
    sv_t, J, Z, abs_rugosity, D, D_punctual_interf, g, theta_ref, eps, ...
    P_0, P_L, theta_wall, P_h, phi, rho, c_v_n, c_v_interf, theta_res, ...
    delta_t )

n = size (sv,1);

%% Cálculo de temperaturas, propriedades do fluido e do escoamento
omega = pi/4 .* ([0.5; ones(n-2, 1); 0.5] .* (J .* D.^2));

theta_nodes = theta_ref * exp( sv ./ c_v_n / rho);

theta_interfaces = theta_ref * exp( ( sv (1:end-1) + sv (2:end) ) ...
    ./ (2 * rho * c_v_interf) );

[lambda_n, c_v_n, c_p_n, mu_n, lambda_interf, c_v_interf, c_p_interf, Pr]...
    = fluid_properties ( theta_nodes, theta_interfaces, xlsfluid );

```

```

Re = 4 * abs(Q) * rho / pi ./ mu_n ./ D;

% [min(Re) mean(Re) median(Re) max(Re)]

Pe = 4 * Q * rho / pi * 0.5 * (J(1:end-1)+J(2:end)) .* c_p_interf ./ ...
    lambda_interf ./ D_punctual_interf .^2;
Pe_n = 4 * Q * rho * J .* c_p_n / pi ./ lambda_n ./ D .^2;

f = friction_factor ( abs_rugosity, D, Re, eps );
H = lambda_n .* Nu (Re, Pr, Pe_n, f) ./ D;

%% Cálculos das funções de peso
flow_beta = upwind (Pe);

weight = diag( [( 3/8 - flow_beta(1)/2 ); ...
    ( 3/4 + (flow_beta(1:end-1) - flow_beta(2:end))/2 ); ...
    (3/8 + flow_beta(end)/2)] ) + diag( 1/8 - flow_beta/2, 1 ) ...
    + diag( 1/8 + flow_beta/2, -1 );

derv_weight = diag( [ -0.5 - flow_beta(1); ...
    - flow_beta(1:end-1) - flow_beta(2:end); 0.5 - flow_beta(end) ] ) ...
    + diag( flow_beta - 0.5, 1 ) + diag( flow_beta + 0.5, -1 );

%% Avaliação dos termos no interior do domínio
press_convective = 8 / pi^2 * rho * Q * (D(1)^(-4) - D(end)^(-4) );
press_dissipation = 8 / pi^2 * rho * abs(Q) * ...
    [0.5 ones(1,n-2) 0.5] * (J .* f) / (D' * D.^4);

press_gravity = - rho * g * Z;

aux_conduction = lambda_interf .* D_punctual_interf .^2 .* ...
    theta_interfaces ./ c_v_interf;
pow_conduction = pi / (4*rho) * ( - diag([aux_conduction; 0]) + ...
    diag(aux_conduction, 1) - diag([0; aux_conduction]) + ...
    diag(aux_conduction, -1) );

pow_convection = Q * derv_weight .* ( diag( theta_nodes ) + ...
    diag( theta_nodes(2:end), 1) + diag( theta_nodes(1:end-1), -1) );

aux_nonhomog = theta_interfaces .* ( sv(1:end-1) + sv(2:end) ) / 2 ...
    ./ c_v_interf;
pow_nonhomog = Q / rho * ...
    ( - diag (J .* [ ( 0.5-flow_beta) .* aux_nonhomog) ; 0]) ...

```

```

+ diag (J .* [ 0; ( (0.5+flow_beta) .* aux_nonhomog)]) ...
+ diag ( J(1:end-1) .* (0.5-flow_beta) .* aux_nonhomog, 1) ...
- diag ( J(2:end) .* (0.5+flow_beta) .* aux_nonhomog, -1) );

pow_wall = weight * ( J .* H .* P_h .* ( theta_nodes - theta_wall ));
pow_generation = pi * rho / 4 * weight * ( J .* phi .* D.^2 );
pow_dissipation = 8 / pi^2 * rho * Q^2 * abs(Q) * weight * ...
    ( J .* f ./ D.^5 );

inertia = 4 * rho / pi * [0.5 ones(1,n-2) 0.5] * (J .* D.^(-2));

matrix_theta = pi / 4 * weight .* ( ...
    diag ( J .* D.^2 .* theta_nodes ./ omega ) + ...
    diag ( J(2:end) .* D(2:end).^2 .* theta_nodes(2:end) ./ ...
        omega(2:end), 1 ) + diag ( J(1:end-1) .* D(1:end-1).^2 .* ...
        theta_nodes(1:end-1) ./ omega(1:end-1), -1 ) );

%% Avaliação dos termos nas fronteiras - causalidades integrais
press_boundary = P_0 - P_L;

% O valor de q_x^0 não importa para esse problema em específico
pow_boundcond = pi / 4 * [ 1 * D(1)^2; zeros(n-2, 1); 0 * D(end)^2 ];

pow_boundconv = Q * diag([-theta_nodes(1); zeros(n-2,1); theta_nodes(end)]);

%% Montagem das matrizes do sistema linear
A1 = ( press_convective - press_dissipation ) / inertia;
b1 = ( press_gravity + press_boundary ) / inertia;

A1 = 1 - A1 * delta_t;
b1 = Q_t + b1 * delta_t;

A2 = pow_conduction + pow_convection + pow_nonhomog - pow_boundconv;
b2 = pow_generation + pow_dissipation - pow_wall - pow_boundcond;

A2 = matrix_theta * diag(omega) - A2 * delta_t;
b2 = matrix_theta * diag(omega) * sv_t + b2 * delta_t;

```

```

%% Implementação de causalidades derivativas
A2(1,:) = [1; zeros(n-1,1)]';
b2(1,1) = rho * c_v_n(1) * log(theta_res / theta_ref );

%% Passo preditivo
Q_P = b1 / A1;

sv_P = thomas ([A2 b2]);

%% Ponto de verificação: Balanço energético para regime permanente
% Balanço => E_entra + E_wall + E_dissipação - E_sai = 0
% ver_W_in = Q_P*theta_nodes(1)*sv_P(1);
% ver_W_wall = (H.*D.*J)'*(theta_wall-theta_nodes)*pi;
% ver_W_diss = sum(8 / pi^2 * rho * Q^2 * abs(Q) * weight * ...
    ( J .* f ./ D.^5 ));
% ver_W_sai = Q_P*theta_nodes(end)*sv_P(end);
%
% bal = ver_W_in + ver_W_wall + ver_W_diss - ver_W_sai;
% tot = ver_W_wall + ver_W_diss;
%
% relat = bal/tot;
% disp([bal tot relat]);

```

B.2.6 Temperatura de entrada

```

function TH_0 = TH_entrada ( TH_ini, TH_ss, t_ini, t_end, t)

k = (t_end - t_ini)*4/30;
dT = TH_ss - TH_ini;

aux = [ t_ini          TH_ini; ...
    (0.5*k + t_ini)  (0.3935*dT + TH_ini); ...
    (1.0*k + t_ini)  (0.6321*dT + TH_ini); ...
    (1.8*k + t_ini)  (0.8347*dT + TH_ini); ...
    (2.7*k + t_ini)  (0.9328*dT + TH_ini); ...
    (3.7*k + t_ini)  (0.9753*dT + TH_ini); ...
    (5.0*k + t_ini)  (0.9933*dT + TH_ini); ...
    (6.0*k + t_ini)  TH_ss; ...
    t_end*1.01      TH_ss]';

TH_0 = lin_interp ( t, aux );

% TH_0 = TH_ss;

```