

**SISTEMA COMPUTACIONAL ORIENTADO A OBJETOS PARA
ANÁLISE TERMOMECAÂNICA FÍSICAMENTE NÃO-LINEAR EM
REGIME TRANSIENTE VIA MÉTODO DOS ELEMENTOS FINITOS**

Guilherme Garcia Botelho

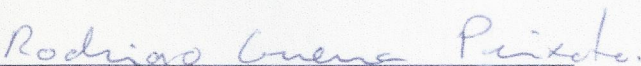
UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ESTRUTURAS

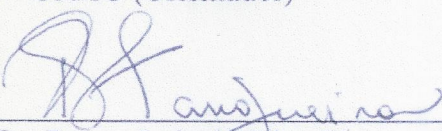
**"SISTEMA COMPUTACIONAL ORIENTADO A OBJETOS PARA
ANÁLISE TERMOMECÂNICA FÍSICAMENTE NÃO-LINEAR EM
REGIME TRANSIENTE VIA MÉTODO DOS ELEMENTOS
FINITOS"**


Guilherme Garcia Botelho


Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Estruturas da Escola de Engenharia da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do título de "Mestre em Engenharia de Estruturas".

Comissão Examinadora:


Prof. Dr. Rodrigo Guerra Peixoto
DEES - UFMG (Orientador)


Prof. Dr. Roque Luiz da Silva Pitangueira
DEES - UFMG (Coorientador)


Prof. Dr. Samuel Silva Penna
DEES - UFMG


Prof. Dr. Luis Paulo Barra
UFJF

Belo Horizonte, 26 de novembro de 2019

B748i

Botelho, Guilherme Garcia.

Sistema computacional orientado a objetos para análise termomecânica fisicamente não-linear em regime transiente via método dos elementos finitos [recurso eletrônico] / Guilherme Garcia Botelho. - 2019.

1 recurso online (xxv, 304 f. : il., color.) : pdf.

Orientador: Rodrigo Guerra Peixoto.

Coorientador: Roque Luiz da Silva Pitangueira.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 215-294.

Bibliografia: f. 295-304.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia de estruturas - Teses. 2. Método dos elementos finitos - Teses. 3. Software livre - Teses. I. Peixoto, Rodrigo Guerra. II. Pitangueira, Roque Luiz da Silva. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 624(043)



PROPEEs

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Programa de Pós-Graduação em Engenharia de Estruturas



UFMG

ATA DA DEFESA DE DISSERTAÇÃO DE MESTRADO EM ENGENHARIA DE ESTRUTURAS Nº: 363 do aluno **Guilherme Garcia Botelho**.

Às **14h** horas do dia **26** do mês de **novembro** de **2019**, reuniu-se, na Escola de Engenharia da Universidade Federal de Minas Gerais - UFMG, a Comissão Examinadora indicada pelo Colegiado do Programa em **18 de novembro de 2019**, para julgar a defesa da Dissertação de Mestrado intitulada "**Sistema Computacional Orientado a Objetos para Análise Termomecânica Fisicamente Não-Linear em Regime Transiente via Método dos Elementos Finitos**", cuja aprovação é um dos requisitos para a obtenção do Grau de MESTRE EM ENGENHARIA DE ESTRUTURAS na área de ESTRUTURAS.

Abrindo a sessão, o Presidente da Comissão, **Prof. Dr. Rodrigo Guerra Peixoto**, após dar a conhecer aos presentes o teor das Normas Regulamentares passou a palavra ao candidato para apresentação de seu trabalho. Seguiu-se a arguição pelos examinadores, com a respectiva defesa do candidato. Logo após, a Comissão se reuniu, sem a presença do candidato e do público, para julgamento e expedição do resultado final. Foram atribuídas as seguintes indicações:

(Aprov./Repr.)

Prof. Dr. Rodrigo Guerra Peixoto - DEES - UFMG (Orientador) APROVADO
Prof. Dr. Roque Luiz da Silva Pitangueira - DEES - UFMG (Coorientador) APROVADO
Prof. Dr. Samuel Silva Penna - DEES - UFMG APROVADO
Prof. Dr. Luis Paulo Barra - UFJF (a distância) APROVADO

Pelas indicações acima, o candidato foi considerado A PROVADO, conforme pareceres em anexo.

O resultado final foi comunicado publicamente ao candidato pelo Presidente da Comissão.

Nada mais havendo a tratar, o Presidente encerrou a reunião e lavrou a presente ATA, que será assinada por todos os membros participantes da Comissão Examinadora. Belo Horizonte, 26 de novembro de 2019.

Rodrigo Guerra Peixoto
Roque Luiz da Silva Pitangueira
Samuel Silva Penna
Luis Paulo Barra

Observações:

- 1) A aprovação do candidato na defesa da Dissertação de Mestrado não significa que o mesmo tenha cumprido todos os requisitos necessários para obtenção do Grau de Mestre em Engenharia de Estruturas;
- 2) Este documento não terá validade sem a assinatura e carimbo do Coordenador do Programa de Pós-Graduação.

Laws of Thermodynamics:

1 - You cannot win

2 - You cannot break even

3 - You cannot get out of the game.

C. P. Snow

A todos os professores que contribuíram para o meu aprendizado.

Índice

Índice	ii
Lista de Tabelas	vi
Lista de Figuras	x
Lista de Símbolos	xi
Resumo	xxiii
Abstract	xxiv
Agradecimentos	xxv
1 INTRODUÇÃO	1
1.1 Objetivos	7
1.1.1 Objetivo geral	7
1.1.2 Objetivos específicos	7
1.2 Organização do texto	9
2 REFERENCIAL TEÓRICO	11
2.1 Levantamento histórico	11
2.2 Primeira Lei da Termodinâmica	15
2.2.1 Sistemas uniformes	15
2.3 Formulação clássica do Problema de Transferência de Calor	18
2.3.1 Modos de transferência de calor	18
2.3.2 Teoria da condução de calor	19
2.3.3 Propriedades térmicas dependentes da temperatura	25
2.4 Formulação do problema termomecânico	25
2.4.1 Conservação da energia aplicada ao problema termomecânico	26
2.4.2 Considerações simplificadoras	33
2.4.3 Problema termomecânico desacoplado em regime quasi-estático	34
3 ABORDAGEM NUMÉRICA	36
3.1 Formulação do Problema de Transferência de Calor pelo Método dos Elementos Finitos	38

3.1.1	Método de Diferenças Finitas para a solução de problemas transientes formulados segundo o MEF	46
3.1.2	Transferência de calor em regime permanente	49
3.1.3	Transferência de calor fisicamente não-linear pelo MEF	50
3.2	Formulação do Problema da Mecânica dos Sólidos pelo Método dos Elementos Finitos	62
3.2.1	Dependência do coeficiente de expansão térmica com a temperatura	67
3.3	Formulação paramétrica do MEF	69
3.3.1	Integração numérica	74
3.4	Problema Termomecânico desacoplado em regime quasi-estático	76
4	IMPLEMENTAÇÃO COMPUTACIONAL	78
4.1	Visão geral do núcleo numérico do INSANE	79
4.2	Requisitos a serem atendidos	81
4.3	Expansões para a solução do PTC	83
4.3.1	Expansões no projeto <code>solution</code>	83
4.3.2	Expansões no pacote <code>assembler</code>	86
4.3.3	Expansões no projeto <code>model</code>	87
4.4	Inclusão do projeto <code>conditions</code>	97
4.4.1	Pacote <code>statevarboundaryconditions</code>	97
4.4.2	Pacote <code>initialconditions</code>	99
4.4.3	Pacote <code>conditions</code>	100
4.5	Expansões para gerenciamento da solução: projeto <code>simulationmanager</code>	102
4.5.1	Pacote <code>simulationmanager</code>	102
4.5.2	Pacote <code>setuper</code>	125
4.5.3	Pacote <code>datamanager</code>	134
4.6	Entidades com comportamento multifísico	141
4.7	Modificações para a solução do PTM	143
4.7.1	Modificações no projeto <code>solution</code>	143
4.7.2	Modificações no projeto <code>assembler</code>	147
4.7.3	Modificações no projeto <code>model</code>	149
4.8	Modificações no projeto <code>persistence</code>	154
5	EXEMPLOS DE VALIDAÇÃO	156
5.1	Problemas de Transferência de Calor	159
5.1.1	Exemplo 1 - Placa retangular com fluxos e temperaturas prescritas no contorno	159
5.1.2	Exemplo 2 - Esfera oca com temperaturas prescritas	164
5.1.3	Exemplo 3 - Barra com aquecimento senoidal nas extremidades	168
5.1.4	Exemplo 4 - Cilindro longo trocando calor por convecção na superfície	173
5.2	Problemas Termomecânicos	176
5.2.1	Exemplo 5 - Viga bi-apoiada com geração de calor	176

5.2.2	Exemplo 6 - Barra bi-engastada em aquecimento uniforme . . .	184
5.2.3	Exemplo 7 - Viga submetida a fluxo de calor prescrito em sua face superior	190
5.2.4	Exemplo 8 - Barra bi-engastada com aquecimento senoidal nas extremidades	198
6	CONSIDERAÇÕES FINAIS	207
6.1	Contribuições deste trabalho	207
6.2	Sugestões para trabalhos futuros	212
A	ANÁLISE DO NÚCLEO NUMÉRICO DO INSANE	215
A.1	Pontos de partida do processo de solução	215
A.2	Classe abstrata Solution	217
A.2.1	Processo de solução de problemas não-lineares	218
A.2.2	Adequações necessárias no pacote solution	221
A.3	Interface Assembler	223
A.4	Classe abstrata Model	225
A.4.1	Classe Node	227
A.4.2	Classe abstrata Element	228
A.4.3	Classe abstrata AnalysisModel	232
A.4.4	Classe abstrata ProblemDriver	234
A.4.5	Classe abstrata ConstitutiveModel	235
A.4.6	Classe abstrata Degeneration	237
A.4.7	Classe abstrata Material	238
A.4.8	Classe abstrata Shape	240
A.4.9	Pacote load	241
A.4.10	Classe abstrata ScalarFunction	243
A.5	Interface Persistence	244
B	SOLUÇÕES ANALÍTICAS DOS EXEMPLOS DE VALIDAÇÃO	247
B.1	Solução analítica do Exemplo 1 - Placa quadrada com temperaturas prescritas nas arestas	247
B.2	Solução analítica do Exemplo 2 - Esfera oca com temperaturas pres- critas nas superfícies	251
B.3	Solução analítica do Exemplo 3 - Barra com aquecimento senoidal nas extremidades	255
B.3.1	Período de elevação da temperatura nas bordas	256
B.3.2	Período de uniformização da temperatura na barra	259
B.4	Solução analítica do Exemplo 4 - Cilindro longo com convecção na superfície	261
B.5	Solução analítica do Exemplo 5 - Viga bi-apoiada sob gradiente tér- mico ao longo da altura	267
B.6	Solução analítica do Exemplo 6 - Barra bi-engastada sob variação de temperatura uniforme	270

B.7	Solução analítica do Exemplo 7 - Viga submetida a fluxo de calor na face superior	274
B.8	Solução analítica do Exemplo 8 - Barra bi-engastada sob aquecimento senoidal nas bordas	280
C	CLASSES EMPREGADAS NOS EXEMPLOS DE VALIDAÇÃO	287
C.1	Exemplo 1	287
C.2	Exemplo 2	288
C.3	Exemplo 3	289
C.4	Exemplo 4	290
C.5	Exemplo 5	291
C.6	Exemplo 6	292
C.7	Exemplo 7	293
C.8	Exemplo 8	294
	Referências Bibliográficas	295

Lista de Tabelas

5.1	Descrição resumida dos testes de validação realizados.	158
5.2	Simbologia adotada para as unidades de medida.	159
5.3	Parâmetros considerados na modelagem do Exemplo 1.	162
5.4	Parâmetros considerados na modelagem do Exemplo 2.	165
5.5	Parâmetros considerados na modelagem do Exemplo 3.	171
5.6	Parâmetros considerados na modelagem do Exemplo 4.	174
5.7	Parâmetros considerados na modelagem do Exemplo 5.	181
5.8	Parâmetros considerados na modelagem do Exemplo 6.	187
5.9	Comparação entre as deformações calculadas analítica e numericamente.	189
5.10	Comparação entre as tensões calculadas analítica e numericamente.	190
5.11	Parâmetros considerados na modelagem do Exemplo 7.	194
5.12	Comparação entre as flechas máximas calculadas analítica e numericamente.	197
5.13	Parâmetros considerados na modelagem do Exemplo 8.	202
C.1	Classes utilizadas na modelagem e análise do Exemplo 1.	287
C.2	Classes utilizadas na modelagem e análise do Exemplo 2.	288
C.3	Classes utilizadas na modelagem e análise do Exemplo 3.	289
C.4	Classes utilizadas na modelagem e análise do Exemplo 4.	290
C.5	Classes utilizadas na modelagem e análise do Exemplo 5.	291
C.6	Classes utilizadas na modelagem e análise do Exemplo 6.	292
C.7	Classes utilizadas na modelagem e análise do Exemplo 7.	293
C.8	Classes utilizadas na modelagem e análise do Exemplo 8.	294

Lista de Figuras

2.1	Publicações sobre termomecânica entre 1836 e 1955 (Boley, 1980). . .	12
2.2	Árvore genealógica do desenvolvimento da termomecânica (Adaptado de Maugin (1999)).	13
2.3	Desenho esquemático da placa considerada no desenvolvimento da teoria da condução de calor.	20
2.4	Volume de controle utilizado na formulação da equação da condução do calor (Bergman et al., 2011).	22
3.1	Fluxograma para solução do PTC transiente fisicamente não-linear. .	55
3.2	Fluxograma para solução do PTC fisicamente não-linear em regime permanente.	57
3.3	Mapeamento bidimensional de alguns elementos finitos (adaptado de Zienkiewicz e Taylor (2000)).	70
3.4	Fluxograma para solução do PTM transiente fisicamente não-linear. .	77
4.1	Padrão de cores utilizado na representação UML.	79
4.2	Diagrama de expansão da interface <code>Step</code>	83
4.3	Diagrama de classe da expansão de <code>Solution</code>	84
4.4	Diagrama de expansão da interface <code>Assembler</code>	86
4.5	Diagrama de expansão da classe <code>AnalysisModel</code>	88
4.6	Diagrama de expansão da classe <code>ProblemDriver</code>	89
4.7	Diagrama de expansão da classe <code>ConstitutiveModel</code>	91
4.8	Diagrama de expansão da classe <code>Material</code>	94
4.9	Diagrama de expansão da classe <code>Shape</code>	95
4.10	Modificações realizadas no pacote <code>load</code>	96
4.11	Diagrama das classes do pacote <code>statevarboundaryconditions</code>	99
4.12	Diagrama das classes do pacote <code>initialconditions</code>	100

4.13	Diagrama das classes do pacote <code>conditions</code>	101
4.14	Diagrama das classes do pacote <code>simulationmanager</code>	103
4.15	Diagrama da classes abstrata <code>SimulationManager</code>	103
4.16	Diagrama UML da classe <code>SolutionOutputMessage</code>	105
4.17	Diagrama da classes <code>CompatibilitySimulationManager</code>	105
4.18	Diagrama da classe <code>FemSimulationManager</code>	106
4.19	Diagrama da classe <code>SinglePhysicsFemSimManager</code>	108
4.20	Diagrama da classe <code>ThermoMechanicalSimulationManager</code>	110
4.21	Diagrama da classe <code>SteadyStateThermoMecSimManager</code>	116
4.22	Diagrama da classe <code>ScaledAndShiftedFunction</code>	123
4.23	Diagrama da classe <code>TimeDepThermoMecSimManager</code>	124
4.24	Diagrama de classes do pacote <code>setuper</code>	125
4.25	Diagrama da classe abstrata <code>Setuper</code>	126
4.26	Diagrama da classe <code>CompatibilitySetuper</code>	127
4.27	Diagrama da classe <code>FemSetuper</code>	129
4.28	Diagrama da classe <code>SinglePhysicsFemSetuper</code>	130
4.29	Diagrama da classe <code>TimeDepThermalSetuper</code>	131
4.30	Diagrama da classe <code>SharedModelSteadyStateTermoMecSetuper</code>	131
4.31	Diagrama das classes no pacote <code>datamanager</code>	134
4.32	Diagrama da classe abstrata <code>DataManager</code>	134
4.33	Diagrama da classe <code>SinglePhysicsDataManager</code>	135
4.34	Diagrama da classe <code>ThermoMechanicalDataManager</code>	136
4.35	Diagrama da classe <code>SharedModelThermoMecDataManager</code>	139
4.36	Diagrama da classe <code>ClonedModelsThermoMecDataManager</code>	141
4.37	Especialização da classe <code>AnalysisModel</code> agregando o comportamento termomecânico.	143
4.38	Modificações realizadas no pacote <code>element</code>	151
5.1	Placa estudada no Exemplo 1.	160
5.2	Representação bidimensional da placa retangular do Exemplo 1.	161
5.3	Distribuição de temperaturas calculada para a placa do Exemplo 1.	163
5.4	Resultados comparativos para a placa do Exemplo 1.	164
5.5	Modelagem computacional do setor da esfera oca do Exemplo 2.	166
5.6	Exemplo 2 - Distribuição de temperaturas calculadas pelo INSANE.	167

5.7	Comparativo entre as soluções analítica e numérica do Exemplo 2. . .	167
5.8	Função que descreve o valor da temperatura prescrita nas bordas da barra.	168
5.9	Barra com temperaturas prescritas nas extremidades - Exemplo 3. . .	169
5.10	Resultados comparativos durante a elevação da temperatura nas bordas.	172
5.11	Comparativo entre as soluções durante o período de uniformização. . .	172
5.12	Cilindro longo exposto à convecção em sua superfície.	173
5.13	Discretização da seção axissimétrica modelada no Exemplo 4.	175
5.14	Comparativo entre as soluções analítica e numérica do Exemplo 4. . .	176
5.15	Viga submetida a geração de calor e temperaturas prescritas - Ex. 5.	177
5.16	Discretização da viga do exemplo 5 contendo as condições de contorno da parcela mecânica.	180
5.17	Distribuição de temperaturas calculada pelo INSANE - Exemplo 5. . .	181
5.18	Comparativo entre os campos de temperatura analítico e numérico. . .	182
5.19	Campo de deslocamentos na direção Y calculado pelo INSANE - Ex. 5.	182
5.20	Comparativo entre as linhas elásticas analítica e numérica.	183
5.21	Tensões normais na direção X calculadas pelo INSANE - Ex. 5. . . .	183
5.22	Comparativo entre os campos de tensão normal na direção axial. . . .	184
5.23	Barra bi-engastada sob aquecimento uniforme - Exemplo 6.	185
5.24	Curva tensão-deformação para a barra do Exemplo 6.	186
5.25	Relação entre a tensão e a deformação ao longo da análise do Exemplo 6. Valores em módulo.	188
5.26	Ampliação da região em regime plástico.	189
5.27	Viga submetida a fluxo de calor na face superior - Exemplo 7.	191
5.28	Discretização da viga do exemplo 7 contendo as condições de contorno da parcela mecânica.	194
5.29	Resultados comparativos das distribuições de temperaturas.	195
5.30	Campo de temperaturas calculado pelo INSANE em $t = 600 \text{ ute}$	196
5.31	Deslocamentos em y calculado pelo INSANE em $t = 600 \text{ ute}$	196
5.32	Resultados comparativos das distribuições de tensões.	197
5.33	Tensões σ_{xx} calculadas pelo INSANE em $t = 600 \text{ ute}$	198
5.34	Barra bi-engastada sob aquecimento senoidal nas extremidades - Exemplo 8.	199

5.35	Resultados comparativos do campo de deslocamentos durante a elevação da temperatura nas bordas.	204
5.36	Resultados comparativos do campo de deslocamentos após o término da elevação da temperatura nas bordas.	204
5.37	Comparativo entre a evolução temporal da deformação calculada analítica e numericamente.	205
5.38	Comparativo entre a evolução temporal da tensão calculada analítica e numericamente.	205
5.39	Ampliação da região próxima ao escoamento no comparativo das tensões.	206
6.1	Classificação das sugestões para trabalhos futuros.	214
A.1	Diagrama de classe para <code>Solution</code>	217
A.2	Diagrama de classe para <code>Step</code>	219
A.3	Diagrama de classe para <code>IterativeStrategy</code>	220
A.4	Diagrama de classe para <code>Assembler</code>	225
A.5	Diagrama de classe para <code>Model</code>	226
A.6	Diagrama de classe para <code>FemModel</code>	226
A.7	Diagrama de classe para <code>Node</code>	228
A.8	Diagrama de classe para <code>Element</code>	228
A.9	Diagrama de classe para <code>AnalysisModel</code>	233
A.10	Diagrama de classe para <code>ProblemDriver</code>	234
A.11	Diagrama de classe para <code>ConstitutiveModel</code>	236
A.12	Diagrama de classe para <code>Degeneration</code>	238
A.13	Diagrama de classe para <code>Material</code>	239
A.14	Diagrama de classe para <code>Shape</code>	241
A.15	Diagrama UML das classe contidas no pacote <code>load</code>	242
A.16	Diagrama de classe para <code>ScalarFunction</code>	244
B.1	Problema modificado com base na simetria do problema original.	256

Lista de Símbolos

Capítulo 2

\mathbf{X}	Variáveis de estado
t	Tempo
t_0	Instante inicial de tempo
t_1	Instante de tempo após o início do processo
W	Trabalho
\mathcal{U}	Energia interna do sistema uniforme
Q	Calor recebido ou perdido pelo corpo
θ	Temperatura
$\Delta\theta$	Diferença de temperaturas
A	Área
h	Espessura
k	Condutividade térmica
q''	Fluxo de calor
\mathbf{D}	Tensor constitutivo
q_x, q_y, q_z	Taxas de transferência de calor nas direções cartesianas

ρ	Massa específica
c_p	Calor específico
\dot{e}	Taxa de geração ou de dissipação de calor
E_g	Energia gerada no volume de controle
\dot{U}	Energia acumulada no volume de controle
\mathbf{b}	Vetor de forças de corpo
\mathbf{p}	Vetor de forças de superfície
\mathbf{u}	Deslocamento
\mathbf{v}	Velocidade
\mathcal{K}	Energia cinética
γ	Função densidade de energia interna
$\boldsymbol{\sigma}$	Tensor de tensões de Cauchy
\mathbf{n}	Vetor normal à superfície
Ω	Domínio espacial do corpo
Γ	Superfície do corpo
$\boldsymbol{\omega}$	Tensor de rotações
$\boldsymbol{\epsilon}$	Tensor de deformações infinitesimais
$\boldsymbol{\xi}_a$	Parcela não-dissipativa da componente de deformação
\mathbf{d}_m	Parcela dissipativa da componente de deformação
$\boldsymbol{\sigma}_a$	Componente de tensão conjugada de $\boldsymbol{\xi}_a$

\mathfrak{s}_a	Componente de tensão conjugada de \mathbf{d}_m
F, Λ	Funções matemáticas para assegurar a integrabilidade
T	Temperatura medida na escala absoluta
S	Entropia do sistema uniforme
η	Função densidade de entropia
φ	Energia livre
$I_\epsilon, II_\epsilon, III_\epsilon$	Invariantes do tensor de deformações infinitesimais
T_0	Temperatura de referência no estado livre de tensões
T'	Temperatura adimensional
δ_{ij}	Delta de Kronecker
λ, μ	Constantes de Lamé
α	Coefficiente de expansão térmica
v_e	Velocidade de propagação das ondas de dilatação

Capítulo 3

\mathbf{D}	Tensor constitutivo
θ	Temperatura
\dot{e}	Taxa de geração ou de dissipação de calor
ρ	Massa específica
c_p	Calor específico
R	Resíduo

ψ	Função peso
E_ψ	Conjunto de funções para a descrição da função peso
E_θ	Conjunto de funções para a descrição do campo de temperaturas
\mathbf{n}	Vetor normal à superfície
$H^1(\Omega)$	Primeiro espaço de Sobolev
\mathbf{q}_s''	Fluxo de calor prescrito na superfície
E_θ^n, E_ψ^n	Sub-espços dos espaços de funções E_ψ e E_θ
\mathbf{N}	Matriz das funções de interpolação do elemento finito
$\boldsymbol{\theta}^e$	Vetor das temperaturas nos nós do elemento finito
$\boldsymbol{\psi}^e$	Vetor dos valores da função peso nos nós do elemento finito
\mathbf{L}	Matriz operadora de derivadas
\mathbf{B}	Matriz das derivadas parciais das funções de interpolação
\mathbf{R}^e	Vetor de resíduo do elemento finito
\mathbf{C}^e	Matriz de capacitância do elemento finito
\mathbf{K}^e	Matriz de condutividade ou de rigidez do elemento finito, dependendo do contexto
$\boldsymbol{\Lambda}^e$	Parcela condutiva da matriz de condutividade do elemento finito
\mathbf{H}^e	Parcela convectiva da matriz de condutividade do elemento finito
\mathbf{F}^e	Vetor das ações externas que atuam sobre o elemento finito
\mathbf{F}_e^e	Vetor de carregamento nodal equivalente relativo à geração de calor no interior do elemento finito

$\mathbf{F}_{q_p}^e$	Vetor de carregamento nodal equivalente devido à prescrição de fluxo de calor por unidade de área no contorno do elemento
$\mathbf{F}_{q_c}^e$	Vetor do carregamento nodal equivalente devido à fluxos de calor concentrados prescritos no contorno
\mathbf{F}_h^e	Vetor do carregamento nodal equivalente devido às trocas de calor por convecção na superfície do elemento finito
h	Coefficiente de troca de calor por convecção
$\boldsymbol{\theta}$	Vetor contendo os valores de temperatura em cada um dos nós do meio discretizado
\mathbf{A}^e	Matriz de indexação dos graus de liberdade do elemento finito
\mathbf{R}	Vetor do resíduo do meio discretizado
\mathbf{F}	Vetor das ações externas do meio discretizado
\mathbf{C}	Matriz de capacitância do meio discretizado
\mathbf{K}	Matriz de condutividade ou de rigidez do meio discretizado, dependendo do contexto
β	Parâmetro de ponderação do método de Newmark
Δt	Tamanho do passo de tempo no método de Newmark
$\widetilde{\mathbf{K}}, \widetilde{\mathbf{F}}$	Matrizes equivalentes no método de Newmark, obtidas a partir das matrizes \mathbf{C} e \mathbf{K} e dos parâmetros β e Δt
$\widetilde{\mathbf{K}}^e, \widetilde{\mathbf{F}}^e$	Matrizes equivalentes no método de Newmark, a nível elemental
\mathbf{M}	Matriz de iteração em problemas fisicamente não-lineares
$\boldsymbol{\delta\theta}$	Vetor do incremento no campo de temperaturas em problemas fisicamente não-lineares

$k'(\theta)$	Condutividade térmica tangente do material
D'	Tensor constitutivo tangente
K'^e	Matriz de condutividade tangente do elemento finito
σ	Tensor de tensões de Cauchy
b	Vetor de forças de corpo
p	Vetor de forças de superfície
$\delta\varepsilon$	Vetor das derivadas parciais da função peso, ou vetor das deformações virtuais.
u	Campo de deslocamentos
u^e	Vetor dos deslocamentos nodais em um elemento finito
ε	Tensor de deformações infinitesimais
ε_{mec}	Tensor das deformações líquidas decorrentes das ações mecânicas externas
ε_θ	Tensor correspondente às deformações ocasionadas pela variação da temperatura
α	Coefficiente de expansão térmica
θ_0	Temperatura de referência do corpo previamente à atuação das ações mecânicas externas
$F_{eq(\theta)}^e$	Carregamento nodal equivalente devidos à variação da temperatura
$F_{eq(b)}^e$	Carregamento nodal equivalente devidos às forças de corpo
$F_{eq(p)}^e$	Carregamento nodal equivalente devidos às forças de superfície

L	Comprimento
T	Temperatura medida na escala absoluta
T_i, T_f	Temperaturas inicial e final
t_i, t_f	Instantes de tempo inicial e final
$\mathbf{x}^e, \mathbf{y}^e, \mathbf{z}^e$	Vetores das coordenadas x, y e z dos nós do elemento finito
ξ, η, ζ	Coordenadas naturais, definidas em um sistema local no elemento finito
\mathbf{J}	Matriz jacobiana
I	Resultado da integração numérica
W_i	Peso associado ao ponto de integração i

Capítulo 5 - Exemplo 5.1.1

a, b, t	Largura, altura e espessura da placa retangular
T_∞	Temperatura do fluido em contato com a aresta BC
T_{AB}	Temperatura prescrita na face lateral que contém a aresta AB
h_c	Coefficiente de troca de calor por convecção
T	Temperatura
x, y	Coordenadas espaciais
h	Razão entre o coeficiente convectivo e a condutividade térmica
α_n	Raízes da equação transcendental $\alpha \tan(\alpha a) = h$

Capítulo 5 - Exemplo 5.1.2

r	Posição radial
-----	----------------

R_a, R_b	Raios interno e externo da esfera oca
T_a, T_b	Temperaturas nas superfícies interna e externa da esfera
k_a, k_b	Condutividades térmicas às temperaturas T_a e T_b

Capítulo 5 - Exemplo 5.1.3

t	Tempo decorrido desde o início da análise
p	Tempo de elevação da temperatura nas bordas da barra
L	Metade do comprimento da barra
θ	Temperatura prescrita nas bordas
α	Difusividade térmica
k	Condutividade térmica
ρ	Massa específica
c_p	Calor específico à pressão constante
s	Posição ao longo da barra
A_m, β_m, C_n	Coefficientes de cálculo

Capítulo 5 - Exemplo 5.1.4

T	Temperatura
t	Tempo decorrido desde o instante inicial
T_∞	Temperatura do fluido em contato com a superfície do cilindro
h	Coefficiente de troca de calor por convecção
T_0	Distribuição de temperatura inicial no cilindro

a	Raio do cilindro
$\lambda(T)$	Condutividade térmica dependente da temperatura
ρ	Massa específica
c_p	Calor específico
β	Parâmetro do método Newmark- β

Capítulo 5 - Exemplo 5.2.1

L	Comprimento da viga
a, b	Altura e largura da seção transversal da viga
\dot{e}	Calor gerado uniformemente no domínio da viga
T_{sup}, T_{inf}	Temperaturas prescritas nas faces superior e inferior da viga
x	Posição na direção longitudinal da viga
y	Posição na direção da altura da viga
k	Condutividade térmica
P_T, M_{Tz}	Esforço normal e momento fletor relacionados à variação da temperatura
γ	Coefficiente de expansão térmica
A	Área da seção transversal
E	Módulo de elasticidade
v	Flecha da viga
σ	Tensão normal na direção longitudinal da viga
ν	Coefficiente de Poisson

Capítulo 5 - Exemplo 5.2.2

t	Tempo decorrido desde o início da análise
p	Tempo de elevação da temperatura nas bordas da barra
L	Metade do comprimento da barra
θ	Temperatura prescrita nas bordas
α	Difusividade térmica
k	Condutividade térmica
ρ	Massa específica
c_p	Calor específico à pressão constante
s	Posição ao longo da barra
A_m, β_m, C_n	Coefficientes de cálculo
ϵ_{mec}	Parcela do tensor de deformações associada aos esforços mecânicos
γ	Coefficiente de expansão térmica
ϵ_y, f_y	Deformação e tensão relacionadas ao início do escoamento
E_0, E_{ep}	Módulo de elasticidade elástico e elastoplástico
σ	Tensão normal na direção longitudinal da barra
A	Área da seção transversal da barra

Capítulo 5 - Exemplo 5.2.3

L	Comprimento da viga
a, b	Altura e largura da seção transversal da viga

q''	Calor prescrito na face superior da viga
t	Tempo decorrido desde o instante inicial
T_0	Temperatura inicial da viga, uniforme
x	Posição na direção longitudinal da viga
y	Posição na direção da altura da viga
k	Condutividade térmica
ρ	Massa específica
c_p	Calor específico
α	Difusividade térmica
P_T, M_{Tz}	Esforço normal e momento fletor relacionados à variação da temperatura
γ	Coefficiente de expansão térmica
A	Área da seção transversal
I_z	Momento de inércia da seção em relação ao eixo z
E	Módulo de elasticidade
v	Flecha da viga
σ_x	Tensão normal na direção longitudinal da viga
ν	Coefficiente de Poisson

Capítulo 5 - Exemplo 5.2.4

t	Tempo decorrido desde o início da análise
p	Tempo de elevação da temperatura nas bordas da barra

L	Metade do comprimento da barra
θ	Temperatura prescrita nas bordas
α	Difusividade térmica
k	Condutividade térmica
ρ	Massa específica
c_p	Calor específico à pressão constante
s	Posição ao longo da barra
A_m, β_m, C_n	Coefficientes de cálculo
$W(m, t)$	Função auxiliar para cálculo
u_T, u_M	Campos de deslocamentos associados à temperatura e aos esforços mecânicos
ϵ_{mec}	Parcela do tensor de deformações associada aos esforços mecânicos
γ	Coefficiente de expansão térmica
ϵ_y, f_y	Deformação e tensão relacionadas ao início do escoamento
E_0, E_{ep}	Módulo de elasticidade elástico e elastoplástico
σ	Tensão normal na direção longitudinal da barra
A	Área da seção transversal da barra

Resumo

Essa dissertação de mestrado se refere à implementação de recursos no ambiente computacional livre INSANE (*INteractive Structural ANalysis Environment*), que é escrito conforme o paradigma de programação orientada a objetos, para a solução de problemas termomecânicos fisicamente não-lineares no regime transiente. A discretização espacial do problema é realizada segundo a formulação paramétrica do Método dos Elementos Finitos e a discretização temporal é realizada por uma expansão em Diferenças Finitas. O escopo se restringe a situações onde a variação da temperatura ocorre durante um regime quasi-estático. Nesse tipo de regime, tanto a variação da temperatura como a variação das ações mecânicas ocorrem de forma lenta e gradual, permitindo que o calor gerado durante o processo de deformação possa ser desprezado, bem como o calor gerado por fenômenos dissipativos. Ao remover esses termos, obtém-se um conjunto de equações diferenciais desacopladas que viabiliza a solução do problema térmico de forma independente do problema mecânico. Todavia, para a solução do problema mecânico é necessário conhecer previamente a solução térmica, uma vez que a variação da temperatura contribui para a deformação do corpo analisado. Para gerir o processo de solução, uma nova entidade denominada *SimulationManager* foi introduzida no projeto orientado a objetos do núcleo numérico do programa. As novas classes, bem como todas as demais modificações realizadas, foram validadas a partir da análise de problemas cuja solução analítica é conhecida.

Palavras-Chave: Análise Termomecânica, Método dos Elementos Finitos, Software Livre

Abstract

This master's thesis refers to the implementations of tools in the free computational environment INSANE (*INteractive Structural ANalysis Environment*), written accordingly with the object-oriented programming paradigm, for the solution of physically non-linear thermomechanical problems in the time-dependent regime. The spatial discretization follows the parametric formulation of the Finite Element Method (FEM), and the temporal discretization follows an expansion in Finite Differences (FD). The scope is limited to situations where the temperature variation occur in a quasi-static regime. In this regime, both the temperature variation and the mechanical actions change slowly, allowing neglect the heat produced during the deformation process, as well as the heat produced by dissipative phenomena. Removing these terms we obtain a set of decoupled partial differential equations, enabling the solution of the thermal problem independently of the mechanical problem. However, to solve the mechanical problem, the thermal solution should be known in advance, since the temperature variation contributes to the strain field in the body. To manage the solution process, we introduced a new entity in the object-oriented design of the software's numerical core, named SimulationManager. The analysis of problems whose analytical solution is previously known validated the new classes, as well as all other modifications made in the source code.

Keywords: Thermomechanical Analysis, Finite Element Method, Free Software

Agradecimentos

À minha família e às pessoas amadas próximas a mim, que incentivaram essa conquista importante.

Ao professor Roque Pitangueira, pelo seu grande apoio desde o início dessa pesquisa, pelas suas contribuições fundamentais ao longo desse trajeto, pelos ensinamentos, pelo seu carisma e pela sua compreensão quanto aos períodos que precisei me distanciar da vida acadêmica para me dedicar à vida profissional.

Ao professor Rodrigo Peixoto, pelo seu envolvimento e dedicação. Sua serenidade e ensinamentos foram essenciais no processo de compreensão da fundamentação teórica e na busca por formas de aplicá-la no contexto desse trabalho.

Aos demais professores do Departamento de Engenharia de Estruturas da UFMG, tanto pela contribuição à minha formação como pelo acolhimento durante essa etapa.

Ao colega Humberto Monteiro, cujas reuniões para buscarmos soluções em conjunto foram muito proveitosas e cujas contribuições foram de fundamental importância.

À equipe do INSANE, tanto pelo companheirismo como pela dedicação e seriedade de seus integrantes, cujas pesquisas suportaram as minhas contribuições.

Aos colegas do departamento, por terem compartilhado momentos de alegria e descontração. Ficarão para a história os Churrascos, as festas juninas, as lembranças e as amizades que seguiram para além da universidade.

Aos funcionários do Departamento de Engenharia de Estruturas da UFMG, por toda prestatividade durante essa caminhada.

Às agências brasileira de fomento à pesquisa, pelo apoio financeiro.

Capítulo 1

INTRODUÇÃO

A termoelasticidade é a área do conhecimento que estuda o comportamento dos corpos deformáveis sob a interação mútua dos campos de deformações e de temperatura (Nowacki, 1970). Ela representa, portanto, uma generalização da Teoria da Elasticidade e da Teoria da Condução do Calor (Parkus, 1976), e se relaciona com outras áreas do conhecimento, como a Termodinâmica e a Matemática Aplicada (Maugin, 1999; Hetnarski e Eslami, 2009).

Problemas termomecânicos ocorrem em diferentes ramos da engenharia e têm recebido considerável atenção da comunidade científica (Johns e Neal, 1965). Ilustrativamente, podem ser mencionadas aplicações práticas ligadas às indústrias siderúrgica (Schacht, 1993), automobilística (Cerit, 2011; Dippel et al., 2014; Jian e Shui, 2017), aeroespacial (Thornton, 1996), ferroviária (Wu et al., 2011) e de construção civil (Kumar e Kodur, 2017). Também destacam-se estudos voltados para a simulação computacional de processos de fabricação por soldagem (Kim et al., 2005; Sluzalec, 2005), por conformação mecânica (Badreddine et al., 2016) e por usinagem (Shih e Yang, 1993; Lei et al., 1999), bem como processos de endurecimento superficial por têmpera (Kang e Im, 2007) e por esmerilhamento (Kolkwitz et al., 2011).

Segundo observado inicialmente em experimentos e depois formulado matematicamente por Duhamel (1837), a atuação de carregamentos dinâmicos em um corpo

ocasiona, além de deslocamentos, variações de temperatura no mesmo. Reciprocamente, a introdução ou remoção de calor de um corpo promove o surgimento de deformações, além de provocar variações de temperatura (Nowacki, 1970).

Conforme detalhadamente descrito por Booley e Weiner (1997), o fluxo de calor que surge durante o processo de deformação ocasiona um aumento da entropia e, portanto, a variação da energia ocorre de uma forma irreversível. Este fenômeno é conhecido como dissipação termoelástica. Essa dissipação é representada na equação corrigida da condução de calor por um termo proporcional à taxa de variação do tensor de deformações, usualmente denominado termo de acoplamento termoelástico.

Todavia, na maioria das aplicações práticas de engenharia é possível simplificar a modelagem do problema sem a introdução de erros significativos nos resultados da análise. Dentre as possíveis simplificações, ressalta-se a omissão do termo de acoplamento termoelástico na equação corrigida da condução de calor, o que desacopla a análise térmica da mecânica.

O desacoplamento dispensa a solução simultânea da equação diferencial parcial da condução de calor do conjunto de equações diferenciais parciais que compreende as equações do movimento, as equações de equilíbrio, a relação constitutiva e as relações cinemáticas. Estratégias de solução para o caso acoplado podem ser encontradas na bibliografia, como em Riedlbauer et al. (2014) e em Bouery (2012).

O acoplamento termoelástico somente pode ser desprezado se não existirem variações bruscas ou descontinuidades no histórico temporal do campo de temperaturas e se os carregamentos mecânicos forem aplicados de forma lenta e gradual, na mesma escala temporal em que ocorre a variação da temperatura.

As condições exigidas para negligenciar o acoplamento termoelástico na equação corrigida da condução de calor correspondem a um tipo particular de regime transiente, denominado regime quasi-estático. Dentro dessas condições, também é usual desprezar os termos inerciais na equação do movimento, que se reduz à equação de

equilíbrio após tal consideração.

Ressalta-se que durante a determinação de uma relação entre o campo de tensões e o campo de deformações, obtém-se um termo associado ao surgimento de deformações devidas à variação da temperatura. Este termo é mantido na lei constitutiva nas análises termomecânicas desacopladas.

Outra simplificação frequentemente realizada consiste em modelar o problema térmico segundo a Lei de Fourier, o que corresponde à teoria clássica da termoelasticidade. Esta simplificação possui três deficiências principais, a saber: velocidade infinita de propagação dos distúrbios termoelásticos, resposta termoelástica insatisfatória de um sólido quando a introdução de calor ocorre de forma extremamente curta e concentrada como em pulsos de laser, e descrição precária do comportamento termoelástico a temperaturas próximas ao zero absoluto (Hetnarski e Ignaczak, 1999).

Segundo apresentado por Ignaczak e Ostoja-Starzewski (2009) e por Hetnarski e Ignaczak (1999), diversos modelos foram desenvolvidos para contornar as deficiências da Lei de Fourier, constituindo uma teoria denominada Termoelasticidade Generalizada. Nestes modelos, a condução do calor passa a ter um comportamento ondulatório no qual a velocidade de propagação da onda de calor é finita. Este fenômeno é denominado Segundo Som. Ilustrativamente, pode-se mencionar como uma aplicação prática da teoria generalizada da termoelasticidade a análise termoelástica dinâmica de meios fissurados durante choques térmicos (Esmati et al., 2018).

As discussões realizadas até o presente momento se referem ao comportamento elástico do material. Caso o material trabalhe em um regime inelástico, também surgirá na equação corrigida da condução de calor um termo associado aos fenômenos dissipativos do processo de deformação inelástica. Conforme discutido em Eraslan e Orcan (2002), o calor gerado por fenômenos dissipativos também pode ser desprezado se a variação de temperatura que ele proporcionar for significativamente inferior à causada pelas ações térmicas externas.

Um outro aspecto importante a ser discutido durante a modelagem de problemas termomecânicos é a dependência das propriedades físicas em relação à temperatura (Hetnarski e Achenbach, 1996). Nos casos em que a variação da temperatura ocorre em uma pequena faixa, estas propriedades podem ser consideradas constantes. Caso contrário, atenção especial deve ser dada à formulação matemática a fim de considerar essa dependência (Barron e Barron, 2012).

Na análise termomecânica acoplada, os problemas podem se tornar fisicamente não-lineares se qualquer uma das propriedades físicas adotadas na modelagem possuir dependência com a temperatura. Adicionalmente, as demais fontes de não-linearidade física, como a dependência da deformação na lei tensão-deformação, também tornam o problema termomecânico fisicamente não-linear (Nicholson, 2003).

Caso o problema termomecânico (PTM) seja modelado de forma desacoplada, existirá não-linearidade na parcela térmica se uma ou mais dentre as propriedades Calor Específico, Densidade e Condutividade Térmica dependerem da temperatura (Carslaw e Jaeger, 1959). Soluções analíticas podem ser encontradas para alguns casos particulares, porém em geral é necessário recorrer a métodos numéricos para resolver o problema termomecânico devido à sua complexidade matemática (Hetnarski e Eslami, 2009).

Diversas técnicas numéricas para a solução de problemas contínuos regidos por equações diferenciais parciais vem sendo desenvolvidas há algumas décadas, sendo o Método dos Elementos Finitos (MEF) uma técnica consolidada disponível para tal finalidade (Zienkiewicz e Taylor, 2000). Segundo Hetnarski e Eslami (2009), o MEF é uma ferramenta poderosa para a solução de problemas termomecânicos, especialmente por sua elevada taxa de convergência.

Além de análises termomecânicas pelo MEF, podem ser encontrados na literatura, a título ilustrativo, trabalhos utilizando o Método dos Elementos Finitos com suavização baseada em faces (Feng et al., 2013), o Método dos Elementos Finitos Generalizado também conhecido como Método dos Elementos Finitos Estendido

(Plews e Duarte, 2016; Esmati et al., 2018) , o Método dos Elementos de Contorno (Matsumoto et al., 2005) , Métodos Sem Malha como o Método de Diferenças Finitas Generalizado (Hosseini, 2013) e o Método Local Petrov-Galerkin (Sladek et al., 2009) e o método Phase-Field (Miehe, Schänzel, e Ulmer, 2015; Miehe, Hofacker, Schänzel, e Aldakheel, 2015), dentre outros métodos numéricos.

Segundo Zienkiewicz e Taylor (2000), o MEF é uma técnica numérica baseada na subdivisão do domínio em componentes individuais, denominados elementos finitos, cujo comportamento é previamente conhecido. Esses componentes são então combinados de forma adequada para reconstruir o sistema original, com o objetivo de fornecer uma aproximação do comportamento do corpo.

A subdivisão do domínio, denominada discretização, simplifica o processo de solução por aproximar o problema contínuo, governado por equações diferenciais parciais, por meio de um problema discreto, regido por um sistema de equações algébricas. Como a qualidade dessa aproximação melhora à medida que o número de componentes individuais aumenta, é necessário resolver um grande sistema de equações nas situações de aplicação prática, justificando a implementação computacional do método.

Belytschko et al. (2014) apresenta uma revisão da evolução histórica do MEF aplicado a análises não-lineares, indicando algumas ferramentas computacionais desenvolvidas nos centros de pesquisa ao longo dos anos.

Os programas de computador desenvolvidos para a solução numérica de problemas de engenharia segundo o MEF podem ser divididos em duas categorias: programas livres ou proprietários. Denomina-se Software Livre aquele que respeita a liberdade e a concordância do usuário. Neste senso, o usuário possui a liberdade de executar, copiar, distribuir, estudar, modificar e melhorar o programa de computador. Quando o usuário não possui controle sobre o programa, diz-se que o mesmo é proprietário (Stallman e Gay, 2015).

Atualmente existem vários projetos de software livre para a solução de problemas

de engenharia regidos por equações diferenciais parciais com o auxílio do MEF. Uma breve lista de alguns desses projetos é apresentada a seguir:

- Agros2D (Karban et al., 2013) ;
- Calculix (Dhondt, 2004) ;
- Elmer (Jussila e Ruokolainen, 2007) ;
- FEniCS (Logg, 2007; Logg et al., 2012) ;
- FreeFEM++ (Hecht, 2012) ;
- GetFEM++ (Putanowicz et al., 2013) ;
- INSANE (Pitangueira et al., 2008);
- LibMesh (Kirk et al., 2006; Ratzet, 2017).

O sistema computacional INSANE - *INteractive Structural ANalysis Environ-ment*, disponível em <https://www.insane.dees.ufmg.br/>, é um software livre multiplataforma, escrito na linguagem de programação Java segundo o paradigma de Programação Orientada a Objetos (POO). O programa é desenvolvido desde 2004 no Laboratório de Software Livre - *INSANE Lab*, pertencente ao Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais (UFMG), com o objetivo de ser utilizado principalmente como um recurso didático e como uma ferramenta de pesquisa.

Além de conter recursos para a solução de problemas utilizando o MEF, o INSANE atualmente também é capaz de resolver problemas da mecânica dos sólidos utilizando as técnicas de discretização: Método dos Elementos Finitos Generalizado/Estendido (Alves et al., 2013) , Método dos Elementos de Contorno (Peixoto et al., 2018) e Métodos sem Malha (Gori et al., 2017).

1.1 Objetivos

1.1.1 Objetivo geral

Esta dissertação tem por objetivo a implementação computacional de recursos no núcleo numérico do ambiente computacional INSANE para a solução de problemas termomecânicos, utilizando como técnica de discretização o MEF. A implementação se restringe ao caso quasi-estático, no qual os termos inerciais nas equações do movimento e o termo de acoplamento termoelástico são desprezados. A implementação também se restringe aos casos nos quais as trocas de calor por radiação podem ser desprezadas. Quanto às propriedades físicas, permite-se uma dependência polinomial entre a condutividade térmica e a temperatura, o que introduz não-linearidade física na parcela térmica. Quanto à parcela mecânica do problema termomecânico, estuda-se o comportamento do corpo na presença de deformações oriundas da variação da temperatura.

1.1.2 Objetivos específicos

A solução de um problema termomecânico desacoplado pode ser subdividida em duas macroetapas, onde a primeira consiste na solução de um Problema de Transferência de Calor (PTC) para obter a variação da distribuição de temperaturas, e a segunda consiste na solução de um problema da mecânica dos sólidos contemplando o efeito da variação da temperatura calculada na primeira etapa.

Os recursos previamente disponíveis no INSANE para a solução do PTC e do PTM com o auxílio do MEF (Botelho et al., 2015) se restringiam à análise linear em regime permanente.

A fim de solucionar o PTC no regime transiente considerando a não-linearidade física introduzida pela dependência entre a condutividade térmica e a temperatura, tornou-se necessário expandir as entidades do núcleo numérico relacionadas à solução do problema de transferência de calor.

A implementação das novas ferramentas relativas ao PTC foi realizada em fases, a saber:

1. PTC fisicamente não-linear em regime estacionário;
2. PTC linear em regime transiente;
3. PTC fisicamente não-linear em regime transiente.

Para generalizar as análises multifísicas, onde a solução de cada física pode ser conduzida por um algoritmo diferente de acordo com as características do problema analisado, é recomendável a existência de uma entidade para controlar o fluxo de execução durante o processo global de obtenção da solução.

Outra questão relativa à multifísica é a necessidade de especificar, sobre a mesma geometria, condições iniciais e condições de contorno específicas para cada uma das físicas envolvidas. Especialmente em relação às condições de contorno, há a necessidade de introduzir na parcela mecânica a condição de contorno correspondente ao efeito da variação da temperatura, calculada na primeira etapa.

A fim de generalizar a implementação para qualquer caso multifísico, propõe-se neste trabalho que o gerenciador da solução também seja responsável pela manipulação das condições iniciais e das condições de contorno.

Por fim, por serem referentes ao mesmo meio, cada física possui suas particularidades no seu respectivo problema discreto. Portanto, também é necessária uma configuração específica para cada física, que também pode ser coordenada pelo gerenciador do processo de solução.

Persegue-se, portanto, como um objetivo específico, a implementação do gerenciador da solução atendendo aos critérios supramencionados.

Concluída a implementação do PTC e existindo um gerenciador do processo de solução, pôde-se avançar para a última etapa deste trabalho, que consiste em implementar algoritmos para atender às seguintes demandas:

- Geração e aplicação automática dos carregamentos oriundos da variação da temperatura no problema mecânico;
- Solução da parcela mecânica de problemas fisicamente não-lineares que contenham carregamentos de variação de temperatura;
- Correção do cálculo das tensões quando houver a presença de deformações oriundas de variações de temperatura.

Após a implementação dessas tarefas, realizou-se a implementação final do PTM em fases como foi realizado para o PTC:

1. PTM linear em regime transiente;
2. PTM fisicamente não-linear em regime estacionário;
3. PTM fisicamente não-linear em regime transiente.

1.2 Organização do texto

Após esse capítulo introdutório, no qual é brevemente detalhada uma visão geral sobre o tema tratado, são apresentados cinco capítulos e três apêndices.

No capítulo 2 é realizada uma revisão histórica da análise termomecânica, acompanhada da formulação matemática clássica do problema. Neste capítulo, as simplificações descritas até o presente momento são detalhadas.

No capítulo 3 é apresentada a formulação do PTC e do PTM segundo o MEF, e no capítulo 4 são apresentadas as modificações realizadas no núcleo numérico do INSANE para atender aos objetivos mencionados neste capítulo introdutório.

Os testes da implementação encontram-se no capítulo 5, e as conclusões relativas a esse trabalho se encontram no capítulo 6, bem como a proposição de trabalhos futuros que continuem a pesquisa apresentada nessa dissertação.

Uma análise detalhada do núcleo numérico do INSANE indicando os pontos que necessitavam de mudanças para que as ferramentas propostas nessa dissertação fossem introduzidas no programa é apresentada no Apêndice A.

O apêndice B contém o desenvolvimento das soluções analíticas dos exemplos de validação tratados no capítulo 5, e o apêndice C contém tabelas indicando as entidades do núcleo numérico empregadas durante a modelagem e solução dos exemplos de validação.

Capítulo 2

REFERENCIAL TEÓRICO

Este capítulo apresenta uma breve revisão teórica sobre a análise termomecânica. Primeiramente é exposto um levantamento histórico da evolução dessa área do conhecimento, seguido de uma revisão sobre os conceitos básicos da termodinâmica clássica e da formulação do problema de transferência de calor em materiais isotrópicos. Posteriormente, é feita uma apresentação da termodinâmica do contínuo em conjunto com a formulação do problema termomecânico acoplado. Ao final, as aproximações simplificadoras são discutidas, explicitando o desacoplamento entre o problema térmico e o problema mecânico no caso particular tratado nessa dissertação de mestrado.

2.1 Levantamento histórico

Segundo detalhadamente descrito por Boley (1980) em (Hasselman e Heller, 1980) e por Hetnarski e Eslami (2009), o desenvolvimento da teoria da termoelasticidade se iniciou com a leitura do trabalho de Duhamel (1837) pela Academia Francesa de Ciências. A teoria apresentada por J. C. Duhamel foi publicada poucos anos após a publicação de C.L.M.H. Navier sobre os fundamentos da teoria da elasticidade (Navier, 1821), lido também pela Academia Francesa de Ciências em 1821 e publicado em 1827, e após a publicação do tratado de J.B.J. Fourier sobre a teoria da transferência de calor (Fourier, 1822).

Pouco avanço ocorreu em um período de aproximadamente 60 anos após a publicação de J. C. Duhamel, como ilustrado na Figura 2.1. Após esse período, as principais contribuições à formulação das equações da termoelasticidade são atribuídas a Neumann (1885), a Almansi (1897), a Tedone (1906) e a Voigt (1910).

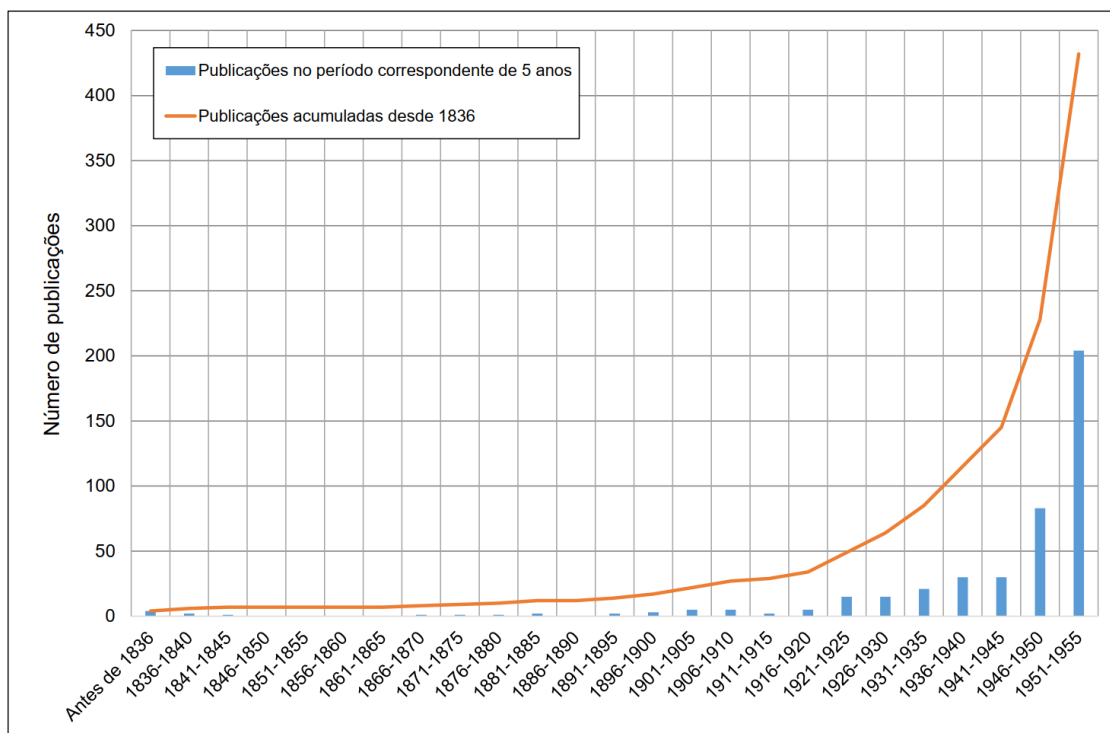


Figura 2.1: Publicações sobre termomecânica entre 1836 e 1955 (Boley, 1980).

Contudo, apesar do trabalho apresentado por J. C. Duhamel conter a derivação das equações para o acoplamento entre o campo de temperaturas e o campo de deformações em um corpo sólido, foi somente 120 anos depois que uma demonstração completa para a equação da condutividade térmica foi apresentada por Biot (1956), que se baseou na Termodinâmica dos Processos Irreversíveis. O trabalho de Biot também inclui métodos básicos tanto para a solução da equação da termoelasticidade, como para a formulação de uma abordagem variacional (Nowacki, 1970).

Uma breve revisão dos principais trabalhos na área da termodinâmica até o estabelecimento da teoria utilizada por Biot para a sua formulação variacional do problema termomecânico é apresentada a seguir.

Como descrito por Maugin (1999), os desenvolvimentos da termodinâmica se iniciaram principalmente em 1824 com o trabalho de Carnot (1986) e com o trabalho de Clapeyron (1834), que construíram as noções de termometria e calorimetria. Estes conceitos introduziram a noção de trabalho, que posteriormente se concretizou na primeira e na segunda lei da termodinâmica, formalmente postuladas por Clausius (1851).

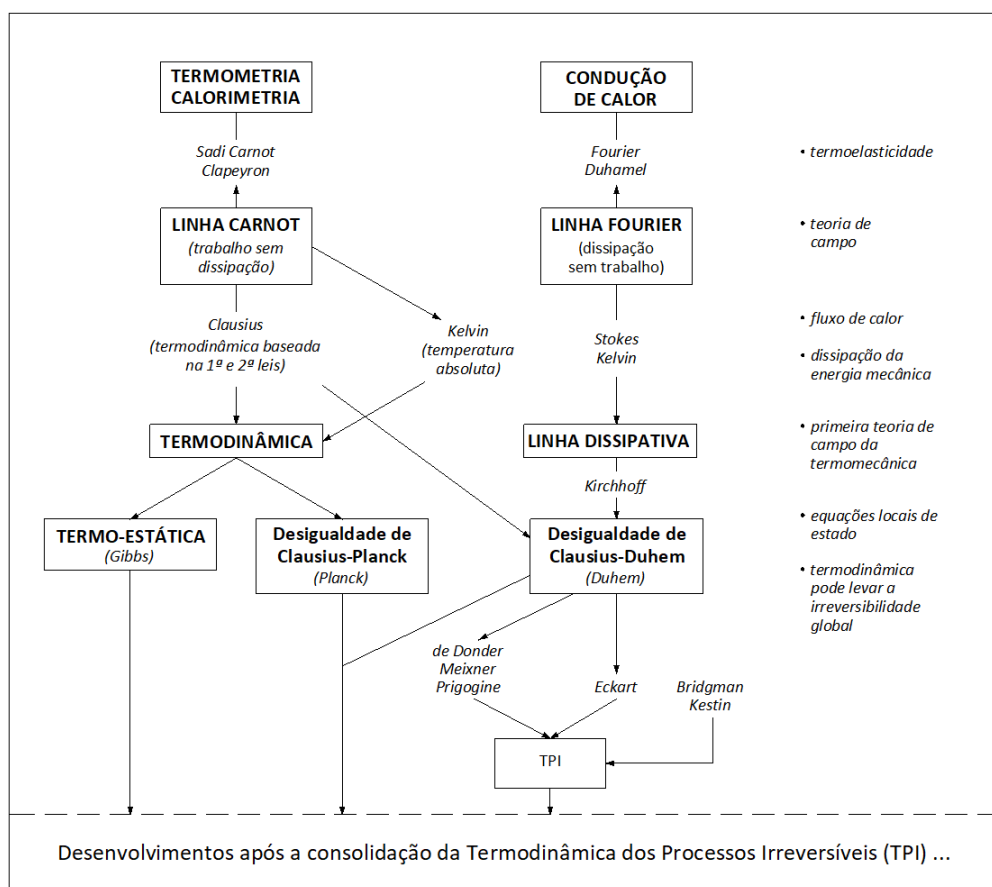


Figura 2.2: Árvore genealógica do desenvolvimento da termomecânica (Adaptado de Maugin (1999)).

Especificamente em relação à segunda lei da termodinâmica, cuja essência é atribuída a R. von Mayer (1842) e a William Thomson, também conhecido como Lord Kelvin (Thomson e Tait, 1867), na maioria dos casos a energia deve mudar a sua natureza para que ela seja conservada. Essa mudança de qualidade, segundo Clausius, ocorre somente em uma direção, implicando em fenômenos não-conservativos.

Atribui-se também a R. Clausius a criação do conceito de entropia, utilizado para qualificar o que muda quando a energia é conservada.

O princípio de conservação da energia foi então desenvolvido detalhadamente por Gibbs (1928), no domínio da termoestática, e o princípio do não-decrécimo da entropia foi amplamente discutido nos trabalhos de Planck (1903), levando à desigualdade de Clausius-Planck.

Os primeiros cruzamentos entre a termodinâmica e a termoelasticidade ocorreram com o trabalho de G. G. Stokes, relativo ao reconhecimento do vetor de fluxo de calor, e com o trabalho de Lord Kelvin sobre a dissipação da energia mecânica, como ilustrado na Figura 2.2.

As contribuições de Kirchhoff, mencionadas em Jakubowska (1982), consolidaram a primeira teoria de campo da termomecânica, e após essa consolidação a referida linha de pesquisa levou à formulação da desigualdade de Clausius-Duhem por Duhem (1911), que introduziu claramente as equações locais de estado enquanto reconhecia que a termodinâmica pode levar a irreversibilidades globais.

Após os trabalhos de Gibbs, Planck e Duhem, Théophile de Donder consolidou uma teoria padronizada para a termodinâmica, denominada Teoria dos Processos Irreversíveis (de Donder e van Rysselberghe, 1936), abreviada como TPI. na Figura 2.2.

Diversos avanços ocorreram posteriormente, em especial quanto à definição do conceito de entropia com as contribuições de Maxwell (1871) e de Ludwig Boltzmann, que teve um dos seus trabalhos mais proeminentes na termodinâmica traduzidos para o inglês por Sharp e Matschinsky (2015), e com a contribuição de Rayleigh (1899), importante na caracterização de processos irreversíveis em sistemas macroscópicos em diversos estados e na definição de funções de dissipação. Também ressalta-se a proposição de uma equação para o balanço de entropia por Bridgman (1943, 1950), que foi utilizada para provar que a dissipação térmica deve ser sempre positiva.

As evoluções descritas até o presente momento são suficientes para caracterizar

historicamente o leitor em relação à formulação matemática da teoria termomecânica clássica em sua forma geral, apresentada na seção 2.4 deste capítulo. Maiores detalhes sobre os desenvolvimentos da termodinâmica após Duhem podem ser consultados em Maugin (1999).

2.2 Primeira Lei da Termodinâmica

Na presente seção, os aspectos termodinâmicos serão revisitados, sendo apresentada a formulação matemática da primeira lei da termodinâmica conforme detalhado em Booley e Weiner (1997).

2.2.1 Sistemas uniformes

Inicia-se definindo variáveis de estado como o conjunto de valores numéricos relacionados às grandezas físicas observáveis utilizadas para descrever o comportamento de um sistema em um dado instante de tempo. Define-se também processo adiabático como sendo aquele que ocorre em um sistema que não troca calor com a sua vizinhança. Adicionalmente, diz-se que um sistema encontra-se em equilíbrio termodinâmico se as variáveis de estado não se modificam ao longo do tempo.

Se um sistema for retirado de um estado de equilíbrio termodinâmico no instante de tempo $t = t_0$, caracterizado pelas variáveis de estado $\mathbf{X} = \mathbf{X}(t_0)$, e levado por meio de um processo adiabático a um novo estado de equilíbrio no instante de tempo $t = t_1$, caracterizado pelas variáveis de estado $\mathbf{X} = \mathbf{X}(t_1)$, pode-se afirmar que o trabalho total realizado pelo sistema, definido a seguir, é independente do processo adiabático particular empregado, ou seja, o trabalho depende somente das variáveis de estado observadas nos extremos do intervalo temporal.

$$W = \int_{t_0}^{t_1} \dot{W}(t) dt \quad (2.1)$$

Denomina-se energia interna (\mathcal{U}) o trabalho realizado pelo sistema em um processo adiabático que retira o sistema de uma condição de referência arbitrária e o leva ao estado em questão. Sendo a referência fixa, \mathcal{U} é uma função de estado definida como:

$$\mathcal{U} = \mathcal{U}(X_1, \dots, X_j) \quad (2.2)$$

Observa que a energia interna é uma propriedade extensiva, ou seja, a energia interna de um sistema compreendido por dois ou mais subsistemas é igual à soma das energias internas de todos os subsistemas.

Considerando um processo arbitrário conectando dois pontos de extremidade caracterizados por $\mathbf{X}(t_0)$ e $\mathbf{X}(t_1)$, onde $\mathcal{U}[\mathbf{X}_1(t_0), \dots, \mathbf{X}_j(t_0)] = \mathcal{U}(t_0)$ e $\mathcal{U}[\mathbf{X}_1(t_1), \dots, \mathbf{X}_j(t_1)] = \mathcal{U}(t_1)$, e no qual o trabalho total seja expresso pela Equação 2.1, pode-se escrever para um processo adiabático:

$$[\mathcal{U}(t_1) - \mathcal{U}(t_0)] - W = 0 \quad (2.3)$$

Nos casos em que o processo não for adiabático, a diferença entre a variação da energia interna e o trabalho não será nula. Essa diferença é então denominada calor (Q):

$$[\mathcal{U}(t_1) - \mathcal{U}(t_0)] - W = Q \quad (2.4)$$

Os valores de W e de Q serão diferentes para diferentes processos conectando os mesmos pontos de extremidade, todavia a soma ($W + Q$) é independente do processo particular empregado visto que ela depende apenas da função de estado \mathcal{U} .

A Equação 2.3 também pode ser expressa na forma de taxa ou na forma diferencial:

$$\dot{\mathcal{U}} = \dot{W} + \dot{Q} \quad (2.5a)$$

$$d\mathcal{U} = dW + dQ \quad (2.5b)$$

Considerando dois sistemas, 1 e 2, em contato perfeito um com o outro, mas separados de suas vizinhanças por fronteiras adiabáticas, tem-se:

$$\dot{\mathcal{U}}_1 = \dot{W}_1 + \dot{Q}_1 \quad (2.6a)$$

$$\dot{\mathcal{U}}_2 = \dot{W}_2 + \dot{Q}_2 \quad (2.6b)$$

Como a energia interna é uma propriedade extensiva, a variação da energia interna do sistema constituído pela união dos dois sistemas 1 e 2 é igual à soma da variação das energias internas desses sistemas separadamente. Adicionalmente, como o sistema composto está impedido de trocar calor com a vizinhança, a variação do calor desse sistema composto deve ser nula. Logo,

$$\dot{\mathcal{U}} = \dot{W}_1 + \dot{W}_2 \quad (2.7)$$

O resultado anterior implica em:

$$\dot{Q}_1 = -\dot{Q}_2 \quad (2.8)$$

Desta forma, é razoável definir calor como a energia sendo transferida de um sistema para o outro. Sendo assim, a Equação 2.5 pode ser interpretada como uma formulação matemática da lei da conservação da energia, usualmente referida como primeira lei da termodinâmica.

2.3 Formulação clássica do Problema de Transferência de Calor

Esta seção apresenta uma breve revisão dos modos de transferência de calor a fim de complementar a definição de calor apresentada na seção anterior. Como essa dissertação de mestrado se insere no âmbito da mecânica dos sólidos, será apresentada na sequência a formulação matemática clássica das equações pertinentes à condução de calor em sólidos, que posteriormente serão utilizadas no desenvolvimento da teoria termomecânica.

Como detalhadamente descrito por Bergman et al. (2011), a transferência de calor ocorre na natureza em três modos distintos: condução, convecção e radiação térmica.

2.3.1 Modos de transferência de calor

2.3.1.1 Condução

A condução, em nível atômico e molecular, pode ser vista como a transferência de energia das partículas mais energéticas para as menos energéticas do material devido às interações entre mesmas. Em um gás que não possua movimento global, ou macroscópico, e que apresente gradiente térmico, esta transferência de energia ocorre a partir da transmissão da energia cinética de partículas com temperatura mais alta para partículas de temperaturas mais baixa, à medida que elas colidem. Um comportamento muito similar pode ser observado nos líquidos, embora as moléculas estejam mais próximas que em um gás e as interações sejam mais fortes e mais frequentes. Analogamente, em um sólido a condução pode ser atribuída à atividade atômica na forma de vibração dos retículos. Portanto, na presença de gradiente de temperatura a transferência de energia por condução deve ocorrer na direção da diminuição da temperatura.

2.3.1.2 Convecção

A transferência de calor por convecção é caracterizada pela combinação de dois fenômenos: transferência de energia devido ao movimento molecular aleatório do fluido e pelo transporte de energia devido à movimentação macroscópica do fluido. A convecção pode ser classificada de acordo com a natureza do escoamento do fluido, sendo subdividida em convecção forçada e convecção natural. A convecção forçada ocorre quando o escoamento é induzido por meios externos, tais como um ventilador, uma bomba, ou ventos atmosféricos. Já a convecção natural ocorre quando o escoamento é induzido por forças de empuxo, que são originadas a partir de diferenças de densidades causadas por variações de temperatura no fluido.

2.3.1.3 Radiação

A radiação térmica é a energia emitida pela matéria que se encontra a uma temperatura não-nula. Independentemente do estado da matéria, a emissão de energia pode ser atribuída a mudanças nas configurações eletrônicas dos átomos ou moléculas que a constituem. A energia do campo de radiação é transportada por ondas eletromagnéticas, portanto, não necessitando de um meio material para a sua propagação.

2.3.2 Teoria da condução de calor

Segundo Carslaw e Jaeger (1959) a teoria da condução do calor foi construída embasada em observações experimentais. O primeiro ensaio, que posteriormente levou à Lei de Fourier, será descrito a seguir.

Considera-se a placa mostrada na Figura 2.3, na qual o comprimento e a largura sejam muito maiores que a espessura e cujas superfícies superior e inferior estejam submetidas a diferentes temperaturas. O centro do sistema de coordenadas se encontra no centro da placa, e seus eixos são perpendiculares às superfícies que formam o contorno da placa.

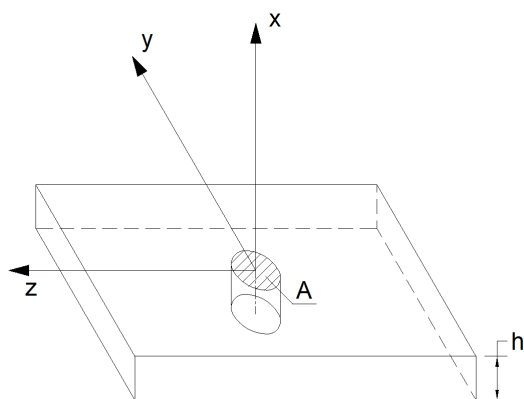


Figura 2.3: Desenho esquemático da placa considerada no desenvolvimento da teoria da condução de calor.

Se as temperaturas nas superfícies superior e inferior forem mantidas constantes ao longo do tempo, a distribuição de temperaturas na placa não irá se alterar. Restringe-se essa diferença de temperaturas a uma faixa na qual as propriedades físicas do material da placa possam ser consideradas constantes.

Dadas as condições descritas acima, escolhe-se como volume de controle um cilindro pertencente ao domínio da placa, distante das bordas, e cujo eixo seja normal às superfícies superior e inferior.

Neste tipo de ensaio observa-se que o calor total Q transmitido através do cilindro após t segundos é diretamente proporcional à diferença de temperaturas entre as superfícies da placa, $\Delta\theta$, à área da seção transversal do cilindro, A , ao tempo de observação, e é inversamente proporcional à espessura da placa, h . Também se observa que o calor transmitido nesse intervalo de tempo se modifica ao se modificar o material da placa. Portanto, introduz-se uma constante k denominada condutividade térmica que possui um valor específico para cada material a cada temperatura. Matematicamente, pode-se expressar:

$$Q = k A t \frac{\Delta\theta}{h} \quad (2.9)$$

Denomina-se fluxo de calor q'' a taxa na qual calor atravessa uma determinada

superfície em um ponto do corpo, expresso por unidade de área. Analisando um caso limite no qual a espessura h da placa tenda a zero, o fluxo de calor que a atravessa é definido como:

$$q''_x = -k \frac{d\theta}{dx} \quad (2.10)$$

O sinal negativo é introduzido para enfatizar que o fluxo de calor é positivo na direção de diminuição da temperatura.

Pode-se mostrar que o fluxo de calor em um ponto genérico P é uma grandeza vetorial, cujas componentes em cada uma das direções independentes do sistema de coordenadas são, para materiais isotrópicos:

$$q''_i = -k\theta_{,i} \quad (2.11)$$

com i correspondente a cada uma das direções espaciais independentes.

A Equação 2.11 pode ser generalizada a fim de contemplar materiais anisotrópicos. Para tal, a condutividade térmica é representada por um tensor de segunda ordem \mathbf{D} e assume-se que cada componente do vetor de fluxo de calor em um ponto é uma função linear das componentes do gradiente de temperatura nesse mesmo ponto (Carslaw e Jaeger, 1959). Dessa forma,

$$q''_i = -D_{ij}\theta_{,j} \quad (2.12)$$

A equação anterior naturalmente contempla o caso particular de materiais isotrópicos, onde o tensor constitutivo é dado por $D_{ij} = k \cdot \delta_{ij}$.

2.3.2.1 Equação da Condução do Calor

O desenvolvimento matemático para a obtenção de uma equação que permita calcular a distribuição de temperaturas em um corpo sólido parte da aplicação da primeira lei da termodinâmica ao elemento diferencial mostrado na Figura 2.4.

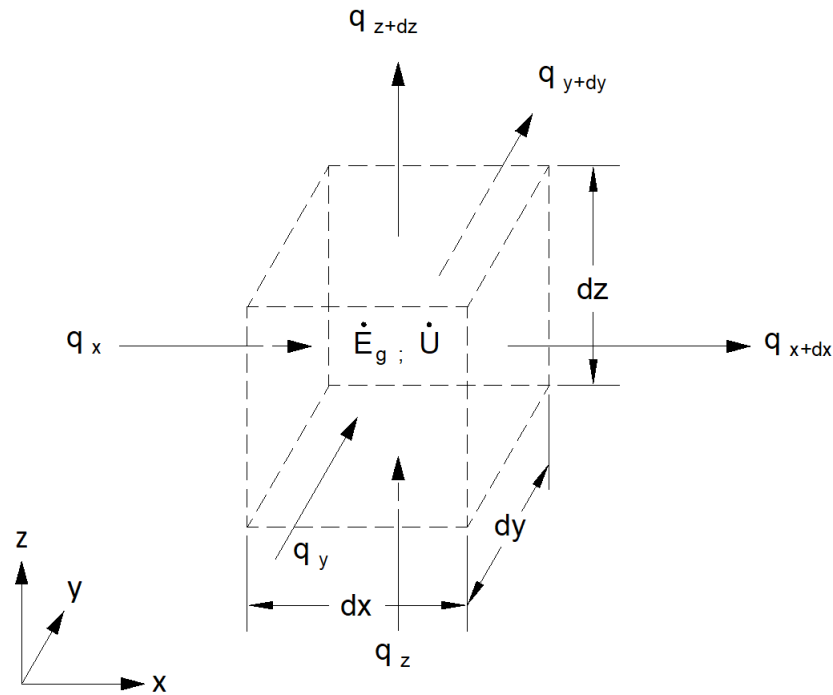


Figura 2.4: Volume de controle utilizado na formulação da equação da condução do calor (Bergman et al., 2011).

Como dentro do elemento diferencial pode ocorrer um processo que gere ou consome energia térmica, tem-se a taxa de variação da energia \dot{E}_g causada por esse fenômeno:

$$\dot{E}_g = \dot{e} \, dx \, dy \, dz \quad (2.13)$$

onde \dot{e} é a taxa na qual energia é gerada ou consumida, por unidade de volume do meio.

Além de poder gerar ou consumir energia, a quantidade de energia interna, \mathcal{U} , acumulada pela matéria no elemento diferencial pode aumentar ou diminuir ao longo do tempo. Essa variação pode ser matematicamente explicitada por:

$$\dot{\mathcal{U}} = \rho \, c_p \, \frac{\partial \theta}{\partial t} \, dx \, dy \, dz \quad (2.14)$$

onde ρ é a massa específica e c_p é o calor específico a pressão constante.

As taxas de transferência de calor por condução nas superfícies de controle que passam pelo ponto $P(x, y, z)$ são q_x , q_y e q_z . Para determinar as taxas correspondentes nas superfícies das faces que passam pelo ponto $P(x + dx, y + dy, z + dz)$, realiza-se uma expansão em série de Taylor. Desprezando os termos de ordem superior à linear, tem-se:

$$q_{x+dx} = q_x + \frac{\partial q_x}{\partial x} dx \quad (2.15a)$$

$$q_{y+dy} = q_y + \frac{\partial q_y}{\partial y} dy \quad (2.15b)$$

$$q_{z+dz} = q_z + \frac{\partial q_z}{\partial z} dz \quad (2.15c)$$

Realizando o balanço de energia descrito pela primeira lei da termodinâmica, Equação 2.5a, ao volume de controle da Figura 2.4, pode-se equacionar:

$$q_x + q_y + q_z + \dot{e} dx dy dz - q_{x+dx} - q_{y+dy} - q_{z+dz} = \rho c_p \frac{\partial \theta}{\partial t} dx dy dz \quad (2.16)$$

Substituindo as equações 2.15a, 2.15b e 2.15c na Equação 2.16, obtém-se:

$$-\frac{\partial q_x}{\partial x} dx - \frac{\partial q_y}{\partial y} dy - \frac{\partial q_z}{\partial z} dz + \dot{e} dx dy dz = \rho c_p \frac{\partial \theta}{\partial t} dx dy dz \quad (2.17)$$

Utilizando a relação entre o fluxo de calor e o gradiente de temperaturas descrito pela Lei de Fourier em materiais isotrópicos, Equação 2.11, e multiplicando pela correspondente área da seção transversal do elemento diferencial em cada uma das direções cartesianas para determinar a taxa de transferência de calor a partir do fluxo de calor, pode-se escrever:

$$q_x = -k dy dz \frac{\partial \theta}{\partial x} \quad (2.18a)$$

$$q_y = -k dy dz \frac{\partial \theta}{\partial y} \quad (2.18b)$$

$$q_z = -k dy dz \frac{\partial \theta}{\partial z} \quad (2.18c)$$

Substituindo as equações 2.18a, 2.18b e 2.18c na Equação 2.17, obtém-se a equação da difusão do calor em coordenadas cartesianas, para materiais isotrópicos:

$$\frac{\partial}{\partial x} \left(k \frac{\partial \theta}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial \theta}{\partial y} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial \theta}{\partial z} \right) + \dot{e} = \rho c_p \frac{\partial \theta}{\partial t} \quad (2.19)$$

que na forma indicial pode ser escrita como:

$$(k\theta_{,i})_{,i} + \dot{e} = \rho c_p \dot{\theta} \quad (2.20)$$

2.3.2.2 Condições de contorno do Problema de Transferência de Calor

Para solucionar a equação diferencial parcial da difusão do calor, Equação 2.19, é necessário introduzir as condições de contorno e a condição inicial.

A condição inicial se refere ao estado que o corpo se encontra no instante de referência para o começo da análise. Devem ser especificadas as temperaturas e os fluxos de calor em todo o domínio neste instante de tempo.

As condições de contorno se referem às condições físicas existentes na fronteira do sólido, podendo essas serem constantes ou variáveis ao longo do tempo. As condições de contorno mais comuns estão listadas abaixo:

1. Temperatura prescrita;
2. Fluxo térmico prescrito na superfície;
3. Troca de calor por convecção na superfície;
4. Troca de calor por radiação na superfície.

Ressalta-se que o caso de uma superfície isolada termicamente equivale matematicamente a prescrever um fluxo de calor nulo na superfície.

Nesta dissertação de mestrado não será abordada a troca de calor por radiação visto que a mesma demanda um tratamento especial devido à sua não-linearidade, o que excede o escopo previsto para este trabalho.

2.3.3 Propriedades térmicas dependentes da temperatura

Segundo Carslaw e Jaeger (1959), rigorosamente falando as propriedades térmicas dos materiais variam com a temperatura. Quando a faixa na qual ocorre a variação da temperatura é estreita, pode-se negligenciar essa dependência.

Quando a dependência das propriedades térmicas com a temperatura não pode ser negligenciada, a Equação 2.19 se torna uma equação diferencial não-linear, cuja solução analítica não é trivial. Algumas soluções podem ser encontradas na literatura para geometrias particulares, todavia em casos genéricos torna-se necessário recorrer a técnicas numéricas para obter a distribuição de temperaturas.

2.4 Formulação do problema termomecânico

Na presente seção será realizada uma breve apresentação da formulação termomecânica acoplada em sua forma mais geral, conforme detalhado em Booley e Weiner (1997). Devido à consideração da presença de campos de temperaturas não uniformes, torna-se necessário estender os aspectos termodinâmicos apresentados na seção 2.2 a sistemas não-uniformes. Para tal, utiliza-se a teoria da termodinâmica do contínuo.

Primeiramente, a primeira lei da termodinâmica será aplicada ao problema termomecânico, onde será considerado o trabalho realizado pelas ações externas e pela resposta mecânica interna do material, a energia cinética adquirida pelo corpo durante o processo de deformação e o calor trocado com a vizinhança.

Em seguida, a parcela dissipativa será isolada na equação da conservação da energia e o teorema matemático de Carathéodory será empregado para que a formulação atenda à segunda lei da termodinâmica.

Para concluir o desenvolvimento, uma função potencial de energia livre será introduzida na equação da conservação da energia, o que irá conduzir às equações acopladas que descrevem o problema termomecânico, a saber: a lei tensão-deformação

contemplando o efeito das deformações devidas à variação da temperatura, e a equação corrigida da equação do calor, que contempla o termo dissipativo termoelástico. Finalmente, serão aplicadas as simplificações que desacoplam os problemas térmico e mecânico.

2.4.1 Conservação da energia aplicada ao problema termo-mecânico

A taxa na qual trabalho é realizado por forças de corpo por unidade de massa \mathbf{b} e forças de superfície \mathbf{p} sobre um corpo compreendido pelo domínio Ω envolto pela superfície Γ pode ser calculada por:

$$\dot{W}(t) = \int_{\Omega} \rho [b_i(\mathbf{X}, t)v_i(\mathbf{X}, t)] dV + \int_{\Gamma} [p_i(\mathbf{X}, t)v_i(\mathbf{X}, t)] dA \quad (2.21)$$

onde \mathbf{v} é a velocidade, $\mathbf{v} \equiv \dot{\mathbf{u}}$.

A energia cinética é dada pela expressão:

$$\mathcal{K} = \frac{1}{2} \int_{\Omega} \rho v_i v_i dV \quad (2.22)$$

Dada a função de densidade de energia interna $\gamma(\mathbf{X}, t)$, medida em energia por unidade de massa, pode-se obter a energia interna total pela integração:

$$\mathcal{U} = \int_{\Omega} \rho \gamma dV \quad (2.23)$$

O equilíbrio entre as forças de superfície e as tensões em um ponto da superfície do corpo é expresso por:

$$p_i = \sigma_{ij} n_j \quad (2.24)$$

onde $\boldsymbol{\sigma}$ é o tensor de tensões de Cauchy e \mathbf{n} é um vetor unitário normal à superfície Γ .

Considerando que ocorre transferência de calor para o corpo através da superfície Γ a uma taxa $-\mathbf{q}''(\mathbf{X}, t)$ por unidade de área, a taxa total de transferência de energia por esse mecanismo é dada por:

$$q_{ent} = - \int_{\Gamma} q''_i n_i dA \quad (2.25)$$

A lei da conservação da energia implica que a taxa na qual as forças de corpo e de superfície realizam trabalho sobre o corpo adicionadas à energia transferida ao corpo por calor é igual ao aumento da soma entre as energias interna e cinética, levando a:

$$\int_{\Gamma} \sigma_{ij} n_j v_i dA + \int_{\Omega} \rho b_i v_i dV - \int_{\Gamma} q''_i n_i dA = \frac{d}{dt} \left(\frac{1}{2} \int_{\Omega} \rho v_i v_i dV + \int_{\Omega} \rho \gamma dV \right) \quad (2.26)$$

Aplicando o teorema da divergência ao primeiro e ao terceiro termos:

$$\int_{\Gamma} \sigma_{ij} v_i n_j dA = \int_{\Omega} (\sigma_{ij} v_i)_{,j} dV \quad (2.27)$$

$$\int_{\Gamma} q''_i n_i dA = \int_{\Omega} q''_{i,i} dV \quad (2.28)$$

Além disso, baseando-se na equação da continuidade que expressa a conservação da massa, pode-se inverter a ordem de diferenciação e de integração nos termos do lado direito da igualdade na Equação 2.26:

$$\frac{d}{dt} \left(\frac{1}{2} \int_{\Omega} \rho v_i v_i dV + \int_{\Omega} \rho \gamma dV \right) = \int_{\Omega} \rho \dot{v}_i v_i dV + \int_{\Omega} \rho \dot{\gamma} dV \quad (2.29)$$

Substituindo as equações 2.27 a 2.29 na equação 2.26 e utilizando um processo de integração por partes, tem-se:

$$\int_{\Omega} [(\sigma_{ij,j} + \rho b_i - \rho \dot{v}_i)] v_i dV + \int_{\Omega} [(\sigma_{ij} v_{i,j} - q''_{i,i} - \rho \dot{\gamma})] dV = 0 \quad (2.30)$$

onde a primeira integral na equação acima é nula por corresponder à equação de equilíbrio. Assumindo que a equação 2.30 seja válida para qualquer ponto do domínio do corpo, tem-se que o integrando dessa equação deve ser identicamente nulo. Logo,

$$\sigma_{ij}v_{i,j} - q''_{i,i} - \rho\dot{\gamma} = 0 \quad (2.31)$$

Efetuada uma decomposição do gradiente de velocidades em sua parcela simétrica e anti-simétrica:

$$v_{i,j} = \frac{1}{2}(v_{i,j} + v_{j,i}) + \frac{1}{2}(v_{i,j} - v_{j,i}) = \dot{\epsilon}_{ij} + \dot{\omega}_{ij} \quad (2.32)$$

sendo ϵ o tensor de deformações infinitesimais.

Como o produto $\sigma_{ij}\omega_{ij}$ é nulo visto que σ_{ij} é simétrico e ω_{ij} é anti-simétrico, a Equação 2.31 pode ser escrita como:

$$\sigma_{ij}\dot{\epsilon}_{ij} - q''_{i,i} = \rho\dot{\gamma} \quad (2.33)$$

Assume-se então uma separação entre as parcelas dissipativas e as parcelas conservativas no produto entre o tensor de tensões e a variação temporal do tensor de deformações. Para tanto, cada componente independente dos respectivos tensores é representada por uma variável característica:

$\xi_a = \textit{conserv} \epsilon_{rs}$: deformações não-dissipativas onde $a = 1, \dots, 6$ e $r, s = 1, 2, 3$;

$d_m = \textit{dissip} \epsilon_{rs}$: deformações dissipativas onde $m = 1, \dots, 6$ e $r, s = 1, 2, 3$;

σ_a : componente de tensão conjugada de ξ_a ;

ς_m : componente de tensão conjugada de d_m .

A Equação 2.33 pode ser reescrita após a separação entre as parcelas conservativa e dissipativa:

$$\varsigma_m \dot{d}_m + \sigma_a \dot{\xi}_a - q''_{i,i} = \rho\dot{\gamma} \quad (2.34)$$

Aplicando o teorema de Caratheodory (1909), abordado em detalhes por Zachmanoglou (1971), às variáveis de estado em um ponto genérico do corpo, obtém-se a equação abaixo para que a mudança de estado em um ponto genérico do corpo ocorra de forma irreversível, acompanhada de dissipação de energia na forma de calor para as vizinhanças do ponto:

$$\frac{1}{\Lambda(\xi_a, \theta)} \left[\left(\rho \frac{\partial \gamma}{\partial \xi_a} - \sigma_a \right) \dot{\xi}_a + \rho \frac{\partial \gamma}{\partial \theta} \dot{\theta} \right] = \dot{F}(\xi_a, \theta) \quad (2.35)$$

onde $\Lambda(\xi_a, \theta)$ e $F(\xi_a, \theta)$ são funções matemáticas a serem determinadas para assegurar a integrabilidade no teorema Carathéodory, como descrito em detalhes em Booley e Weiner (1997).

Reescrevendo a Equação 2.34 em termos das novas variáveis Λ e F , e entendendo que $\gamma = \gamma(\xi_a, \theta)$:

$$\rho \dot{\gamma} - \sigma_a \dot{\xi}_a = \varsigma_m d_m - q''_{i,i} = \Lambda(\xi_1, \dots, \xi_6, \theta) \dot{F}(\xi_1, \dots, \xi_6, \theta) \quad (2.36)$$

Restringindo a aplicação do teorema de Carathéodory a um processo onde $d_m(t) = 0$, isto é, um processo não dissipativo, pode-se reescrever a Equação 2.36 como:

$$-q''_{i,i} = \Lambda(\xi_1, \dots, \xi_6, \theta) \dot{F}(\xi_1, \dots, \xi_6, \theta) \quad (2.37)$$

A integração da Equação 2.37 sobre o volume do corpo, utilizando o teorema da divergência, leva a:

$$\dot{Q} = \int_{\Omega} \Lambda(\xi_1, \dots, \xi_6, \theta) \dot{F}(\xi_1, \dots, \xi_6, \theta) dV \quad (2.38)$$

onde \dot{Q} é a taxa na qual energia é transferida para o sistema na forma de calor.

Assumindo que o sistema seja uniforme e que o calor seja transferido de forma lenta e gradual, a equação anterior pode ser reescrita como:

$$\dot{Q} = \Lambda \dot{F} V \quad (2.39)$$

onde V é o volume total do sistema, ou como:

$$\dot{Q} = \Lambda^* \dot{F} M \quad (2.40)$$

sendo M é a massa total do sistema e $\Lambda^*(\xi_a, \theta) = \Lambda(\xi_a, \theta)/\rho(\xi_a, \theta)$.

Pode-se mostrar que existe uma única função Λ^* a menos de uma constante multiplicadora que atenda ao Teorema de Carathéodory, e que essa função depende unicamente da variável de estado θ e não poder mudar de sinal. Essa função é conhecida como temperatura absoluta, denominada $T(\theta)$.

A função $\dot{F}(\xi_a, \theta)$ correspondente a $T(\theta)$ é portanto uma função de estado única para cada sistema.

A função Densidade de Entropia, por sua vez, corresponde diretamente à função $F(\xi_a, \theta)$, sendo denominada $\eta(\xi_a, \theta)$, ou seja:

$$\eta(\xi_a, \theta) = F(\xi_a, \theta) \quad (2.41)$$

Definida a função de estado intensiva Densidade de Entropia, reescreve-se a Equação 2.36 utilizando, por motivos de conveniência, T no lugar de θ como variável de estado:

$$\sigma_a \dot{\xi}_a + \rho T \dot{\eta} = \rho \dot{\gamma} \quad (2.42a)$$

$$\varsigma_m d_m - q''_{i,i} = \rho T \dot{\eta} \quad (2.42b)$$

Introduz-se então uma função de energia livre, $\varphi(\epsilon_{ij}, T)$ como definido a seguir:

$$\varphi(\epsilon_{ij}, T) = \gamma(\epsilon_{ij}, T) - T\eta(\epsilon_{ij}, T) \quad (2.43)$$

Substituindo a expressão para a energia livre na Equação 2.42a:

$$\left(\sigma_{ij} - \rho \frac{\partial \varphi}{\partial \epsilon_{ij}} \right) \dot{\epsilon}_{ij} - \rho \left(\eta + \frac{\partial \varphi}{\partial T} \right) \dot{T} = 0 \quad (2.44)$$

Devido ao fato dos coeficientes de ϵ_{ij} e T serem independentes, e devido à simetria do tensor de tensões, ambos os parênteses devem ser identicamente nulos, o que leva a:

$$\sigma_{ij} = \rho \frac{\partial \varphi}{\partial \epsilon_{ij}} \quad (2.45)$$

$$\eta = -\frac{\partial \varphi}{\partial T} \quad (2.46)$$

A energia livre pode ser expressa em função dos invariantes do tensor de deformações, I_ϵ , II_ϵ , III_ϵ , dos ângulos dos eixos principais e da temperatura absoluta. No caso de um material isotrópico, a energia livre é independente dos ângulos de orientação dos eixos principais do tensor de deformações, logo:

$$\varphi(\epsilon_{ij}, T) = \varphi(I_\epsilon, II_\epsilon, III_\epsilon, T) \quad (2.47)$$

Assim, a Equação 2.45 pode ser reescrita como:

$$\sigma_{ij} = \rho \left(\frac{\partial \varphi}{\partial I_\epsilon} \frac{\partial I_\epsilon}{\partial \epsilon_{ij}} + \frac{\partial \varphi}{\partial II_\epsilon} \frac{\partial II_\epsilon}{\partial \epsilon_{ij}} + \frac{\partial \varphi}{\partial III_\epsilon} \frac{\partial III_\epsilon}{\partial \epsilon_{ij}} \right) \quad (2.48)$$

Considerando um estado de referência à temperatura $T = T_0$ para o qual o material se encontre isento de tensões, expande-se φ em uma série de potências em função dos invariantes do tensor de deformações, I_ϵ , II_ϵ , III_ϵ e da variável T' , onde:

$$T' = (T - T_0)/T_0 \quad (2.49)$$

Então:

$$\begin{aligned} \varphi(I_\epsilon, II_\epsilon, III_\epsilon, T') = \frac{1}{\rho} & (a_0 + a_1 I_\epsilon + a_2 II_\epsilon + a_3 III_\epsilon + a_4 T' + a_5 I_\epsilon^2 + \\ & a_6 II_\epsilon^2 + a_7 III_\epsilon^2 + a_8 T' I_\epsilon + \dots) \end{aligned} \quad (2.50)$$

Substituindo a Equação 2.50 na Equação 2.48, utilizando as expressões dos invariantes do tensor de deformações em função das componentes desse tensor e considerando que para um material linear em regime de pequenos deslocamentos os produtos entre ϵ_{ij} e T' possam ser desprezados, obtém-se:

$$\sigma_{ij} = (2a_5 + a_2)\delta_{ij}\epsilon_{kk} - a_2\epsilon_{ij} + a_8\delta_{ij}T' \quad (2.51)$$

As constantes a_2 , a_5 e a_8 podem ser expressas em termos das constantes de Lamé, λ e μ , e em termos do coeficiente de expansão térmica α , o que leva à tradicional relação tensão-deformação para materiais sólidos elásticos e lineares na presença de variações de temperatura:

$$\sigma_{ij} = \lambda\delta_{ij}\epsilon_{kk} + 2\mu\epsilon_{ij} - (3\lambda + 2\mu)\delta_{ij}\alpha(T - T_0) \quad (2.52)$$

Partindo do princípio de que a geração de entropia é sempre positiva, na ausência de variáveis dissipativas obtém-se uma relação funcional linear entre o fluxo de calor e o gradiente de temperaturas. Para um material isotrópico, essa relação se reduz à lei de Fourier.

A equação da conservação da energia pode ser colocada em uma forma mais conveniente, para sólidos elásticos lineares na ausência de variáveis dissipativas, partindo-se da Equação 2.42b:

$$-q''_{i,i} = \rho T \dot{\eta} = \rho T \left(\frac{\partial \eta}{\partial \epsilon_{ij}} \dot{\epsilon}_{ij} + \frac{\partial \eta}{\partial T} \dot{T} \right) \quad (2.53)$$

Substituindo a formulação para a função de densidade de entropia descrita na Equação 2.46 na Equação 2.53, obtém-se:

$$-q''_{i,i} = -\rho T \left(\frac{\partial^2 \varphi}{\partial \epsilon_{ij} \partial T} \dot{\epsilon}_{ij} + \frac{\partial^2 \varphi}{\partial T^2} \dot{T} \right) \quad (2.54)$$

Da definição de calor específico a deformação constante, caracterizado por um processo onde $\dot{\epsilon}_{ij} = 0$, tem-se:

$$c_p = -\frac{\partial^2 \varphi}{\partial T^2} T \quad (2.55)$$

Considerando também a relação da Equação 2.45, pode-se mostrar que:

$$\frac{\partial^2 \varphi}{\partial \epsilon_{ij} \partial T} = \frac{1}{\rho} \frac{\partial \sigma_{ij}}{\partial T} \quad (2.56)$$

Substituindo as equações 2.55 e 2.56 na equação 2.54:

$$-q''_{i,i} = \rho c_p \dot{T} - T \frac{\partial \sigma_{ij}}{\partial T} \dot{\epsilon}_{ij} \quad (2.57)$$

Aplicando a relação tensão deformação da Equação 2.52 na Equação 2.57, utilizando a Lei de Fourier, incluindo o termo de geração de energia, \dot{e} , e considerando $T = T_0 + T' T_0 \approx T_0$ obtém-se a equação corrigida, ou acoplada, da condução de calor para o problema termoelástico:

$$(k\theta_{,i})_{,i} + \dot{e} = \rho c_p \dot{T} + (3\lambda + 2\mu) \alpha T_0 \dot{\epsilon}_{kk} \quad (2.58)$$

onde α é o coeficiente de expansão térmica do material.

A Equação 2.58 é uma extensão da Equação 2.20 aplicada ao caso termomecânico, e seu último termo é denominado termo de acoplamento termoelástico por implicar na necessidade de resolver as equações da mecânica dos sólidos simultaneamente à equação da condução do calor.

2.4.2 Considerações simplificadoras

Reescrevendo a Equação 2.58 introduzindo um parâmetro adimensional Ψ :

$$(k\theta_{,i})_{,i} = \rho c_p \dot{T} \left[1 + \Psi \left(\frac{\lambda + 2\mu}{3\lambda + 2\mu} \right) \left(\frac{\dot{\epsilon}_{kk}}{\alpha \dot{T}} \right) \right] \quad (2.59)$$

onde Ψ é definido por:

$$\Psi = \frac{(3\lambda + 2\mu)^2 \alpha^2 T_0}{\rho^2 c_p v_e^2} \quad (2.60)$$

O parâmetro v_e é a velocidade de propagação das ondas de dilatação no meio elástico, definido como:

$$v_e = \sqrt{(\lambda + 2\mu)/\rho} \quad (2.61)$$

O acoplamento termoelástico pode então ser desprezado quando:

$$\frac{\dot{\epsilon}_{kk}}{3\alpha\dot{T}} \ll \left(\frac{\lambda + 2\mu/3}{\lambda + 2\mu} \right) \frac{1}{\Psi} \quad (2.62)$$

Segundo Booley e Weiner (1997), em termos práticos, para negligenciar o acoplamento termoelástico é necessário atender à desigualdade $\Psi \ll 1$, o que ocorre para a maioria dos metais, desde que as taxas de deformação sejam da mesma ordem de grandeza das taxas de variação da temperatura. O segundo critério implica na necessidade do histórico temporal dos deslocamentos seguir o histórico da variação de temperaturas de tal forma que não ocorram atrasos ou vibrações no movimento do corpo.

2.4.3 Problema termomecânico desacoplado em regime quasi-estático

No problema termomecânico desacoplado em regime quasi-estático tanto o termo de acoplamento termoelástico na Equação 2.58 como os termos inerciais da equação do movimento podem ser desprezados.

Nessa condição, o problema é solucionado em duas etapas:

Etapa 1: Obter a evolução da distribuição de temperatura segundo a teoria clássica da condução do calor:

$$(k\theta_{,i})_{,i} + \dot{e} = \rho c_p \dot{\theta} \quad (2.63)$$

para a condição inicial em todo o domínio do corpo $\mathbf{X} \in \Omega$, em $t = 0$:

$$\begin{cases} \theta(x, y, z, t = 0) \text{ conhecida em } \Omega \\ q(x, y, z, t = 0) \text{ conhecida em } \Omega \end{cases}$$

e com as condições de contorno na superfície do corpo $\mathbf{X} \in \Gamma = \Gamma_\theta \cup \Gamma_q \cup \Gamma_h$:

$$\begin{cases} \theta(x, y, z, t) \text{ conhecida em } \Gamma_\theta \\ q_i n_i(x, y, z, t) \text{ conhecido em } \Gamma_q \\ q_{hi} n_i(x, y, z, t) = h(\theta - \theta_\infty) \text{ conhecido em } \Gamma_h \end{cases}$$

Etapa 2: Obter a evolução do campo de deslocamentos em função da variação temporal das ações mecânicas e da temperatura:

$$(\lambda + \mu)u_{i,ij} + \mu u_{j,kk} + b_j - (3\lambda + 2\mu)\alpha\theta_{,j}(t) = 0 \quad (2.64)$$

Para as condições de contorno na superfície do corpo $\mathbf{X} \in \Gamma = \Gamma_u \cup \Gamma_p$:

$$\begin{cases} u_i(x, y, z, t) \text{ conhecida em } \Gamma_u \\ \sigma_{ij}(x, y, z, t)n_j(x, y, z, t) = p_i(x, y, z, t) \text{ conhecido em } \Gamma_p \end{cases}$$

Nas equações acima, Γ_θ e Γ_u se referem às porções da superfície do corpo onde atuam as condições de contorno de Dirichlet, ou de primeira espécie, e Γ_q e Γ_p às porções da superfície do corpo onde atuam as condições de contorno de Neuman, ou de segunda espécie, e Γ_h à porção da superfície do corpo onde atua a condição de contorno de Robin, ou de terceira espécie. Ressalta-se que a formulação apresentada neste capítulo se restringe a materiais isotrópicos.

Capítulo 3

ABORDAGEM NUMÉRICA

A teoria referente ao problema termomecânico desacoplado em regime quasi-estático, discutida resumidamente no Capítulo 2, resulta em um conjunto de equações diferenciais parciais em função das variáveis espaciais do problema e da variável tempo. Nesta situação, diz-se que o problema é contínuo visto que, ao se resolver o referido conjunto de equações, é obtida uma solução analítica válida para qualquer ponto do domínio em qualquer instante de tempo. Entretanto, para a maioria dos casos de aplicação prática, a obtenção da solução analítica não é trivial, ou até impossível, especialmente em corpos cuja geometria seja complexa.

Uma alternativa à solução analítica do conjunto de equações diferenciais parciais consiste em calcular uma solução aproximada em pontos do domínio espacial e do domínio temporal, com o auxílio de métodos numéricos, e então interpolar os valores calculados nesses pontos para estender essa aproximação ao restante do corpo e do período de tempo estudado.

Nessa dissertação de mestrado são empregados métodos que realizam uma subdivisão tanto do espaço como do período de tempo em que se deseja conhecer a resposta aproximada. A porção espacial do problema é discretizada e analisada segundo o Método dos Elementos Finitos e a porção temporal segundo o Método de Diferenças Finitas, ambos métodos consagrados na literatura.

A apresentação desses métodos nesse capítulo se iniciará com a discussão do

Método dos Elementos Finitos aplicado à porção espacial do Problema de Transferência de Calor, seguida da apresentação do Método de Diferenças finitas para o tratamento da porção temporal dos problemas de condução de calor em regime transiente. Nesse momento, também serão discutidas as simplificações relativas à análise térmica em regime permanente.

Posteriormente, será mostrada a formulação do Método de Newton para a solução numérica da equação da condução do calor quando a condutividade térmica apresenta dependência em relação à temperatura, o que torna o problema não-linear.

A seção relativa ao problema de transferência de calor será concluída com uma discussão da aplicação conjunta dos métodos de Newton, Elementos Finitos e Diferenças Finitas para a solução de problemas de transferência de calor não-lineares no regime transiente. Nessa seção também serão apontadas as considerações simplificadoras para o caso particular do PTC não-linear em regime permanente.

Na sequência, a formulação do problema mecânico segundo o MEF, contemplando a variação da temperatura no corpo, é apresentada em detalhes. Uma breve discussão sobre o procedimento de cálculo das deformações térmicas, quando o coeficiente de expansão térmica possui dependência polinomial com a temperatura, é realizada ao final da respectiva seção.

Uma breve revisão acerca da formulação paramétrica e da utilização da técnica de integração numérica dentro do contexto do MEF é apresentada após as formulações do PTC e do PTM, onde são tratadas algumas particularidades dos problemas abordados nessa dissertação de mestrado.

O capítulo será concluído com a apresentação de uma das abordagens possíveis à solução do problema multifísico termomecânico em regime quasi-estático contemplando os métodos numéricos supramencionados, sob o ponto de vista de fluxo de execução.

3.1 Formulação do Problema de Transferência de Calor pelo Método dos Elementos Finitos

O problema contínuo de condução de calor em corpos sólidos, descrito no Capítulo 2, é regido pela equação:

$$(D_{ij}\theta_{,j})_{,i} + \dot{e} = \rho c_p \dot{\theta} \quad (3.1)$$

onde \mathbf{D} é o tensor constitutivo, que representa a relação entre as componentes do fluxo de calor e do gradiente de temperaturas em um ponto. Em materiais isotrópicos, o tensor \mathbf{D} torna-se diagonal com todos os termos iguais à condutividade térmica do material, isso é, $D_{ij} = k \cdot \delta_{ij}$.

A Equação 3.1 pode ser reescrita na forma:

$$R(\theta) = (D_{ij}\theta_{,j})_{,i} + \dot{e} - \rho c_p \dot{\theta} = 0 \quad (3.2)$$

onde R é denominado resíduo. Para que a solução buscada seja válida, esse resíduo deve ser nulo em qualquer ponto do domínio e em qualquer instante do período de tempo analisado.

Como detalhadamente descrito por Bergheau e Fortunier (2008), existe uma família de métodos de resíduos ponderados que se baseiam na condição de cancelamento do resíduo:

$$\int_{\Omega} \psi R(\theta) dV = 0 \quad (3.3)$$

onde ψ são funções peso, pertencentes a um conjunto de funções E_{ψ} , integráveis no domínio Ω . Diz-se que a função $\theta(x, y, z, t)$ buscada, quando admissível, deve possuir derivadas de segunda ordem integráveis em Ω . O conjunto de funções θ admissíveis é denominado E_{θ} .

A formulação do método de resíduos ponderados aplicada ao problema de transferência de calor leva, portanto, ao problema de buscar uma função $\theta \in E_{\theta}$ para

qualquer função $\psi \in E_\psi$ tal que:

$$\int_{\Omega} \psi[(D_{ij}\theta_{,j})_{,i} + \dot{\epsilon} - \rho c_p \dot{\theta}] dV = 0 \quad (3.4)$$

As mesmas condições iniciais e condições de contorno discutidas no Capítulo 2 devem ser atendidas pelo problema de valor inicial e de contorno descrito acima.

Caso o campo de temperaturas $\theta(x, y, z, t)$ seja uma solução da Equação 3.2, ele automaticamente será uma solução da Equação 3.4. Caso contrário, a escolha dos conjuntos de funções E_ψ e E_θ irá definir se o campo de temperaturas θ será uma solução da Equação 3.4. Esta abordagem é adequada para a busca de uma solução aproximada para o campo de temperaturas.

A integral de volume na Equação 3.4 pode ser modificada pela integração por partes seguida da aplicação do teorema da divergência:

$$\int_{\Omega} \psi[(D_{ij}\theta_{,j})_{,i}] dV = \int_{\Gamma} \psi(D_{ij}\theta_{,j} n_i) dA - \int_{\Omega} \psi_{,i}(D_{ij}\theta_{,j}) dV \quad (3.5)$$

Substituindo esse resultado na Equação 3.4, obtém-se a Formulação Integral Fraca do problema:

$$\int_{\Omega} \psi(\dot{\epsilon} - \rho c_p \dot{\theta}) dV + \int_{\Gamma} \psi(D_{ij}\theta_{,j} n_i) dA - \int_{\Omega} \psi_{,i}(D_{ij}\theta_{,j}) dV = 0 \quad (3.6)$$

Ao escrever o problema na Formulação Integral Fraca, os requisitos relacionados às funções que podem descrever o campo de temperaturas são reduzidos, uma vez que não existem mais derivadas segundas dessas funções na Equação 3.6. Por outro lado, se observa que nessa formulação existem derivadas primeiras tanto das funções θ como das funções ψ , o que eleva o rigor quanto à seleção do conjunto de funções admissíveis E_ψ .

Segundo Bergheau e Fortunier (2008), os conjuntos de funções admissíveis E_θ e E_ψ devem pertencer ao primeiro espaço de Sobolev, $H^1(\Omega)$, que fornece funções quadraticamente integráveis. Por apresentarem requisitos similares, as funções ψ

podem ser escolhidas como variações admissíveis das funções E_θ . Portanto, se $\theta \in E_\theta$, então para qualquer ψ em E_ψ , $\theta + \psi \in E_\theta$.

Na Formulação Integral Fraca, as condições de contorno de Dirichlet, ou essenciais, devem ser obrigatoriamente introduzidas no espaço de funções admissíveis para o campo de temperaturas, E_θ . Ademais, selecionam-se funções peso ψ que sejam nulas no contorno Γ_θ .

O termo correspondente à integral de superfície na Equação 3.6 pode ser reduzido fazendo uso da propriedade supramencionada das funções ψ no contorno Γ do problema, onde ψ é nula em Γ_θ , região onde são prescritas as condições de contorno essenciais, e pode ser diferente de zero em Γ_q onde são prescritas as condições de contorno naturais:

$$\int_{\Gamma} \psi(D_{ij}\theta_{,j} n_i) dA = \int_{\Gamma_\theta} \underbrace{\psi}_{=0}(D_{ij}\theta_{,j} n_i) dA + \int_{\Gamma_q} \psi(D_{ij}\theta_{,j} n_i) dA = \int_{\Gamma_q} \psi q_s'' dA \quad (3.7)$$

onde q_s'' é o fluxo de calor correspondente à condição de contorno de Neumann, ou natural, prescrita na superfície Γ_q .

Uma solução aproximada pode ser obtida trabalhando em sub-espacos E_θ^n e E_ψ^n de E_θ e E_ψ , respectivamente, de dimensão finita n . A partir de um elemento particular de E_θ , θ^* , é possível expressar todos os elementos do sub-espaco E_θ^n na forma: $\theta = \theta^* + \psi$. O Método de Galerkin faz uso dessa propriedade ao considerar um espaco de funções E_θ^n definido a partir do espaco E_ψ^n .

Portanto, tal método consiste em substituir o problema contínuo de cancelamento do resíduo por um problema discreto de dimensão n , no qual deve-se buscar $\theta \in E_\theta^n$ tal que para qualquer função $\psi \in E_\psi^n$:

$$\int_{\Omega} \psi(\dot{\epsilon} - \rho c_p \dot{\theta}) dV + \int_{\Gamma_q} \psi q_s'' dA - \int_{\Omega} \psi_{,i}(D_{ij}\theta_{,j}) dV = 0 \quad (3.8)$$

Ao se incrementar a dimensão n de sub-espacos E_θ^n e E_ψ^n , se obtém uma melhor

aproximação da solução, que converge para a solução analítica quando o número de funções nos sub-espacos tende ao infinito.

O Método dos Elementos Finitos é um caso particular do Método de Galerkin. No MEF os sub-espacos de aproximação E_θ^n e E_ψ^n são construídos sobre a subdivisão da geometria do corpo, e cada subdomínio geométrico é denominado Elemento Finito.

Para construir as aproximações nos elementos finitos são utilizados os valores do campo de temperaturas em pontos específicos do subdomínio geométrico, que passam a constituir as incógnitas do problema. Esses pontos são denominados nós, e por isso diz-se que o Método dos Elementos Finitos pertence a uma família de métodos de aproximação nodal por sub-domínios.

As funções que realizam a aproximação local são denominadas funções de forma, funções de aproximação, ou funções interpoladoras, e serão aqui referenciadas por N . Cada nó do modelo discreto possui uma função de interpolação associada a si. Essas funções devem atender a alguns requisitos particulares tais como:

- Possuir valor unitário sobre o nó correspondente à função;
- Ser nula em sub-domínios geométricos que não contenham o nó correspondente à função;
- Ser nula nas fronteiras do sub-domínio geométrico que não conterem o nó correspondente à função;
- Ser definida e continuamente diferenciável no sub-domínio geométrico.

Além dos requisitos sobre as funções de aproximação, a união dos subdomínios Ω^e dos elementos deve resultar no domínio do corpo Ω e não deve ocorrer sobreposição dos mesmos. Em geral, os subdomínios são figuras geométricas básicas como linhas, triângulos, quadriláteros, tetraedros e hexaedros.

Inicia-se o desenvolvimento do MEF realizando a aproximação do campo de temperatura θ no interior do elemento finito por uma interpolação dos valores de

temperatura nos nós do elemento. Essa interpolação é feita utilizando as funções N , associadas a cada um dos nós do elemento, agrupadas no vetor linha \mathbf{N} .

Agrupando as temperaturas nos nós do elemento em um vetor coluna $\boldsymbol{\theta}^e$, pode-se interpolar o campo de temperaturas no domínio do elemento finito pelo produto matricial:

$$\theta = \mathbf{N} \boldsymbol{\theta}^e \quad (3.9)$$

onde $\mathbf{N} = \{N_1 \ N_2 \ \dots \ N_n\}$ e $\boldsymbol{\theta}^e = \{\theta^e_1 \ \theta^e_2 \ \dots \ \theta^e_n\}^T$.

A função peso no interior de um elemento finito também pode ser construída a partir dos seus valores nodais, ou seja,

$$\psi = \boldsymbol{\psi}^{eT} \mathbf{N}^T \quad (3.10)$$

onde $\boldsymbol{\psi}^e$ é o vetor dos pesos nos nós do elemento, escrito como $\boldsymbol{\psi}^e = \{\psi^e_1 \ \psi^e_2 \ \dots \ \psi^e_n\}^T$.

O número de componentes no vetor linha \mathbf{N} e nos vetores coluna $\boldsymbol{\theta}^e$ e $\boldsymbol{\psi}^e$ é dado pelo número de graus de liberdade do elemento finito.

O gradiente do campo de temperaturas no interior do domínio do elemento finito também pode ser calculado utilizando as funções de interpolação:

$$\theta_{,i} = (\mathbf{N} \boldsymbol{\theta}^e)_{,i} = (\mathbf{L} \mathbf{N}^T) \boldsymbol{\theta}^e \quad (3.11)$$

onde \mathbf{L} é um operador de derivadas, que no caso tridimensional em um sistema cartesiano é expresso como:

$$\mathbf{L} = \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix} \quad (3.12)$$

O resultado do produto entre \mathbf{L} e \mathbf{N} é uma matriz, que será referida como \mathbf{B} .

Logo,

$$\theta_{,i} = \mathbf{B} \boldsymbol{\theta}^e \quad (3.13)$$

Analogamente, o gradiente da função peso é obtido como:

$$\psi_{,i} = (\boldsymbol{\psi}^{eT} \mathbf{N}^T)_{,i} = \boldsymbol{\psi}^{eT} (\mathbf{L} \mathbf{N})^T = \boldsymbol{\psi}^{eT} \mathbf{B}^T \quad (3.14)$$

Em seguida, as integrais presentes na Equação 3.8, que operam sobre todo o domínio Ω do corpo, são decompostas nas somas de funções integrais sobre os domínios Ω^e de cada um dos elementos finitos. As aproximações para ψ e θ , definidas nas equações 3.9 e 3.10, são então substituídas na formulação:

$$\int_{\Omega} \psi (\dot{\epsilon} - \rho c_p \dot{\theta}) dV = \sum_{e=1}^m \boldsymbol{\psi}^{eT} \left(\int_{\Omega^e} \mathbf{N}^T \dot{\epsilon} dV - \int_{\Omega^e} \rho c_p \mathbf{N}^T \mathbf{N} dV \right) \quad (3.15a)$$

$$\int_{\Gamma_q} \psi q_s'' dA = \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Gamma^e \cap \Gamma_q} \mathbf{N}^T q_s'' dA \quad (3.15b)$$

$$\int_{\Omega} \psi_{,i} D_{ij} \theta_{,j} dV = \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} \theta^e dV \quad (3.15c)$$

onde m é o número de elementos finitos que compõem a malha, obtidos no processo de discretização espacial.

Substituindo as expressões acima na Equação 3.8, obtém-se:

$$\sum_{e=1}^m \boldsymbol{\psi}^{eT} \mathbf{R}^e = 0 \quad (3.16)$$

onde os termos \mathbf{R}^e do somatório são os vetores de resíduo associados a cada um dos elementos finitos:

$$\begin{aligned} \mathbf{R}^e = & \int_{\Omega^e} \mathbf{N}^T \dot{\epsilon} dV - \int_{\Omega^e} \rho c_p \mathbf{N}^T \mathbf{N} \dot{\theta}^e dV + \\ & \int_{\Gamma^e \cap \Gamma_q} \mathbf{N}^T q_s'' dA - \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} \theta^e dV \end{aligned} \quad (3.17)$$

A definição do resíduo do elemento finito expressa pela Equação 3.17 ainda pode ser modificada considerando que a troca de calor no contorno q_s'' pode ocorrer tanto por fluxo de calor prescrito q_p'' como por convecção, $q_h'' = h(\theta_{\infty} - \theta)$, logo:

$$\begin{aligned}
\mathbf{R}^e = & \int_{\Omega^e} \mathbf{N}^T \dot{e} dV - \int_{\Omega^e} \rho c_p \mathbf{N}^T \mathbf{N} \dot{\boldsymbol{\theta}}^e dV + \\
& \int_{\Gamma^e \cap \Gamma_{q_p}} \mathbf{N}^T q_p'' dA + \int_{\Gamma^e \cap \Gamma_{q_h}} \mathbf{N}^T h \theta_\infty dA - \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} \boldsymbol{\theta}^e dV \\
& - \int_{\Gamma^e \cap \Gamma_{q_h}} \mathbf{N}^T h \mathbf{N} \boldsymbol{\theta}^e dA
\end{aligned} \quad (3.18)$$

onde $\Gamma_q = \Gamma_{q_p} \cup \Gamma_{q_h}$, sendo Γ_{q_p} a porção do contorno onde foram prescritos fluxos de calor e Γ_{q_h} a porção que troca calor com um fluido por convecção.

Para que a relação expressa na Equação 3.16 seja válida para qualquer função ψ , caracterizada pelos coeficientes ψ_i nos nós que compõem a malha, o resíduo deve ser identicamente nulo.

No processo de formulação, as funções interpoladoras, os valores instantâneos de temperatura nos nós e os valores da variação temporal da temperatura nos nós em cada elemento finito foram agrupados no vetor linha \mathbf{N} e nos vetores coluna $\boldsymbol{\theta}^e$ e $\dot{\boldsymbol{\theta}}^e$, o que faz com que as integrais que descrevem o resíduo em cada elemento finito resultem em matrizes e vetores ao serem avaliadas. As matrizes e vetores resultantes possuem sentido físico, e serão aqui denominadas matrizes de capacitância e de condutividade térmica, \mathbf{C}^e e \mathbf{K}^e respectivamente, e vetor das ações externas \mathbf{F}^e , onde:

$$\mathbf{C}^e = \int_{\Omega^e} \rho c_p \mathbf{N}^T \mathbf{N} dV \quad (3.19)$$

$$\mathbf{K}^e = \mathbf{\Lambda}^e + \mathbf{H}^e \Rightarrow \begin{cases} \mathbf{\Lambda}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \\ \mathbf{H}^e = \int_{\Gamma^e \cap \Gamma_{q_h}} \mathbf{N}^T h \mathbf{N} dA \end{cases} \quad (3.20)$$

$$\mathbf{F}^e = \mathbf{F}_{\dot{e}}^e + \mathbf{F}_{q_p''}^e + \mathbf{F}_h^e \Rightarrow \begin{cases} \mathbf{F}_{\dot{e}}^e = \int_{\Omega^e} \mathbf{N}^T \dot{e} dV \\ \mathbf{F}_{q_p''}^e = \int_{\Gamma^e \cap \Gamma_{q_p}} \mathbf{N}^T q_p'' dA \\ \mathbf{F}_h^e = \int_{\Gamma^e \cap \Gamma_{q_h}} \mathbf{N}^T h \theta_\infty dA \end{cases} \quad (3.21)$$

onde $\mathbf{F}_{\dot{e}}^e$ é o vetor de carregamento nodal equivalente relativo à geração de calor no interior do elemento, $\mathbf{F}_{q_p''}^e$ é o vetor de carregamento nodal equivalente devido à

prescrição de fluxo de calor por unidade de área no contorno do elemento e \mathbf{F}_h^e é o carregamento nodal equivalente devido às trocas de calor por convecção na superfície do elemento finito.

O resíduo em um elemento finito pode então ser representado pela equação matricial:

$$\mathbf{R}^e = \mathbf{F}^e - \mathbf{C}^e \cdot \dot{\boldsymbol{\theta}}^e - \mathbf{K}^e \cdot \boldsymbol{\theta}^e = 0 \quad (3.22)$$

O somatório das contribuições de cada elemento finito na Equação 3.16 resulta no resíduo global. Esse processo de retornar ao resíduo total a partir dos resíduos de cada elemento é usualmente definido como etapa de montagem do sistema global. Tal sistema global assume então a seguinte forma:

$$\mathbf{R} = \mathbf{F} - \mathbf{C} \dot{\boldsymbol{\theta}} - \mathbf{K} \boldsymbol{\theta} \quad (3.23)$$

onde \mathbf{F} é o vetor das ações externas do meio discretizado, \mathbf{C} é a matriz de capacitância do meio discretizado e \mathbf{K} é a matriz de condutividade do meio discretizado.

Ao impor que as componentes do vetor de resíduos do meio discretizado devem ser nulas, obtém-se o sistema de equações diferenciais ordinárias na variável temporal:

$$\mathbf{C} \dot{\boldsymbol{\theta}} + \mathbf{K} \boldsymbol{\theta} = \mathbf{F} \quad (3.24)$$

A última etapa do processo de solução pelo MEF consiste em condicionar o sistema descrito na Equação 3.24, o que pode ser realizado de diversas formas, sendo as mais comuns a aplicação do Método da Eliminação, que será empregado nesse trabalho e descrito a seguir, ou a aplicação do Método da Penalidade ou do Método dos Multiplicadores de Lagrange. Esses métodos são detalhados em Bergheau e Fortunier (2008).

No Método da Eliminação realiza-se um particionamento do sistema a partir de uma reordenação das equações, onde são agrupadas em uma porção as equações relacionadas aos graus de liberdade cujo valor é conhecido a priori, $\boldsymbol{\theta}_c$, e na outra porção

as demais equações, referentes aos nós onde não se conhece o valor da temperatura, θ_d .

$$\left(\begin{Bmatrix} \mathbf{F}_c \\ \mathbf{F}_d \end{Bmatrix} - \begin{bmatrix} \mathbf{C}_{dd} & \mathbf{C}_{dc} \\ \mathbf{C}_{cd} & \mathbf{C}_{cc} \end{bmatrix} \begin{Bmatrix} \dot{\theta}_d \\ \dot{\theta}_c \end{Bmatrix} - \begin{bmatrix} \mathbf{K}_{dd} & \mathbf{K}_{dc} \\ \mathbf{K}_{cd} & \mathbf{K}_{cc} \end{bmatrix} \begin{Bmatrix} \theta_d \\ \theta_c \end{Bmatrix} \right) = 0 \quad (3.25)$$

O sistema da equação anterior é resolvido em duas etapas. Na primeira, trata-se a porção superior desse sistema para determinar os valores desconhecidos do campo de temperaturas ao longo do tempo. Essa solução é obtida ao se resolver a equação diferencial ordinária:

$$\mathbf{C}_{dd} \dot{\theta}_d + \mathbf{K}_{dd} \theta_d = \mathbf{F}_c - \mathbf{C}_{dc} \dot{\theta}_c - \mathbf{K}_{dc} \theta_c \quad (3.26)$$

Após a solução da Equação 3.26 trata-se a porção inferior da Equação 3.25 para determinar o calor transferido através do contorno do problema nas regiões em que o campo de temperaturas foi prescrito. Para tal, substitui-se $\dot{\theta}_d$ e θ_d , previamente calculados, na equação a seguir:

$$\mathbf{F}_d = \mathbf{C}_{cd} \dot{\theta}_d + \mathbf{C}_{cc} \dot{\theta}_c + \mathbf{K}_{cd} \theta_d + \mathbf{K}_{cc} \theta_c \quad (3.27)$$

3.1.1 Método de Diferenças Finitas para a solução de problemas transientes formulados segundo o MEF

A formulação do Problema de Transferência de calor segundo o Método dos Elementos Finitos na seção anterior resultou em sistemas de equações diferenciais ordinárias na variável temporal. Em geral, a solução desses sistemas de forma analítica não é simples, e por isso emprega-se um método numérico específico para essa função.

Segundo Bergheau e Fortunier (2008), a técnica numérica usualmente empregada para a solução do referido sistema de equações diferenciais ordinárias é o Método de

Diferenças Finitas. Outras técnicas, como a superposição modal, também podem ser empregadas e estão descritas resumidamente em Bergheau e Fortunier (2008).

A aplicação do Método de Diferenças Finitas para a solução das equações 3.26 e 3.27 resulta em uma família de métodos de integração direta (Reddy, 2005), que se diferenciam principalmente pela forma de aproximar a variação temporal da temperatura. Nesses métodos, o intervalo de tempo entre o início e o término da análise é discretizado, e a solução no instante de tempo seguinte é calculada de forma aproximada utilizando os valores correspondentes ao instante atual.

O método de Newmark, exposto detalhadamente por Logan (2011), é um dos métodos pertencentes à família de métodos de integração direta e foi o método numérico empregado nesse trabalho. No método de Newmark, as distribuições de temperatura em dois instantes de tempo consecutivos (t) e ($t + 1$) podem ser ponderadas por um parâmetro que varia entre 0 e 1, denominado β , e pelo intervalo de tempo Δt através da equação:

$$\boldsymbol{\theta}_{(t+1)} = \boldsymbol{\theta}_{(t)} + [(1 - \beta)\dot{\boldsymbol{\theta}}_{(t)} + \beta \dot{\boldsymbol{\theta}}_{(t+1)}] \Delta t \quad (3.28)$$

sendo que para $\beta = 0$ o método de Newmark corresponde ao método de diferenças finitas progressivas, para $\beta = 1/2$ ao método de diferenças centrais, para $\beta = 2/3$ ao esquema de Galerkin, e para $\beta = 1$ ao método de diferenças finitas regressivas (Reddy, 2005).

Escrevendo a Equação 3.24 para os instantes de tempo $t_{(i)}$ e $t_{(i+1)}$:

$$(1 - \beta)(\mathbf{K} \boldsymbol{\theta}_{(t)} + \mathbf{C} \dot{\boldsymbol{\theta}}_{(t)}) = (1 - \beta) \mathbf{F}_{(t)} \quad (3.29a)$$

$$\beta (\mathbf{K} \boldsymbol{\theta}_{(t+1)} + \mathbf{C} \dot{\boldsymbol{\theta}}_{(t+1)}) = \beta \mathbf{F}_{(t+1)} \quad (3.29b)$$

Observa-se que a Equação 3.29a foi multiplicada em ambos os lados por $(1 - \beta)$ e que a Equação 3.29b foi multiplicada em ambos os lados por β . Somando a Equação

3.29a à Equação 3.29b obtém-se:

$$\mathbf{C}[(1 - \beta)\dot{\boldsymbol{\theta}}_{(t)} + \beta\dot{\boldsymbol{\theta}}_{(t+1)}] + \mathbf{K}[(1 - \beta)\boldsymbol{\theta}_{(t)} + \beta\boldsymbol{\theta}_{(t+1)}] = (1 - \beta)\mathbf{F}_{(t)} + \beta\mathbf{F}_{(t+1)} \quad (3.30)$$

As derivadas temporais da temperatura na Equação 3.30 podem ser eliminadas isolando o termo entre colchetes na Equação 3.28 e o substituindo na Equação 3.30, o que resulta em:

$$\frac{\mathbf{C}(\boldsymbol{\theta}_{(t+1)} - \boldsymbol{\theta}_{(t)})}{\Delta t} + \mathbf{K}[(1 - \beta)\boldsymbol{\theta}_{(t)} + \beta\boldsymbol{\theta}_{(t+1)}] = (1 - \beta)\mathbf{F}_{(t)} + \beta\mathbf{F}_{(t+1)} \quad (3.31)$$

Rearranjando a Equação 3.31, pode-se escrever a distribuição de temperaturas no instante de tempo seguinte em função da distribuição de temperaturas no instante corrente e das ações nestes instantes de tempo:

$$\left(\frac{1}{\Delta t}\mathbf{C} + \beta\mathbf{K}\right)\boldsymbol{\theta}_{(t+1)} = \left[\frac{1}{\Delta t}\mathbf{C} - (1 - \beta)\mathbf{K}\right]\boldsymbol{\theta}_{(t)} + (1 - \beta)\mathbf{F}_{(t)} + \beta\mathbf{F}_{(t+1)} \quad (3.32)$$

A Equação 3.32 pode ser representada na forma compacta:

$$\widetilde{\mathbf{K}}\boldsymbol{\theta}_{(t+1)} = \widetilde{\mathbf{F}}\boldsymbol{\theta}_{(t)} + (1 - \beta)\mathbf{F}_{(t)} + \beta\mathbf{F}_{(t+1)} \quad (3.33)$$

onde as matrizes $\widetilde{\mathbf{K}}$ e $\widetilde{\mathbf{F}}$ são:

$$\widetilde{\mathbf{K}} = \left(\frac{1}{\Delta t}\mathbf{C} + \beta\mathbf{K}\right) \quad (3.34)$$

$$\widetilde{\mathbf{F}} = \left[\frac{1}{\Delta t}\mathbf{C} - (1 - \beta)\mathbf{K}\right] \quad (3.35)$$

Para solucionar o sistema de equações algébricas descrito pela Equação 3.33, é necessário passar pela etapa de condicionamento, que corresponde a aplicar métodos como o Método da Eliminação, descrito na seção anterior, ou o Método da Penalidade ou o Método dos Multiplicadores de Lagrange (Bergheau e Fortunier, 2008).

A aplicação do Método da Eliminação, que faz uso da propriedade dos coeficientes da função peso serem nulos nos nós submetidos a condições de contorno essenciais, resulta no particionamento:

$$\begin{bmatrix} \widetilde{\mathbf{K}}_{dd} & \widetilde{\mathbf{K}}_{dc} \\ \widetilde{\mathbf{K}}_{cd} & \widetilde{\mathbf{K}}_{cc} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\theta}_{d(t+1)} \\ \boldsymbol{\theta}_{c(t+1)} \end{Bmatrix} = \begin{bmatrix} \widetilde{\mathbf{F}}_{dd} & \widetilde{\mathbf{F}}_{dc} \\ \widetilde{\mathbf{F}}_{cd} & \widetilde{\mathbf{F}}_{cc} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\theta}_{d(t)} \\ \boldsymbol{\theta}_{c(t)} \end{Bmatrix} + (1 - \beta) \begin{Bmatrix} \mathbf{F}_{c(t)} \\ \mathbf{F}_{d(t)} \end{Bmatrix} + \beta \begin{Bmatrix} \mathbf{F}_{c(t+1)} \\ \mathbf{F}_{d(t+1)} \end{Bmatrix} \quad (3.36)$$

onde os índices d se referem às equações associadas a nós cuja temperatura seja desconhecida a priori, e os índices c às equações associadas a nós cuja temperatura tenha sido prescrita por meio da condição de contorno essencial.

As temperaturas desconhecidas $\boldsymbol{\theta}_d$ no instante de tempo $(t + 1)$ são então calculadas a partir das equações da porção superior do sistema de equações algébricas descrito na Equação (3.36):

$$\widetilde{\mathbf{K}}_{dd} \boldsymbol{\theta}_{d(t+1)} = \widetilde{\mathbf{F}}_{dd} \boldsymbol{\theta}_{d(t)} + \widetilde{\mathbf{F}}_{dc} \boldsymbol{\theta}_{c(t)} + (1 - \beta) \mathbf{F}_{c(t)} + \beta \mathbf{F}_{c(t+1)} - \widetilde{\mathbf{K}}_{dc} \boldsymbol{\theta}_{c(t+1)} \quad (3.37)$$

Obida as temperaturas no instante de tempo $(t + 1)$, pode-se calcular os fluxos de calor no instante $(t + 1)$ pela porção inferior da Equação (3.36):

$$\beta \mathbf{F}_{d(t+1)} = \widetilde{\mathbf{K}}_{cd} \boldsymbol{\theta}_{d(t+1)} + \widetilde{\mathbf{K}}_{cc} \boldsymbol{\theta}_{c(t+1)} - \widetilde{\mathbf{F}}_{cd} \boldsymbol{\theta}_{d(t)} - \widetilde{\mathbf{F}}_{cc} \boldsymbol{\theta}_{c(t)} - (1 - \beta) \mathbf{F}_{d(t)} \quad (3.38)$$

3.1.2 Transferência de calor em regime permanente

Como dito anteriormente, as equações 3.26 e 3.27 constituem sistemas de equações diferenciais ordinárias em relação à variável tempo. Nesse tipo de problema, o corpo possui uma configuração inicial que se modificará ao longo do tempo enquanto as ações externas perturbarem o seu equilíbrio.

Se após um certo instante as condições de contorno permanecerem constantes, a distribuição de temperaturas irá se estabilizar em uma nova configuração. Nesta situação, diz-se que o sistema atingiu o regime permanente.

Caso não haja interesse em conhecer a evolução da distribuição de temperaturas ao longo do tempo, e sim determinar qual é a distribuição de temperaturas final na nova condição de equilíbrio, pode-se simplificar as equações 3.26 e 3.27, já que no regime permanente $\dot{\theta}_c$ e $\dot{\theta}_d$ são nulos.

Ao negligenciar os termos referentes às taxas de variação da temperatura nas equações 3.26 e 3.27, são obtidos sistemas de equações algébricas nos quais as respectivas soluções podem ser facilmente obtida sequencialmente:

$$\mathbf{K}_{dd} \theta_d = \mathbf{F}_c - \mathbf{K}_{dc} \theta_c \quad (3.39)$$

$$\mathbf{F}_d = \mathbf{K}_{cc} \theta_c + \mathbf{K}_{cd} \theta_d \quad (3.40)$$

3.1.3 Transferência de calor fisicamente não-linear pelo MEF

A discussão apresentada nas seções anteriores desse capítulo se baseia nas seguintes considerações:

- Os valores máximo e mínimo da distribuição de temperaturas estão contidos em uma faixa dentro da qual pode-se admitir que as propriedades físicas do material sejam constantes, ou seja, não é necessário descrever essas propriedades a partir de funções matemáticas dependentes da temperatura;
- As condições de contorno do problema não possuem dependência com a temperatura do corpo, ou possuem dependência descrita por uma relação linear com a temperatura do corpo, que é o caso da convecção com valor do coeficiente de filme constante.

Conforme melhor descrito por Reddy (2005) e por Bergheau e Fortunier (2008), quando os critérios acima não são atendidos diz-se que o problema de transferência de calor é fisicamente não-linear.

Essa dissertação de mestrado se restringe ao caso onde somente a condutividade térmica apresenta dependência com a temperatura e onde as condições de contorno atendem ao critério supramencionado.

Nesse tipo de problema fisicamente não-linear é necessário conhecer a distribuição de temperaturas no corpo para montar a matriz de condutividade térmica do meio discretizado com exatidão.

Como a distribuição de temperaturas é a informação buscada, deve-se recorrer a um processo de solução iterativo que se inicie a partir de uma estimativa do campo de temperaturas. Essa estimativa é então utilizada para calcular uma aproximação da matriz \mathbf{K} , e a cada iteração são calculadas melhores aproximações tanto da distribuição de temperaturas como da matriz de condutividade. O processo iterativo continua até que um critério de convergência seja atingido.

O método iterativo empregado nesse trabalho para a solução de problemas de transferência de calor fisicamente não-lineares é o Método de Newton, que será descrito em detalhes a seguir.

3.1.3.1 Método de Newton aplicado à formulação do MEF do problema de transferência de calor fisicamente não-linear

Conforme discutido na seção 3.1, o Método dos Elementos Finitos consiste no cancelamento do resíduo, sob uma perspectiva integral, em um domínio discretizado.

Segundo Reddy (2005) e Bergheau e Fortunier (2008), a formulação do Método de Newton para a solução do sistema de equações não-lineares parte da expansão em série de Taylor do vetor de resíduo na iminência da solução de forma a correlacionar a distribuição de temperaturas da iteração corrente com a distribuição de temperaturas calculada na iteração anterior. A expansão em série de Taylor do resíduo, truncada no termo linear, pode ser escrita como:

$$R(\boldsymbol{\theta})_i^{(r)} = R(\boldsymbol{\theta})_i^{(r-1)} + \sum_{j=1}^n \left(\frac{\partial R_i}{\partial \theta_j} \right)^{(r-1)} \left(\theta_j^{(r)} - \theta_j^{(r-1)} \right) \quad (3.41)$$

Na iminência da solução, pode-se considerar que o resíduo é nulo, o que implica em igualar o lado esquerdo da Equação 3.41 a zero. As derivadas parciais presentes na equação acima podem ser dispostas em uma matriz, referenciada nesse trabalho como \mathbf{M} , e o somatório da Equação 3.41 pode ser representado por um produto entre a matriz \mathbf{M} e o vetor $\delta\boldsymbol{\theta}$, que corresponde ao último termo entre parênteses na equação anterior. Logo,

$$\mathbf{M}^{(r-1)} \delta\boldsymbol{\theta} = \mathbf{R}(\boldsymbol{\theta})^{(r-1)} \quad (3.42)$$

onde os termos da matriz \mathbf{M} são definidos como:

$$M_{ij}^{(r-1)} = - \left(\frac{\partial R_i}{\partial \theta_j} \right)^{(r-1)} \quad (3.43)$$

O vetor $\delta\boldsymbol{\theta}$ representa, portanto, o incremento que deve ser dado na distribuição de temperaturas calculada na iteração anterior para que se caminhe em direção à solução do problema.

Para o caso da condução de calor em regime transiente, o resíduo após a discretização temporal por Diferenças Finitas segundo o Método de Newmark é obtido a partir da reorganização da Equação 3.33 e pode ser expresso por:

$$\mathbf{R}(\boldsymbol{\theta})_{(t+1)}^{(r-1)} = \tilde{\mathbf{F}} \boldsymbol{\theta}_{(t)} + (1 - \beta)\mathbf{F}_{(t)} + \beta\mathbf{F}_{(t+1)} - \tilde{\mathbf{K}}^{(r-1)} \boldsymbol{\theta}_{(t+1)} \quad (3.44)$$

Substituindo a Equação 3.44 na Equação 3.43 e utilizando as expressões para $\tilde{\mathbf{K}}$ e para $\tilde{\mathbf{F}}$ dadas pelas equações 3.34 e 3.35, pode se escrever a contribuição de cada elemento finito \mathbf{M}^e à matriz de iteração global \mathbf{M} :

$$M_{ij}^{e(r-1)} = - \left(\frac{\partial}{\partial \theta_j^e(t+1)} \right) \left(\sum_{m=1}^n \tilde{F}_{im}^e \theta_m^e(t) + (1 - \beta)F_i^e(t) + \beta F_i^e(t+1) - \sum_{m=1}^n \tilde{K}_{im}^{e(r-1)} \theta_m^e(t+1) \right) \quad (3.45)$$

Como as derivadas parciais são tomadas em relação aos instantes de tempo $t + 1$, e como os vetores de carregamento \mathbf{F}^e não dependem da temperatura do corpo, a equação acima se reduz a:

$$M_{ij}^{e(r-1)} = \frac{\partial}{\partial \theta_j^e(t+1)} \sum_{m=1}^n \tilde{K}_{im}^{e(r-1)} \theta_m^e(t+1) \quad (3.46)$$

Aplicando a regra da cadeia nos termos do somatório da Equação 3.46, tem-se:

$$M_{ij}^{e(r-1)} = \sum_{m=1}^n \left(\frac{\partial \tilde{K}_{im}^{e(r-1)}}{\partial \theta_j^e(t+1)} \theta_m^e(t+1) + \tilde{K}_{im}^{e(r-1)} \frac{\partial \theta_m^e(t+1)}{\partial \theta_j^e(t+1)} \right) \quad (3.47)$$

onde a derivação parcial no segundo termo no somatório se reduz à identidade se $m = j$ e a zero se $m \neq j$. Logo,

$$M_{ij}^{e(r-1)} = \sum_{m=1}^n \frac{\partial \tilde{K}_{im}^{e(r-1)}}{\partial \theta_j^e(t+1)} \theta_m^e(t+1) + \tilde{K}_{ij}^{e(r-1)} \quad (3.48)$$

O termo da derivação parcial presente na Equação 3.48 pode ser expandido a partir da substituição da definição de $\tilde{\mathbf{K}}$ dada pela Equação 3.34:

$$\frac{\partial \tilde{K}_{im}^{e(r-1)}}{\partial \theta_j^e(t+1)} = \frac{\partial}{\partial \theta_j^e(t+1)} \left(\frac{1}{\Delta t} C_{im}^e + \beta K_{im}^{e(r-1)} \right) = \beta \frac{\partial K_{im}^{e(r-1)}}{\partial \theta_j^e(t+1)} \quad (3.49)$$

Na equação acima a derivação de \mathbf{C} em relação a $\boldsymbol{\theta}$ é nula devido à consideração realizada nesse trabalho de que somente a condutividade térmica apresenta dependência com a temperatura.

Combinando as equações 3.34, 3.48 e 3.49, obtém-se finalmente a expressão para a matriz de iteração no problema de transferência de calor fisicamente não-linear em regime transiente:

$$M_{ij}^{e(r-1)} = \frac{1}{\Delta t} C_{ij}^e + \beta \left(K_{ij}^e + \sum_{m=1}^n \frac{\partial K_{im}^{e(r-1)}}{\partial \theta_j^e(t+1)} \theta_m^e(t+1) \right) \quad (3.50)$$

Em cada iteração, resolve-se o sistema de equações algébricas descrito na Equação 3.42 para determinar o valor de $\boldsymbol{\delta\theta}$. Nesta etapa, a matriz \mathbf{M} é montada segundo

a Equação 3.50 e o resíduo \mathbf{R} que é montado segundo a Equação 3.44, ambos utilizando a distribuição de temperaturas $\boldsymbol{\theta}_{(t+1)}^{(r-1)}$ calculada na iteração anterior, ou seja, na iteração $(r - 1)$.

Determinado o valor de $\delta\boldsymbol{\theta}$, incrementa-se o campo de temperaturas segundo a equação:

$$\boldsymbol{\theta}_{(t+1)}^{(r)} = \boldsymbol{\theta}_{(t+1)}^{(r-1)} + \delta\boldsymbol{\theta} \quad (3.51)$$

Com a distribuição de temperaturas atualizada, recalcula-se o vetor de resíduos para fins de verificação da convergência da iteração. Usualmente adota-se como critério de convergência avaliar se a norma do vetor de resíduo é inferior à uma tolerância pré-estabelecida. Também pode-se adotar como critério complementar verificar se a norma do incremento do campo de temperaturas é inferior à sua respectiva tolerância.

Atendidos os critérios de convergência, encerra-se o processo iterativo no passo de tempo atual para dar início ao cálculo da distribuição de temperaturas no próximo passo de tempo. Caso contrário, a matriz de iteração é atualizada para a nova distribuição de temperaturas para que um novo incremento $\delta\boldsymbol{\theta}$ seja calculado.

A matriz \mathbf{M} de iteração, que nessa dissertação foi obtida aplicando o Método de Newton, pode ser calculada de outras formas, como o Método da Iteração Direta, ou Método de Picard, ou como o Método de Newton Modificado (Reddy, 2005).

O processo completo de solução de um problema de transferência de calor fisicamente não-linear em regime transiente é ilustrado na figura a seguir. Nessa figura, as etapas foram identificadas com algarismos romanos para futura referência.

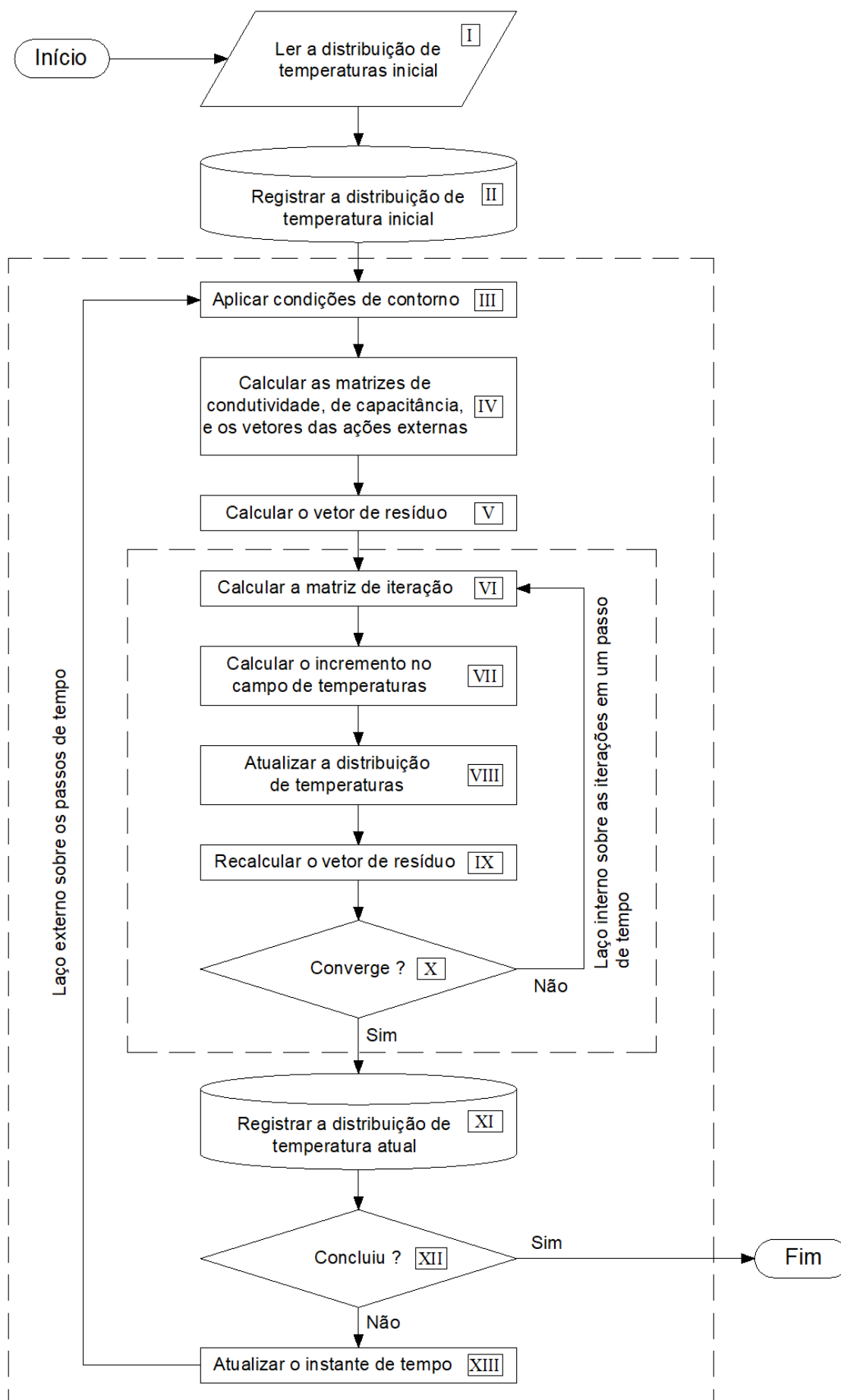


Figura 3.1: Fluxograma para solução do PTC transiente fisicamente não-linear.

3.1.3.2 Problema de Transferência de Calor fisicamente não-linear em regime permanente

Na seção 3.1.2 foram discutidas as simplificações aplicáveis quando as condições de contorno se estabilizam após o início da análise. Estas simplificações permitem ao calculista determinar a distribuição de temperaturas no regime permanente a partir da solução do sistema de equações algébricas descrito na Equação 3.39.

Quando a condutividade térmica do material depende da temperatura buscada, a Equação 3.39 é não-linear e a solução pode ser obtida utilizando um método numérico como o Método de Newton, descrito na seção anterior.

No caso particular do PTC fisicamente não-linear em regime permanente, o resíduo pode ser expresso como:

$$\mathbf{R}(\boldsymbol{\theta}) = \mathbf{F} - \mathbf{K}(\boldsymbol{\theta}) \boldsymbol{\theta} \quad (3.52)$$

Os termos da matriz \mathbf{M}^e de iteração do elemento finito podem ser obtidos a partir da substituição da Equação 3.52 na Equação 3.43:

$$M_{ij}^{e(r-1)} = \frac{\partial}{\partial \theta_j^e} \left(\sum_{m=1}^n K_{im}^e \theta_m^e - F_i^e \right) \quad (3.53)$$

Como as ações externas não dependem da temperatura, obtém-se:

$$M_{ij}^{e(r-1)} = \sum_{m=1}^n \left(\frac{\partial K_{im}^e}{\partial \theta_j^e} \theta_m^e + K_{im}^e \frac{\theta_m^e}{\partial \theta_j^e} \right) = \sum_{m=1}^n \frac{\partial K_{im}^e}{\partial \theta_j^e} \theta_m^e + K_{ij}^e \quad (3.54)$$

Para obter a distribuição de temperaturas em regime permanente, calculam-se os incrementos ao campo de temperaturas a cada iteração conforme a Equação 3.42 até que o critério de convergência seja atendido.

O processo de solução do PTC fisicamente não-linear em regime permanente está ilustrado a seguir. Novamente, identificou-se as etapas com algarismos romanos para futura referência.

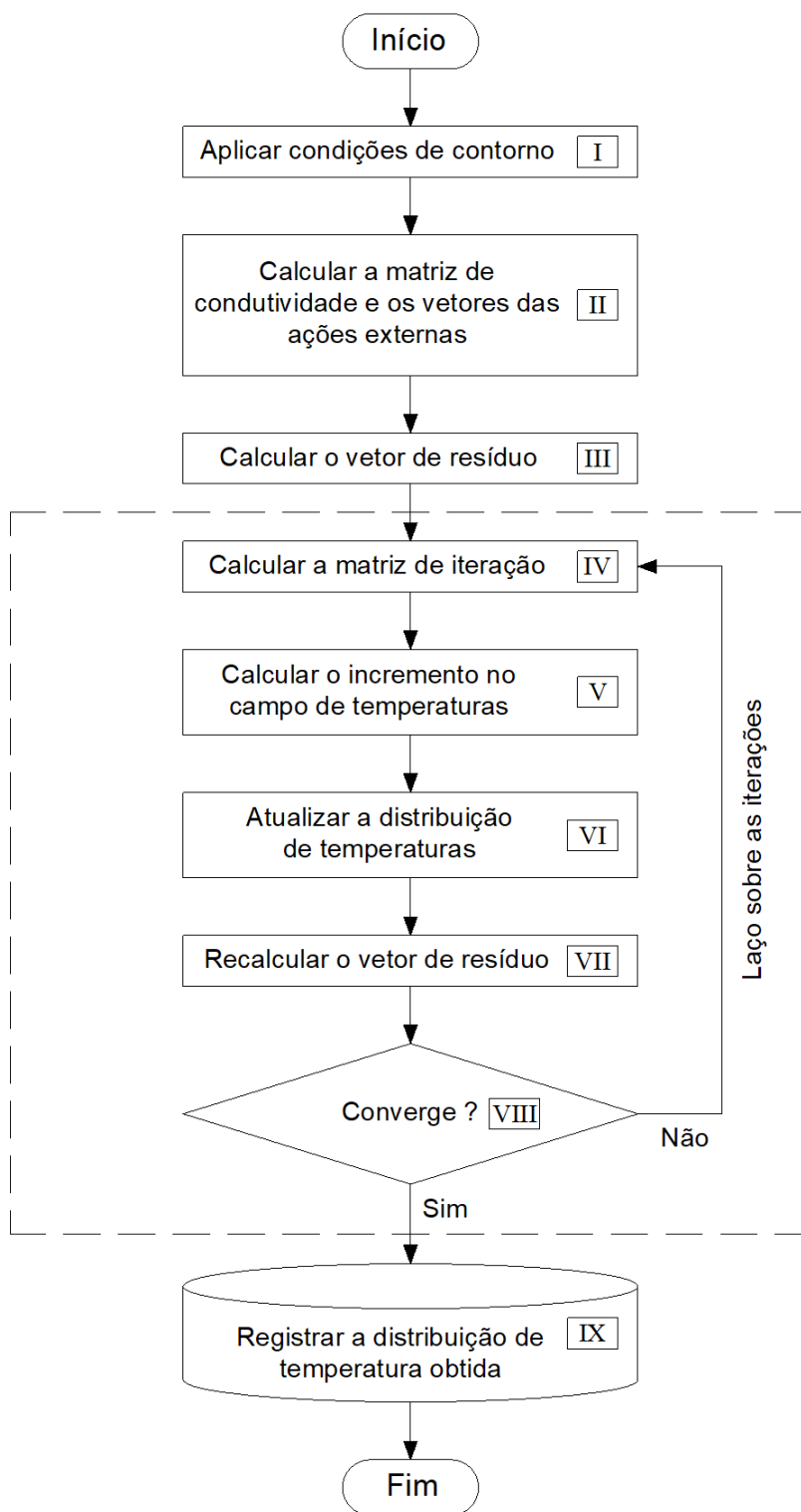


Figura 3.2: Fluxograma para solução do PTC fisicamente não-linear em regime permanente.

3.1.3.3 Considerações sobre a dependência entre a condutividade térmica e a temperatura

Nas duas sub-seções anteriores foram obtidas expressões para uma matriz denominada matriz de iteração, que é utilizada no cálculo dos incrementos no campo de temperaturas para diminuir a norma do resíduo a cada iteração em problemas fisicamente não-lineares.

Nas expressões para as matrizes de iteração do problema em regime transiente e do problema em regime permanente, equações 3.50 e 3.54, respectivamente, são encontradas derivadas parciais da matriz de condutividade em relação às temperaturas nodais do elemento finito.

Nesta sub-seção será realizado um desenvolvimento desses termos para o caso em que a dependência entre a condutividade térmica e a temperatura é expressa por um polinômio, como descrito na equação abaixo:

$$k(\theta) = a_0 + a_1 \theta + a_2 \theta^2 + \dots + a_n \theta^n \quad (3.55)$$

onde a_0 , a_1 , a_2 , \dots , a_n são valores escalares, coeficientes do polinômio $k(\theta)$.

Reforça-se que o desenvolvimento aqui apresentado se restringe a materiais isotrópicos.

A matriz de condutividade é calculada segundo a Equação 3.20. Os termos dessa matriz podem ser calculados separadamente:

$$K_{ij}^e = \int_{\Omega^e} \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j dV + \int_{\Gamma^e \cap \Gamma_q} \mathbf{N}_i^T h \mathbf{N}_j dA \quad (3.56)$$

onde os índices i e j referem-se à linha i e à coluna j das matrizes \mathbf{B} , \mathbf{N} e \mathbf{N}^T

Primeiramente, transfere-se a derivação parcial sobre termos da matriz de condutividade nas equações 3.50 e 3.54 para o interior do integrando da Equação 3.56:

$$\frac{\partial K_{im}^e}{\partial \theta_j^e} = \int_{\Omega^e} \mathbf{B}_i^T \frac{\partial \mathbf{D}}{\partial \theta_j^e} \mathbf{B}_m dV \quad (3.57)$$

A parcela convectiva de \mathbf{K} , correspondente ao segundo termo à direita da igualdade na Equação 3.56, não possui dependência com as temperaturas nodais do elemento e por isso a sua derivada em relação a θ^e é nula.

Além disso, se observa na parcela condutiva que a matriz \mathbf{B} não possui dependência com a temperatura, portanto o operador de derivadas $\partial/\partial\theta_j^e$ atua somente sobre a matriz constitutiva, matriz \mathbf{D} .

Em materiais isotrópicos, a matriz constitutiva é uma matriz diagonal na qual a condutividade térmica é igual em todas as direções. Ao aplicar o operador de derivadas sobre \mathbf{D} , obtém-se:

$$\frac{\partial \mathbf{D}}{\partial \theta_j^e} = \frac{\partial}{\partial \theta_j^e} \begin{bmatrix} k(\theta) & 0 & 0 \\ 0 & k(\theta) & 0 \\ 0 & 0 & k(\theta) \end{bmatrix} = \begin{bmatrix} \partial k(\theta)/\partial \theta_j^e & 0 & 0 \\ 0 & \partial k(\theta)/\partial \theta_j^e & 0 \\ 0 & 0 & \partial k(\theta)/\partial \theta_j^e \end{bmatrix} \quad (3.58)$$

Como a temperatura é uma grandeza que varia dentro do domínio do elemento finito, a condutividade térmica também irá assumir um valor específico em cada ponto desse domínio quando ela depender da temperatura. Consequentemente, a matriz constitutiva também irá variar dentro do domínio do elemento finito.

No Método dos Elementos Finitos o campo de temperaturas no interior do elemento é calculado a partir dos valores que a temperatura assume nos nós, por meio das funções de interpolação conforme apresentado na Equação 3.9.

Substituindo a Equação 3.9 na Equação 3.55, obtém-se:

$$k(\theta) = a_0 + a_1 (\mathbf{N} \boldsymbol{\theta}^e) + a_2 (\mathbf{N} \boldsymbol{\theta}^e)^2 + \dots + a_n (\mathbf{N} \boldsymbol{\theta}^e)^n \quad (3.59)$$

As derivadas parciais de $k(\theta)$ em relação aos valores nodais de temperatura no elemento finito, θ_j^e , podem então ser calculadas:

$$\begin{aligned} \frac{\partial k(\theta)}{\partial \theta_j^e} &= a_1 \frac{\partial (\mathbf{N} \boldsymbol{\theta}^e)}{\partial \theta_j^e} + 2 a_2 (\mathbf{N} \boldsymbol{\theta}^e) \frac{\partial (\mathbf{N} \boldsymbol{\theta}^e)}{\partial \theta_j^e} + \dots \\ &+ n a_n (\mathbf{N} \boldsymbol{\theta}^e)^{(n-1)} \frac{\partial (\mathbf{N} \boldsymbol{\theta}^e)}{\partial \theta_j^e} \end{aligned} \quad (3.60)$$

Claramente, os termos $\frac{\partial(\mathbf{N} \boldsymbol{\theta}^e)}{\partial \theta_j^e}$ são iguais a N_j , logo:

$$\frac{\partial k(\boldsymbol{\theta})}{\partial \theta_j^e} = (a_1 + 2 a_2 (\mathbf{N} \boldsymbol{\theta}^e) + \dots + n a_n (\mathbf{N} \boldsymbol{\theta}^e)^{(n-1)}) N_j \quad (3.61)$$

Retornando a relação $\theta = \mathbf{N} \boldsymbol{\theta}^e$ na equação anterior, obtém-se:

$$\frac{\partial k(\theta)}{\partial \theta_j^e} = (a_1 + 2 a_2 \theta + \dots + n a_n \theta^{(n-1)}) N_j = k'(\theta) N_j \quad (3.62)$$

onde $k'(\theta)$ será referenciada ao longo dessa dissertação como condutividade térmica tangente do material, sendo correspondente ao termo entre parênteses na equação anterior, isto é:

$$k'(\theta) = (a_1 + 2 a_2 (\mathbf{N} \boldsymbol{\theta}^e) + \dots + n a_n (\mathbf{N} \boldsymbol{\theta}^e)^{(n-1)}) \quad (3.63)$$

Analogamente, será introduzida a nomenclatura matriz constitutiva tangente para a matriz diagonal cujos termos não nulos são a condutividade térmica tangente do material:

$$\mathbf{D}' = \begin{bmatrix} k'(\theta) & 0 & 0 \\ 0 & k'(\theta) & 0 \\ 0 & 0 & k'(\theta) \end{bmatrix} \quad (3.64)$$

A Equação 3.58 pode então ser reescrita utilizando a matriz constitutiva tangente, levando a:

$$\frac{\partial \mathbf{D}}{\partial \theta_j^e} = \frac{\partial}{\partial \theta_j^e} \begin{bmatrix} k(\theta) & 0 & 0 \\ 0 & k(\theta) & 0 \\ 0 & 0 & k(\theta) \end{bmatrix} = \mathbf{D}' \cdot N_j \quad (3.65)$$

Substituindo a expressão acima na Equação 3.57, tem-se:

$$\frac{\partial K_{im}^e}{\partial \theta_j^e} = \int_{\Omega_e} \mathbf{B}_i^T \mathbf{D}' \mathbf{B}_m N_j dV \quad (3.66)$$

Retornando agora às equações 3.50 e 3.54, deve-se desenvolver o seguinte somatório:

$$\sum_{m=1}^n \frac{\partial K_{im}^e}{\partial \theta_j^e} \theta_m^e = \frac{\partial K_{i1}^e}{\partial \theta_j^e} \theta_1^e + \dots + \frac{\partial K_{in}^e}{\partial \theta_j^e} \theta_n^e \quad (3.67)$$

Assim, substituindo a Equação 3.66 na Equação 3.67:

$$\sum_{m=1}^n \frac{\partial K_{im}^e}{\partial \theta_j^e} \theta_m^e = \int_{\Omega_e} (\mathbf{B}_i^T \mathbf{D}' \mathbf{B}_1) (N_j \theta_1^e) + \dots + (\mathbf{B}_i^T \mathbf{D}' \mathbf{B}_n) (N_j \theta_n^e) dV \quad (3.68)$$

Pode-se notar que a soma no integrando da Equação 3.68 pode ser representada pelo produto interno entre dois vetores, o que resulta em um escalar:

$$\sum_{m=1}^n \frac{\partial K_{im}^e}{\partial \theta_j^e} \theta_m^e = \int_{\Omega_e} \left\{ \mathbf{B}_i^T \mathbf{D}' \mathbf{B}_1 \quad \dots \quad \mathbf{B}_i^T \mathbf{D}' \mathbf{B}_n \right\} \begin{Bmatrix} N_j \theta_1^e \\ \vdots \\ N_j \theta_n^e \end{Bmatrix} dV = K'_{ij} \quad (3.69)$$

Dessa forma, o cálculo de todos os termos da matriz \mathbf{K}'^e pode ser realizado a partir da integração do produto matricial mostrado abaixo:

$$\mathbf{K}'^e = \int_{\Omega_e} \begin{bmatrix} \mathbf{B}_1^T \mathbf{D}' \mathbf{B}_1 & \dots & \mathbf{B}_1^T \mathbf{D}' \mathbf{B}_n \\ \vdots & \ddots & \vdots \\ \mathbf{B}_n^T \mathbf{D}' \mathbf{B}_1 & \dots & \mathbf{B}_n^T \mathbf{D}' \mathbf{B}_n \end{bmatrix} \begin{bmatrix} N_1 \theta_1^e & \dots & N_n \theta_1^e \\ \vdots & \ddots & \vdots \\ N_1 \theta_n^e & \dots & N_n \theta_n^e \end{bmatrix} dV \quad (3.70)$$

A equação acima pode então ser condensada :

$$\mathbf{K}'^e = \int_{\Omega_e} (\mathbf{B}^T \mathbf{D}' \mathbf{B}) (\boldsymbol{\theta}^e \mathbf{N}) dV \quad (3.71)$$

As matrizes de iteração utilizadas no processo de solução do problema não-linear em regime transiente e em regime permanente, equações 3.50 e 3.54, respectivamente, podem finalmente ser reescritas:

$$M_{ij}^{e(r-1)} = \frac{1}{\Delta t} C_{ij}^e + \beta \left(K_{ij}^{e(r-1)} + K'_{ij}{}^{e(r-1)} \right) \quad (3.72)$$

para o problema em regime transiente, e

$$M_{ij}^{e(r-1)} = K_{ij}^{e(r-1)} + K_{ij}^{re(r-1)} \quad (3.73)$$

para o problema em regime permanente.

3.2 Formulação do Problema da Mecânica dos Sólidos pelo Método dos Elementos Finitos

A formulação do problema mecânico segundo o MEF segue os mesmos princípios da formulação apresentada para a condução de calor na seção 3.1 desse capítulo. A abordagem adotada consiste em cancelar o resíduo no domínio do corpo, sob uma perspectiva integral, como enunciado na Equação 3.3.

Como essa dissertação de mestrado se restringe ao regime quasi-estático, que é um caso particular do regime transiente, o desenvolvimento apresentado nessa seção não irá considerar a contribuição dos termos inerciais.

Na mecânica dos sólidos a variável de estado é o deslocamento, que é um campo vetorial e será referido por \mathbf{u} .

Conforme detalhado por Zienkiewicz e Taylor (2000) e por Fish e Belytschko (2007), a formulação do problema da mecânica dos sólidos segundo o MEF parte da equação de equilíbrio em um elemento infinitesimal:

$$\sigma_{ij,j} + b_i = 0 \quad (3.74)$$

onde \mathbf{b} é o vetor das forças de corpo.

As funções peso são agrupadas em um vetor linha $\boldsymbol{\psi}$, e são independentes para cada uma das direções do campo de deslocamentos. Pode-se então escrever a forma integral forte:

$$\int_{\Omega} \psi_i (\sigma_{ij,j} + b_i) dV = 0 \quad (3.75)$$

Integrando o primeiro termo da equação acima por partes e aplicando o teorema da divergência, obtém-se:

$$\int_{\Omega} \psi_i \sigma_{ij,j} dV = \int_{\Gamma} \psi_i \sigma_{ij} n_j dA - \int_{\Omega} \psi_{i,j} \sigma_{ij} dV \quad (3.76)$$

Substituindo a Equação 3.76 na Equação 3.75:

$$\int_{\Gamma} \psi_i \sigma_{ij} n_j dA - \int_{\Omega} \psi_{i,j} \sigma_{ij} dV + \int_{\Omega} \psi_i b_i dV = 0 \quad (3.77)$$

Na equação acima, o produto $\sigma_{ij} n_j$ no contorno do corpo deve se igualar à componente p_i do vetor de forças de superfície \mathbf{p} .

A forma integral fraca do problema de valor de contorno referente ao meio contínuo é equivalente à formulação do princípio dos trabalhos virtuais, com as funções peso correspondendo aos deslocamentos virtuais (Zienkiewicz e Taylor, 2000).

Organizando as componentes do tensor de tensões em um vetor, o produto $\psi_{i,j} \sigma_{ij}$ pode ser representado pelo produto interno entre um vetor linha denominado $\delta\boldsymbol{\varepsilon}$, que também pode ser interpretado como as deformações virtuais, e o vetor $\boldsymbol{\sigma}$.

Dessa forma, a Equação 3.76 pode ser reescrita na forma matricial:

$$\int_{\Omega} \delta\boldsymbol{\varepsilon} \boldsymbol{\sigma} dV - \int_{\Omega} \boldsymbol{\psi} \mathbf{b} dV - \int_{\Gamma_p} \boldsymbol{\psi} \mathbf{p} dA = 0 \quad (3.78)$$

onde a integral de superfície foi reduzida à porção Γ_p do contorno baseando-se na propriedade de que $\boldsymbol{\psi}$ é nula na porção Γ_u de Γ .

Seguindo os preceitos do Método dos Elementos Finitos apresentados no início desse capítulo, desenvolve-se uma aproximação para o campo de deslocamentos no interior de um elemento finito conforme a relação:

$$\mathbf{u} = \mathbf{N} \mathbf{u}^e \quad (3.79)$$

Analogamente, aplica-se o mesmo expediente às funções peso:

$$\boldsymbol{\psi} = \boldsymbol{\psi}^{eT} \mathbf{N}^T \quad (3.80)$$

Desenvolve-se então os operadores sobre o campo de deslocamentos e sobre as funções peso:

$$\boldsymbol{\varepsilon} = (\mathbf{LN}) \mathbf{u}^e = \mathbf{B} \mathbf{u}^e \quad (3.81)$$

$$\delta \boldsymbol{\varepsilon} = \mathbf{L} (\boldsymbol{\psi}^{eT} \mathbf{N}^T) = \boldsymbol{\psi}^{eT} (\mathbf{LN})^T = \boldsymbol{\psi}^{eT} \mathbf{B}^T \quad (3.82)$$

Substituindo as equações 3.80 e 3.82 na Equação 3.78 e decompondo as integrações sobre o domínio e sobre o contorno do corpo em somatórios das integrais sobre os subdomínios:

$$\begin{aligned} & \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Omega^e} \mathbf{B}^T \boldsymbol{\sigma} dV - \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Omega^e} \mathbf{N}^T \mathbf{b} dV - \\ & \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Gamma^e \cap \Gamma_p} \mathbf{N}^T \mathbf{p} dA = 0 \end{aligned} \quad (3.83)$$

Conforme discutido em detalhes em Borelli et al. (2011), pode-se realizar uma decomposição do tensor de deformações em uma parcela associada à alteração geométrica do corpo ocasionada exclusivamente pelas ações mecânicas externas e uma parcela exclusivamente associada à variação da temperatura:

$$\varepsilon_{ij} = \varepsilon_{ij}^{mec} + \varepsilon_{ij}^{\theta} \quad (3.84)$$

onde $\boldsymbol{\varepsilon}$ é o tensor de deformações infinitesimais, correspondente à deformação total, $\boldsymbol{\varepsilon}_{mec}$ é o tensor das deformações líquidas decorrentes das ações mecânicas externas e $\boldsymbol{\varepsilon}_{\theta}$ é o tensor correspondente às deformações ocasionadas pela variação da temperatura.

No caso de materiais termicamente isotrópicos, as componentes do tensor $\boldsymbol{\varepsilon}_{\theta}$ podem ser expressas por:

$$\varepsilon_{ij}^{\theta} = \alpha(\theta - \theta_0)\delta_{ij} \quad (3.85)$$

em que α é o coeficiente de expansão térmica do material, θ_0 é a temperatura de referência na qual o ponto material está isento de tensões e θ é a temperatura corrente do corpo, obtida da solução do problema de transferência de calor.

A lei constitutiva do problema termomecânico, Equação 2.52, pode então ser representada na forma matricial, com os tensores $\boldsymbol{\varepsilon}$ e $\boldsymbol{\varepsilon}_{\theta}$ representados por vetores:

$$\boldsymbol{\sigma} = \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\theta}) \quad (3.86)$$

Substituindo a Equação 3.86 na Equação 3.83, obtém-se:

$$\begin{aligned} \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Omega^e} \mathbf{B}^T \mathbf{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\theta}) dV - \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Omega^e} \mathbf{N}^T \mathbf{b} dV - \\ \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Gamma^e \cap \Gamma_p} \mathbf{N}^T \mathbf{p} dA = 0 \end{aligned} \quad (3.87)$$

Substitui-se então a Equação 3.81 na Equação 3.87 e realiza-se uma reorganização dos termos:

$$\begin{aligned} \sum_{e=1}^m \boldsymbol{\psi}^{eT} \left(\int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{u}^e dV \right) - \sum_{e=1}^m \boldsymbol{\psi}^{eT} \left(\int_{\Omega^e} \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_{\theta} dV \right) - \\ \sum_{e=1}^m \boldsymbol{\psi}^{eT} \int_{\Omega^e} \mathbf{N}^T \mathbf{b} dV - \sum_{e=1}^m \boldsymbol{\psi}^{eT} \left(\int_{\Gamma^e \cap \Gamma_p} \mathbf{N}^T \mathbf{p} dA \right) = 0 \end{aligned} \quad (3.88)$$

Colocando as funções peso em evidência, a equação acima pode ser reescrita como:

$$\sum_{e=1}^m \boldsymbol{\psi}^{eT} \mathbf{R}^e = \mathbf{0} \quad (3.89)$$

onde o resíduo associado a cada um dos elementos finitos pode ser representado por:

$$\mathbf{R}^e = \left(\int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \right) \mathbf{u}^e - \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_{\theta} dV - \int_{\Omega^e} \mathbf{N}^T \mathbf{b} dV - \int_{\Gamma^e \cap \Gamma_p} \mathbf{N}^T \mathbf{p} dA \quad (3.90)$$

Na equação anterior, o vetor contendo os valores de deslocamento nos nós do elemento finito \mathbf{u}^e pôde ser removido do integrando visto que eles são grandezas pontuais e por isso não variam no domínio de integração.

Para que a relação descrita na Equação 3.89 seja válida para qualquer conjunto de funções $\boldsymbol{\psi}$, caracterizadas pelos coeficientes ψ_i para cada uma das direções cartesianas independentes nos nós que compõem a malha, o resíduo deve ser identicamente nulo, logo:

$$\mathbf{K}^e \mathbf{u}^e = \mathbf{F}^e = \mathbf{F}_{eq(\theta)}^e + \mathbf{F}_{eq(b)}^e + \mathbf{F}_{eq(p)}^e \quad (3.91)$$

sendo:

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} dV \quad (3.92)$$

$$\mathbf{F}_{eq(\theta)}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \boldsymbol{\varepsilon}_\theta dV \quad (3.93)$$

$$\mathbf{F}_{eq(b)}^e = \int_{\Omega^e} \mathbf{N}^T \mathbf{b} dV \quad (3.94)$$

$$\mathbf{F}_{eq(p)}^e = \int_{\Gamma^e \cap \Gamma_p} \mathbf{N}^T \mathbf{p} dV \quad (3.95)$$

onde \mathbf{K}^e é a matriz de rigidez do elemento finito e $\mathbf{F}_{eq(\theta)}^e$, $\mathbf{F}_{eq(b)}^e$ e $\mathbf{F}_{eq(p)}^e$ são os vetores dos carregamentos nodais equivalentes devidos à variação da temperatura, às forças de corpo e às forças de superfície, respectivamente.

Os símbolos \mathbf{K}^e e \mathbf{F}^e foram utilizados, em diferentes contextos, nas formulações segundo o MEF para o problema de transferência de calor e para o problema da mecânica dos sólidos. Apesar de terem significado físico diferente, esses termos possuem uma natureza matemática similar. O leitor deve atentar, portanto, ao contexto em que esses símbolos forem utilizados para se referirem às respectivas equações que os descrevem.

Combinando os resíduos em cada um dos elementos finitos e igualando o resultado a zero obtém-se o sistema de equações global:

$$\mathbf{K}\mathbf{u} = \mathbf{F} \quad (3.96)$$

onde \mathbf{F} é o vetor das ações externas do meio discretizado e \mathbf{K} é a matriz de rigidez do meio discretizado.

O sistema global precisa passar pela etapa final de condicionamento, onde são introduzidas as condições de contorno essenciais. Aplicando o Método da Eliminação, descrito anteriormente, obtém-se:

$$\begin{bmatrix} \mathbf{K}_{dd} & \mathbf{K}_{dc} \\ \mathbf{K}_{cd} & \mathbf{K}_{cc} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_d \\ \mathbf{u}_c \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_c \\ \mathbf{F}_d \end{Bmatrix} \quad (3.97)$$

onde \mathbf{u}_c é a parcela do vetor \mathbf{u} contendo os graus de liberdade cujo valor da variável de estado é conhecido a priori e \mathbf{u}_d é a parcela onde esses valores são desconhecidos.

O sistema da Equação 3.97 deve ser solucionado em duas etapas. Na primeira, resolve-se a porção superior referente aos graus de liberdade desconhecidos:

$$\mathbf{K}_{dd} \mathbf{u}_d = \mathbf{F}_c - \mathbf{K}_{dc} \mathbf{u}_c \quad (3.98)$$

Obtido o campo de deslocamentos, pode-se calcular as reações de apoio solucionando a porção inferior da 3.97:

$$\mathbf{F}_d = \mathbf{K}_{cd} \mathbf{u}_d + \mathbf{K}_{cc} \mathbf{u}_c \quad (3.99)$$

3.2.1 Dependência do coeficiente de expansão térmica com a temperatura

A lei constitutiva associada à resposta mecânica do material, Equação 2.52 apresentada no capítulo anterior, se restringe ao caso em que a variação da temperatura

ocorre em uma faixa estreita na qual o coeficiente de expansão térmica pode ser considerado constante.

De forma criteriosa, o coeficiente de expansão térmica é definido como a mudança relativa no comprimento L , ou outra dimensão linear, por unidade de mudança da temperatura absoluta T enquanto a tensão é mantida constante no material (Barron e Barron, 2012). Esse enunciado pode ser expresso matematicamente como:

$$\alpha(T) = \frac{1}{L} \left(\frac{\partial L}{\partial T} \right)_{\sigma} = \left(\frac{\partial \varepsilon}{\partial T} \right)_{\sigma} \quad (3.100)$$

Portanto, a parcela de deformação relacionada exclusivamente à variação de temperatura em materiais isotrópicos em cada uma das direções principais é expressa como:

$$\varepsilon_{\theta} = \int_{T_i}^{T_f} \alpha(T) dT \quad (3.101)$$

No caso particular em que a relação entre o coeficiente de expansão térmica e a temperatura possa ser representada matematicamente por uma função polinomial, tem-se:

$$\varepsilon_{\theta} = \alpha_0 \int_{T_i}^{T_f} dT + \alpha_1 \int_{T_i}^{T_f} T dT + \alpha_2 \int_{T_i}^{T_f} T^2 dT + \dots + \alpha_n \int_{T_i}^{T_f} T^n dT \quad (3.102)$$

onde $\alpha(T) = \alpha_0 + \alpha_1 T + \alpha_2 T^2 + \dots + \alpha_n T^n$.

Integrando a Equação 3.102, pode-se representar a deformação correspondente por:

$$\varepsilon_{\theta} = \sum_{j=1}^n \frac{\alpha_{j-1}}{j} (T_f^j - T_i^j) \quad (3.103)$$

Na análise transiente, a temperatura varia continuamente, e a deformação térmica pode então ser representada por uma função temporal por meio de uma mudança de variáveis:

$$\varepsilon_{\theta}(t) = \int_{t_i}^{t_f} \alpha(T(t)) \frac{\partial T(t)}{\partial t} dt \quad (3.104)$$

Para a relação polinomial entre o coeficiente de expansão térmica e a temperatura:

$$\begin{aligned} \varepsilon_{\theta}(t) = & \alpha_0 \int_{t_i}^{t_f} \frac{\partial T(t)}{\partial t} dt + \alpha_1 \int_{t_i}^{t_f} T(t) \cdot \frac{\partial T(t)}{\partial t} dt + \\ & \alpha_2 \int_{t_i}^{t_f} T^2(t) \cdot \frac{\partial T(t)}{\partial t} dt + \dots + \alpha_n \int_{t_i}^{t_f} T^n(t) \frac{\partial T(t)}{\partial t} dt \end{aligned} \quad (3.105)$$

Os termos da equação acima podem ser integrados por partes, resultando em:

$$\varepsilon_{\theta}(t) = \sum_{j=1}^n \frac{\alpha_{j-1}}{j} \cdot (T^j(t_f) - T^j(t_i)) \quad (3.106)$$

A formulação apresentada na Equação 3.106 é útil na definição do carregamento nodal equivalente relativo à variação de temperatura em problemas dependentes do tempo, Equação 3.93.

Ressalta-se que devido à diferença entre potências da variável temperatura nas equações 3.103 e 3.106, deve-se utilizar a escala de temperatura absoluta tanto para a realização dos cálculos como para representar a relação polinomial entre a variação do coeficiente de expansão térmica e a temperatura.

3.3 Formulação paramétrica do MEF

Conforme detalhado nas seções anteriores, ao discretizar o problema contínuo segundo o Método dos Elementos Finitos são obtidas matrizes que representam o comportamento do subdomínio, \mathbf{C}^e e \mathbf{K}^e , e vetores que representam a influência das ações externas sobre o subdomínio, \mathbf{F}^e , cujos termos são calculados a partir de integrais que operam sobre o domínio ou sobre o contorno do elemento finito.

Para algumas geometrias regulares, as referidas integrais podem ser avaliadas analiticamente em função de parâmetros como as coordenadas dos nós ou como a

largura e a altura do elemento. Todavia, em casos de aplicação prática frequentemente não é possível subdividir o domínio utilizando geometrias regulares, o que implica na necessidade de distorcer as geometrias de referência em formas arbitrárias (Zienkiewicz e Taylor, 2000).

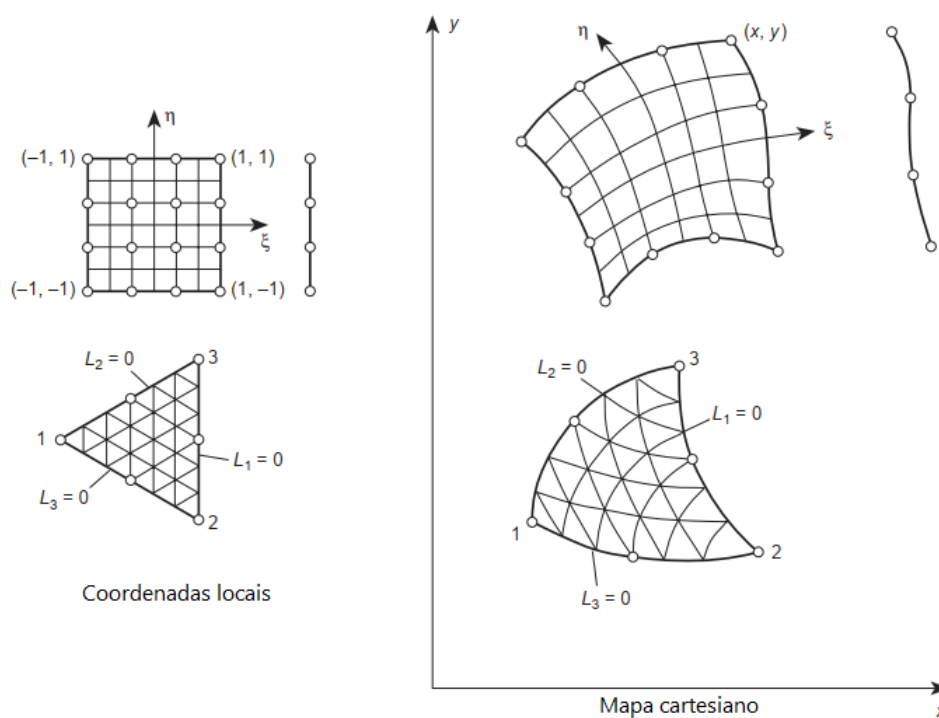


Figura 3.3: Mapeamento bidimensional de alguns elementos finitos (adaptado de Zienkiewicz e Taylor (2000)).

A formulação paramétrica do MEF se mostra atraente tanto por viabilizar o cálculo sobre geometrias complexas e genéricas, como por ser adequada à implementação em programas de computador.

Segundo descrito por Zienkiewicz e Taylor (2000), na formulação paramétrica é realizada uma transformação de coordenadas a fim de se obter uma geometria parametrizada, descrita no sistema de coordenadas local do elemento finito. Essa transformação de coordenadas é denominada mapeamento, e está mostrada ilustrativamente para o caso de problemas bidimensionais na Figura 3.3 .

O mapeamento é responsável por estabelecer uma correspondência biunívoca

entre o elemento finito em seu sistema local de coordenadas e o elemento finito em sua forma distorcida, definida no sistema de referência. Matematicamente, pode-se escrever:

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} f_x(\xi, \eta, \zeta) \\ f_y(\xi, \eta, \zeta) \\ f_z(\xi, \eta, \zeta) \end{Bmatrix} \quad (3.107)$$

Como discutido anteriormente, a variável de estado no interior de um elemento finito é aproximada utilizando os valores nodais e as funções de interpolação, funções N . Essa mesma metodologia pode ser utilizada na formulação paramétrica para descrever a geometria do elemento em função das coordenadas do nós.

Agrupando as coordenadas dos nós em cada uma das direções independentes, em vetores separados \mathbf{x}^e , \mathbf{y}^e e \mathbf{z}^e , pode-se aproximar a geometria do elemento por:

$$x = \mathbf{N} \mathbf{x}^e \quad (3.108)$$

$$y = \mathbf{N} \mathbf{y}^e \quad (3.109)$$

$$z = \mathbf{N} \mathbf{z}^e \quad (3.110)$$

onde as funções de forma na formulação paramétrica são escritas no sistema de coordenadas do elemento.

O cálculo dos termos das matrizes e vetores \mathbf{C}^e , \mathbf{K}^e e \mathbf{F}^e contempla derivações das funções de interpolação em relação às direções cartesianas. Como na formulação paramétrica as funções de interpolação são escritas em relação ao sistema de coordenadas do elemento finito, torna-se necessário aplicar a regra da cadeia, como mostrado nas equações abaixo:

$$\begin{pmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{pmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{pmatrix} = \mathbf{J} \begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{pmatrix} \quad (3.111)$$

onde \mathbf{J} é denominada matriz Jacobiana.

Substituindo as equações 3.108, 3.109 e 3.110 na matriz jacobiana, obtém-se:

$$\mathbf{J} = \sum_{i=1}^n \begin{bmatrix} N_{i,\xi} x_i & N_{i,\xi} y_i & N_{i,\xi} z_i \\ N_{i,\eta} x_i & N_{i,\eta} y_i & N_{i,\eta} z_i \\ N_{i,\zeta} x_i & N_{i,\zeta} y_i & N_{i,\zeta} z_i \end{bmatrix} \quad (3.112)$$

As derivadas das funções de interpolação em relação às coordenadas cartesianas podem ser obtidas a partir da inversão da matriz jacobiana:

$$\begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{pmatrix} = \mathbf{J}^{-1} \begin{pmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{pmatrix} \quad (3.113)$$

No processo de transformação de coordenadas, a geometria é parametrizada para que os limites de integração, em geral, sejam -1 e 1 a fim de facilitar a integração numérica por quadratura de Gauss.

A relação entre os diferenciais nas coordenadas originais e no sistema de coordenadas natural é definida pelo determinante da matriz jacobiana:

$$dx \, dy \, dz = |\mathbf{J}| \, d\xi \, d\eta \, d\zeta \quad (3.114)$$

As matrizes \mathbf{C}^e e \mathbf{K}^e e os vetores \mathbf{F}^e serão apresentados a seguir na forma paramétrica tanto para o problema de condução de calor como para o problema da mecânica dos sólidos.

3.3.0.1 Problema de transferência de calor

Na forma paramétrica, as equações 3.19, 3.20 e 3.21 assumem o formato:

$$\mathbf{C}^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \rho c_p \mathbf{N}^T(\xi, \eta, \zeta) \mathbf{N}(\xi, \eta, \zeta) |\mathbf{J}| d\xi d\eta d\zeta \quad (3.115)$$

$$\begin{aligned} \mathbf{K}^e = & \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T(\xi, \eta, \zeta) \mathbf{D}(\xi, \eta, \zeta) \mathbf{B}(\xi, \eta, \zeta) |\mathbf{J}| d\xi d\eta d\zeta + \\ & \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T(\xi, \eta, \zeta) h \mathbf{N}(\xi, \eta, \zeta) |\mathbf{J}|_{\xi=cte} d\eta d\zeta \end{aligned} \quad (3.116)$$

$$\mathbf{F}_{\dot{\epsilon}}^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T(\xi, \eta, \zeta) \dot{\epsilon} |\mathbf{J}| d\xi d\eta d\zeta \quad (3.117)$$

$$\mathbf{F}_{q_p''}^e = \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T(\xi, \eta, \zeta) q_p'' |\mathbf{J}| d\eta d\zeta \quad (3.118)$$

$$\mathbf{F}_{q_c}^e = \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T(\xi, \eta, \zeta) |_{\xi=cte} q_c |\mathbf{J}|_{\xi=cte} d\eta d\zeta \quad (3.119)$$

$$\mathbf{F}_h^e = \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T(\xi, \eta, \zeta) h \theta_{\infty} |\mathbf{J}|_{\xi=cte} d\eta d\zeta \quad (3.120)$$

$$\mathbf{K}'^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (\mathbf{B}^T(\xi, \eta, \zeta) \mathbf{D}'(\xi, \eta, \zeta) \mathbf{B}(\xi, \eta, \zeta)) \cdot (\boldsymbol{\theta}^e(\xi, \eta, \zeta) \mathbf{N}(\xi, \eta, \zeta)) |\mathbf{J}| d\xi d\eta d\zeta \quad (3.121)$$

3.3.0.2 Problema da mecânica dos sólidos

A forma paramétrica das equações relativas à matriz de rigidez, Equação 3.92, e aos vetores de força, equações 3.93 a 3.95, é apresentada a seguir:

$$\mathbf{K}^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T(\xi, \eta, \zeta) \mathbf{D} \mathbf{B}(\xi, \eta, \zeta) |\mathbf{J}| d\xi d\eta d\zeta \quad (3.122)$$

$$\mathbf{F}_{\boldsymbol{\theta}}^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T(\xi, \eta, \zeta) \mathbf{D} \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}(\xi, \eta, \zeta) |\mathbf{J}| d\xi d\eta d\zeta \quad (3.123)$$

$$\mathbf{F}_{\mathbf{b}}^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T(\xi, \eta, \zeta) \mathbf{b} |\mathbf{J}| d\xi d\eta d\zeta \quad (3.124)$$

$$\mathbf{F}_p^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T(\xi, \eta, \zeta) \mathbf{p} |\mathbf{J}| d\xi d\eta d\zeta \quad (3.125)$$

3.3.1 Integração numérica

As integrais presentes nas equações 3.115 a 3.125 geralmente são calculadas numericamente quando o processo de cálculo é implementado em um programa de computador. Usualmente, emprega-se a Quadratura de Gauss-Legendre para efetuar a integração numérica. Essa técnica é explicada em detalhes dentro do contexto do Método dos Elementos Finitos pelos autores Zienkiewicz e Taylor (2000), Navarra (1995), Weaver e Johnston (1984), entre outros.

Ao se realizar uma integração numérica, o domínio de integração é substituído por pontos específicos e por pesos associados, sendo o resultado obtido a partir de um somatório do valor que o integrando assume nesses pontos, denominados pontos de integração, multiplicados pelo fator peso associado, conforme mostrado a seguir para domínios unidimensionais:

$$I = \int_{-1}^1 f(\xi) d\xi \approx \sum_{i=1}^p W_i f(\xi_i) \quad (3.126)$$

onde W_i é o peso associado ao ponto de integração ξ_i .

Na Quadratura de Gauss-Legendre o número e a disposição dos pontos de integração e os respectivos fatores-peso são fixados para polinômios de várias ordens, visando à melhor precisão para a aproximação. Neste método, os pontos de integração usualmente são denominados pontos de Gauss. Quando o integrando é polinomial, uma quadratura contendo n pontos de Gauss integra com exatidão um polinômio de grau $2n - 1$.

Para efetuar a integração em mais de uma direção independente, é usual adotar a integração unidimensional de Gauss sucessivamente para cada uma das direções (Zienkiewicz e Taylor, 2000). Neste caso, devem ser introduzidos em cada direção um número de pontos de Gauss suficiente considerando o grau do polinômio obtido

naquela direção, considerando as variáveis associadas às demais direções como constantes. As bibliográficas relacionadas à aplicação da integração numérica no contexto do MEF, citadas previamente, dispõem de tabelas indicando a posição dos pontos de integração e os respectivos valores dos fatores peso para diversas geometrias de elementos finitos.

3.3.1.1 Particularidades da análise térmica fisicamente não-linear

No caso da transferência de calor fisicamente não-linear, deve-se atentar à elevação de ordem do integrando tanto na matriz de condutividade \mathbf{K} como da matriz de condutividade não-linear \mathbf{K}' devido às matrizes \mathbf{D} e \mathbf{D}' conterem potências da matriz \mathbf{N} nos seus termos. Essas potências se devem à dependência entre a condutividade térmica e a temperatura, descrita nesse trabalho por uma relação polinomial. As expressões para a condutividade térmica $k(\theta)$ e para a condutividade térmica tangente $k'(\theta)$ podem ser consultadas nas equações 3.59 e 3.63, respectivamente.

3.3.1.2 Particularidades da análise mecânica

Quando houver dependência entre o coeficiente de expansão térmica e a temperatura segundo uma relação polinomial, deve-se atentar à eventual elevação de ordem do integrando do carregamento nodal equivalente relacionado à variação de temperatura, $\mathbf{F}_{eq(\theta)}^e$.

Assim como no problema de transferência de calor fisicamente não-linear, deve-se assegurar que número de pontos de integração esteja compatível com a ordem do integrando durante o cálculo de $\mathbf{F}_{eq(\theta)}^e$, para que não haja perda de precisão.

3.4 Problema Termomecânico desacoplado em regime quasi-estático

Na análise termomecânica tanto a distribuição de temperaturas como as ações mecânicas externas podem variar continuamente ao longo do tempo. Como discutido anteriormente, em casos particulares nos quais essas variações ocorram de forma lenta e gradual o problema termomecânico pode ser simplificado ao se negligenciar o termo de acoplamento termoelástico na equação da condução do calor e o termo inercial na equação do movimento, o que corresponde a estudá-lo no regime quasi-estático. Tal consideração permite que o campo de temperaturas seja obtido independentemente da solução do problema da mecânica dos sólidos. Além disso, ao se desprezar os efeitos inerciais, a solução do problema mecânico em um instante de tempo qualquer pode ser calculada considerando somente a variação total das ações mecânicas externas e a variação total do campo de temperatura entre o instante inicial e o instante corrente.

Como será discutido nos próximos capítulos, diversas abordagens podem ser empregadas para a obtenção das soluções térmica e mecânica nos instantes de tempo desejados. Uma delas, que foi a considerada nesse trabalho, consiste em avançar alternadamente nas análises térmica e mecânica como resumido na figura a seguir. As etapas da análise foram identificadas com algarismos arábicos para futura referência. Três macroetapas também foram identificadas com as letras A, B e C correspondendo, respectivamente, à macroetapa de cálculo da distribuição de temperaturas ao longo do tempo, à macroetapa de transporte e manipulação de informações entre as físicas envolvidas e à macroetapa de cálculo do campo de deslocamentos na solução mecânica.

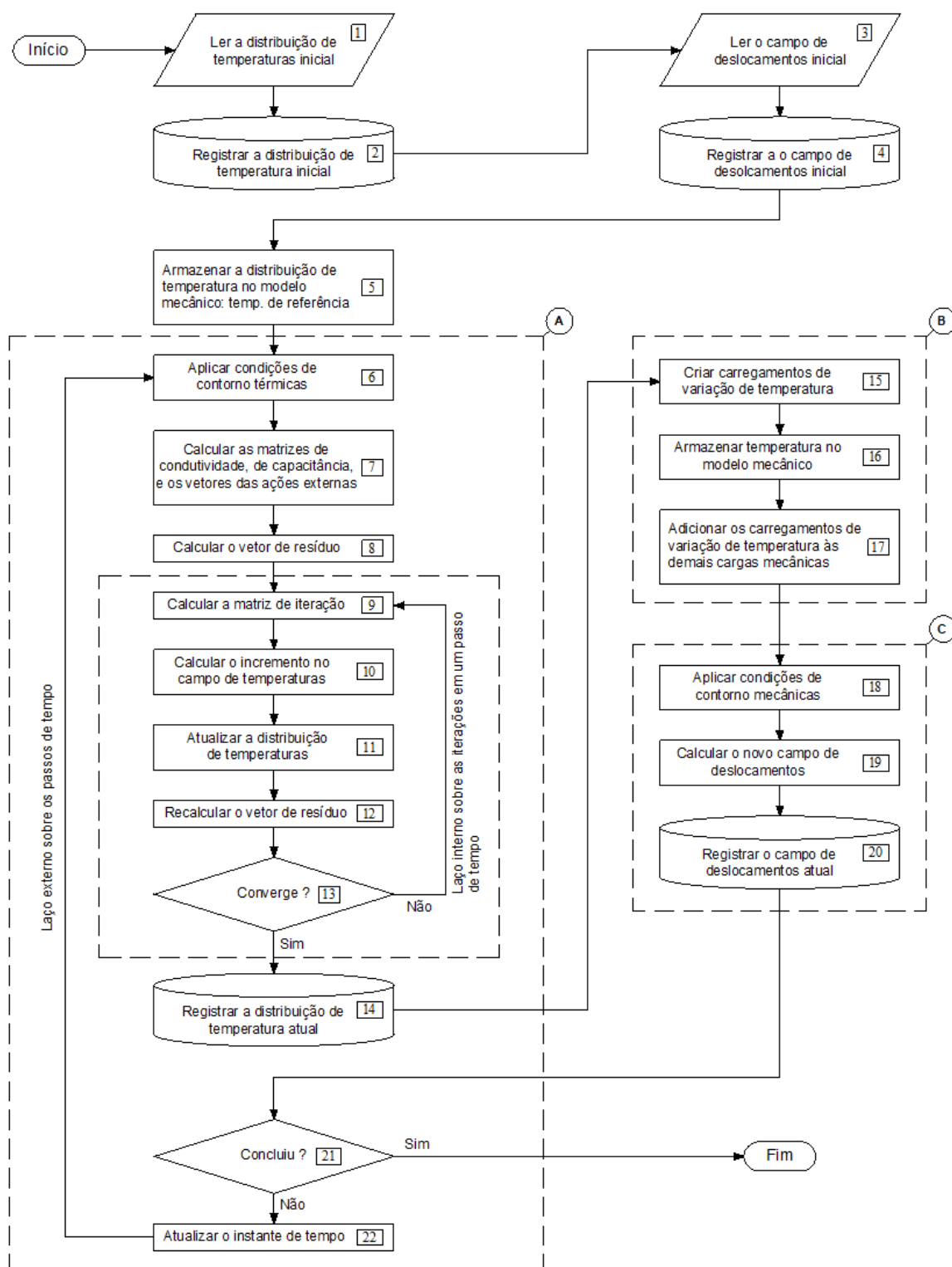


Figura 3.4: Fluxograma para solução do PTM transiente fisicamente não-linear.

Capítulo 4

IMPLEMENTAÇÃO COMPUTACIONAL

O Capítulo 1 realiza uma breve apresentação do sistema computacional INSANE, que é um programa de computador livre, desenvolvido majoritariamente na linguagem JAVA no Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais. As modificações e expansões realizadas nesse programa para atingir os objetivos desse trabalho são apresentadas em detalhes nesse capítulo.

Inicia-se com uma visão geral sobre as entidades que compõem o INSANE, exposta na seção 4.1. Em seguida, a lista dos requisitos que devem ser atendidos para viabilizar a solução de problemas multifísicos termomecânicos no âmbito do Método dos Elementos Finitos utilizando o INSANE é brevemente discutida na seção 4.2. Esses requisitos foram determinados a partir de uma análise do projeto orientado a objetos do núcleo numérico do INSANE, que está disponível no Apêndice A. A partir dessa análise, pôde-se identificar os trechos de código passíveis de reutilização com ou sem alterações, bem como novas funcionalidades que precisavam ser introduzidas.

A descrição detalhada da implementação computacional se inicia na seção 4.3 com a apresentação dos novos recursos para solução de problemas de transferência de calor. Na sequência, são mostradas na seção 4.4 as novas classes para a representação das condições iniciais e das condições de contorno essenciais, e as novas classes para a representação dos conjuntos formados pela condição inicial e pelas condições de contorno essenciais e naturais.

Na seção 4.5, a entidade responsável pelo gerenciamento do fluxo de execução durante o processo de solução é abordada. Essa entidade trabalha em conjunto com duas novas entidades também introduzidas nesse trabalho: um configurador do ambiente de solução e um gerenciador da estrutura de dados. Dedicar-se a seção 4.6 às estratégias adotadas para conferir um comportamento multifísico às entidades que são dependentes da física sendo solucionada.

Por fim, as novas classes para tratar das particularidades do problema termomecânico em regime quasi-estático na presença de não-linearidade física na parcela mecânica são detalhadas na seção 4.7, e as modificações necessárias na interface de entrada e saída de dados para lidar com as novas informações são abordadas na seção 4.8.

Para melhor visualização das contribuições desse trabalho ao núcleo numérico do INSANE, as classes modificadas durante o desenvolvimento são representadas em amarelo e as novas classes introduzidas são representadas em verde, como ilustrado na Figura 4.1.

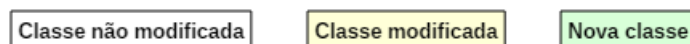


Figura 4.1: Padrão de cores utilizado na representação UML.

4.1 Visão geral do núcleo numérico do INSANE

O INSANE pode ser dividido basicamente em três grandes aplicações independentes: pré-processador, processador e pós-processador. Essas aplicações são responsáveis, respectivamente, pela geração da estrutura de dados que representa o problema a ser analisado computacionalmente, pela solução desse problema, e pela exibição dos resultados obtidos.

Como o trabalho apresentado nessa dissertação se concentrou no processador, não serão abordadas as aplicações de pré e de pós-processamento. Entretanto, ressalta-se

que o núcleo se comunica com essas aplicações.

O núcleo numérico do INSANE é resultado das contribuições de diversos trabalhos, dos quais citam-se em especial Penna (2011) e da Fonseca (2008). Esse núcleo era composto, previamente a esse trabalho, principalmente pelas entidades **Assembler**, **Model**, **Persistence** e **Solution**, que não tiveram os seus propósitos alterados.

De uma forma geral, após o disparo do processo de solução pelo usuário, o INSANE percorre as combinações de carregamentos definidas para a análise e, para cada uma delas, solicita à entidade **Solution** que obtenha os resultados desejados. Para calcular a solução, é necessário obter uma representação algébrica do sistema de equações correspondente ao meio discretizado, e para isso a entidade **Solution** realiza solicitações à entidade **Assembler** para obter as matrizes e os vetores desejados.

Como foi discutido no capítulo anterior, o comportamento de um meio discretizado pode ser determinado pelo arranjo apropriado das contribuições de cada subdomínio que constitui o meio. A organização dessas contribuições dos subdomínios é realizada pela entidade **Assembler**. Todavia, o comportamento de cada subdomínio não é calculado pelo **Assembler**, e sim pela entidade **Model**. Essa estratégia confere ao **Assembler** um maior nível de abstração, uma vez que o processo de montagem das matrizes e dos vetores do sistema de equações algébricas que representa o meio discretizado é indiferente ao problema físico abordado. Portanto, ao longo do processo de montagem, a entidade **Assembler** se comunica continuamente com a entidade **Model** para obter as informações necessárias.

A entidade **Model** contém toda a estrutura de dados relacionada ao meio discretizado, e tanto o seu preenchimento para a realização dos cálculos, como o registro dos resultados nela armazenados após a conclusão do processo de solução é realizado pela entidade **Persistence**.

4.2 Requisitos a serem atendidos

Dentre os requisitos discutidos detalhadamente no Apêndice A, distinguem-se as seis grandes demandas listadas a seguir.

Requisito 1: Introdução de novas classes para a representação e solução do problema de transferência de calor contemplando a dependência temporal e a não-linearidade física;

Requisito 2: Introdução de novas classes para a representação das condições iniciais e condições de contorno essenciais. Nesta mesma linha, também observou-se ser necessário criar novas classes para representar combinações multifísicas, onde são especificados conjuntos de condições iniciais e condições de contorno naturais e essenciais para cada uma das físicas.

Requisito 3: Criação de um mecanismo de gerenciamento que proporcione controle do fluxo de execução durante o processo de solução, permitindo dessa forma:

- Colocar o processo de solução do PTC em estado de espera ao final de um passo de tempo;
- Solicitar à persistência o registro dos resultados obtidos no passo de tempo corrente da solução térmica;
- Transferir informações entre a análise térmica e a mecânica, introduzindo no comportamento mecânico os efeitos da modificação da distribuição de temperaturas. Isso implica tanto na criação e aplicação de carregamentos de variação de temperatura na análise mecânica, como na eventual alteração das propriedades constitutivas mecânicas;
- Realizar a solução mecânica para o passo de tempo corrente e colocá-la em estado de espera após a conclusão referente ao passo

de tempo corrente;

- Solicitar à persistência o registro dos resultados obtidos no passo de tempo corrente da solução mecânica;
- Retornar o fluxo de execução à solução térmica para que seja obtida a distribuição de temperaturas no passo de tempo seguinte.

Requisito 4: Criação de mecanismos para configurar o ambiente de solução em cada caso particular. Dentre as configurações possíveis, ressalta-se a aplicação das condições iniciais e de contorno aplicáveis ao PTC e ao PTM, bem como a manipulação das entidades que tenham comportamento multifísico.

Requisito 5: Criação de mecanismos para representar e gerir a estrutura de dados do modelo discretizado no caso multifísico. Como detalhado no Apêndice A, no regime permanente a mesma instância do modelo pode ser utilizada para ambas as físicas, desde que seja realizada uma adequação nas entidades que se comportem de maneira diferente de acordo com a física ativa. Essa manipulação é realizada durante a transição entre a análise térmica e a mecânica. Para soluções em regime transiente, propõe-se utilizar duas instâncias do modelo discreto, uma para cada física, sendo uma cópia da outra com as devidas adequações nas entidades que possuam comportamento dependente da física. Tal escolha permite uma maior reutilização de código.

Requisito 6: Introdução de novas classes para tratarem as particularidades da solução mecânica quando ela for fisicamente não-linear e contemplar efeitos de variação de temperatura. Esse requisito se deve à forma específica de introduzir gradativamente os carregamentos de variação de temperatura segundo os algoritmos incrementais-iterativos e à forma específica de verificar a obtenção do equilíbrio ao final da iteração.

Reforça-se aqui que detalhes sobre a análise mecânica fisicamente não-linear não foram abordados no Capítulo 3. Entretanto, recursos para esse tipo de análise encontram-se disponíveis no INSANE e foram utilizados nesse trabalho a partir de adaptações simples, demonstrando as potencialidades de um projeto baseado na programação orientada a objetos.

4.3 Expansões para a solução do PTC

4.3.1 Expansões no projeto solution

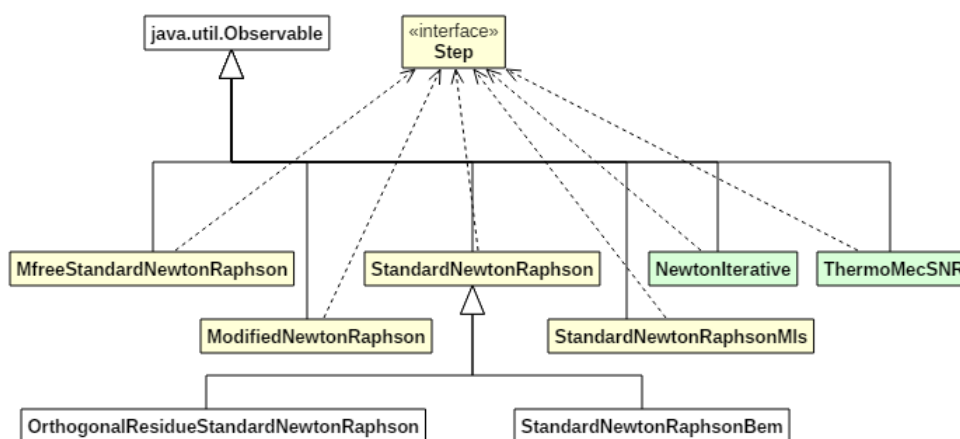


Figura 4.2: Diagrama de expansão da interface Step.

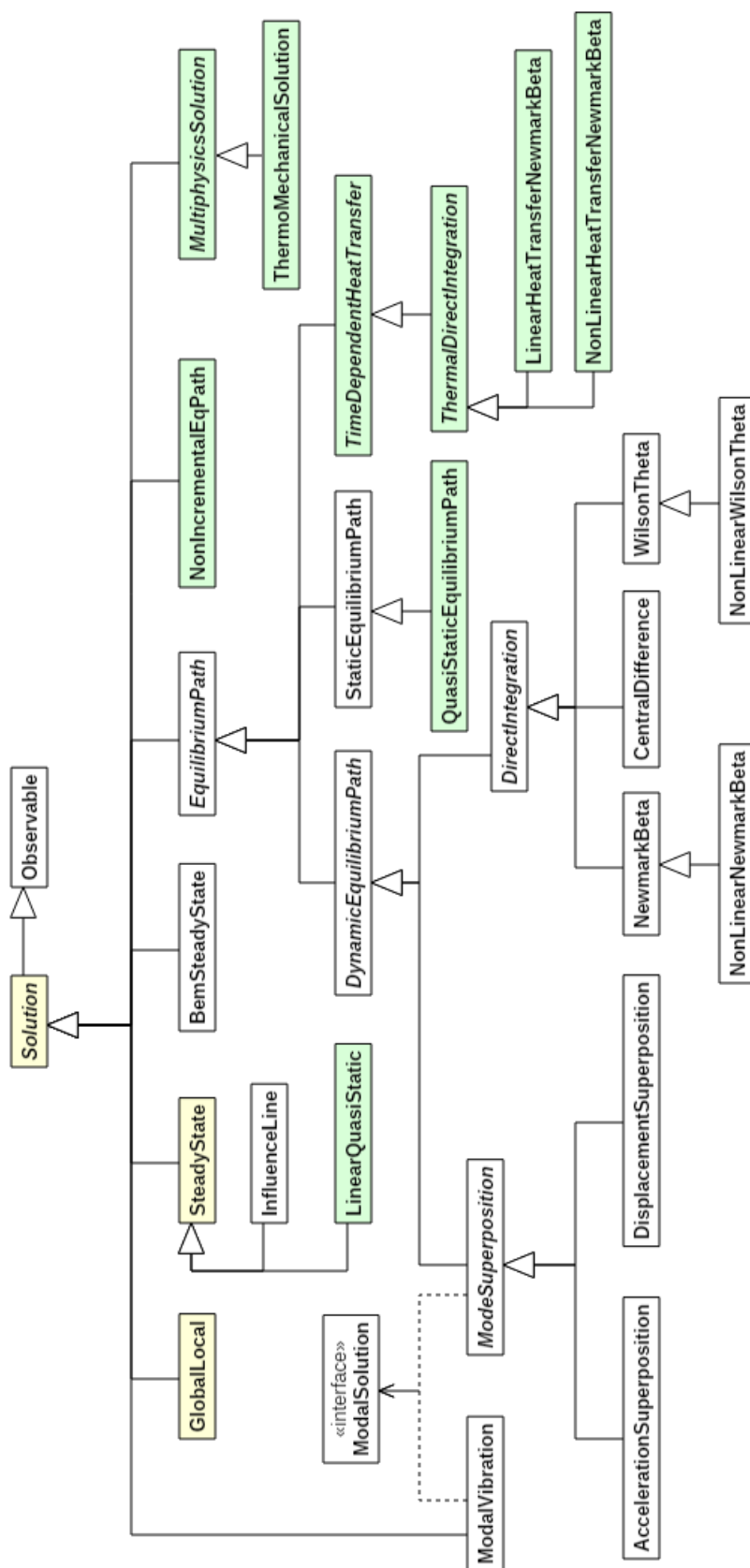


Figura 4.3: Diagrama de classe da expansão de `Solution`.

Quanto aos algoritmos para cálculo da solução do PTC, foram introduzidas seis novas classes no projeto `solution`:

- `TimeDependentHeatTransfer`: Classe abstrata para agrupar algoritmos para solução do PTC no domínio dependente do tempo. Contém apenas três atributos sendo um deles uma referência a um objeto do tipo `Assembler`, e os outros dois referências à condição de contorno natural e à condição de contorno essencial. Também encontram-se nessa classe métodos de acesso e modificação a esses atributos;
- `ThermalDirectIntegration`: Classe abstrata para agrupar algoritmos de solução, pela técnica de integração direta, de problemas de transferência de calor dependentes do tempo. Contém referências às submatrizes que constituem a matriz de capacitância e a matriz de condutividade, equações 3.19 e 3.20, parâmetros relativos ao método de integração direta como o tamanho do passo de tempo, Δt , e o tempo final da análise, t_f e dois campos auxiliares, sendo um para os instantes de tempo atual e seguinte, (t) e $(t + 1)$, e um para o vetor das ações externas, \mathbf{F} . Os métodos implementados são utilizados para acesso e modificação dos atributos;
- `LinearHeatTransferNewmarkBeta`: Classe que implementa o algoritmo de integração direta para problemas de condução de calor fisicamente lineares em regime transiente, descrito no item 3.1.1.
- `NonLinearHeatTransferNewmarkBeta`: Especialização da classe `LinearHeatTransferNewmarkBeta` para a solução do PTC fisicamente não-linear conforme o algoritmo descrito na seção 3.1.3.1.
- `NonIncrementalEqPath`: Classe que tem por objetivo iniciar o processo de solução de problemas de transferência de calor fisicamente não-lineares no regime permanente, cujo algoritmo de solução é implementado em um objeto do

tipo `Step`. Possui um comportamento semelhante ao da classe `StaticEquilibriumPath`, sendo a principal diferença a remoção da parte incremental do processo de solução incremental-iterativo.

- **NewtonIterative**: Classe que implementa a interface `Step` para solucionar problemas de transferência de calor fisicamente não-lineares em regime permanente segundo o algoritmo descrito no item 3.1.3.2.

Os diagramas de classe ilustrando a expansão no pacote `solution` são mostrados nas figuras 4.2 e 4.3. Nestes diagramas podem ser notadas outras novas classes que são aplicáveis à solução multifísica e serão abordadas posteriormente.

4.3.2 Expansões no pacote assembler

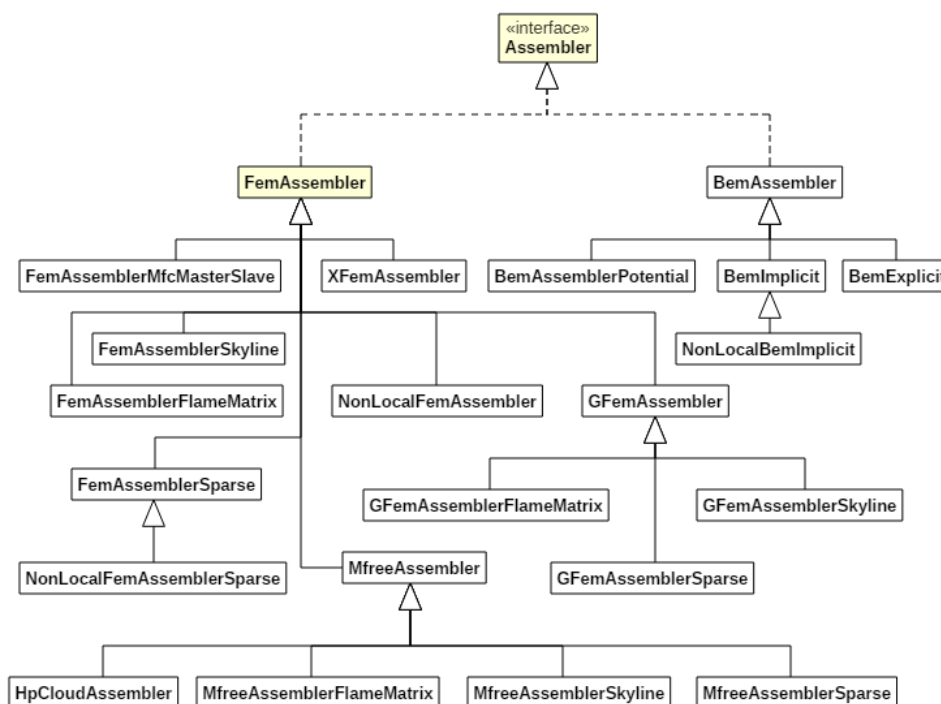


Figura 4.4: Diagrama de expansão da interface `Assembler`.

Para o cálculo de problemas de transferência de calor em regime transiente, dois novos métodos foram inseridos na interface `Assembler` e implementados na

classe `FemAssembler`, a saber: `setDerivedXp` e `getXp`, como destacado na Figura 4.4. O primeiro método é responsável por registrar ao longo do processo de solução os valores da taxa de variação temporal da temperatura nos graus de liberdade prescritos quando a condição de contorno essencial não é constante. O segundo método mencionado acima é utilizado para obter o valor dos graus de liberdade prescritos na condição inicial, e também possui implementações para que os valores prescritos sejam obtidos para uma combinação específica das condições de contorno essenciais no instante inicial ou no instante de tempo informado.

4.3.3 Expansões no projeto `model`

4.3.3.1 Expansões no pacote `analysisModel`

Uma nova classe herdeira de `AnalysisModel` foi introduzida para descrever modelos de análise axissimétricos para a transferência de calor. Essa classe, denominada `HeatTransferAxisymmetric`, pode ser visualizada no diagrama de classes da Figura 4.5. As outras classes adicionadas serão descritas posteriormente. As alterações nas classes existentes e as demais classes inseridas nesse pacote também serão explicadas nas próximas seções desse capítulo.

4.3.3.2 Expansões no pacote problemDriver

A classe `HeatTransferPd`, previamente implementada para problemas de transferência de calor lineares em regime permanente foi expandida com a implementação do método `getB`, responsável pelo cálculo da matriz de capacitância do elemento, equações 3.19 e 3.20. Para o cálculo das matrizes e vetores adicionais, presentes no processo de solução fisicamente não-linear, matriz \mathbf{K}' dada pela Equação 3.71 e vetor \mathbf{R} dado pela Equação 3.23, especializou-se a classe `HeatTransferPd` na classe `NonLinearHeatTransfer`, mostrada no diagrama de classes da Figura 4.6. Essa nova classe contém métodos que atendem às necessidades do problema de transferência de calor fisicamente não-linear tanto em regime permanente como do em regime transiente. As alterações na classe `Parametric` e as novas classes para o tratamento do problema multifísico serão discutidas posteriormente.

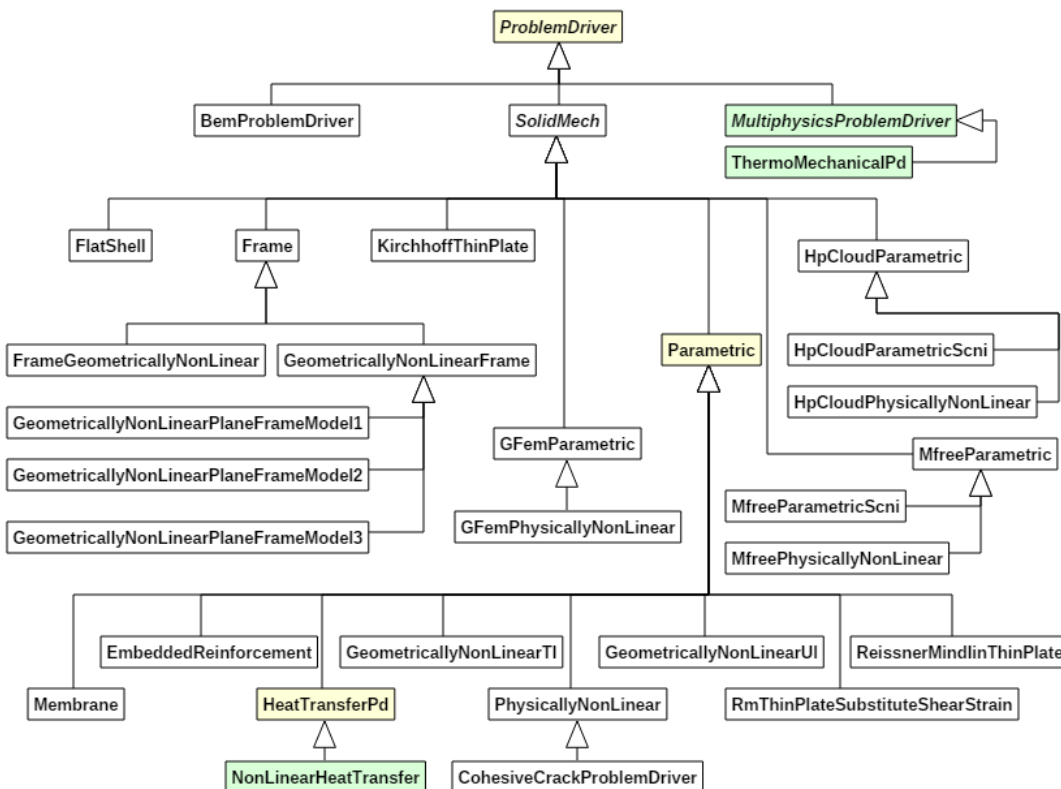


Figura 4.6: Diagrama de expansão da classe `ProblemDriver`.

4.3.3.3 Expansões no pacote `constitutiveModel`

Para representar a lei constitutiva nos problemas de condução de calor formulados segundo a Lei de Fourier, introduziu-se uma nova classe denominada `HeatTransfer-ConstitutiveModel`. Visto que outros modelos constitutivos para transferência de calor podem ser implementados posteriormente, criou-se também uma superclasse denominada `ThermalConstitutiveModels` meramente com o objetivo de agrupar suas herdeiras sob um tipo comum. O diagrama de classes da Figura 4.7 ilustra essa expansão. Esse diagrama também contém classes para o tratamento do problema multifísico, que serão explicadas nas seções que se seguem.

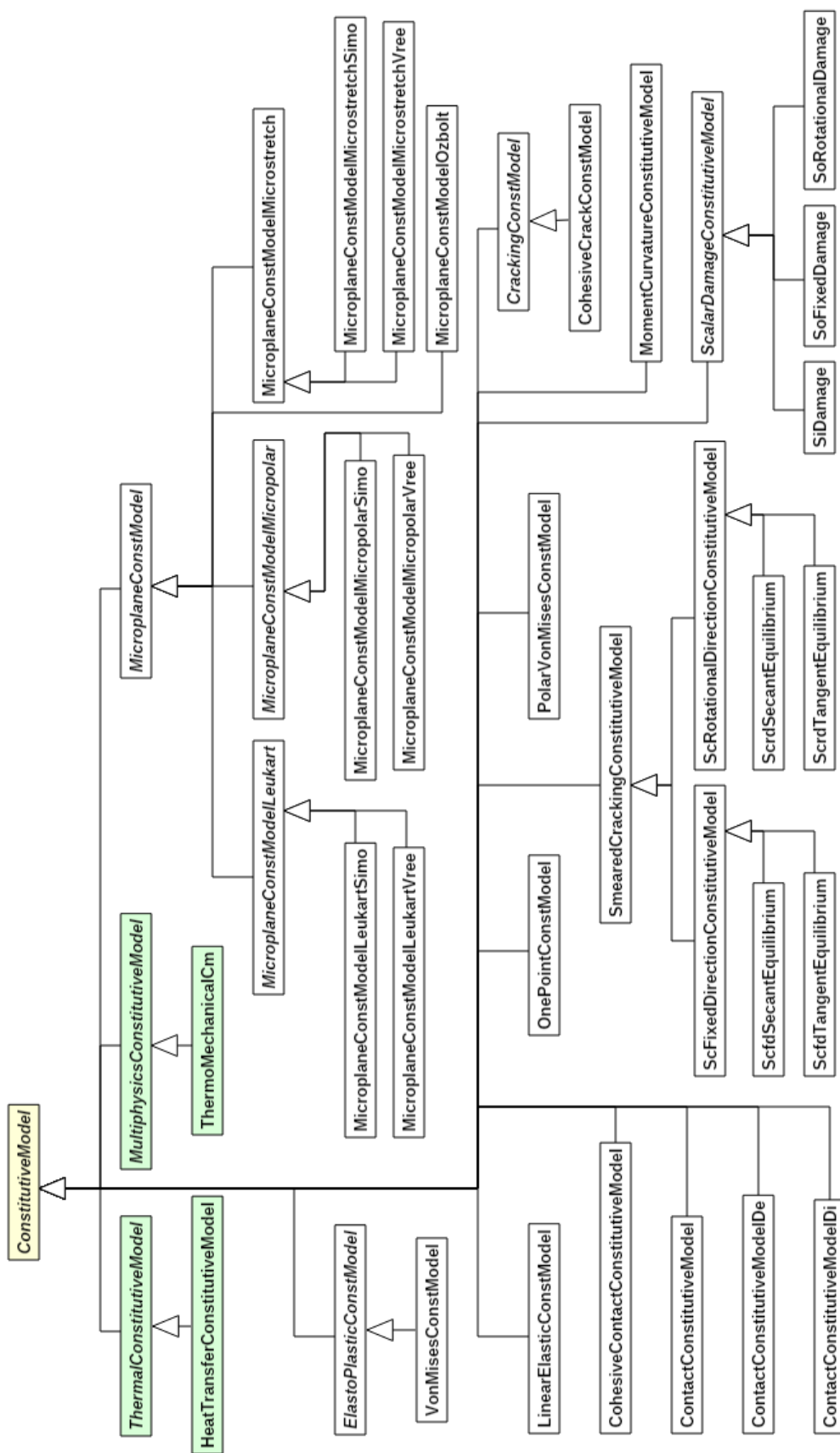


Figura 4.7: Diagrama de expansão da classe ConstitutiveModel.

4.3.3.4 Expansões no pacote `Material`

Para representar materiais lineares e isotrópicos dentro do âmbito da condução de calor, criou-se a classe `ThermalIsotropicMaterial`. Essa classe contém os atributos básicos para descrever as propriedades físicas relacionadas à transferência de calor e métodos para modificar ou para retornar esses valores. A implementação realizada nessa classe atende aos casos em que o meio possui um comportamento linear. Também foram adicionadas três chaves na classe `Material`, a saber: `CONDUCTIVITY`, `SPECIFIC_HEAT` e `EMISSIVITY`.

A condução de calor fisicamente não-linear é caracterizada pela dependência entre as propriedades constitutivas do PTC e a temperatura, que é a grandeza buscada durante o processo de solução. Para conferir essa funcionalidade de dependência em relação à temperatura, criou-se uma interface no pacote `Material`, denominada `TemperatureDependable`. Essa interface especifica quatro métodos, a saber:

- *getMaterialValues*: Retorna o valor de uma propriedade física armazenado no mapa de um objeto da classe `Material` dada a chave que identifica a propriedade;
- *getTemperature*: Retorna o valor de temperatura armazenado no objeto que implementa a interface `TemperatureDependable`. Essa é a temperatura considerada para a avaliação das propriedades físicas do material;
- *getTemperatureDependenceIntegrationOrder*: Informa o máximo grau das funções que descrevem a dependência entre as propriedades físicas e a temperatura. Esse valor é utilizado para determinar o número de pontos de integração durante a etapa de integração numérica das matrizes e vetores que representam o comportamento do meio;

- *setTemperature*: Atribui um novo valor de temperatura ao campo que armazena essa informação no objeto que implementa a interface `TemperatureDependable`.

A interface `TemperatureDependable` foi então implementada na classe abstrata `TemperatureDependableMaterial`, que estende diretamente a classe abstrata `Material`. A classe `TemperatureDependableMaterial` funciona como uma cápsula capaz de adicionar a funcionalidade de dependência em relação à temperatura a qualquer um dos materiais previamente implementados no INSANE. Para tal, essa classe possui dois atributos: a temperatura na qual as propriedades físicas devem ser avaliadas e uma referência a um objeto do tipo `Material`. Na prática, toda vez que um método é evocado na classe `TemperatureDependableMaterial`, ela atualiza as propriedades físicas do material armazenado em si conforme a temperatura que está armazenada em seus atributos, e em seguida evoca o mesmo método em seu objeto interno do tipo `Material`. Essa classe possui dois métodos abstratos, que se referem a forma de cálculo das propriedades físicas e suas derivadas em relação à temperatura dado o valor da temperatura. Essa abstração flexibiliza a forma de descrever a relação entre as propriedades físicas e a temperatura, que pode ser dada por uma tabela de valores, por um polinômio, entre outras possibilidades.

Como essa dissertação de mestrado se concentra no caso particular em que a dependência em relação à temperatura é descrita por uma relação polinomial, criou-se a classe `PolynomialTempDepMaterial`, que estende a classe `TemperatureDependableMaterial` incluindo como um novo atributo um mapa para armazenar os coeficientes dos polinômios para cada uma das propriedades físicas, e métodos para o cálculo das propriedades físicas e suas derivadas a partir do valor de temperatura armazenado na sua superclasse.

Por fim, a classe `PolynomialTempDepMaterial` é especializada em duas novas classes, uma utilizada para a análise térmica e uma para a análise mecânica. Essas duas classes diferem entre si especialmente na forma de sobrecarregarem o método

getPt, que retorna as propriedades tangentes do material. Na versão para o PTC, deve-se utilizar a derivada em relação à temperatura do polinômio que descreve essa dependência, como descrito pela equação 3.64 na seção 3.1.3.3. Já no caso da mecânica dos sólidos, utiliza-se a o polinômio em si, e não a sua derivada, para avaliar o valor da propriedade física segundo enunciado na Equação 3.102.

A novas classes e a nova interface adicionada podem ser observadas no diagrama da Figura 4.8. As classes para aplicações multifísicas serão abordadas posteriormente.

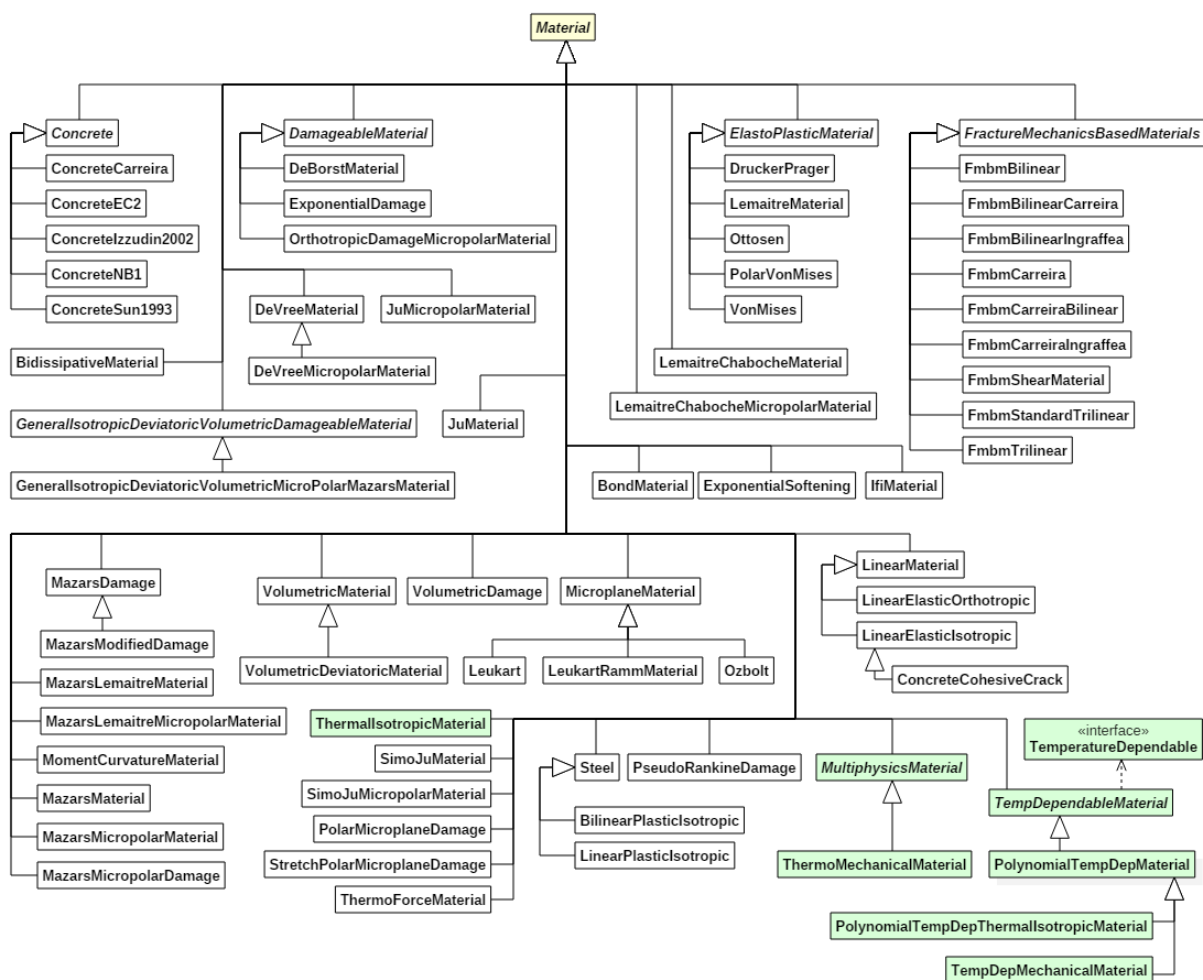


Figura 4.8: Diagrama de expansão da classe Material.

4.3.3.5 Expansões no pacote shape

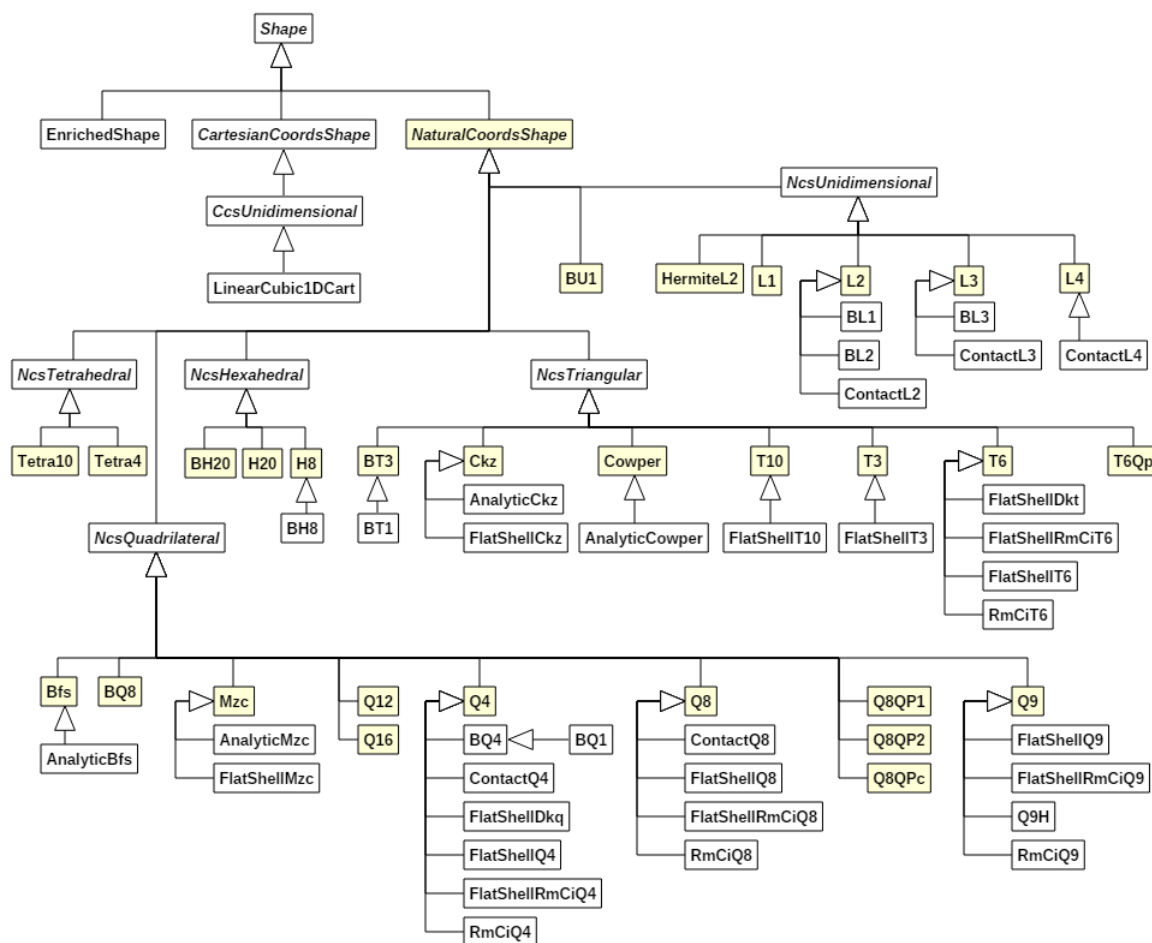


Figura 4.9: Diagrama de expansão da classe Shape.

Conforme discutido na seção 3.3.1.1, quando a condutividade térmica varia com a temperatura segundo uma relação polinomial, os termos da matriz de constitutiva secante e da matriz constitutiva tangente são polinômios, o que eleva o grau do integrando das matrizes de condutividade.

Como a implementação apresentada nesse texto se refere à formulação paramétrica do MEF, adicionou-se um método abstrato na classe `NaturalCoordsShape` denominado `getMaxPolynomialOrder()` cuja finalidade é informar a máxima ordem das funções de interpolação. Tendo acesso a essa informação, objetos do tipo `ProblemDriver` conseguem calcular corretamente o número de pontos de integração que

devem ser usados durante o cálculo das matrizes de condutividade linear e não-linear. Todas as classes não abstratas no primeiro nível de herança de `NaturalCoordsShape` tiveram o método `getMaxPolynomialOrder` implementado, conforme representado na Figura 4.9.

4.3.3.6 Expansões no pacote load

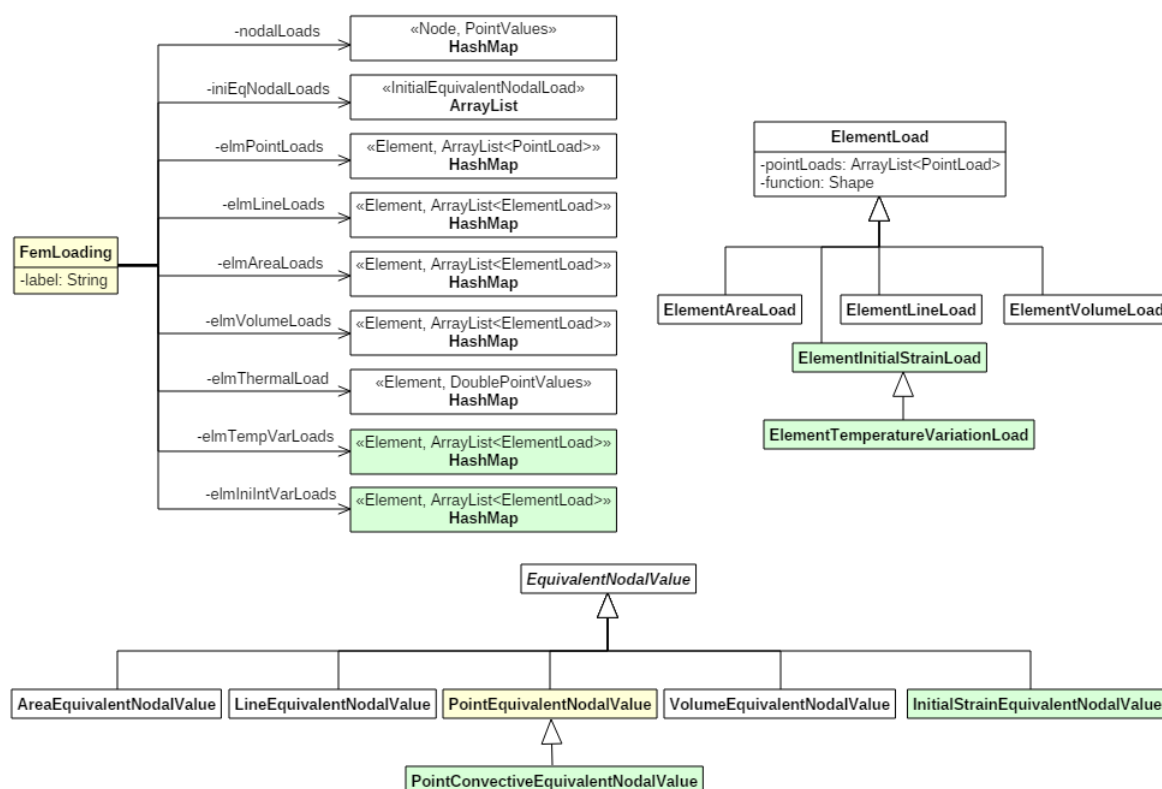


Figura 4.10: Modificações realizadas no pacote load.

Nos problemas de transferência de calor, os carregamentos degenerados em um único ponto do domínio do elemento podem ser de duas naturezas diferentes: fluxos de calor prescritos ou convecções prescritas. A diferenciação entre esses carregamentos pode ser facilmente realizada pelo número de componentes que eles contêm em sua descrição, sendo necessário somente um valor para descrever um fluxo prescrito e dois valores para prescrever uma convecção prescrita. Todavia, o modo de cálculo do carregamento nodal equivalente é diferente entre esses casos, e por isso tornou-se

necessário incluir a classe `PointConvectiveEquivalentNodalValue`, destacada na Figura 4.10.

4.4 Inclusão do projeto `conditions`

No Apêndice A foi colocada a necessidade de flexibilizar a forma de armazenar as condições de contorno essenciais na estrutura de dados, bem como a forma de aplicá-las. Também foi apontada a importância de iterar sobre conjuntos que contenham, em cada conjunto, as condições de contorno naturais e essenciais, de uma forma que ambas possam variar ao longo do tempo, além da condição inicial.

Com esse objetivo, criou-se um novo projeto no ambiente de desenvolvimento do INSANE, denominado `conditions`. Esse projeto é composto por três pacotes, sendo um para agrupar as classes que representam as condições de contorno essenciais, um para agrupar as classes para representarem as condições iniciais e um para agrupar as classes que representam os conjuntos de condições de contorno e condições iniciais. Esses pacotes serão detalhados a seguir. Ressalta-se que não foi necessário criar um pacote para representar as condições de contorno naturais, que já estão implementadas no INSANE no pacote `load` dentro do projeto `model`.

4.4.1 Pacote `statevarboundaryconditions`

Neste pacote encontram-se as classes para a representação das condições de contorno essenciais e suas combinações. Cada uma das classes é brevemente descrita a seguir:

- **`StateVariableBoundaryCondition`**: Essa classe abstrata representa de forma genérica uma condição de contorno essencial e contém apenas a assinatura de quatro métodos, responsáveis por:
 - retornar, em um vetor de valores booleanos, quais graus de liberdade têm os seus valores prescritos pela condição;

- obter a lista das entidades submetidas à essa condição de contorno essencial. No caso do MEF, essas entidades são os nós;
 - obter os valores prescritos pela condição de contorno à variável de estado;
 - combinar um objeto de condição de contorno essencial com outro por meio de uma comparação lógica do tipo “Ou”. Esse método visa a eliminação de conflitos quando duas condições de contorno essenciais são aplicadas sobre a mesma entidade, fazendo com que essas condições se complementem. O tratamento de conflitos será abordado com mais detalhes adiante, podendo também ser possível que o usuário opte por ignorar o conflito mantendo a condição que for aplicada primeiro ou por sobrescrever as informações correntes com a condição que for aplicada por último. Para a identificação da estratégia de solução de conflito, três constantes são declaradas nessa classe com os nomes *COMBINE_STRATEGY*, *IGNORE_STRATEGY* e *REPLACE_STRATEGY*.
- **NodalStateVariableBoundaryCondition**: descreve a condição de contorno essencial quando ela é aplicada sobre um nó. Essa classe é uma especialização da classe **StateVariableBoundaryCondition**.
 - **StateVariableBoundaryConditionCase**: representa um caso de uma condição de contorno essencial. O caso contém um objeto do tipo **StateVariableBoundaryCondition** ao qual é atribuída uma função escalar que pode conferir a possibilidade de variação temporal.
 - **StateVariableBoundaryConditionCombination**: essa classe contém uma lista de casos de condições de contorno essenciais, representando portanto uma combinação, além de conter métodos para manipular essa lista.

Essas classes são ilustradas nos diagramas da Figura 4.11.

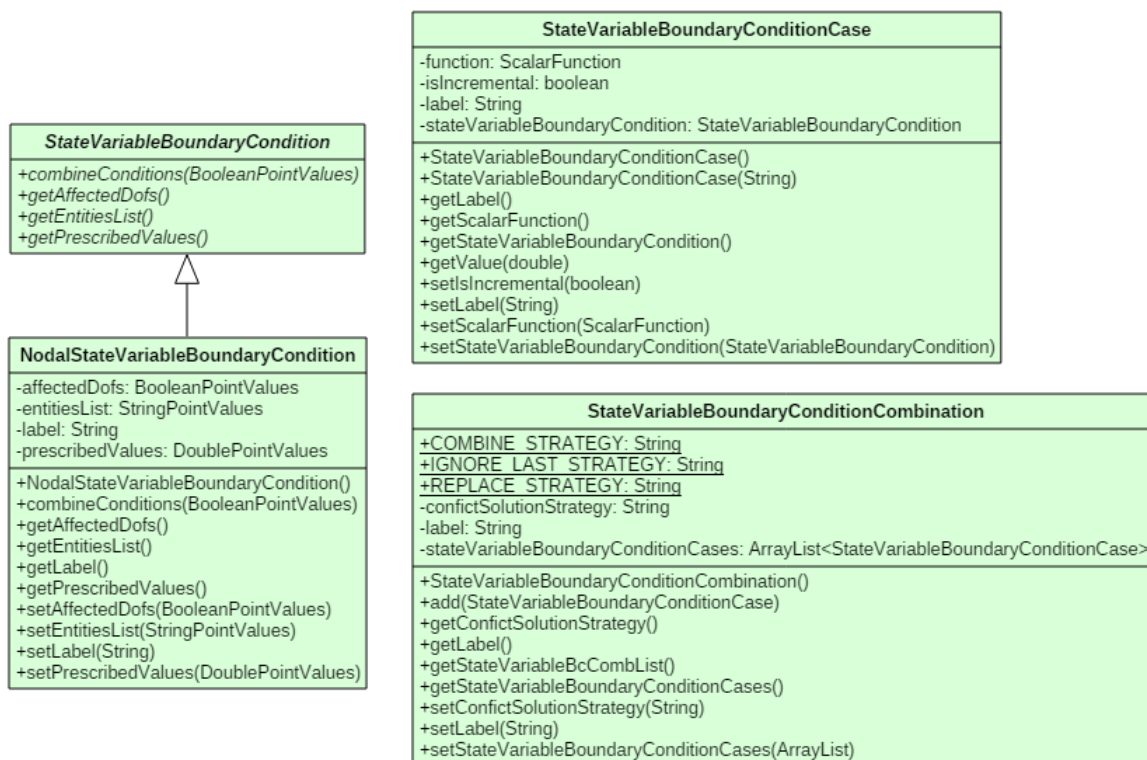


Figura 4.11: Diagrama das classes do pacote `statevarboundaryconditions`.

4.4.2 Pacote `initialconditions`

As condições iniciais são representadas pelas classes expostas na Figura 4.12, descritas a seguir:

- `StateVariableInitialCondition`: contém dois métodos abstratos para retornarem as listas das entidades submetidas à essa condição inicial e os valores que foram prescritos;
- `NodalStateVariableInitialCondition`: especializa a classe abstrata `StateVariableInitialCondition`. Nessa implementação as condições iniciais são definidas sobre os nós do modelo discreto. As entidades submetidas à condição inicial descrita por objetos dessa classe são armazenadas em uma lista cujos itens são do tipo `String` e os valores prescritos para cada um dos graus de

liberdade do nó são armazenados em uma lista de pontos flutuantes de dupla precisão. A classe também conta com métodos para acesso e manipulação desses atributos.

- **InitialConditionSet**: representa um conjunto de condições iniciais, que são armazenados em uma lista de objetos do tipo `StateVariableInitialCondition`. O agrupamento de diversos objetos do tipo `StateVariableInitialCondition` é necessário quando os valores iniciais não são uniformes ao longo da malha.

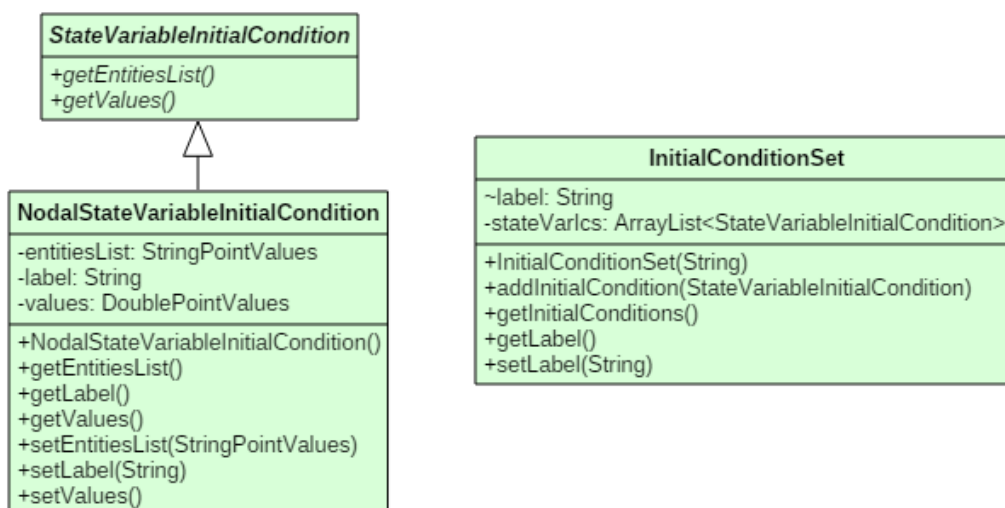


Figura 4.12: Diagrama das classes do pacote `initialconditions`.

4.4.3 Pacote `conditions`

O pacote `conditions` contém quatro classes, detalhadas a seguir:

- **BoundaryConditionsSet**: descreve uma combinação contemplando as condições de contorno natural e essencial, armazenando somente os rótulos na forma de texto que identificam essas condições de contorno.
- **CombinationSet**: contém atributos para representar um conjunto compreendido pelas condições de contorno e pela condição inicial. Tanto a referência

ao conjunto de condições de contorno como ao conjunto de condições iniciais é feito por um rótulo do tipo `String`.

- **MultiphysicsSet**: essa classe é utilizada para armazenar as referências a conjuntos de condições em ambientes multifísicos. O conjunto associado a cada física é referenciado por um rótulo no mesmo formato das demais classes desse pacote previamente descritas.
- **Conditions**: essa classe é uma estrutura de dados para conter em listas e mapas as informações relacionadas às condições iniciais, às condições de contorno essenciais e suas combinações, aos conjuntos contendo apenas as condições de contorno natural e essencial e aos conjuntos contendo a condição inicial e as condições de contorno natural e essencial. Nessa classe também são implementados métodos para acesso e modificação das informações armazenadas nessa estrutura de dados.

Essas classes podem ser visualizadas no diagrama de classes na Figura 4.13.

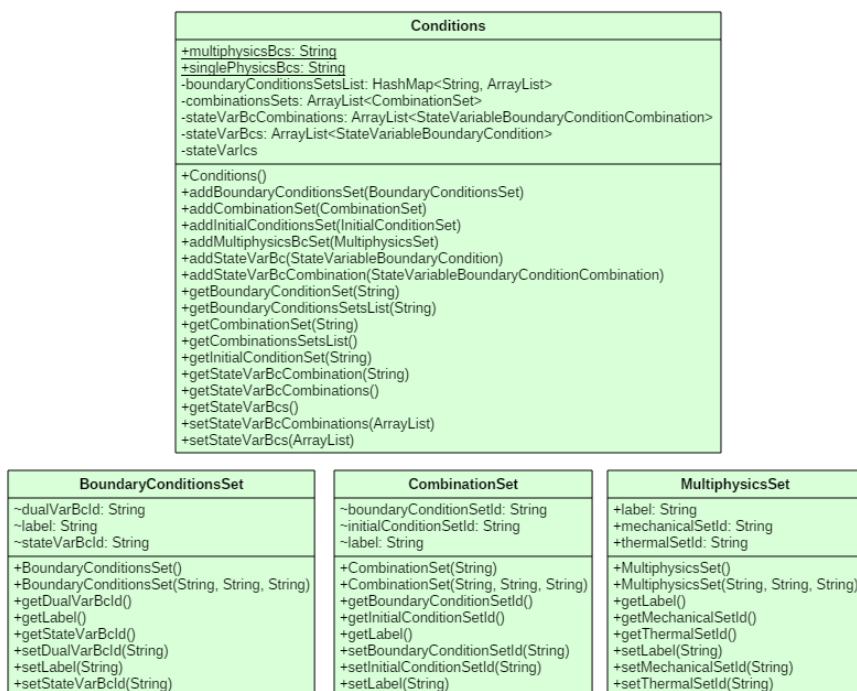


Figura 4.13: Diagrama das classes do pacote `conditions`.

4.5 Expansões para gerenciamento da solução: projeto `simulationmanager`

Como discutido anteriormente na seção A.2.2 , uma das estratégias possíveis para a solução de problemas multifísicos corresponde à solução conduzida de forma intercalada em duas ou mais classes de solução, o que demanda um maior controle do fluxo de execução. Além desse controle, é necessário automatizar a transferência de dados entre as físicas sendo solucionadas, o que justifica a implementação de um gerenciador da estrutura de dados do modelo discreto.

Em geral, os resultados obtidos em uma física em um determinado momento se comportam como dados de entrada para a continuação da solução na física seguinte. Essa entrada de dados normalmente se manifesta na forma de uma condição de contorno, que deve ser aplicada automaticamente durante essa transição entre as físicas. Essa tarefa de manipulação das condições de contorno está associada com uma configuração do ambiente de solução, e por isso foram encapsuladas em classes responsáveis por essas configurações.

Para atingir o objetivo principal de gerenciar a solução multifísica, inseriu-se uma nova entidade denominada `SimulationManager` no núcleo numérico do INSANE. Essa entidade foi projetada para trabalhar em conjunto com duas novas entidades: um `DataManager` para gerenciar a estrutura de dados e um `Setuper` para configurar o ambiente de solução. Cada uma dessas entidades será descrita nas sub-seções a seguir.

4.5.1 Pacote `simulationmanager`

Esse pacote agrupa as classes ilustradas no diagrama da Figura 4.14, responsáveis pelo gerenciamento do processo de solução, e serão descritas nas seções a seguir.

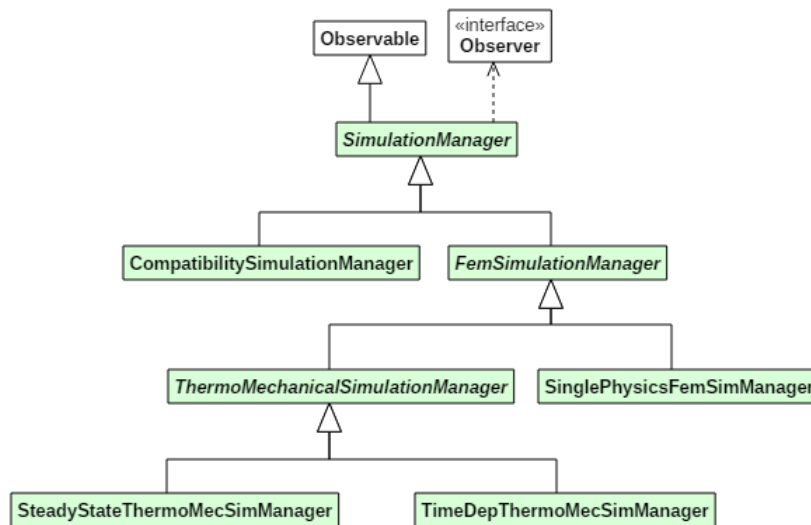


Figura 4.14: Diagrama das classes do pacote `simulationmanager`.

4.5.1.1 Classe abstrata `SimulationManager`

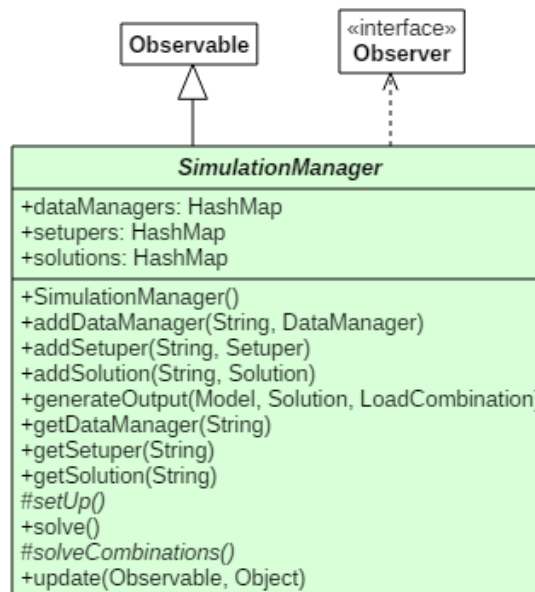


Figura 4.15: Diagrama da classes abstrata `SimulationManager`.

A classe abstrata `SimulationManager` é tanto uma observadora como uma classe observada dentro do padrão *Observer-Observable*, atuando como uma intermediadora entre as notificações provenientes dos objetos de solução e os seus originais observadores. Essa mudança de arquitetura permite que sejam tomadas ações toda

vez que os objetos de solução concluírem um passo, seja ele um passo temporal, um passo no processo iterativo ou a conclusão de uma análise linear em regime permanente.

Três atributos são declarados na superclasse `SimulationManager`, sendo eles três mapas contendo como valores objetos do tipo `Solution`, `DataManager` e `Setuper`. Além dos métodos de acesso e manipulação desses atributos, ressaltam-se os métodos descritos a seguir:

- *createInstance*, que corresponde a uma fábrica de instâncias de objetos do tipo `SimulationManager`;
- *solve*, que é o ponto de partida do processo de solução. Na classe `SimulationManager` esse método evoca dois métodos abstratos da própria classe;
- *setUp*, que é a primeira ação executada pelo método *solve*, correspondendo à configuração do ambiente de solução;
- *solveCombinations*, que é a segunda ação executada pelo método *solve()*, onde são percorridas as combinações, e para cada uma dispara-se uma solução.
- *generateOutput*, que notifica os observadores do `SimulationManager`, onde um deles é a persistência, para a geração do arquivo de saída de dados. Nesse processo, um objeto do tipo `SolutionOutputMessage`, cujo diagrama está apresentado na Figura 4.16, é criado contendo dentro de si uma referência ao modelo de elementos finitos que foi analisado, uma referência ao objeto de solução que disparou a notificação e uma referência ao objeto de combinação de carregamento ao qual a solução se refere.

O próximo nível da hierarquia de `SimulationManager` contém duas classes, uma para manter a compatibilidade com todos os trabalhos já realizados previamente no projeto INSANE, denominada `CompatibilitySimulationManager` e uma para descrever em linhas gerais o comportamento do `SimulationManager` dentro do âmbito

do Método dos Elementos Finitos, denominada `FemSimulationManager`.

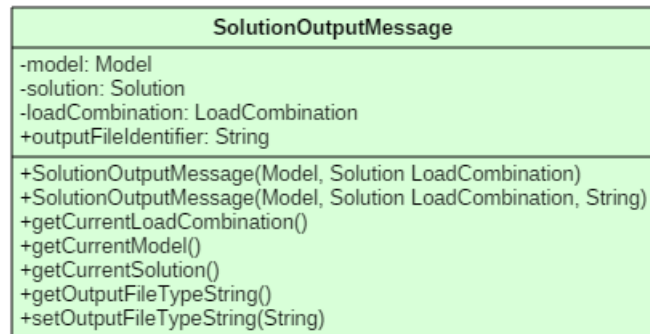


Figura 4.16: Diagrama UML da classe `SolutionOutputMessage`.

4.5.1.2 Classe `CompatibilitySimulationManager`

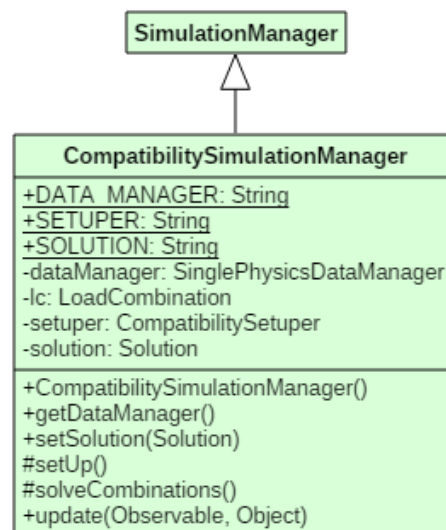


Figura 4.17: Diagrama da classes `CompatibilitySimulationManager`.

A classe `CompatibilitySimulationManager` esquematizada na Figura 4.17 opera com uma classe de solução, e tanto o seu *setuper* como o seu *data manager* são pré-definidos, dos tipos `CompatibilitySetuper` e `SinglePhysicsDataManager`, respectivamente.

O método `solveCombinations` foi implementado conforme o conceito original do

INSANE, onde a lista de combinações de carregamentos é percorrida sequencialmente, evocando a cada uma delas o método *execute* do objeto de solução. Previamente à execução em cada combinação, o *setuper* realiza uma atualização da combinação de carregamento corrente no objeto de solução.

Quanto à etapa de configuração prévia, a única atividade realizada é a criação e inicialização do objeto do tipo **Assembler** adequado, conforme o tipo do modelo e de solução. Após sua inicialização, o *assembler* recebe a referência ao objeto de modelo, e em seguida o objeto de solução recebe uma referência ao objeto *assembler* recém criado. Esse procedimento finaliza a criação da cadeia de comunicação entre solução, *assembler* e modelo.

Em relação à implementação do método *update* da classe **CompatibilitySimulationManager**, a única ação executada é a chamada ao método *generateOutput* implementado na superclasse **SimulationManager**, passando como parâmetros o modelo armazenado no objeto *datamanager*, o objeto de solução e a referência à combinação de carregamento corrente.

4.5.1.3 Classe abstrata FemSimulationManager

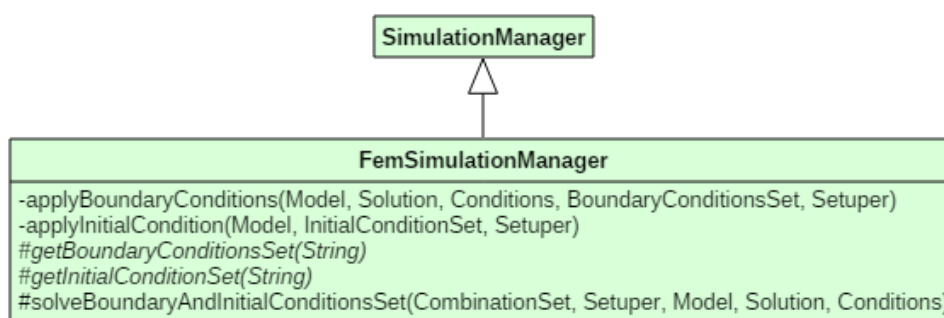


Figura 4.18: Diagrama da classe **FemSimulationManager**.

A classe **FemSimulationManager** mostrada na Figura 4.18 representa um estágio intermediário na implementação do **SimulationManager**, apenas introduzindo funcionalidades relacionadas às condições iniciais e de contorno que serão comuns a

todas as suas herdeiras que seguirem a metodologia prevista pelo MEF.

Nessa classe são introduzidos cinco novos métodos, sendo dois deles protegidos e abstratos, dois deles privados, e um protegido.

Os métodos privados *applyBoundaryConditions* e *applyInitialCondition* recebem todas as informações necessárias à aplicação dessas condições por parâmetro, tais como os objetos de modelo, solução, de condições iniciais e de contorno, o conjunto de condições iniciais corrente, o conjunto de condições de contorno corrente e uma referência ao *setuper*. Dessa forma, em suas implementações os referidos métodos apenas obtêm dos parâmetros as informações requeridas e delegam a função de aplicar as condições iniciais e de contorno ao *setuper*.

Os métodos abstratos introduzidos em `FemSimulationManager` são o método *getBoundaryConditionsSet* e o método *getInitialConditionSet*, que apenas obterão os objetos relativos aos conjuntos de condições de contorno e iniciais dados rótulos na forma de texto passados como parâmetro. Esses métodos são abstratos porque a estrutura de dados para conter esse tipo de informação ainda não está implementada nas classes abstratas `SimulationManager` e `FemSimulationManager`, entretanto esses métodos já são utilizados na implementação do método *solveBoundaryAndInitialConditionsSet*, explicado no parágrafo a seguir.

O método *solveBoundaryAndInitialConditionsSet* prepara e dispara a solução de um problema definido para um modelo discretizado segundo o MEF, dado o conjunto de condições inicial e de contorno. Este método foi implementado nesse nível hierárquico para ser utilizado pelas herdeiras de `FemSimulationManager` independentemente do problema ser multifísico ou não. O método basicamente obtém os objetos que descrevem as condições iniciais e de contorno com o auxílio dos métodos abstratos supramencionados, e em seguida realiza a aplicação dessas condições por meio dos métodos privados implementados na própria classe. Após essas etapas, a solução é disparada.

A seguir, serão detalhadas as duas classes filhas de `FemSimulationManager`, das

quais uma é aplicável a modelos que envolvem uma única física e a outra é empregada para o caso multifísico termomecânico.

4.5.1.4 Classe `SinglePhysicsFemSimManager`

Com o intuito de permitir a utilização da nova forma de combinar as condições de contorno e as condições iniciais em análises realizadas em uma única física, criou-se a classe `SinglePhysicsFemSimManager`, ilustrada na Figura 4.19, que possui como atributos uma referência a um objeto de solução, uma referência a um *setuper* do tipo `SinglePhysicsFemSetuper` e uma referência a um gerenciador da estrutura de dados do tipo `SinglePhysicsDataManager`.

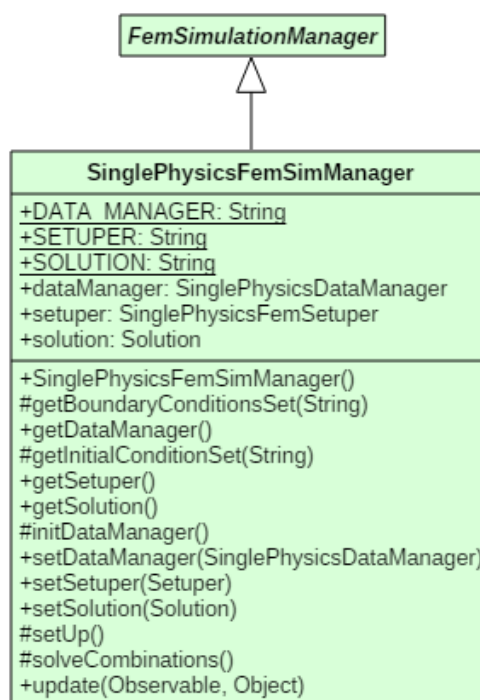


Figura 4.19: Diagrama da classe `SinglePhysicsFemSimManager`.

O método *solveCombinations* percorre os conjuntos de combinações, que contemplam tanto as condições de contorno essencial e natural e a condição inicial, e para cada um deles solicita a solução do problema evocando o método *solveBoundaryAndInitialConditionsSet*, que foi implementado na superclasse `FemSimulationManager`.

Após o término da análise da combinação corrente, o iterador dos conjuntos de combinações verifica se há uma próxima combinação a ser analisada, e caso exista, o *setuper* é acionado para redefinir todos os parâmetros do ambiente de solução para os seus valores padrão, de forma que a análise de uma combinação não interfira na análise seguinte.

Ao término da análise em cada combinação, o objeto de solução notifica os seus observadores, o que dispara o método *update* sobrecarregado em `SinglePhysicsFemSimManager`, visto que a própria classe é um dos observadores da solução. Esse registro na lista de observadores é feito no método *setSolution*. A versão implementada do método *update* solicita a criação do arquivo de saída pela chamada ao método *generateOutput*, implementado na superclasse `SimulationManager`.

A etapa de configuração prévia do ambiente de solução é descrita na implementação do método *setUp*. A primeira ação nesse método é delegar ao objeto *setuper* que defina a classe adequada de *assembler* a ser instanciada e que após a sua inicialização crie a cadeia de comunicação entre o *assembler*, o modelo e o objeto de solução. Em seguida, a referência ao objeto de *assembler* é armazenada no gerenciador da estrutura de dados.

Os métodos *getInitialConditionSet* e *getBoundaryConditionsSet*, que haviam sido declarados abstratos no nível hierárquico anterior, foram implementados na classe `SinglePhysicsFemSimManager`. Para fornecerem informações relativas às condições iniciais e de contorno, esses métodos consultam o gerenciador da estrutura de dados do tipo `SinglePhysicsDataManager`.

4.5.1.5 Classe abstrata `ThermoMechanicalSimulationManager`

As especificações gerais para o gerenciamento da solução em problemas multifísicos termomecânicos, implementadas na classe abstrata `ThermoMechanicalSimulationManager` que está apresentada na Figura 4.20, são descritas a seguir.

Essa classe possui duas referências protegidas a objetos de solução, sendo um

utilizado para a solução do PTC e um para os cálculos mecânicos contemplando os efeitos da variação de temperatura. Para tratar as notificações enviadas por esses objetos, a classe possui dois observadores, dos tipos `ThermalSolutionObserver` e `MechanicalSolutionObserver`, ambos declarados como campos privados e implementados em classes internas da classe `ThermoMechanicalSimulationManager`. A escolha pela utilização de classes internas se deve à capacidade de acesso dos observadores aos métodos e atributos da classe `ThermoMechanicalSimulationManager`, mesmo que eles sejam privados.

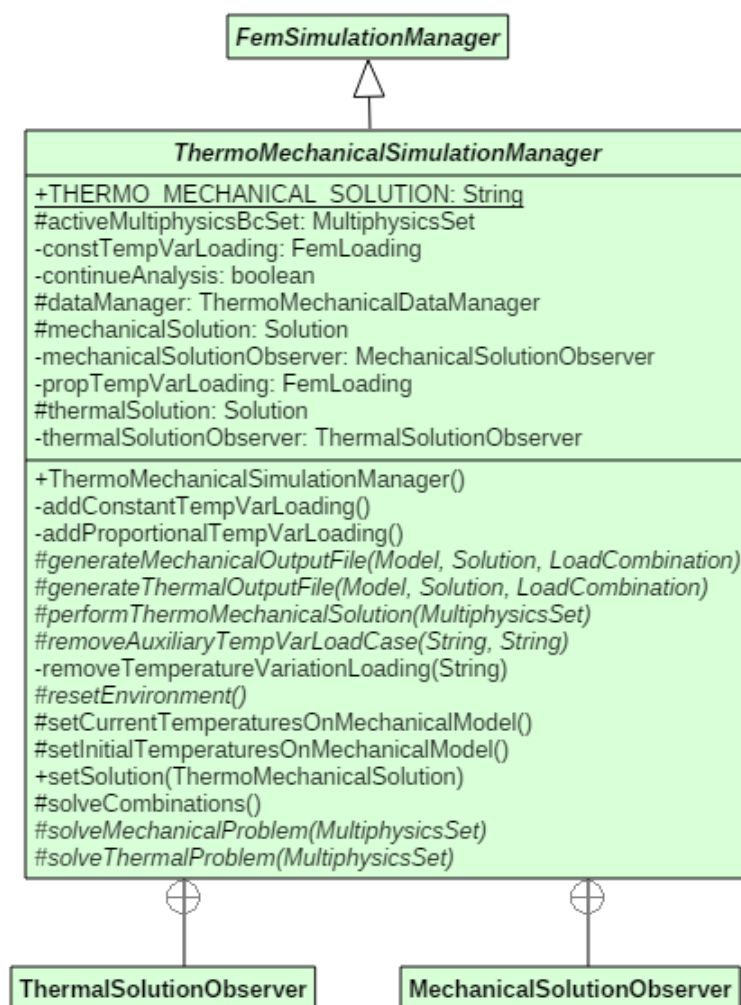


Figura 4.20: Diagrama da classe `ThermoMechanicalSimulationManager`.

Além de compartilhar as referências aos objetos de solução com as suas herdeiras, os objetos de `ThermoMechanicalSimulationManager` também compartilham

com elas uma referência a um conjunto de condições de análise multifísico, do tipo `MultiphysicsSet`, e o seu gerenciador da estrutura de dados, declarado do tipo `ThermoMechanicalDataManager`.

Dentre os novos métodos introduzidos na classe, sete deles são abstratos e estão listados abaixo:

- *generateMechanicalOutputFile*, responsável pela geração dos arquivos de saída de dados relacionados à análise mecânica;
- *generateThermalOutputFile*, idem ao anterior, porém aplicável ao PTC;
- *performThermoMechanicalSolution*, utilizado para iniciar o processo de solução do problema termomecânico;
- *removeAuxiliaryTempVarLoadCase*, empregado para remover o carregamento de variação de temperatura criado automaticamente durante a transição entre a análise térmica e a análise mecânica. Essa ação evita o acúmulo de objetos de carregamento em análises realizadas em mais de um passo de tempo;
- *resetEnvironment*, empregado para reinicializar o ambiente de solução para a sua condição padrão após a solução de uma combinação multifísica;
- *solveMechanicalProblem*, responsável por iniciar a solução do problema da mecânica dos sólidos;
- *solveThermalProblem*, idem ao anterior, porém aplicável ao PTC.

Quanto à manipulação dos carregamentos de variação de temperatura, a classe implementa três novos métodos privados para auxiliar nessa tarefa: *addConstantTempVarLoading*, *addProportionalTempVarLoading* e *removeTemperatureVariationLoading*. Esses métodos são evocados pelo observador do objeto que calcula a solução do problema de transferência de calor, como será descrito nos parágrafos a seguir.

A classe também conta com um método público utilizado para definir quais serão as classes de solução empregadas para a análise térmica e para a análise mecânica, método *setSolution*, e dois métodos auxiliares protegidos que delegam as suas funções ao gerenciador da estrutura de dados, métodos *setCurrentTemperaturesOnMechanicalModel* e *setInitialTemperaturesOnMechanicalModel*. Este último método é utilizado pelo método *solveCombinations*, que percorre as combinações de carregamento multifísicas solucionando uma a uma.

O método *solveCombinations* possui uma implementação simples, e trabalha em conjunto com as implementações dos observadores das soluções, que realizam o controle do fluxo de execução durante o processo de solução multifísico. O detalhamento da implementação do método *solveCombinations* e dos método *update* nos observadores dos objetos de solução, apresentado nos próximos parágrafos, faz referência às etapas indicadas no fluxograma apresentado na Figura 3.4.

O método *solveCombinations* percorre as combinações multifísicas e, em cada uma delas, realiza as seguintes ações:

- lê a distribuição inicial de temperaturas, correspondente ao item 1 do fluxograma, e a atribui no modelo mecânico, correspondente ao item 5 do fluxograma;
- inicia o processo de solução termomecânico com a chamada ao método *performThermoMechanicalSolution*. As implementações do método *performThermoMechanicalSolution* serão descritas nas próximas seções durante o detalhamento das classes herdeiras da classe `ThermoMechanicalSimulationManager`. Basicamente, elas realizam preparações para a solução multifísica, e em seguida disparam a solução do problema de transferência de calor, evocando o método *solveThermalProblem*. Esse último método identifica quais são as condições de contorno do problema térmico e solicita a solução do PTC a partir da chamada ao método *solveBoundaryAndInitialConditionsSet*, implementado na

superclasse `FemSimulationManager`, englobando assim o item 2 do fluxograma e todo o bloco identificado pela letra A;

- verifica se existem outras combinações multifísicas a serem solucionadas após a conclusão da análise termomecânica corrente, e caso existam, realiza uma reconfiguração do ambiente de solução para os seus valores padrão utilizando o método `resetEnvironment`.

Os observadores dos objetos de solução atuam toda vez que esses objetos emitem suas notificações, o que coincide com os momentos em que são realizadas as operações de registro em disco dos resultados calculados, itens 2, 4, 14 e 20 do fluxograma. Como nos itens 2 e 4 somente ocorrem registros das condições iniciais do problema térmico e do problema mecânico, o desvio do fluxo de execução do código nestes momentos não será discutido, visto que o comportamento do algoritmo é igual ao que ocorre nos itens 14 e 20, em cada uma das físicas. Portanto, para proporcionar uma melhor compreensão será explicado abaixo a sequência de ações realizadas pelo observador da solução do PTC assim que ele é notificado após o término do cálculo de uma nova distribuição de temperaturas, item 14 do fluxograma.

Ao receber essa notificação, o observador do objeto de solução do PTC executa as seguintes ações:

- gera o arquivo de saída contendo a distribuição de temperaturas recém calculada, item 14 do fluxograma;
- cria os carregamentos de variação de temperatura, item 15 do fluxograma;
- cria um caso de carregamento contendo os carregamentos de variação de temperatura, item 15 do fluxograma;
- adiciona o caso de carregamento criado à combinação de carregamento mecânico sendo analisada, item 17 do fluxograma;

- atualiza as temperaturas no modelo mecânico com valores recém calculados na análise térmica, item 16 do fluxograma;
- realiza a solução do problema mecânico contendo os carregamentos de variação de temperatura, bloco C do fluxograma. Neste momento o observador do problema de transferência de calor entra em estado de espera, enquanto são realizadas as etapas 18, 19 e 20 do fluxograma. Assim que a análise mecânica é concluída, deve ser realizado o registro em disco desses resultados, item 20 do fluxograma, e para tal uma notificação é disparada ao observador da solução mecânica. Basicamente, esse observador gera o arquivo de saída de dados da análise mecânica e retorna o fluxo de execução ao observador do problema de condução de calor.
- concluída a solução do problema mecânico, o observador do PTC remove os carregamentos auxiliares de variação de temperatura recentemente criados e em seguida retorna o fluxo de execução do código a quem o notificou, nesse caso o objeto de solução do problema térmico. Esse retorno ocorre na transição entre o bloco C e o passo 21 do fluxograma, momento no qual o objeto de solução térmica irá verificar se ainda existem outros passos a serem calculados ou se a análise como um todo já foi concluída.

O procedimento acima descrito se referiu ao problema termomecânico em regime transiente, entretanto ressalta-se que o mesmo algoritmo é aplicável ao caso em regime estacionário, que corresponderá a uma única execução do bloco identificado pela letra A no fluxograma da Figura 3.4.

A classe abstrata `ThermoMechanicalSimulationManager` possui duas herdeiras, sendo uma para problemas em regime permanente e a outra para problemas em regime transiente. Essas classes serão descritas nas duas seções seguintes.

4.5.1.6 Classe `SteadyStateThermoMecSimManager`

A classe `SteadyStateThermoMecSimManager` mostrada na Figura 4.21 é utilizada na solução dos problemas termomecânicos em regime permanente, sejam eles fisicamente lineares ou não-lineares, podendo a não-linearidade se manifestar em somente uma das físicas ou em ambas. Como em problemas em regime permanente a solução em cada física é realizada uma única vez devido à ausência dos passos de tempo típicos da análise transiente, pode-se utilizar o mesmo modelo discreto nas duas análises.

Nessa abordagem, após o devido registro em disco da distribuição de temperaturas em regime permanente e criação dos carregamentos de variação de temperatura, as variáveis armazenadas nos nós são sobrescritas com os valores padrão para que esses espaços de memória sejam usados para receberem os valores calculadas na análise mecânica.

Entretanto, devem ser realizadas adequações durante a transição entre a análise térmica e a mecânica devido à presença de entidades no modelo que têm um comportamento diferente de acordo com a física sendo analisada, a saber: `ProblemDriver`, `AnalysisModel`, `ConstitutiveModel`, `Material`.

Com o intuito de atribuir um comportamento multifísico às entidades supramencionadas, foram introduzidas as novas classes `ThermoMechanicalPd`, `ThermoMechanicalAm`, `ThermoMechanicalCm` e `ThermoMechanicalMaterial`. Para encapsular em um único local os objetos de solução a serem utilizadas no caso termomecânico, incluiu-se também uma nova classe denominada `ThermoMechanicalSolution`. Essas classes serão explicadas na seção 4.6. Em linhas gerais, elas possuem dentro de si dois objetos, um que descreve o comportamento térmico e um que descreve o comportamento mecânico, e também contêm um atributo que determina qual a física ativa. Além disso, todos os métodos das respectivas superclasse são sobrecarregados para realizarem um redirecionamento das chamadas aos seus objetos interiores conforme a física ativa.

Para tratar as particularidades do compartilhamento da estrutura de dados do modelo para a solução em ambas as físicas, a classe `SteadyStateThermoMecSimManager` possui uma referência a um objeto do tipo `SharedModelSteadyStateThermoMecSetuper`, que realiza as configurações necessárias antes do início da solução em cada física. A classe também inicializa, por meio de uma chamada ao método `initDataManager` no seu construtor, um objeto do tipo `SharedModelThermoMecDataManager` para o gerenciamento dessa estrutura de dados. Para determinar se a física ativa é a térmica ou a mecânica, a classe utiliza um campo privado.

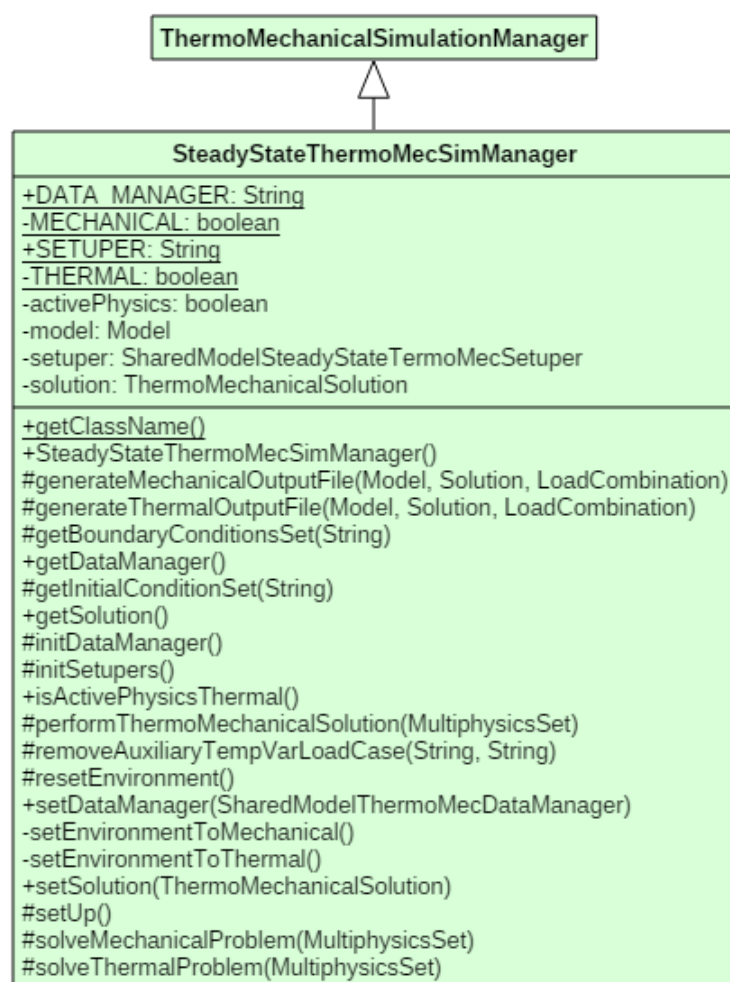


Figura 4.21: Diagrama da classe `SteadyStateThermoMecSimManager`.

Descrevendo o fluxo de execução com o auxílio do fluxograma da Figura 3.4, a solução do problema termomecânico se inicia com o cálculo da distribuição de

temperaturas para o regime permanente, correspondente às etapas 5 a 14, prossegue com a criação dos carregamentos de variação de temperatura nas etapas 15 a 17, e conclui com a solução do problema mecânico nas etapas 18 a 20. Portanto, nesse caso não há continuidade de cálculos na análise térmica após a conclusão da análise mecânica, como mencionado anteriormente.

Como discutido nas seções anteriores, na nova abordagem o processo de solução se inicia no método *solve* da classe `SimulationManager`, que evoca os métodos *setUp* e *solveCombinations*. O método *solveCombinations* foi implementado na superclasse `ThermoMechanicalSimulationManager`, e o método *setUp* foi implementado na classe descrita nessa seção, conforme apresentado a seguir.

A primeira etapa a ser realizada na preparação do ambiente de solução se refere à inicialização do objeto *assembler*, que pode ser compartilhado pelos objetos de solução, e à atribuição desse *assembler* nesses objetos de solução. Para tal, o método *setUp* evoca o método *setUpAssembler* do *setuper* da classe, fornecendo como parâmetros o modelo de elementos finitos compartilhado e o objeto de solução multifísico, que contém dentro de si dois objetos do tipo `Solution`, um para o PTC e um para o PTM. Criado e atribuído às soluções, a referência ao objeto de assembler é atribuída no *data manager* para concluir a ponte de comunicação entre soluções, *assembler* e modelo.

Após a etapa inicial de configurações gerais, o método *solveCombinations* da superclasse `ThermoMechanicalSimulationManager` é chamado e as combinações multifísicas são percorridas. A cada uma delas, a solução termomecânica é iniciada no método *performThermoMechanicalSolution*. Na presente classe, esse método possui apenas uma linha, na qual é realizada uma chamada ao método *solveThermalProblem*.

Para resolver a parcela do problema multifísico relacionada ao PTC, o método

solveThermalProblem, implementado na `SteadyStateThermoMecSimManager`, obtém as condições inicial e de contorno, e em seguida evoca o método *solveBoundaryAndInitialConditionsSet* implementado na classe `FemSimulationManager` fornecendo todos os parâmetros necessários.

A solução térmica será então calculada pelo objeto de solução escolhido para essa física, e ao término dos cálculos esse objeto irá disparar uma notificação aos seus observadores, sendo um deles um objeto do tipo `ThermalSolutionObserver` implementado como uma classe interna da classe `ThermoMechanicalSimulationManager`. Ao receber a notificação, as seguintes ações são realizadas pelo observador:

- Geração do arquivo de saída de dados do PTC, realizada pelo método *generateThermalOutputFile*, implementado na classe `SteadyStateThermoMecSimManager`;
- Criação automática dos carregamentos de variação de temperatura através dos métodos *addConstantTempVarLoading* e *addProportionalTempVarLoading*, implementados na classe `ThermoMechanicalSimulationManager`. Esses métodos delegam a criação desses carregamentos ao objeto de gerenciamento da estrutura de dados, do tipo `SharedModelThermoMecDataManager`;
- Atualização das temperaturas nos nós do modelo a ser utilizado para a solução do problema mecânico, feita pelo método *setCurrentTemperaturesOnMechanicalModel*, implementado na classe `ThermoMechanicalSimulationManager`. A responsabilidade dessa atualização também é delegada ao *data manager*;
- Solução do problema mecânico a partir da chamada ao método *solveMechanicalProblem*, implementado na classe `SteadyStateThermoMecSimManager`. Esse método executa as seguintes atividades:
 - Adequação do ambiente de solução, feita pelo *setuper* e disparada pelo método *setEnvironmentToMechanical*, implementado na própria classe

`SteadyStateThermoMecSimManager`;

- Obtenção das condições inicial e de contorno do problema mecânico;
 - Reatribuição dos valores padrão às variáveis pertinentes à análise, realizada pelo método *resetToDefaultValues* do *setuper*. Essa etapa tem o objetivo de evitar que os valores previamente calculados na análise térmica interfiram de forma indevida nos resultados da análise mecânica;
 - Aplicação das condições inicial e de contorno, e solução do problema mecânico contemplando os efeitos da variação de temperatura por meio da chamada ao método *solveBoundaryAndInitialConditionsSet* descrito na classe `FemSimulationManager`. Após a conclusão da análise mecânica, o observador do tipo `MechanicalSolutionObserver` é notificado, o que desencadeia a geração do arquivo de saída de dados da análise mecânica pela chamada ao método *generateMechanicalOutputFile*, implementado na classe `SteadyStateThermoMecSimManager`. Após o registro em disco, o fluxo de execução retorna ao observador da análise térmica para a realização do último passo, descrito a seguir;
- Remoção dos casos de carregamento de variação de temperatura da combinação de carregamento mecânica, bem como remoção dos carregamentos de variação de temperaturas da lista de carregamentos do modelo mecânico. Essa etapa é realizada pelos métodos *removeAuxiliaryTempVarLoadCase* e *removeTemperatureVariationLoading*.

No caso de existir mais de uma combinação multifísica a ser calculada, o método *resetEnvironment* é chamado entre as análises para que o ambiente seja ajustado para a solução de problemas de transferência de calor. Para tal, o método *setEnvironmentToThermal* é evocado, seguido do método *resetToDefaultValues* do *setuper* para a atribuição dos valores padrão às variáveis armazenadas nos nós.

4.5.1.7 Classe `TimeDepThermoMecSimManager`

Para gerenciar a solução de problemas termoemcânicos em regime quasi-estático, implementou-se a classe `TimeDepThermoMecSimManager`, que possui uma organização similar à da classe `SteadyStateThermoMecSimManager` descrita na seção anterior. Devido a essas semelhanças, somente as particularidades relativas aos problemas com dependência temporal serão discutidas nessa seção.

Ao optar por conduzir a solução em ambas as físicas de forma paralelizada, ou seja, sem a necessidade de concluir toda a análise térmica em todos os passos de tempo para então iniciar o cálculo mecânico, torna-se interessante a utilização simultânea de duas instâncias do modelo discreto visto que nesse caso não é adequado sobrescrever as informações durante as trocas de físicas porquê ao início de cada passo de tempo deve-se dar continuidade aos cálculos realizados no passo de tempo anterior. Uma alternativa que poderia ser empregada para utilizar um único modelo é a alocação de espaços auxiliares de memória para armazenar as variáveis históricas necessárias a cada física para, no passo de tempo corrente, dar continuidade aos cálculos realizados no passo de tempo anterior. Todavia, nessa abordagem a economia de memória não é significativa perante à utilização de dois modelos e se observa uma redução de desempenho causada pela repetida reconfiguração do ambiente de solução a cada transição de física, em cada passo de tempo.

A fim de eliminar a necessidade de criar um mecanismo de mapeamento geométrico nesse primeiro momento, decidiu-se por criar os modelos discretos a serem utilizados em cada física a partir da geração de cópias de um modelo de referência, garantindo assim a mesma posição e identificação dos nós e dos elementos finitos. Esse procedimento de clonagem do modelo é facilitado com a utilização das novas entidades introduzidas no núcleo numérico que possuem comportamento multifísico, a saber `ThermoMechanicalPd`, `ThermoMechanicalAm`, `ThermoMechanicalCm`, `ThermoMechanicalMaterial` e `ThermoMechanicalSolution`. Assim, essas entidades multifísicas são atribuídas inicialmente ao modelo de referência a ser utilizado

por todas as físicas para a geração das respectivas cópias, e após a clonagem passam a ter somente o comportamento da física à que lhes for atribuída. Tal configuração inicial é realizada pelo método *setUp*.

Outra estratégia importante empregada durante a concepção do gerenciador de solução do problema termomecânico em regime quasi-estático se refere à manipulação do carregamento mecânico a ser aplicado em cada instante de tempo. Quando o comportamento mecânico do problema é fisicamente não-linear e as ações mecânicas externas variam ao longo do tempo, torna-se necessário separar a cada passo temporal a parcela do carregamento mecânico previamente equilibrada no início do passo e a parcela a ser aplicada de forma incremental ao longo da solução não-linear no passo de tempo corrente.

Matematicamente, pode-se dizer que a parcela de carregamento a ser equilibrada no passo de tempo pode ser calculada a partir da diferença entre o valor do carregamento no instante em que se deseja obter o campo de deslocamentos, ou seja, no final do passo de tempo, e o valor que esse mesmo carregamento assume no início do passo de tempo.

No INSANE, a variação temporal de um carregamento é definida a partir da atribuição de uma função escalar ao caso de carregamento que o contém. Dentro do contexto das análises que possuem dependência temporal, as funções escalares recebem como parâmetro um instante de tempo e retornam um valor escalar, que é o fator de multiplicação das cargas de referência. Assim, para calcular a diferença entre o carregamento no final e no início do passo de tempo pode-se substituir o caso de carregamento original pela subtração de dois casos de carregamento auxiliares, onde um corresponde aos valores no final do passo de tempo e o outro ao carregamento no início do passo de tempo. Como no INSANE os casos de carregamento são somados automaticamente, tal subtração é realizada ao se multiplicar o carregamento no início do passo de tempo por -1 .

Decidiu-se durante a fase de implementação substituir cada um dos casos de

carregamento presentes na combinação de carregamento corrente aplicável à parcela mecânica por três casos de carregamentos auxiliares conforme descrito abaixo:

- PRIMEIRO CASO DE CARREGAMENTO AUXILIAR (CCA1): é uma cópia do caso de carregamento de referência contendo exatamente a mesma função escalar, que retorna o valor do fator de multiplicação das cargas no instante de tempo em que se deseja obter o campo de deslocamentos, ou seja, no final do passo de tempo corrente;
- SEGUNDO CASO DE CARREGAMENTO AUXILIAR (CCA2): é uma cópia do carregamento de referência contendo uma função escalar modificada que, ao ser acionada para informar o fator de multiplicação do carregamento no instante de tempo referente ao final do passo de tempo, retorna esse valor em um instante de tempo defasado, cuja defasagem é um atraso igual à duração do passo de tempo. Após calcular o valor do fator de multiplicação do carregamento no instante de tempo deslocado, essa função ainda multiplica o resultado por um fator de escala igual a -1 . Assim, a soma $CCA1 + CCA2$ igual ao carregamento a ser equilibrado no passo de tempo;
- TERCEIRO CASO DE CARREGAMENTO AUXILIAR (CCA3): é uma cópia do carregamento de referência contendo uma função escalar modificada semelhante à do segundo caso de carregamento auxiliar, porém sem multiplicar o resultado de sua avaliação por -1 . Esse caso de carregamento auxiliar retorna a parcela do carregamento previamente equilibrada no início do passo de tempo, identificada no INSANE como parcela constante do carregamento na análise fisicamente não-linear. Reforça-se que a soma $CCA1 + CCA2 + CCA3$ leva ao carregamento total no final do passo de tempo.

Essa função escalar modificada é representada por uma nova classe introduzida no núcleo numérico, denominada `ScaledAndShiftedFunction`, que estende a classe `ScalarFunction`. A classe `ScaledAndShiftedFunction` possui como atributos uma

referência a uma função escalar convencional a ser operada, o fator de multiplicação e a defasagem temporal. Nessa classe, o método *getValue* foi sobrecarregado para retornar o valor da função armazenada na classe em um instante de tempo deslocado, multiplicado pelo referido fator de escala.

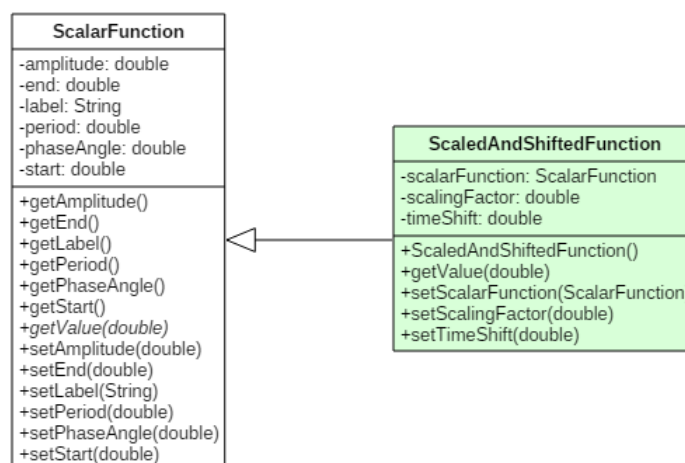


Figura 4.22: Diagrama da classe *ScaledAndShiftedFunction*.

A fim de manter a combinação de carregamento de referência inalterada, optou-se por introduzir os casos de carregamento resultantes do desmembramento descrito acima em uma combinação de carregamento auxiliar. Esse processo de criação da combinação de carregamento auxiliar e dos casos de carregamento auxiliares a partir dos casos de referência é realizado no método *mountAuxiliaryLoadCombination*, evocado pelo método *performThermoMechanicalSolution*, ambos implementados na classe *TimeDepThermoMecSimManager*, cujo diagrama é mostrado na Figura 4.23.

Após obter a combinação de carregamento da análise mecânica e usá-la como referência para a criação da combinação de carregamento auxiliar contendo o desmembramento entre as cargas constantes e as cargas proporcionais, o método *performThermoMechanicalSolution* dispara o início do processo de solução termomecânico pela chamada ao método *solveThermalProblem*. Ao término da solução dos passos de tempo do PTC, o observador do problema térmico é acionado para criar os carregamentos de variação de temperatura e direcionar o fluxo de execução para

a solução do problema mecânico. Ao ser concluída, a solução mecânica notificará o seu observador, retornando o fluxo de execução à solução térmica para dar continuidade ao cálculo da distribuição de temperaturas no instante de tempo seguinte. Essa estratégia de utilização dos observadores para controle do fluxo de execução foi explicada em detalhes na seção anterior.

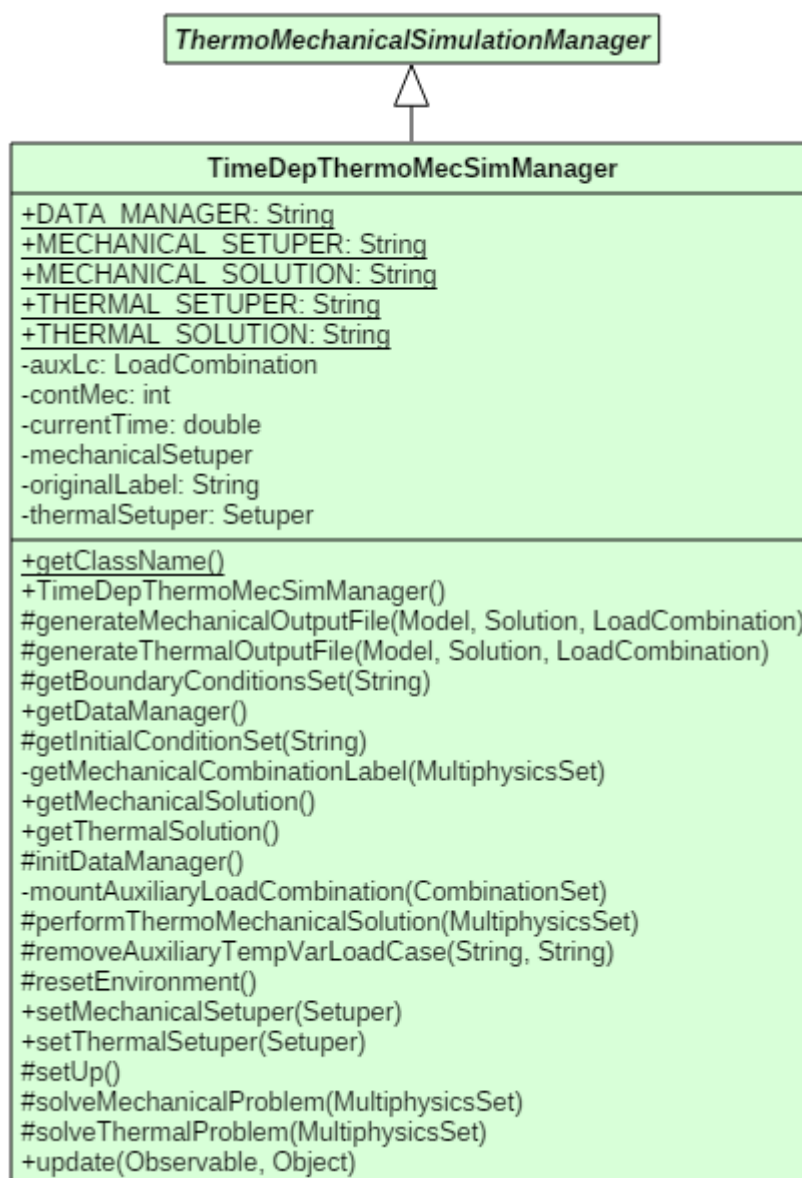


Figura 4.23: Diagrama da classe `TimeDepThermoMecSimManager`.

O método *solveThermalProblem* implementado na classe `TimeDepThermoMecSimManager` é semelhante ao método implementado na classe `SteadyStateThermoMecSimManager`, discutida anteriormente, sendo a única diferença os parâmetros passados ao método *solveBoundaryAndInitialConditionsSet*, que nesse caso contém referências a objetos específicos para o caso em que o modelo de elementos finitos não é compartilhado por ambas as físicas.

O método *solveMechanicalProblem*, por sua vez, possui uma implementação específica na classe `TimeDepThermoMecSimManager`. O método inicia com a atualização do instante de tempo corrente no objeto de solução da parcela mecânica e em seguida a inicia. Observa-se nesse método que as condições de contorno e o preparo para a solução mecânica ocorre somente se for o primeiro passo de tempo. Nos demais, apenas o método *execute* do objeto de solução é evocado para dar continuidade à análise concluída no passo de tempo anterior.

Por fim, a última particularidade dessa classe a ser discutida se refere à geração dos arquivos de saída de dados, realizada pelos métodos *generateThermalOutputFile* e *generateMechanicalOutputFile*. No regime transiente o nome dos arquivos passa a conter tanto o instante de tempo como o passo iterativo ao qual se referem.

4.5.2 Pacote setuper

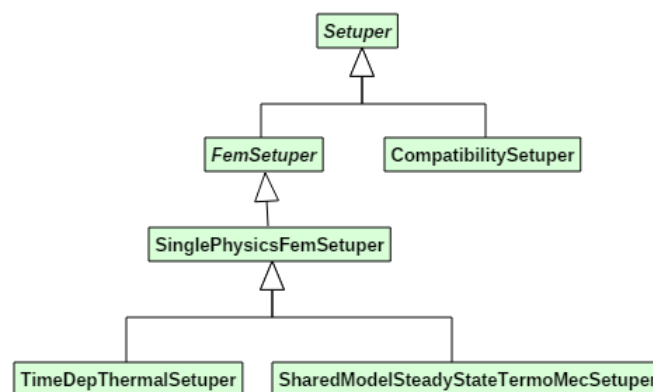


Figura 4.24: Diagrama de classes do pacote `setuper`.

As classes para configuração do ambiente de solução são agrupadas no pacote `setuper`, cuja organização hierárquica é ilustrada na Figura 4.24. Nas próximas sub-seções serão discutidas as particularidades de cada uma dessas classes.

4.5.2.1 Classe abstrata `Setuper`

Essa classe especifica um conjunto mínimo de funcionalidades comuns às classes responsáveis pela configuração do ambiente de solução. Dentre os métodos mostrados na Figura 4.25, destacam-se:

- A inicialização do objeto *assembler* e a criação da ponte de comunicação entre os objetos de modelo e de solução, desempenhada pelos métodos *initAssembler* e *setUpAssembler*;
- A aplicação de condições de contorno no modelo discreto, feita pelo método *applyBoundaryConditions*;
- A aplicação da condição inicial, realizada pelo método *applyInitialConditions*;
- A reinicialização do ambiente de solução aos seus valores padrão, conduzida pelo método *resetToDefaultValues*. Como discutido no Apêndice A, essa reinicialização tem a função de prevenir que os resultados calculados em uma combinação prévia interfiram nos cálculos da combinação corrente.

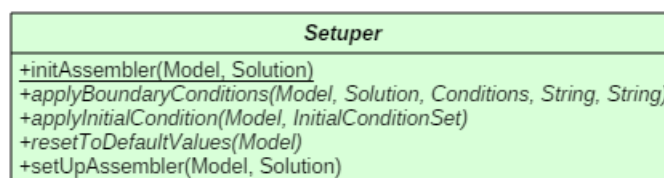


Figura 4.25: Diagrama da classe abstrata `Setuper`.

Nesse nível hierárquico, somente os métodos *initAssembler* e *setUpAssembler* possuem corpo, sendo os demais abstratos. Como o método *initAssembler* se comporta como uma fábrica de instâncias, optou-se por torná-lo um método de classe, ou

seja, um método estático que não requer a existência de um objeto do tipo `Setuper` em tempo de execução para ser evocado.

4.5.2.2 Classe `CompatibilitySetuper`

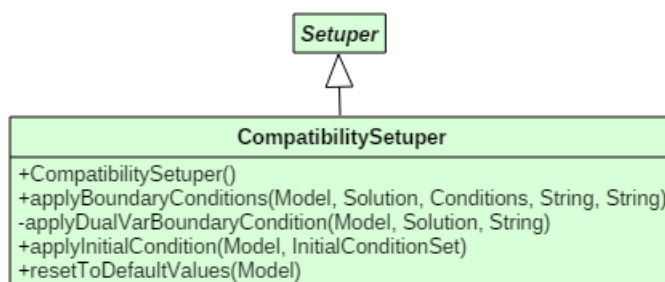


Figura 4.26: Diagrama da classe `CompatibilitySetuper`.

O *setuper* de compatibilidade mostrado na Figura 4.26 é empregado quando as condições inicial e de contorno são descritas sem a utilização das classes especializadas para essa finalidade, como era feito previamente a esse trabalho. Nessa forma convencional, as condições iniciais e a condição de contorno essencial são definidas diretamente sobre os nós e não podem se modificar nem mesmo quando há a troca da combinação de carregamentos em uma análise contendo mais de uma combinação de carregamentos a ser calculada.

Assim, o método responsável por aplicar as condições de contorno essencial e natural, método *applyBoundaryConditions*, apenas aplica a condição de contorno natural, já que as informações relacionadas à condição inicial e à condição de contorno essencial já foram definidas durante a etapa de pré-processamento para geração do modelo discreto.

Para aplicar a condição de contorno natural, o método privado *applyDualVarBoundaryCondition* obtém a combinação de carregamentos corrente a partir de uma consulta ao modelo e em seguida atribui essa combinação ao objeto de solução.

Por não terem aplicabilidade nesse cenário em que somente as condições de contorno naturais se modificam ao longo das combinações a serem analisadas, os

métodos *resetToDefaultValues* e *applyInitialCondition*, que eram abstratos no nível hierárquico anterior, são implementados na classe `CompatibilitySetuper` sem executarem nenhuma ação.

4.5.2.3 Classe abstrata `FemSetuper`

A classe abstrata `FemSetuper` inclui de uma forma geral os recursos para a configuração do ambiente de solução em problemas modelados segundo o Método dos Elementos Finitos e que utilizam a nova abordagem proposta nessa dissertação para a manipulação das condições inicial e de contorno.

O método *applyBoundaryConditions*, que era abstrato no nível hierárquico da superclasse `Setuper`, é implementado evocando sequencialmente dois métodos protegidos introduzidos na classe `FemSetuper`: *applyStateVarBoundaryCondition* e *applyDualVarBoundaryCondition*. Esses métodos são responsáveis, respectivamente, pela aplicação das condições de contorno essenciais e naturais.

Assim como na classe `CompatibilitySetuper`, o método *applyDualVarBoundaryCondition* obtém a combinação de carregamento corrente a partir de uma consulta ao modelo e em seguida atribui essa combinação no objeto de solução.

O método *applyStateVarBoundaryCondition* possui duas assinaturas, sendo a diferença entre elas um parâmetro adicional especificando o instante de tempo para o qual as condições de contorno essenciais devem ser avaliadas antes de serem aplicadas ao modelo discreto. Na implementação atual, a assinatura que não contém o parâmetro de tempo evoca o mesmo método atribuindo o valor zero para o parâmetro de tempo.

Na versão completa do método *applyStateVarBoundaryCondition*, os casos presentes na combinação de condições essenciais são percorridos e, a cada um deles, os valores armazenados no caso são aplicados aos nós contidos na lista armazenada pelo objeto `StateVariableBoundaryCondition`. A cada nó a receber a condição

essencial é feita uma verificação de conflito com eventuais valores atribuídos previamente nesse nó, sendo o conflito tratado por uma das estratégias informadas pelo usuário durante a criação do caso de condição essencial.

O outro método abstrato da classe `Setuper` que é implementado na classe `FemSetuper` é o método *applyInitialCondition*. Esse método percorre a lista de condições iniciais e, a cada uma, atribui os respectivos valores nos nós contidos na lista armazenada pelo objeto `NodalStateVariableInitialCondition`.

A classe `FemSetuper` não pode ser utilizada por não implementar o método abstrato *resetToDefaultValues*, introduzido na superclasse `Setuper`. Essa implementação é realizada nas classes herdeiras de `FemSetuper`.

Também pode ser encontrada em `FemSetuper`, conforme Figura 4.27, uma fábrica de instâncias para objetos aplicáveis à configuração do ambiente de solução em problemas descritos conforme o MEF. As três herdeiras de `FemSetuper` serão descritas a seguir.

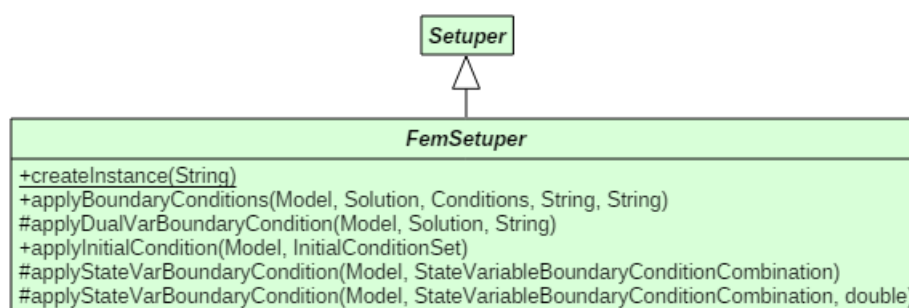


Figura 4.27: Diagrama da classe `FemSetuper`.

4.5.2.4 Classe `SinglePhysicsFemSetuper`

Para realizar as configurações necessárias em análises que contemplem somente uma física e que descrevam as condições inicial e de contorno conforme a metodologia proposta nesse trabalho, utiliza-se a classe `SinglePhysicsFemSetuper` exposta na Figura 4.28.

Essa classe estende `FemSetuper` introduzindo a implementação do método *resetToDefaultValues*, que basicamente percorre a lista de nós do modelo evocando neles o método *assignInitialState*. Em seguida, os valores armazenados pelas chaves *RESTRAINTS* e *PRE_STATE_VARIABLE* do mapa de valores do nó são alterados para os seus valores padrão: falso para os valores da chave *RESTRAINTS* e zero para os valores da chave *PRE_STATE_VARIABLE*.

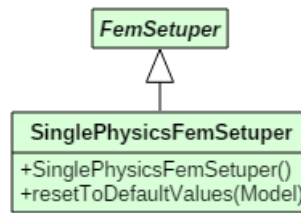


Figura 4.28: Diagrama da classe `SinglePhysicsFemSetuper`.

4.5.2.5 Classe `TimeDepThermalSetuper`

A classe `TimeDepThermalSetuper`, mostrada na Figura 4.29, estende a classe `SinglePhysicsFemSetuper`, sendo a única diferença, em relação à sua classe mãe, a sobrecarga do método *applyBoundaryConditions*. Na versão especializada, após delegar a aplicação das condições de contorno ao mesmo método de sua superclasse, o objeto contendo a combinação das condições de contorno essenciais é atribuído no objeto de solução específico para a solução de problemas de transferência de calor no regime transiente, `TimeDependentHeatTransfer`.

Esse passo é necessário visto que durante o cálculo da solução no referido tipo de problema é necessário utilizar os valores da distribuição de temperaturas nos instantes de tempo t e $t + 1$, $\mathbf{T}_{c(t)}$ e $\mathbf{T}_{c(t+1)}$ respectivamente, como descrito na Equação 3.36. Para obter esses valores, o objeto de solução realiza uma requisição ao *assembler* por meio do método *getXp* informando a combinação de condições essenciais e o instante de tempo. Devido à sua especificidade, essa classe deve ser utilizada somente no contexto da solução de problemas de transferência de calor em regime transiente.

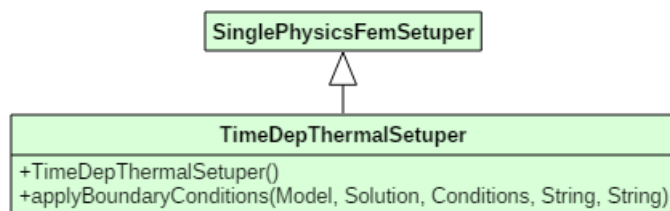


Figura 4.29: Diagrama da classe TimeDepThermalSetuper.

4.5.2.6 Classe SharedModelSteadyStateTermoMecSetuper

A outra herdeira de SinglePhysicsFemSetuper ilustrada na Figura 4.30 é a classe SharedModelSteadyStateTermoMecSetuper, responsável pela configuração do ambiente de solução nas análises multifísicas termomecânicas em regime permanente. Nessas análises, a estrutura de dados do modelo discreto é compartilhada entre as físicas envolvidas, como descrito na seção 4.5.1.6.

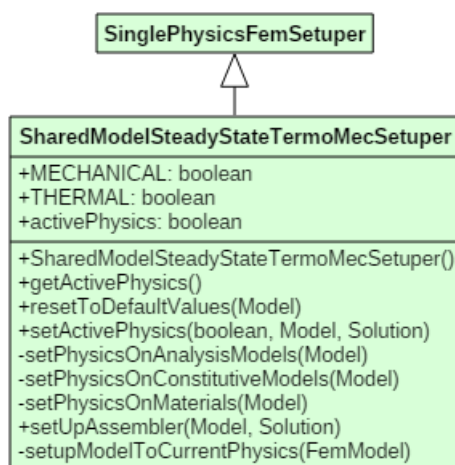


Figura 4.30: Diagrama da classe SharedModelSteadyStateTermoMecSetuper.

A classe possui um campo privado para indicar a física que se encontra ativa e duas constantes booleanas, uma para cada física. Dentre os métodos herdados, os métodos *resetToDefaultValues* e *setUpAssembler* foram sobrecarregados. Nessa classe também são introduzidos dois métodos públicos, *getActivePhysics* e *setActivePhysics*, e quatro métodos privados. Esses métodos serão descritos logo após o detalhamento dos métodos sobrecarregados.

Na sua versão especializada, o método sobrecarregado *setUpAssembler* recebe como parâmetros uma referência ao modelo e uma ao objeto de solução, que deve ser do tipo `ThermoMechanicalSolution`. Desse objeto de solução multifísico são extraídas as referências aos objetos responsáveis pela solução térmica e pela solução mecânica.

Após a obtenção das referências aos objetos de solução para cada física, o método define o estado das entidades multifísicas dos tipos `ThermoMechanicalPd`, `ThermoMechanicalAm`, `ThermoMechanicalSolution`, `ThermoMechanicalCm` e `ThermoMechanicalMaterial` para apresentarem comportamento mecânico.

Em seguida, um objeto do tipo `Assembler` é criado e recebe uma referência ao modelo. Feito isso, a referência ao *assembler* recebe uma referência ao objeto de solução da parcela mecânica e o método *init* do *assembler* é evocado. Nesse momento, o modelo de elementos finitos passa por sua inicialização, que contém etapas que irão apresentar um comportamento específico conforme a física ativa. Essa inicialização completa garante a atribuição de todas as referências nas entidades presentes no núcleo numérico, evitando assim a ocorrência de ponteiros nulos.

Concluída a inicialização para a parcela mecânica, o *setuper* realiza esse mesmo procedimento para a parcela térmica. Para tal, a física ativa é modificada no *setuper* para a transferência de calor, assim como em todas as entidades que possuem comportamento multifísico. Após esse ajuste, a referência ao *assembler* é atribuída no objeto de solução do PTC, o que também desencadeia o processo de inicialização do modelo, utilizando alguns objetos diferentes já que o comportamento está ajustado para a parcela térmica. Reforça-se que o mesmo objeto de *assembler* é compartilhado por ambas as físicas.

Quanto à sobrecarga do método *resetToDefaultValues*, ele difere da versão implementada na classe `SinglePhysicsFemSetuper` por apresentar um comportamento diferente conforme a física ativa. Isso se deve ao número de graus de liberdade por

nó, que em geral não é o mesmo na análise térmica e na mecânica. Todavia, a implementação segue os mesmos preceitos da sua versão para análises em uma única física, que corresponde a chamar o método *assignInitialState* em cada nó e atribuir os valores padrão em cada uma das chaves presentes no mapa dos valores nodais.

Os métodos públicos introduzidos na classe `SharedModelSteadyStateThermoMecSetuper` são o método *getActivePhysics*, que apenas informa qual é a física ativa, e o método *setActivePhysics*, que desempenha as etapas listadas a seguir. Primeiramente, o método atualiza a variável booleana que representa a física ativa segundo a informação recebida por parâmetro. Na sequência, as entidades multifísicas são ajustadas para apresentarem o comportamento desejado por quem evocou o método. Esse ajuste é feito com o auxílio dos métodos privados *setPhysicsOnAnalysisModels*, *setPhysicsOnConstitutiveModels*, *setPhysicsOnMaterials* e *setupModelToCurrentPhysics*.

Dentre os quatro métodos privados citados acima, os três primeiros são similares: percorrem a lista das respectivas entidades armazenadas no modelo e a cada vez que identificam uma entidade multifísica, ajustam a mesma para a física informada no parâmetro. Já o método *setupModelToCurrentPhysics* realiza as seguintes etapas:

- Inicializa as chaves dos nós com os valores informados pelo objeto *problem driver* por meio do método *initKeys* do objeto de modelo;
- Em cada um dos nós do modelo, reinicia o vetor de rótulos dos graus de liberdade por meio da chamada ao método *clearDofLabels* do nó;
- Em cada elemento finito do modelo, obtém os rótulos dos nós conforme o modelo e análise do mesmo e os atribui aos nós do elemento evocando o método *setDofLabels*. Em seguida, preenche os índices de cada um dos graus de liberdade dos nós dos elementos por meio do método *setElementGlobalIndexes*;

4.5.3 Pacote datamanager

As classes para o gerenciamento da estrutura de dados envolvidas na solução de um problema discreto são agrupadas no pacote `datamanager`, cuja organização hierárquica é ilustrada na Figura 4.31. Nas próximas sub-seções serão discutidas as particularidades de cada uma das classes contidas nesse pacote.

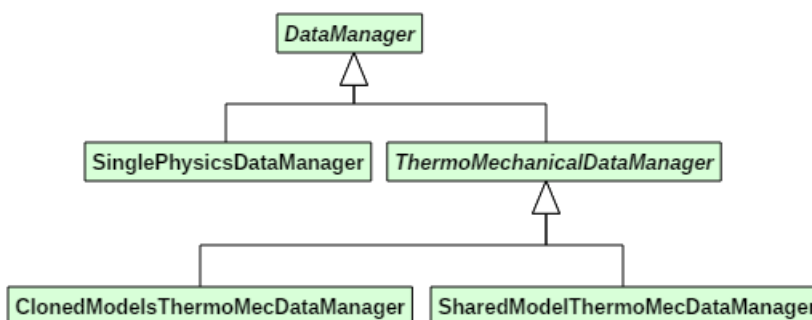


Figura 4.31: Diagrama das classes no pacote `datamanager`.

4.5.3.1 Classe abstrata DataManager

Essa classe, mostrada na Figura 4.32, contém três mapas, onde um armazena entidades do tipo `Assembler`, outro armazena entidades do tipo `Model` e o outro armazena entidades do tipo `Conditions`. A classe também possui métodos para a manipulação e obtenção dos itens desses mapas. Chaves do tipo `String` são utilizadas para identificar os itens nesses mapas.

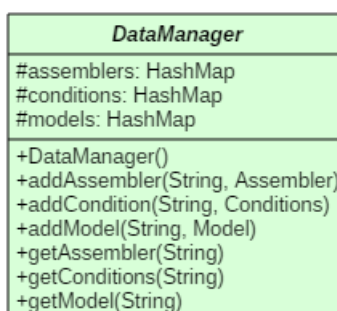


Figura 4.32: Diagrama da classe abstrata `DataManager`.

A classe foi especializada nas suas quatro herdeiras, descritas nas próximas seções.

4.5.3.2 Classe SinglePhysicsDataManager

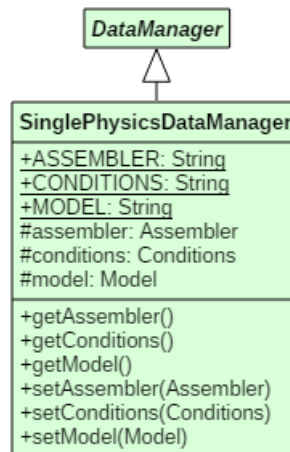


Figura 4.33: Diagrama da classe `SinglePhysicsDataManager`.

Como a classe `SinglePhysicsDataManager` ilustrada na Figura 4.33 é utilizada em análises envolvendo somente uma física nas quais as condições iniciais e de contorno são especificadas utilizando as classes do projeto `conditions`, pode-se trabalhar com apenas uma referência a cada um dos tipos `Assembler`, `Model` e `Conditions`. Essas referências são armazenadas pelos três campos da classe `SinglePhysicsDataManager`.

Os métodos implementados nesse *data manager* são públicos e são utilizados para atribuir objetos aos campos da classe, conforme os métodos *setAssembler*, *setConditions* e *setModel*, e para consultar os valores neles armazenados, segundo os métodos *getAssembler*, *getConditions* e *getModel*. Ressalta-se que os métodos utilizados na atribuição de objetos aos campos da classe também inserem esses objetos na lista correspondente da superclasse `DataManager`.

4.5.3.3 Classe abstrata ThermoMechanicalDataManager

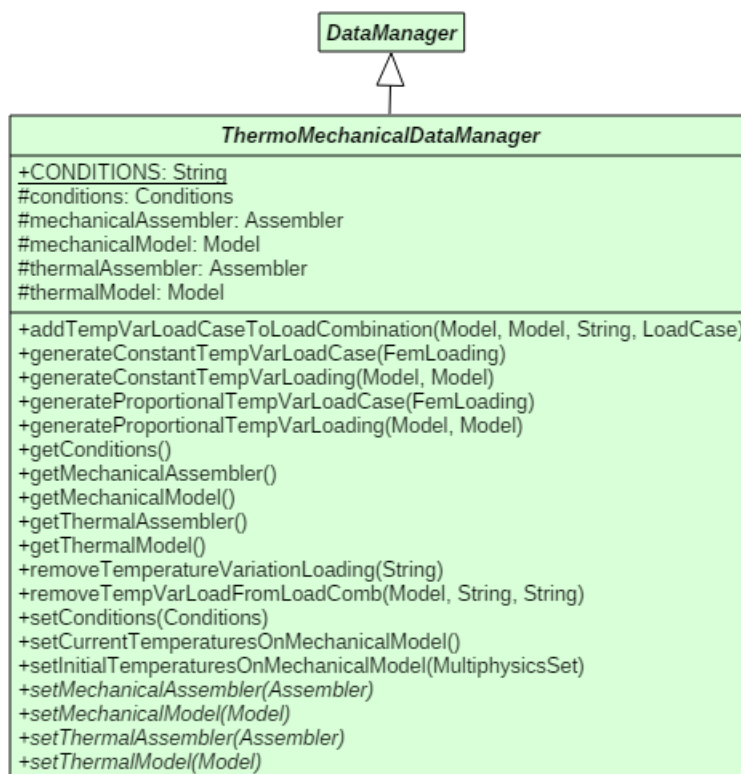


Figura 4.34: Diagrama da classe ThermoMechanicalDataManager.

Conforme discutido anteriormente, no caso termomecânico implementado nesse trabalho é necessário transferir informações da análise térmica para a mecânica. Essas informações se referem tanto aos carregamentos de variação de temperatura como à atualização da distribuição de temperaturas nos nós da discretização utilizada no cálculo mecânico. Assim, o *data manager* termomecânico exposto na Figura 4.34 se encarrega tanto de armazenar as estruturas de dados para ambas as físicas como de gerenciar essa transmissão de informação.

Para armazenar as referências às entidades da estrutura de dados de cada física, a classe contém seis campos: dois modelos, dois objetos de condições e dois *assemblers*. Também são introduzidos métodos para a consulta e para a manipulação das referências armazenadas nesses campos.

Quanto aos carregamentos de variação de temperatura, foram introduzidos métodos associados à parcela desses carregamentos que deve ser aplicada de forma incremental, denominada parcela proporcional, e métodos associados à parcela dos carregamentos previamente equilibrados, denominada parcela constante. Essa separação se torna necessária nos casos em que a solução mecânica possui não-linearidade. Em cada um desses dois grupos de métodos, implementou-se um método para criar os objetos de carga, e um método para introduzir essas cargas em um caso de carregamento. Os métodos criados para essa finalidade estão listados a seguir:

- *generateConstantTempVarLoading*;
- *generateConstantTempVarLoadCase*;
- *generateProportionalTempVarLoading*;
- *generateProportionalTempVarLoadCase*.

Para incluir um caso de carregamento de variação de temperatura na combinação de carregamento corrente da análise mecânica, utiliza-se o método *addTempVarLoadCaseToLoadCombination*, e para removê-lo da combinação de carregamento, utiliza-se o método *removeTempVarLoadFromLoadComb*. Quanto à remoção dos carregamentos, também é implementado um método para remover carregamentos de variação de temperatura da lista de carregamentos do modelo utilizado na análise mecânica, método *removeTemperatureVariationLoading*.

Em relação à atualização da distribuição de temperaturas no modelo utilizado durante a análise mecânica, foram introduzidos os métodos *setInitialTemperaturesOnMechanicalModel* e *setCurrentTemperaturesOnMechanicalModel*, onde o primeiro é responsável por atribuir em cada nó a temperatura no instante inicial e o segundo, por atribuir o valor da temperatura no instante corrente.

A classe `ThermoMechanicalDataManager` não pode ser instanciada porque os métodos para a atribuição dos modelos e dos *assemblers* foram declarados como

abstratos. Não é possível implementar esses métodos nesse momento visto que o modelo pode tanto ser compartilhado entre as físicas, como ocorre na implementação para o regime permanente, como pode ser utilizado um modelo discreto para cada física, situação esta necessária no regime transiente. Devido a isso, a classe `ThermoMechanicalDataManager` possui duas herdeiras, uma para o caso que trabalha sobre um modelo compartilhado, e uma para o caso que opera sobre dois modelos.

4.5.3.4 Classe `SharedModelThermoMecDataManager`

Nas análises em regime permanente, o modelo discreto é compartilhado pelas físicas envolvidas. Devido à essa simplificação, somente uma referência ao modelo compartilhado e uma referência a um objeto de *assembler* são necessárias. Essas referências são introduzidas como campos privados da classe `SharedModelThermoMecDataManager`.

O construtor dessa classe primeiramente evoca os construtores das suas superclasses e em seguida iguala as referências aos modelos térmico e mecânico, herdadas da classe abstrata `ThermoMechanicalDataManager`, à referência do modelo compartilhado, introduzida na classe `SharedModelThermoMecDataManager` apresentada na Figura 4.35. O mesmo procedimento é realizado para as referências aos objetos de *assembler* térmico e mecânico, herdadas do nível hierárquico anterior.

Além dessa particularidade na inicialização, os métodos para obtenção e para a definição dos objetos *model* e *assembler* foram implementados retornando a referência ao respectivo objeto compartilhado, métodos com o prefixo *get*, e definindo essa referência no campo privado da classe que o contém, métodos com o prefixo *set*. Ao definir uma referência a um objeto do tipo *model* ou *assembler*, esses métodos também adicionam o objeto recebido como parâmetro nos mapas declarados na superclasse `DataManager`.

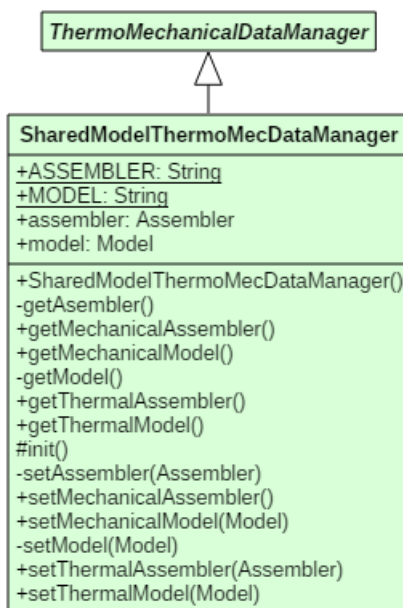


Figura 4.35: Diagrama da classe `SharedModelThermoMecDataManager`.

4.5.3.5 Classe `ClonedModelsThermoMecDataManager`

A classe `ClonedModelsThermoMecDataManager`, cujos métodos e atributos são mostrados na Figura 4.36, foi concebida para trabalhar com modelos independentes nas análises térmica e mecânica. Na implementação realizada nesse trabalho, os modelos para cada uma das físicas são criados a partir de um mesmo modelo de base que possui comportamento multifísico. Essa estratégia torna a etapa de pré-processamento mais simples e assegura que a discretização dos modelos seja idêntica, o que dispensa a implementação de algoritmos de mapeamento geométrico.

Essa geração do modelo multifísico se inicia na persistência, especificamente no método `fillSimulationManagerFromFile` da classe `PersistenceAsXML`. Nesse método, ao se identificar que a análise deve possuir dois modelos, sendo um deles uma cópia do outro, um *data manager* do tipo `ClonedModelsThermoMecDataManager` é criado, recebendo em seguida uma referência ao objeto de modelo armazenado pela persistência. Após essa etapa, uma nova referência ao modelo é criada utilizando um novo método introduzido na persistência, denominado `cloneModel`. Esse método irá

ler novamente o arquivo XML, criando uma nova instância de objeto do tipo `FemModel` e a preenchendo conforme as informações especificadas no arquivo. Concluída a geração do novo modelo, essa referência é atribuída no *data manager* ao campo referente ao modelo mecânico. Nesse ponto, ambos os campos da classe `ClonedModelsThermoMecDataManager` possuem uma referência a uma instância diferente do modelo, porém o conteúdo desses modelos é idêntico.

Para concluir a geração dos modelos independentes para cada uma das físicas, o método *removeDuality* é evocado pelo objeto *setuper* antes do início da análise. Esse método primeiramente remove o comportamento termomecânico no modelo térmico por meio do método *removeDualityOnThermalModel*, e em seguida realiza esse mesmo procedimento no modelo mecânico, evocando o método *removeDualityOnMechanicalModel*.

Cada um desses dois métodos percorre todas as entidades do modelo removendo o comportamento multifísico e atribuindo nelas a entidade relacionada à física desejada, extraída do objeto multifísico correspondente. Para tal, os métodos *removeDualityOnThermalModel* e *removeDualityOnMechanicalModel* utilizam os métodos privados relacionados abaixo, implementados na classe `ClonedModelsThermoMecDataManager`:

- *removeDualityOnElements*;
- *removeDualityOnDegenerations*;
- *removeDualityOnMaterials*;
- *removeDualityOnProblemDriver*;
- *removeDualityOnGlobalAnalysisModels*;
- *removeDualityOnAnalysisModels*;
- *removeDualityOnConstitutiveModels*;

Por fim, os métodos para atribuição e consulta aos objetos *model* e *assembler*, que foram declarados como abstratos no nível hierárquico anterior, são implementados e trabalham com as referências independentes dessas entidades para a parcela térmica e para a parcela mecânica.

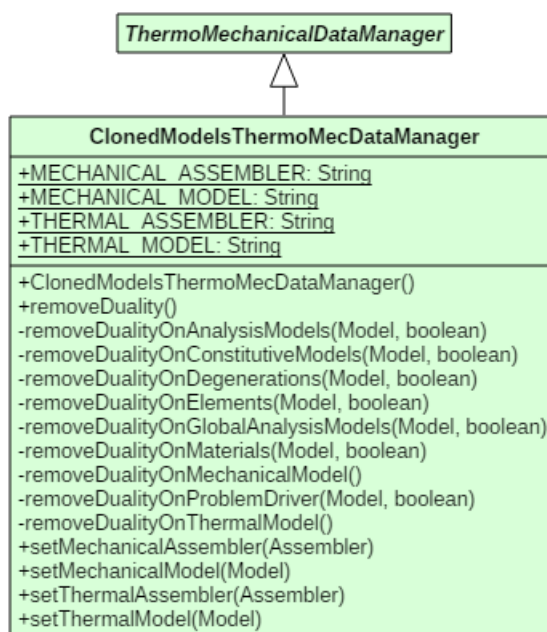


Figura 4.36: Diagrama da classe `ClonedModelsThermoMecDataManager`.

4.6 Entidades com comportamento multifísico

Essa seção descreve a técnica empregada para atribuir um comportamento dependente da física ativa em algumas entidades do núcleo numérico. Como descrito na seção anterior, esse comportamento multifísico é útil tanto quando se deseja utilizar o mesmo modelo discreto para as soluções nas físicas envolvidas, como para criar um único modelo de referência para a geração de cópias para cada uma das físicas, que são especializadas em seguida para que apresentem somente o comportamento desejado.

Para atingir esse objetivo, primeiramente cria-se uma versão especializada da

superclasse que se deseja agregar um comportamento multifísico utilizando o mecanismo de herança. Logo, para cada uma das entidades dependentes de física presentes no núcleo numérico do INSANE foram criadas as seguintes classes especializadas:

- `AnalysisModel` especializada em `ThermoMechanicalAm`;
- `ConstitutiveModel` especializada em `ThermoMechanicalCm`;
- `Material` especializada em `ThermoMechanicalMaterial`;
- `ProblemDriver` especializada em `ThermoMechanicalPd`;
- `Solution` especializada em `ThermoMechanicalSolution`.

Em cada uma das classes especializadas para o caso termomecânico existem três campos: dois objetos internos do mesmo tipo da superclasse e uma variável booleana para indicar qual é a física ativa. Um desses objetos pode ser uma instância de uma classe que possua comportamento aplicável à transferência de calor e o outro uma instância de uma classe aplicável à mecânica dos sólidos. Essa estratégia confere uma maior flexibilidade na hora de determinar como será o comportamento em cada física, além de reutilizar as classes previamente implementadas no núcleo numérico.

Além da introdução dos objetos para cada uma das físicas, criou-se um mecanismo de redirecionamento de métodos. Esse mecanismo se baseia na sobrecarga de todos os métodos da superclasse, onde nas versões especializadas desses métodos é feita uma verificação da física ativa, e conforme o resultado dessa verificação evoca-se o método com o mesmo nome no objeto interno correspondente.

A figura a seguir ilustra essa técnica para a entidade `AnalysisModel`. Raciocínio análogo é aplicável às demais entidades multifísicas mencionadas acima.

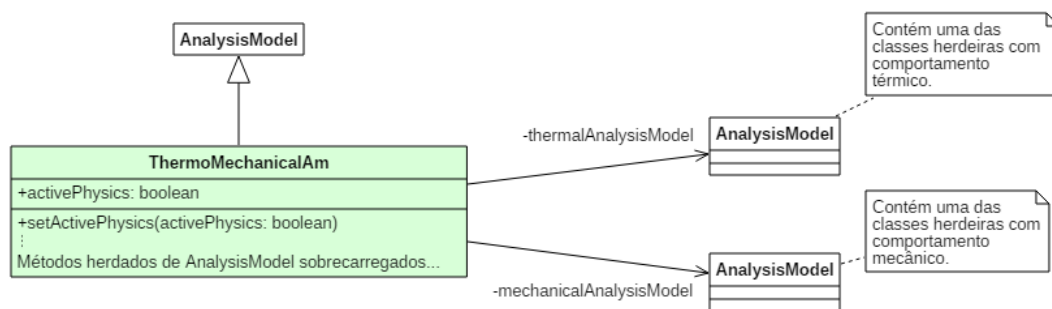


Figura 4.37: Especialização da classe `AnalysisModel` agregando o comportamento termomecânico.

4.7 Modificações para a solução do PTM

Na primeira seção deste capítulo foram discutidas as novas classes e as modificações realizadas nas classes existentes para o tratamento de problemas de transferência de calor que podem apresentar tanto não-linearidade física como dependência temporal. Em seguida, foram discutidas as novas classes introduzidas para proporcionar um maior controle do fluxo de execução durante o processo de solução multifísico, e as novas classes responsáveis por conferirem um comportamento multifísico a algumas entidades do núcleo numérico. Nesta seção, serão apresentadas as demais modificações realizadas, específicas para a solução do problema termomecânico nas condições abordadas nessa dissertação.

4.7.1 Modificações no projeto `solution`

O projeto `solution` é composto por diversos pacotes, e somente os pacotes que sofreram alterações são abordados nesse texto. Esses pacotes que receberam a introdução de novas classes são os pacotes `solution` e `step`.

4.7.1.1 Modificações no pacote `solution`

No pacote `solution`, foram introduzidas duas classes para lidarem com problemas termomecânicos no regime quasi-estático: `LinearQuasiStatic` e `QuasiStaticEquilibriumPath`. Essas classes estão ilustradas na Figura 4.3, apresentada previamente.

A classe `LinearQuasiStatic` é semelhante à classe `SteadyState`, sendo a principal diferença entre elas a inclusão de um parâmetro no método `execute`, que passa a calcular a solução para o instante de tempo informado nesse parâmetro. Como discutido anteriormente, no regime quasi-estático os termos inerciais da equação do movimento são desprezados, porém a variação lenta das ações externas é considerada nos cálculos. Nesse cenário, uma solução para um determinado instante de tempo é equivalente a uma solução em regime permanente considerando a variação das ações externas entre o instante inicial, ou de referência, e o instante desejado.

Analogamente, a classe `QuasiStaticEquilibriumPath` é semelhante à classe `StaticEquilibriumPath`, sendo inclusive uma herdeira dessa classe. Suas principais diferenças se referem ao cálculo dos carregamentos proporcional e constante, que na versão especializada leva em conta o instante de tempo e a combinação de carregamento a partir da qual esses carregamentos serão avaliados, e à atribuição de uma referência à combinação de carregamento corrente no objeto `step`.

Tendo em vista que a obtenção de parcelas proporcionais de carregamentos de variação de temperatura não pode ser realizada apenas por uma multiplicação dos valores armazenados no objeto de carregamento por um escalar, torna-se necessário recalcular o valor da parcela proporcional ao longo do processo incremental-iterativo de minimização do resíduo nas análises em regime quasi-estático que possuem não-linearidade física na parcela mecânica. Portanto, optou-se por fornecer ao objeto `step` uma referência à combinação de carregamento corrente, para que ele realize a avaliação da parcela de cargas associada à variação de temperatura da forma correta.

Observa-se também que a implementação realizada na classe `QuasiStaticEquilibriumPath` flexibiliza a análise para que ela possa ser realizada para mais de uma combinação de carregamento. A implementação da classe `StaticEquilibriumPath` presume que somente uma combinação de carregamento será analisada e que ela se encontra na primeira posição da lista de combinações de carregamento armazenada pelo modelo.

Quanto às modificações nas classes existentes, elas serão abordadas a seguir para cada uma das respectivas classes.

Na classe abstrata `Solution`, foram incluídos três métodos abstratos: *getOutputFileIdentifier*, *getOutputFileSeqNumber* e *setAssemblerWithoutInit*. Os dois primeiros são empregados na geração do nome dos arquivos de saída, e o terceiro é utilizado quando se deseja atribuir o objeto *assembler* no objeto de solução sem realizar uma inicialização completa do modelo discreto. Naturalmente, esses métodos foram implementados nas herdeiras de `Solution`.

Devido à necessidade de obter os carregamentos tradicionais da mecânica dos sólidos separadamente dos carregamentos de variação de temperatura conforme os motivos previamente mencionados, decidiu-se introduzir novos métodos na classe `Assembler`, que serão descritos em detalhes nas seções seguintes. Dentre eles, citam-se os métodos *getHp* e *getHu*, que retornam os carregamentos nodais equivalentes devido à variação de temperatura nos graus de liberdade cujo valor da variável de estado foi prescrito e nos graus de liberdade em que essa variável é desconhecida. Para contemplarem os efeitos da variação de temperatura, foram realizadas modificações nas classes `SteadyState` e `GlobalLocal`, incluindo as chamadas aos referidos métodos durante a etapa de montagem do vetor das ações externas e durante o cálculo dos carregamentos nodais equivalentes.

4.7.1.2 Modificações no pacote `step`

O pacote `step`, que agrupa as classes responsáveis por realizarem os cálculos iterativos durante as soluções não-lineares, recebeu a nova classe `ThermoMecSNR` para abordar problemas termomecânicos segundo o método incremental-iterativo de Newton-Raphson. Essa classe pode ser visualizada no diagrama da Figura 4.2. Ressalta-se que os problemas termomecânicos introduzem uma complexidade adicional no que se refere à avaliação da convergência, visto que os carregamentos de variação de temperatura são auto-equilibrados.

Além dos critérios complementares de verificação da convergência, a classe `ThermoMecSNR` realiza o cálculo das parcelas constante e proporcional do carregamento de variação de temperatura separadamente dos demais carregamentos da mecânica dos sólidos, bem como conduz o processo iterativo considerando essa segregação durante o cálculo dos incrementos no campo de deslocamentos e durante o cálculo do resíduo.

Quanto às modificações nas classes previamente implementadas, foram introduzidos os métodos `resetStep`, `getLoadCombination` e `setLoadCombination` na interface `Step`, onde o primeiro é responsável por reconfigurar o objeto `step` para a sua condição padrão, passo necessário quando se deseja calcular mais de uma combinação, e os dois últimos são necessários para fornecer a referência à combinação de carregamento corrente a objetos do tipo `ThermoMecSNR`, como previamente discutido na sub-seção anterior. Esses métodos também tiveram que ser introduzidos nas classes que implementam a interface `Step`.

Na classe `Convergence`, introduziu-se uma verificação adicional no método `checkConvergence`, que considera que houve convergência quando a norma de ambos os vetores recebidos por parâmetro é inferior ao menor valor de ponto flutuante representável. Esse caso pode ocorrer durante a verificação do critério de convergência em deslocamentos em casos particulares de problemas termo-mecânicos.

Um exemplo que ilustra a necessidade de introduzir essa verificação adicional é

o caso de uma barra cujas extremidades estejam impedidas de deslocar e submetida exclusivamente a uma variação uniforme de temperatura. Independentemente do incremento de cargas associadas à variação de temperatura, o incremento no campo de deslocamentos será nulo, assim como o campo de deslocamentos totais. O modo anterior de verificação não indicaria a ocorrência de convergência, visto que ao dividir a norma desses campos uma pela outra o resultado seria indeterminado devido à divisão por zero, fazendo com que a comparação entre o resultado e a tolerância sempre retorne o valor *false*.

Apesar da classe `ClosestPointProjection` não se encontrar no pacote `step`, decidiu-se mencionar uma modificação realizada nessa classe devido à proximidade dela em relação aos assuntos abordados nesse momento. Essa modificação se refere ao aumento em uma ordem de grandeza no número máximo de iterações realizadas por esse algoritmo, empregado em problemas envolvendo plasticidade. Tal alteração se demonstrou necessária devido à ocorrência de uma redução na taxa de convergência desse algoritmo quando há a presença de deformações de variação de temperatura, conseqüentemente exigindo um número maior de iterações durante o cálculo das tensões.

4.7.2 Modificações no projeto assembler

O projeto `assembler` recebeu modificações na interface `Assembler` e na sua classe herdeira indireta `FemAssembler` no que se refere tanto à montagem dos vetores de carregamento nodal equivalente devido à variação de temperatura, como no que se refere à montagem dos vetores de carregamento proporcional e de carregamento constante, aplicáveis à análise fisicamente não-linear na presença de variações de temperatura.

Como será melhor discutido posteriormente nesse capítulo, foi realizada uma alteração na forma de representar os carregamentos de variação de temperatura. Na

implementação atual, esses carregamentos devem atuar em todo o domínio do elemento finito, diferentemente da implementação dos carregamentos da mecânica dos sólidos, que podem atuar em porções do elemento. Essa simplificação acerca dos carregamentos de variação de temperatura permite o domínio espacial do carregamento a partir da geometria do elemento, o que dispensa a existência de uma classe diferente de carregamento de variação de temperatura para cada nível de degeneração espacial, a saber: carregamentos tridimensionais, bidimensionais e unidimensionais. Com essa redução do número de classes de carregamento associadas à variação de temperatura, tornou-se necessário alterar os métodos existentes *addLoading*, *initLoading* e *setLoading*.

Devido à separação dos carregamentos de variação de temperatura dos demais carregamentos da mecânica dos sólidos, tornou-se necessário implementar os métodos *getH*, *getHp* e *getHu* para a obtenção dos vetores de carregamento nodal equivalente associados à expansão ou contração do corpo.

Para a solução dos problemas termomecânicos fisicamente não-lineares em regime quasi-estático, é necessário informar tanto o parâmetro de tempo como a combinação de carregamentos para a qual deve-se montar os carregamentos proporcionais e os carregamentos constantes. Para tal, foram introduzidas novas assinaturas dos métodos *setProportionalLoad* e *setConstantLoad*.

Quanto ao cálculo dos carregamentos constante e proporcional devidos à variação de temperatura, introduziu-se o método *setTempVarLoading*, que possui uma assinatura pública contendo um número maior de parâmetros e três assinaturas privadas que utilizam um número menor de parâmetros, sendo essas últimas utilizadas pelo método público durante a montagem dos vetores desejados.

Os métodos adicionados na interface **Assembler** também foram introduzidos nas demais superclasses associadas ao Método dos Elementos de Contorno, para que elas possam ser instanciadas, porém esses métodos não possuem conteúdo em suas implementações.

4.7.3 Modificações no projeto `model`

Nesta seção serão apresentadas as modificações nos pacotes pertencentes ao projeto `model` para viabilizar a solução termomecânica.

4.7.3.1 Modificações na classe `FemModel`

Foi realizada uma modificação no método *init* da classe `FemModel`, no qual incluiu-se uma linha para registrar a referência ao objeto *problem driver* nos elementos durante o processo de inicialização do modelo discreto. Esse passo é importante durante as análises termomecânicas em regime permanente, nas quais o mesmo modelo discreto é empregado na solução do problema de transferência de calor e na solução do problema da mecânica dos sólidos.

4.7.3.2 Modificações na classe `Node`

Foram incluídas duas chaves na classe para representarem a temperatura inicial e a temperatura corrente no mapa de valores do nó. Essas chaves receberam os nomes, respectivamente, de `TEMPERATURE` e `INITIAL_TEMPERATURE`. Como discutido previamente, essas chaves são necessárias durante o cálculo termomecânico fisicamente não-linear em regime quasi-estático.

4.7.3.3 Modificações na classe `Element`

Devido à simplificação da maneira de representar os carregamentos de variação de temperatura, que passam a atuar obrigatoriamente sobre todo o domínio do elemento, foram removidas as chaves relativas às implementações prévias dos carregamentos de variação de temperatura e de deformação inicial que possuíam atrelamento à geometria dos seus domínios de atuação.

Ademais, devido à separação dos carregamentos de variação de temperatura em relação aos demais carregamentos da mecânica dos sólidos, incluiu-se o método *getH*,

que delega ao *problem driver* o cálculo do carregamento nodal equivalente devido à variação de temperatura.

Por fim, para que as tensões sejam calculadas corretamente nos casos em que existem deformações provenientes da variação de temperatura, foram realizadas modificações nos métodos *getMpDualInternalVariables*, *getPointDualInternalVariables*, *getPointInternalVariables* e *getPointInternalVariables*. Nessas modificações, optou-se por realizar a subtração das deformações térmicas nos métodos que calculam o valor da deformação, de forma que eles somente retornem deformações oriundas dos esforços mecânicos. Devido a essa decisão, as implementações dos métodos que realizavam o cálculo das tensões a partir das deformações foram modificados para a sua forma original, prévia à introdução do cálculo termomecânico no INSANE, visto que já estão recebendo por parâmetro a deformação previamente corrigida.

Além das modificações na classe **Element**, tornou-se necessário modificar o método *getPointInternalVariables* na classe **Bar**, herdeira de **Element**, visto que ela sobrecarrega o referido método. As modificações realizadas nas classes desse pacote estão indicadas na Figura 4.38.

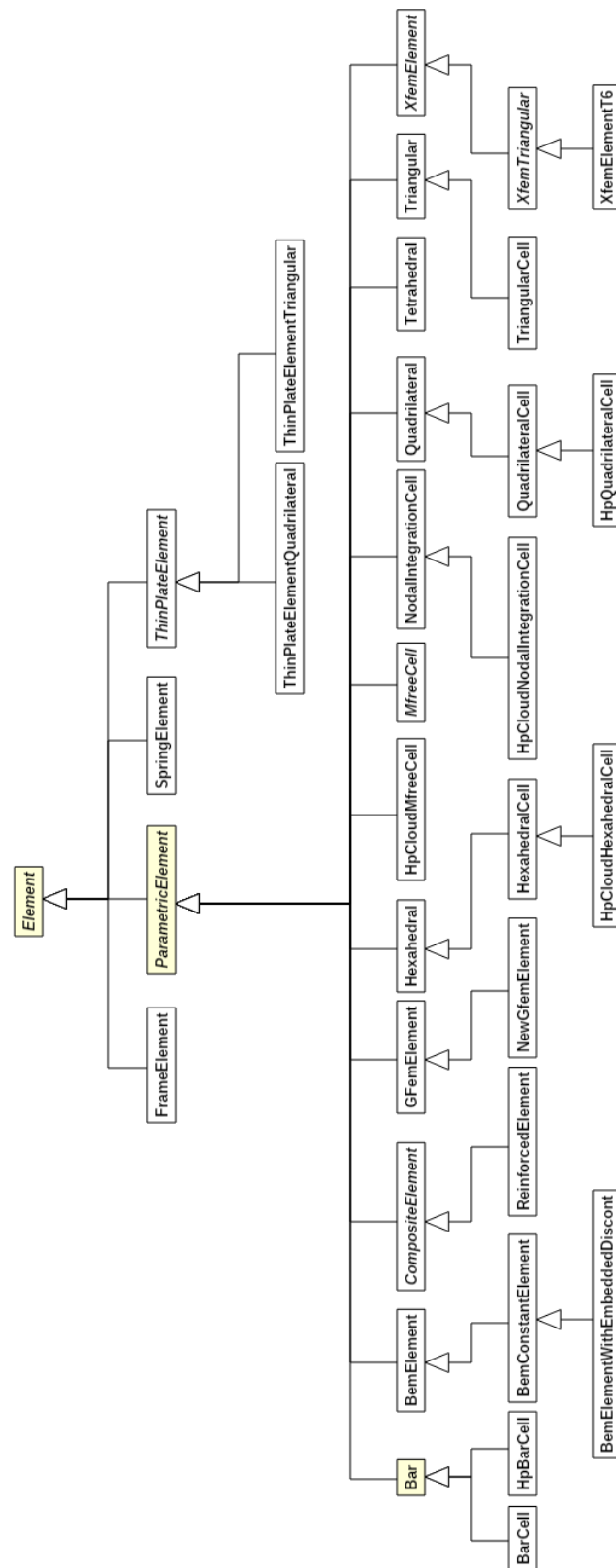


Figura 4.38: Modificações realizadas no pacote element.

4.7.3.4 Modificações no pacote `analysismodel`

Quando a temperatura varia em um ponto do material, surgem deformações oriundas dessa variação de temperatura, que são representadas pelo tensor de deformações térmicas mencionado na Equação 3.84, como discutido em detalhes no Capítulo 3 dessa dissertação. As componentes desse tensor dependem, além da variação de temperatura, do tipo de material e do modelo de análise.

Como a implementação realizada nesse trabalho se restringe a materiais isotrópicos, assumiu-se por simplificação que as deformações angulares devidas à variação de temperatura serão sempre nulas e que as deformações lineares possuirão sempre o mesmo valor. Portanto, resta somente a responsabilidade ao modelo de análise informar, a quem for calcular o tensor de deformações ocasionadas pela variação de temperatura, quais são as componentes não nulas. Para essa finalidade, introduziu-se o método *getTemperatureAffectedInternalVariables* de na classe `AnalysisModel`, sendo esse método sobrecarregado nas classes herdeiras indicando quais são as componentes lineares e quais são as angulares por meio de um vetor de valores booleanos. Essas modificações estão indicadas na Figura 4.5.

4.7.3.5 Modificações no pacote `problemdriver`

A classe `ProblemDriver` foi modificada recebendo a implementação do método abstrato *getH*, responsável pelo cálculo do carregamento nodal equivalente devido à variação de temperatura no interior do domínio do elemento finito, conforme já discutido anteriormente. Somente a classe `Parametric` recebeu uma implementação do método, enquanto as demais classes herdeiras de `ProblemDriver` implementam o método para que não sejam abstratas, mas apenas retornam um ponteiro nulo. Essa modificação nas classes existentes está indicada na Figura 4.6

4.7.3.6 Modificações no pacote load

O pacote load sofreu uma alteração significativa em relação à versão documentada em Botelho et al. (2015). Na versão aqui descrita, fez-se a simplificação de que os carregamentos de campo de deformações iniciais prescritas, e conseqüentemente os carregamentos de variação de temperatura, devem atuar em todo o domínio do elemento finito. Tal simplificação permite que o domínio espacial do objeto de carregamento seja obtido da geometria do elemento finito, sendo necessário portanto informar somente os valores nodais de deformação inicial ou de variação de temperatura.

Devido a essa modificação, foram removidas as seguintes classes:

- ElementAreaInitialStrainLoad;
- ElementAreaTemperatureVariationLoad;
- AreaInitialStrainEquivalentNodalValue;
- AreaTemperatureVariationEquivalentNodalValue;
- ElementLineInitialStrainLoad;
- ElementLineTemperatureVariationLoad;
- LineInitialStrainEquivalentNodalValue;
- LineTemperatureVariationEquivalentNodalValue;
- ElementVolumeInitialStrainLoad;
- ElementVolumeTemperatureVariationLoad;
- VolumeInitialStrainEquivalentNodalValue;
- VolumeTemperatureVariationEquivalentNodalValue.

Essas doze classes listadas acima foram substituídas pelas classes a seguir:

- `ElementInitialStrainLoad`;
- `ElementTemperatureVariationLoad`;
- `InitialStrainEquivalentNodalValue`.

Ressalta-se que a classe `ElementTemperatureVariationLoad` é uma especialização da classe `ElementInitialStrainLoad`, e por isso a classe `InitialStrainEquivalentNodalValue` é capaz de atender a ambos os tipos de carregamentos. O cálculo do carregamento nodal equivalente devido à presença de um campo de deformações iniciais é dado pela Equação 3.123.

As novas classes introduzidas para a representação dos carregamentos de deformação inicial e de variação de temperatura, bem como a nova classe para o cálculo do carregamento nodal equivalente devido a campos de deformação inicial, foram mostradas na Figura 4.10.

4.8 Modificações no projeto persistence

A classe `PersistenceAsXML` foi modificada para atender à nova arquitetura aqui proposta. Para tal, tornou-se necessário incluir métodos para a leitura das informações relacionadas às condições inicial e de contorno prescritas sobre a variável de estado, métodos `fillStateVarBcListFromFile` e `fillStateVarIcListFromFile`, e para a leitura das informações relacionadas ao *simulation manager*, método `fillSimulationManagerFromFile`. No que se refere às entidades multifísicas, incluiu-se o método `fillMultiphysicsEntitiesFromFile`, cuja leitura ocorre previamente à leitura do *problem driver* e do modelo de análise global. Também foram realizadas modificações no método `fillLoadingsFromFile` devido à modificação da forma de representar carregamentos de deformação inicial e variação de temperatura. Quanto às demais classes incluídas no núcleo numérico para a solução de problemas térmicos e termomecânicos, também foram realizadas adequações no algoritmo de leitura do arquivo

XML para que as mesmas sejam reconhecidas e devidamente instanciadas quando necessário.

Os métodos previamente existentes no INSANE relacionados a objetos que possuem comportamento dependente de física foram adequados para que reconheçam as novas classes. Nesses casos, optou-se por incluir as fábricas de instâncias implementadas nos métodos *getSolutionFromOMEElement* e *getMaterialFromOMEElement*, que são evocadas durante a leitura do arquivo XML.

Como discutido na seção 4.5.3.5, para a realização da análise termomecânica fisicamente não-linear em regime transiente optou-se por utilizar dois modelos independentes para cada física, porém ambos criados a partir de um único modelo discreto de referência. A criação da cópia desse modelo de referência é efetuada pelo método *cloneModel*, introduzido nesse trabalho.

Capítulo 5

EXEMPLOS DE VALIDAÇÃO

Ao longo desse capítulo são apresentados os testes realizados para validar a implementação descrita no capítulo anterior. Oito exemplos são abordados, sendo quatro referentes ao problema de transferência de calor e quatro referentes ao problema termomecânico. Esses exemplos são analisados utilizando o INSANE, e em seguida as soluções numéricas obtidas são comparadas com os valores esperados, fornecidos pelas respectivas soluções analíticas desses problemas.

Em cada um desses dois grandes grupos são abordados dois casos em regime permanente e dois em regime transiente. Em cada um desses regimes temporais, analisa-se um problema em que as propriedades físicas do meio possuem comportamento linear e um em que essas propriedades manifestam não-linearidade.

No caso dos problemas de transferência de calor, a não-linearidade se manifesta pela dependência entre a condutividade térmica e a temperatura, enquanto nos problemas termomecânicos introduziu-se como não-linearidade física o comportamento elastoplástico do material.

A seção 5.1 apresenta os quatro problemas de transferência de calor. Para o PTC com comportamento linear em regime permanente, analisa-se na sub-seção 5.1.1 a distribuição de temperaturas em uma placa quadrada submetida a temperaturas e a fluxos de calor prescritos em suas extremidades. Ainda no regime permanente, porém levando em consideração a não-linearidade física, discute-se na sub-seção 5.1.2 a obtenção do perfil de temperaturas em uma esfera oca onde as temperaturas são

prescritas nas superfícies interna e externa, e na qual a condutividade térmica varia linearmente com a temperatura.

Na sequência, discutem-se os dois problemas de transferência de calor em regime transiente. O primeiro deles possui comportamento linear, e se refere a uma barra submetida a um aquecimento senoidal em suas extremidades, tratada na subseção 5.1.3. O último problema discutido na seção 5.1 corresponde a um cilindro submetido à troca de calor por convecção em sua superfície. Nesse problema, abordado na subseção 5.1.4, a condutividade térmica é representada por um polinômio de primeiro grau em função da temperatura.

A seção 5.2 apresenta os quatro problemas termomecânicos utilizados para a validação da implementação. O primeiro a ser discutido é uma viga bi-apoiada cujo material tem comportamento linear. Essa viga, melhor detalhada na sub-seção 5.2.1, é submetida a uma distribuição de temperaturas parabólica assimétrica ao longo da sua altura transversal, o que ocasiona a flexão da viga acompanhada de expansão na direção axial. Em seguida, discute-se na sub-seção 5.2.2 a resposta mecânica de uma barra bi-engastada submetida a uma variação de temperatura uniforme ao longo de todo o seu comprimento. Nesse exemplo, a variação de temperatura é suficiente para que a deformação ocasionada pela variação de temperatura exceda a deformação limite do regime elástico da barra, introduzindo deformações plásticas por compressão.

Os problemas termomecânicos em regime transiente são abordados na sequência. Essa discussão se inicia na sub-seção 5.2.3 ao se analisar a resposta elástica ao longo do tempo de uma viga submetida a um fluxo de calor em sua face superior. Essa viga possui comportamento fisicamente linear. Por fim, estuda-se na sub-seção 5.2.4 o comportamento mecânico da mesma barra abordada na sub-seção 5.2.2, porém nesse caso obtém-se a resposta no regime transiente entre a condição inicial e a distribuição de temperaturas uniforme que é obtida quando o regime permanente é alcançado. Como as condições finais dos exemplos abordados nas sub-seções 5.2.2

e 5.2.3 são iguais, espera-se que ambos exemplos apresentem o mesmo resultado no término da análise. Esses exemplos estão resumidos na tabela a seguir.

Tabela 5.1: Descrição resumida dos testes de validação realizados.

ID	Tipo	Descrição	Regime	Comportamento	Seção
1	PTC	Placa retangular com fluxos e temperaturas prescritas no contorno	Permanente	Linear	5.1.1
2	PTC	Esfera oca com temperaturas prescritas nas superfícies interna e externa	Permanente	Não-Linear	5.1.2
3	PTC	Barra com aquecimento senoidal nas extremidades	Transiente	Linear	5.1.3
4	PTC	Cilindro longo trocando calor por convecção na superfície	Transiente	Não-Linear	5.1.4
5	PTM	Viga bi-apoiada com geração de calor em seu domínio	Permanente	Linear	5.2.1
6	PTM	Barra bi-engastada com aquecimento uniforme	Permanente	Não-Linear	5.2.2
7	PTM	Viga submetida a fluxo de calor prescrito em sua face superior	Transiente	Linear	5.2.3
8	PTM	Barra bi-engastada com aquecimento senoidal nas extremidades	Transiente	Não-Linear	5.2.4

Por não ser objetivo dessa dissertação de mestrado a implementação de algoritmos referentes à plasticidade computacional, a teoria associada não é abordada nesse trabalho. Para maiores detalhes quanto à fundamentação teórica e à respectiva implementação desse recurso no INSANE, refere-se o leitor aos trabalhos de Penna (2011) e Oliveira (2016).

Ao longo dos exemplos foram utilizados símbolos para representar as unidades de medida de maneira genérica, conforme detalhado na tabela abaixo.

Tabela 5.2: Simbologia adotada para as unidades de medida.

Símbolo	Descrição
uc	unidade de comprimento
ut	unidade de temperatura relativa
up	unidade de potência
ute	unidade de tempo
uta	unidade de temperatura absoluta
um	unidade de massa
ue	unidade de energia
uf	unidade de força

No que se refere às classes empregadas para a definição dos modelos numéricos e para a obtenção das soluções utilizando o INSANE em cada um dos exemplos de validação, um resumo pode ser encontrado no Apêndice C dessa dissertação de mestrado.

5.1 Problemas de Transferência de Calor

5.1.1 Exemplo 1 - Placa retangular com fluxos e temperaturas prescritas no contorno

O primeiro exemplo de validação apresentado nesse capítulo tem como objetivo principal validar a implementação previamente realizada, que já viabilizava a solução de problemas de transferência de calor lineares e em regime permanente. Além

da validação do núcleo numérico para esse tipo de aplicação, esse exemplo também valida a implementação do gerenciador de solução de compatibilidade, que é automaticamente instanciado pelo núcleo numérico quando não é declarada a utilização de um tipo específico de gerenciador. O gerenciador de compatibilidade tem como objetivo garantir que todas as análises que eram realizadas previamente no INSANE sem a declaração de um *simulation manager* continuem sendo executadas sem nenhuma alteração.

O problema escolhido se refere à placa retangular mostrada na Figura 5.1. Essa placa possui as suas faces perpendiculares ao eixo Z impedidas de trocar calor com a vizinhança. Devido a essa condição de contorno, pode-se simplificar o problema e representar a placa somente por um plano, paralelo ao sistema de eixos XY . Escolhe-se, portanto, o domínio definido pelos pontos $ABCD$. A largura desse domínio é a , a altura b e a espessura da placa, não representada no modelo bidimensional é t .

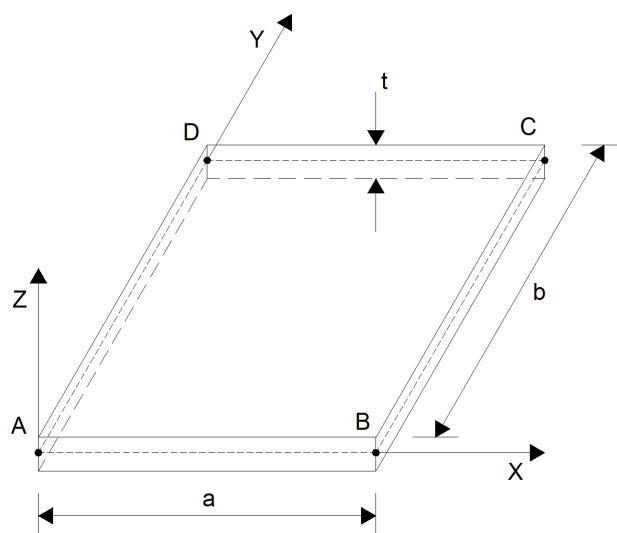


Figura 5.1: Placa estudada no Exemplo 1.

Além das condições de contorno acima descritas, são definidas as seguintes condições de contorno nas demais faces da placa:

- Superfícies adiabáticas nas faces laterais que contêm as arestas AD e CD ;

- Temperatura prescrita na face lateral que contém a aresta AB, cujo valor é T_{AB} ;
- Troca de calor por convecção na face lateral que contém a aresta BC. O coeficiente de troca de calor por convecção é h_c e a temperatura do fluido é $T_\infty = 0$.

Pode-se observar que o problema proposto contempla condições de contorno de primeira, segunda e terceira espécie. Essas condições de contorno estão ilustradas na Figura 5.2.

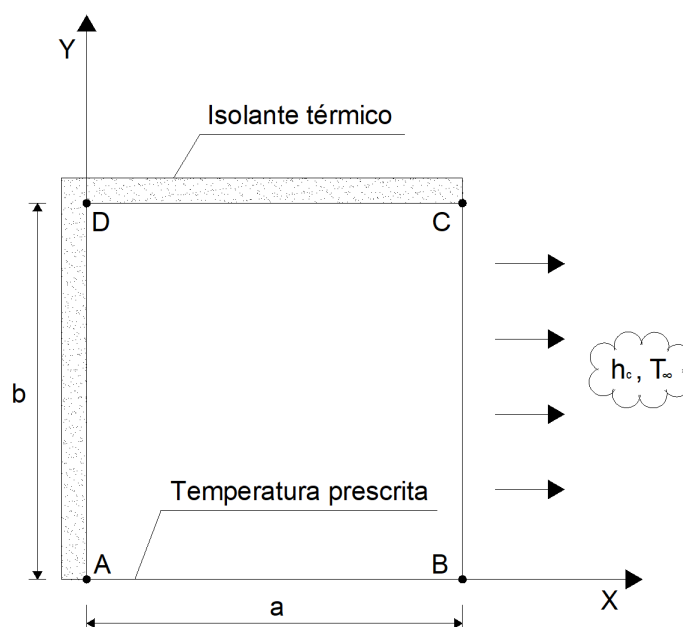


Figura 5.2: Representação bidimensional da placa retangular do Exemplo 1.

A solução analítica desse problema é apresentada por Carslaw e Jaeger (1959) e desenvolvida em detalhes no Apêndice B. O campo de temperaturas obtido dessa solução é dado por:

$$T(x, y) = 2 h T_{AB} \sum_{n=1}^{\infty} \frac{\cos(\alpha_n x) \cosh[\alpha_n (b - y)]}{[(\alpha_n^2 + h^2)a + h] \cos(\alpha_n a) \cosh(\alpha_n b)} \quad (5.1)$$

onde h é um coeficiente dado pela divisão entre o coeficiente convectivo h_c e a condutividade térmica k , logo $h = h_c/k$ e α_n são as raízes da equação transcendental:

$$\alpha \tan(\alpha a) = h \quad (5.2)$$

Para viabilizar a geração de um modelo numérico para comparar os resultados obtidos pelo INSANE com os resultados analíticos, são atribuídos os valores apresentados na tabela a seguir aos parâmetros do problema:

Tabela 5.3: Parâmetros considerados na modelagem do Exemplo 1.

Parâmetro	Símbolo	Valor	Unidade
Largura da placa	a	1	uc
Altura da placa	b	1	uc
Espessura da placa	t	0,01	uc
Temperatura na face AB	T_{ab}	100	ut
Coefficiente convectivo	h_c	10	$up/(uc^2)$
Condutividade térmica	k	1	$up/(uc \cdot ut)$

O domínio retangular foi subdividido em 10.000 elementos finitos quadrilaterais de quatro nós, de mesmo tamanho, correspondendo a uma subdivisão de cada lado do retângulo em 100 partes iguais. O problema discreto resultante possui, portanto, 10.201 graus de liberdade, onde 101 desses graus de liberdade são prescritos e os demais desconhecidos.

As condições de contorno essenciais foram atribuídas diretamente no mapa de valores nodais, seguindo a metodologia convencional para presecrever esses valores em modelos discretos no INSANE. Decidiu-se impor as condições de contorno essenciais dessa forma para comprovar que a introdução de uma nova metodologia para tal não comprometeu os recursos previamente existentes.

O campo de temperaturas obtido da simulação computacional utilizando o INSANE é ilustrado na Figura 5.3.

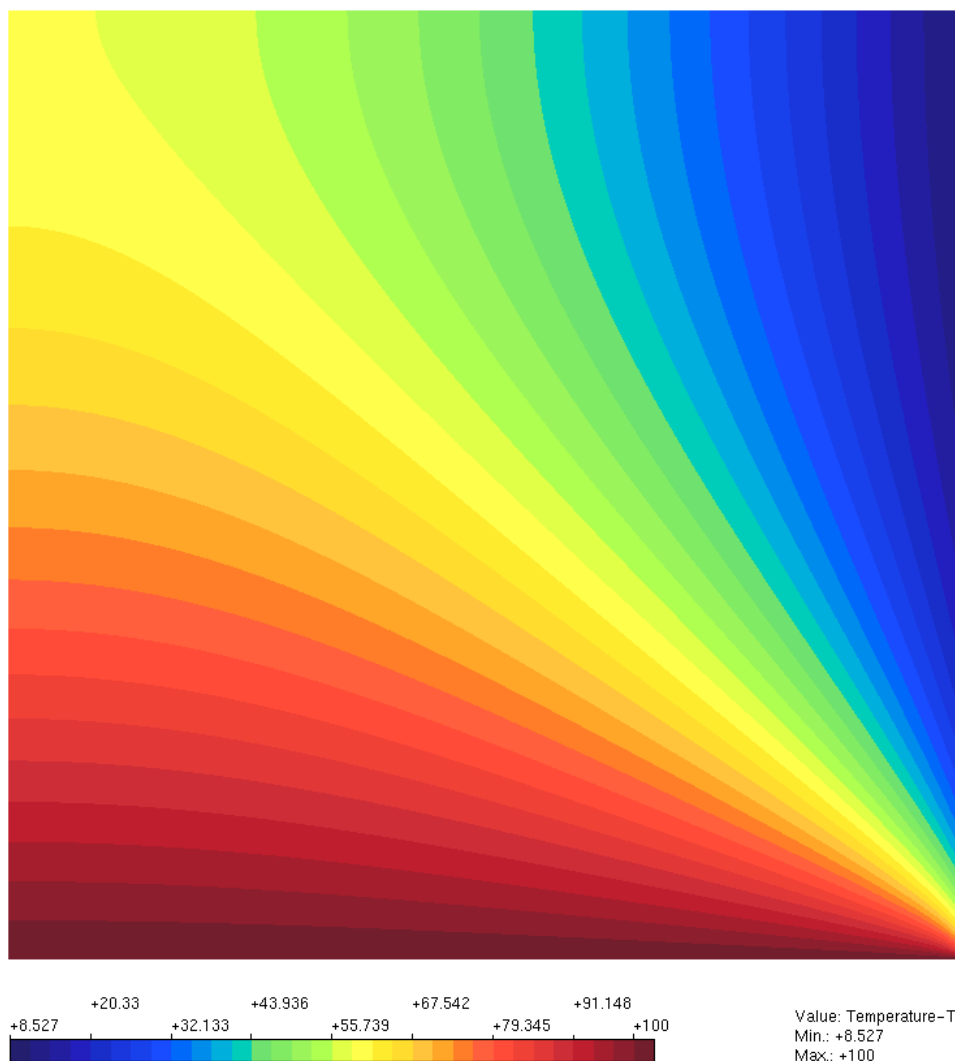


Figura 5.3: Distribuição de temperaturas calculada para a placa do Exemplo 1.

O gráfico a seguir ilustra a comparação entre a solução analítica e a solução numérica calculada pelo INSANE ao longo dos caminhos \overrightarrow{AD} , \overrightarrow{DC} e \overrightarrow{CB} . A solução analítica foi calculada considerando 200 termos na série da Equação 5.1.

Pode-se observar nesse gráfico que as soluções analítica e numérica se igualaram, o que valida a implementação realizada para o caso do PTC linear em regime permanente. Também é validada nesse exemplo a criação automática do gerenciador de solução do tipo `CompatibilitySimulationManager` quando o usuário não especifica essa entidade durante a etapa de pré-processamento.

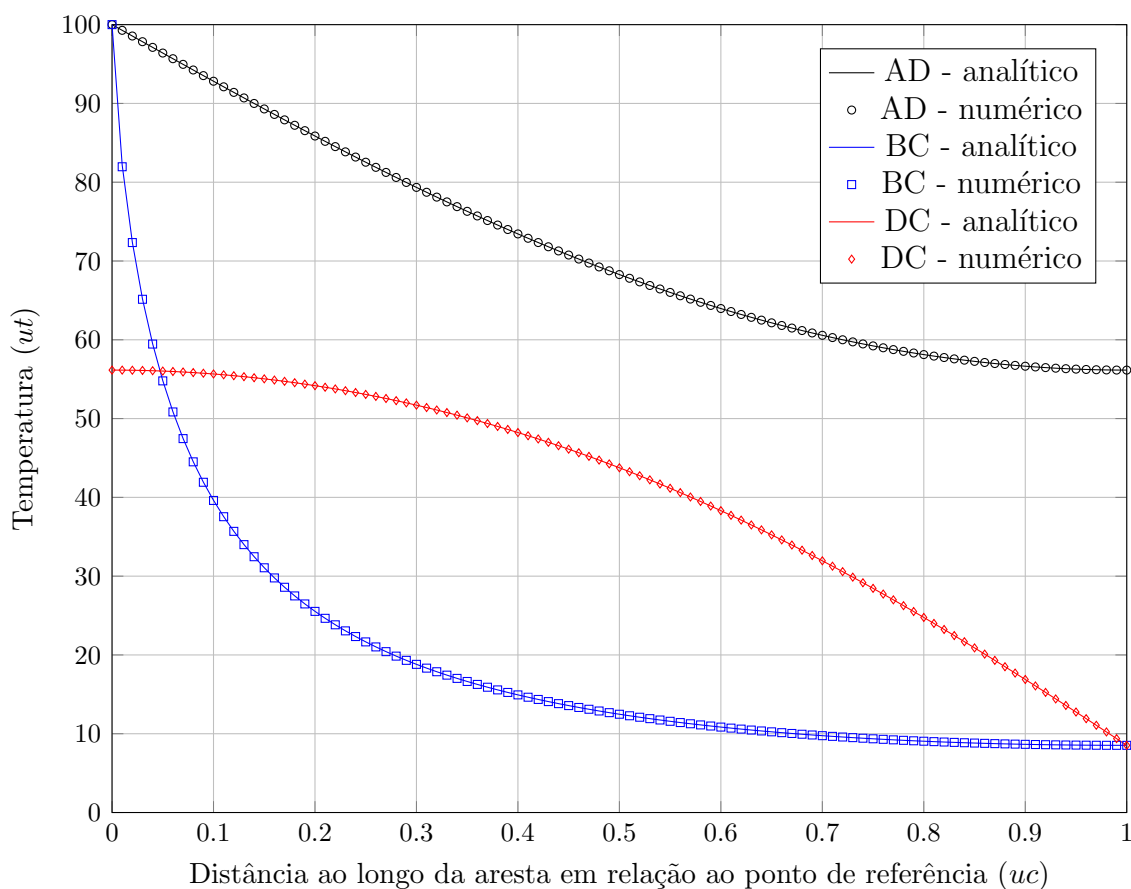


Figura 5.4: Resultados comparativos para a placa do Exemplo 1.

5.1.2 Exemplo 2 - Esfera oca com temperaturas prescritas

Na análise desse exemplo busca-se a distribuição de temperaturas em uma esfera oca submetida a temperaturas prescritas em suas superfícies interna e externa. O comportamento do material é considerado não-linear, onde a condutividade térmica varia com a temperatura segundo um polinômio de primeiro grau.

A solução analítica desse problema pode ser obtida seguindo a metodologia descrita em (Danish et al., 2011), sendo apresentada em detalhes no Apêndice B. O raio interno da esfera é R_a e o raio externo R_b . As temperaturas interna e externa são, respectivamente, T_a e T_b . À temperatura T_a a condutividade térmica é igual a k_a e à temperatura T_b , igual a k_b . Dessa forma, a função que descreve a variação da condutividade térmica com a temperatura pode ser expressa na forma:

$$k(T) = k_a + (k_b - k_a) \frac{T - T_a}{T_b - T_a} \quad (5.3)$$

O campo de temperaturas, válido no intervalo $R_a < r < R_b$, é dado pela expressão a seguir, onde r é a posição radial ao longo da parede da esfera oca:

$$T(r) = T_a + k_a (T_b - T_a) \left(\frac{-1 + \sqrt{1 + \left(\frac{k_b^2 - k_a^2}{k_a^2} \right) \left(\frac{R_b}{R_b - R_a} \right) \left(\frac{r - R_a}{r} \right)}}{k_b - k_a} \right) \quad (5.4)$$

Os valores relacionados na tabela abaixo são atribuídos às variáveis do problema, e então um modelo numérico é desenvolvido para que os resultados calculados pelo INSANE sejam comparados aos resultados analíticos.

Tabela 5.4: Parâmetros considerados na modelagem do Exemplo 2.

Parâmetro	Símbolo	Valor	Unidade
Raio Interno	R_a	100	uc
Raio Externo	R_b	300	uc
Temperatura na superfície interna	T_a	0	ut
Temperatura na superfície externa	T_b	1000	ut
Condutividade térmica à 0 ut	k_a	20	$up/(uc \cdot uta)$
Condutividade térmica à 1000 ut	k_b	1020	$up/(uc \cdot uta)$

Devido à simetria do problema, o calor flui somente na direção radial, o que permite modelar numericamente somente um setor dessa esfera. Dessa forma, escolheu-se modelar um setor de 5,625 graus, que foi subdividido em 306 elementos finitos hexaédricos de oito nós. Nessa discretização, mostrada na Figura 5.5, foram dispostos 35 pontos igualmente espaçados ao longo da espessura da parede da esfera oca. Nas direções circunferencial e azimutal, foram dispostos 4 pontos igualmente espaçados, o que totaliza 560 nós no modelo numérico. Cada um desses nós possui somente um grau de liberdade, que é a temperatura.

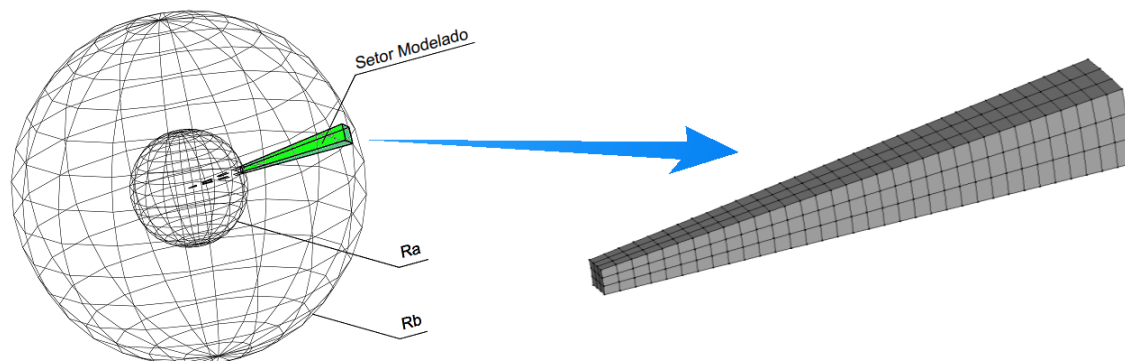


Figura 5.5: Modelagem computacional do setor da esfera oca do Exemplo 2.

Nesse modelo discreto, utilizou-se a nova metodologia para aplicar as condições de contorno essenciais. Para tal, criou-se um grupo de condição de contorno essencial para os nós situados na superfície interna, ao qual foi atribuído o valor prescrito de T_a , e um grupo para os nós situados na superfície externa, que recebeu o valor de T_b . Cada um desses grupos foi incluído em um caso de condição de contorno essencial, e em seguida esses dois casos foram combinados em uma combinação de condições de contorno essenciais.

Quanto à solução, ela foi realizada utilizando uma nova classe introduzida no núcleo numérico, denominada `NonIncrementalEqPath`, trabalhando em conjunto com um objeto `step` do tipo `NewtonIterative`. Nesse exemplo, se utiliza a entidade para gerenciamento da solução nos casos em que há somente uma física a ser resolvida, sendo empregado um objeto do tipo `SinglePhysicsFemSimManager` trabalhando em conjunto com um objeto do tipo `SinglePhysicsFemSetuper`.

No que se refere aos resultados obtidos, o critério de convergência escolhido (de norma do resíduo inferior a 1×10^{-15}) foi alcançado após 11 iterações. A distribuição de temperaturas obtida no setor modelado é mostrada na Figura 5.6.

Um comparativo entre as temperaturas calculadas analiticamente e as obtidas numericamente pelo INSANE ao longo da parede da esfera oca é ilustrado no gráfico apresentado na Figura 5.7. Pode-se observar que os resultados convergiram para a solução esperada para o problema.

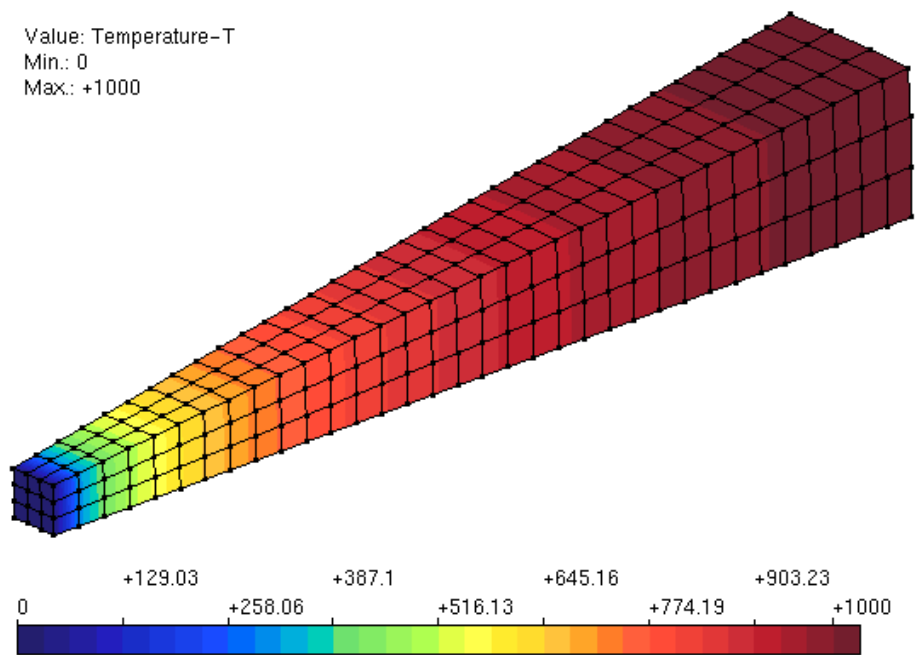


Figura 5.6: Exemplo 2 - Distribuição de temperaturas calculadas pelo INSANE.

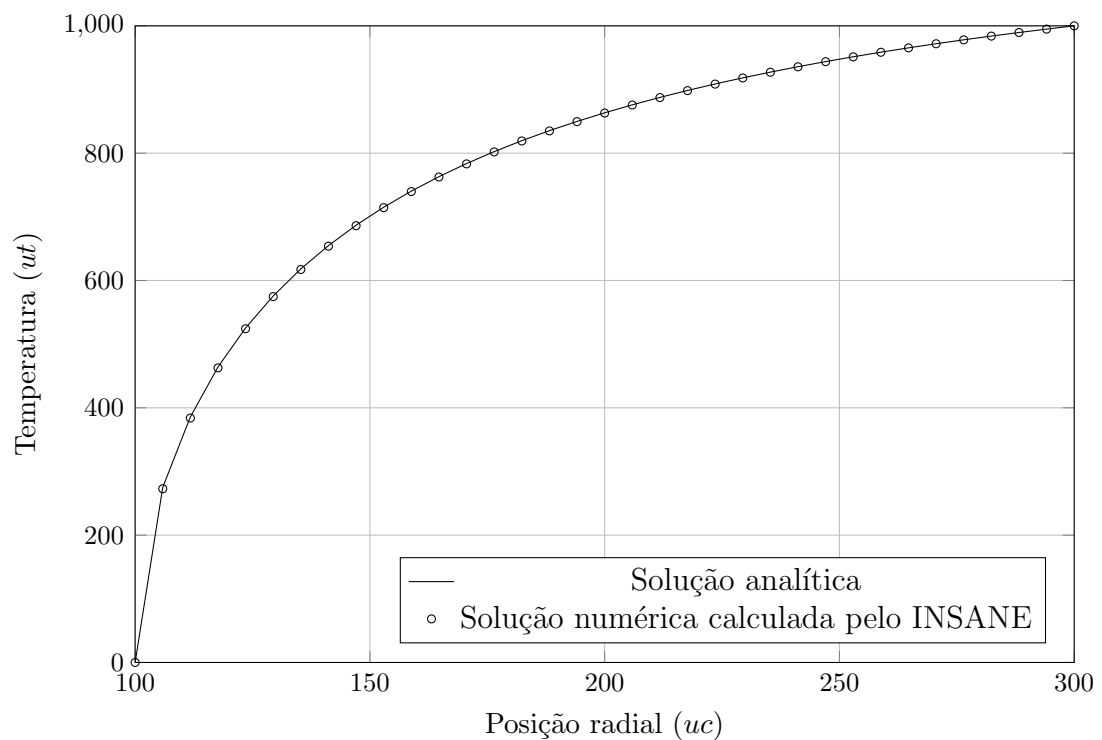


Figura 5.7: Comparativo entre as soluções analítica e numérica do Exemplo 2.

5.1.3 Exemplo 3 - Barra com aquecimento senoidal nas extremidades

Para validar as classes criadas para a solução de problemas de transferência de calor lineares em regime transiente, realizou-se a análise de uma barra impedida de trocar calor com o ambiente ao longo do seu comprimento. Nessa barra, impõe-se que temperatura nas extremidades cresça segundo uma função senoidal até que o pico dessa função seja atingido no instante de tempo denotado por p . Após concluir o período de elevação da temperatura nas bordas, ela é mantida constante nesses locais e aguarda-se a estabilização do sistema. Esse comportamento da condição de contorno é ilustrado na figura a seguir.

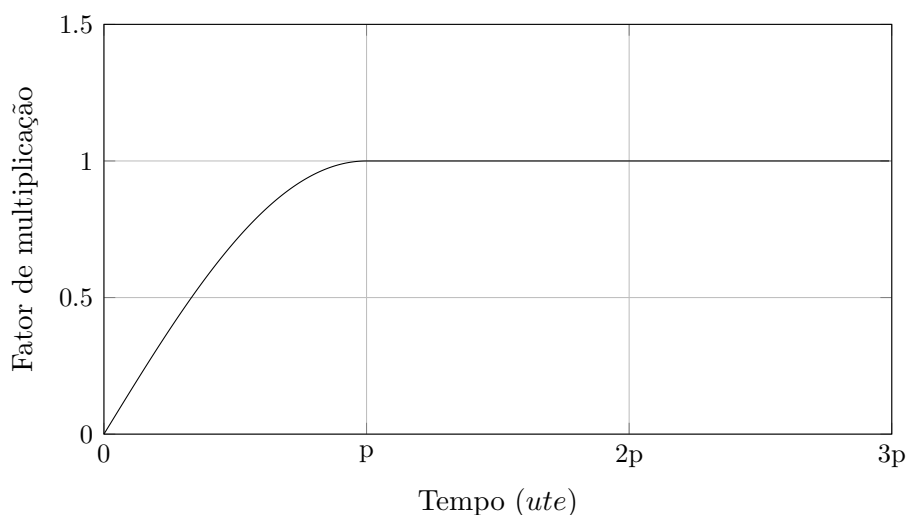


Figura 5.8: Função que descreve o valor da temperatura prescrita nas bordas da barra.

Após tempo suficiente, a temperatura ao longo de toda a barra deve se igualar ao valor prescrito em suas extremidades. A condição inicial desse problema é uma distribuição de temperaturas uniforme e igual a zero. Quanto aos valores de temperatura prescritos no contorno, restringe-se o estudo ao caso em que eles são iguais em ambas as extremidades, conferindo simetria nos resultados em relação ao centro da barra.

A solução analítica desse problema é desenvolvida em duas fases. Para a primeira fase, onde as condições de contorno variam segundo uma função harmônica, segue-se a metodologia apresentada por Groulx (2007). A segunda fase, que corresponde à uniformização da temperatura na barra, constitui um problema onde as condições de contorno não mais variam com o tempo. Nesse período, segue-se a metodologia apresentada por Boyce e DiPrima (2006). O desenvolvimento detalhado da solução analítica é apresentado no Apêndice B.

Na barra em questão, o comprimento é $2L$, a temperatura nas bordas é θ , a posição longitudinal na barra, medida em relação à borda esquerda, é denotada por s e o tempo é representado pelo símbolo t . A difusividade térmica é denotada por α , sendo calculada pela divisão da condutividade térmica k pelo produto entre a massa específica ρ e o calor específico à pressão constante c_p .

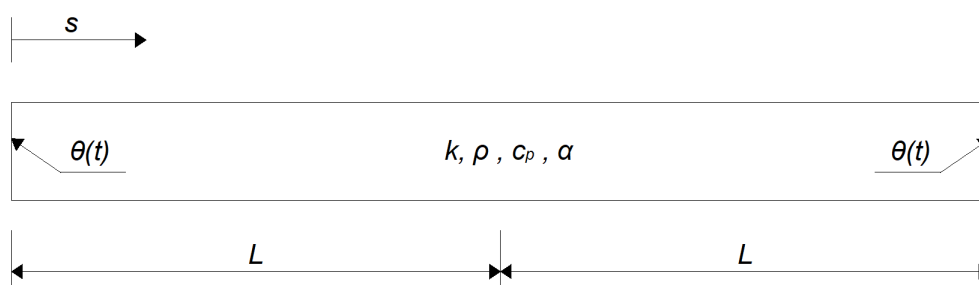


Figura 5.9: Barra com temperaturas prescritas nas extremidades - Exemplo 3.

A distribuição de temperaturas $T(s, t)$ no interior da barra, $0 \leq s \leq 2L$, durante o trecho de elevação da temperatura nas bordas, $0 \leq t \leq p$, é dada pela expressão a seguir:

$$T(s, t) = \theta \operatorname{sen} \left(\frac{\pi t}{2p} \right) + \frac{2\theta}{p} \sum_{m=1}^{\infty} \left\{ \frac{(-1)^m}{(2m-1)} \cdot \frac{\cos(\beta_m |L-s|)}{(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2} \left[\alpha\beta_m^2 \cos \left(\frac{\pi t}{2p} \right) \right] \right\} + \frac{2\theta}{p} \sum_{m=1}^{\infty} \left\{ \frac{(-1)^m}{(2m-1)} \cdot \frac{\cos(\beta_m |L-s|)}{(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2} \left[\left(\frac{\pi}{2p} \right) \operatorname{sen} \left(\frac{\pi t}{2p} \right) - \alpha\beta_m^2 e^{(-\alpha\beta_m^2 t)} \right] \right\} \quad (5.5)$$

Para o período de estabilização $p \leq t \leq \infty$ a solução no interior da barra é dada por:

$$T(s, t) = \theta + \sum_{n=1}^{\infty} C_n e^{-n^2 \pi^2 \alpha (t-p)/4L^2} \operatorname{sen} \left(\frac{n\pi s}{2L} \right) \quad (5.6)$$

onde:

$$C_n = \sum_{m=1}^{\infty} A_m \frac{\cos(\beta_m L)}{2} \left[\frac{1 - \cos(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) - \beta_m} + \frac{1 - \cos(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) + \beta_m} \right] + \sum_{m=1}^{\infty} A_m \frac{\operatorname{sen}(\beta_m L)}{2} \left[\frac{\operatorname{sen}(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) - \beta_m} - \frac{\operatorname{sen}(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) + \beta_m} \right] \quad (5.7)$$

e os coeficientes A_m e β_m são:

$$A_m = \left(\frac{2\theta}{pL} \right) \left[\frac{(-1)^m}{(2m-1)} \right] \left[\frac{\left(\frac{\pi}{2p}\right) - \alpha\beta_m^2 e^{-\alpha\beta_m^2 p}}{(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2} \right] \quad (5.8)$$

$$\beta_m = (2m-1) \frac{\pi}{2L} \quad (5.9)$$

Nas séries acima, observa-se a ocorrência de indeterminações quando $n = 2m - 1$ nos termos:

$$\frac{1 - \cos(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) + \beta_m} \quad \text{e} \quad \frac{\operatorname{sen}(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) - \beta_m}$$

Ao avaliar o limite quando $n \rightarrow 2m - 1$, conclui-se ao aplicar o teorema de l'Hospital que o primeiro termo é igual a zero e o segundo é igual a $2L$. Portanto, durante o cálculo da solução analítica deve-se substituir esses valores quando a relação entre n e m for causar indeterminação.

A comparação entre a solução calculada numericamente pelo INSANE e a solução analítica é realizada para um modelo conforme os parâmetros listados na Tabela 5.5.

Tabela 5.5: Parâmetros considerados na modelagem do Exemplo 3.

Parâmetro	Símbolo	Valor	Unidade
Comprimento da barra	$2L$	1	uc
Tempo para a elevação da temperatura nas bordas	p	3600	ute
Varição de temperatura nas bordas	$\delta\theta$	50	ut
Massa específica	ρ	2702	um/uc^3
Calor específico	c_p	903	$ue/(um \cdot ut)$
Condutividade térmica	k	237	$up/(uc \cdot ut)$
Difusividade térmica	α	$97,1 \times 10^{-6}$	uc^2/ute

A barra modelada foi subdividida em 30 elementos finitos paramétricos unidimensionais de quatro nós, o que totalizou 91 graus de liberdade. O algoritmo de integração direta escolhido para solucionar o problema foi o método de Newmark- β , onde se utilizou um passo de tempo de 1 ute e um parâmetro β igual a 0,5.

Em relação às classes do núcleo numérico do INSANE empregadas na solução desse problema, ressalta-se a utilização da nova classe de solução `LinearHeatTransferNewmarkBeta`, do *simulation manager* do tipo `SinglePhysicsFemSimManager` e do *setuper* `TimeDepThermalSetuper`. Quanto à condição inicial e quanto às condições de contorno essenciais, utilizou-se a nova metodologia proposta nesse trabalho para prescrevê-las sobre a discretização.

A análise computacional foi realizada até um instante de tempo igual a 10800 ute , o que corresponde a três vezes o tempo de elevação da temperatura nas bordas. A aproximação da solução analítica, ilustrada nos gráficos comparativos a seguir, se refere ao truncamento das séries das equações 5.6 e 5.7 nos valores máximos de m e n iguais a 8. Observa-se nesses gráficos que a solução numérica atingiu os valores calculados analiticamente, o que valida a implementação das ferramentas para a solução desse tipo de problema utilizando o INSANE.

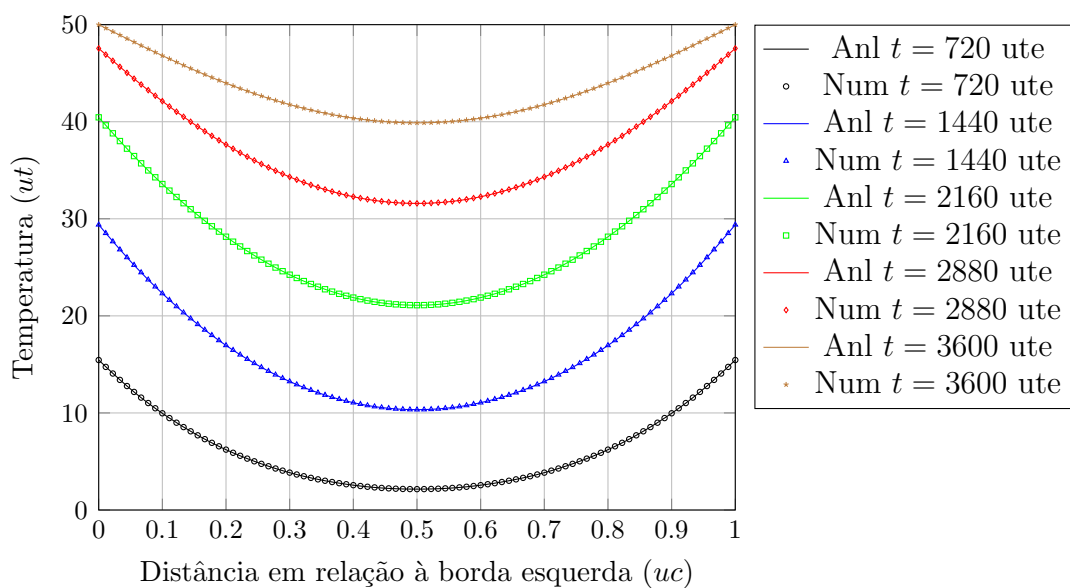


Figura 5.10: Resultados comparativos durante a elevação da temperatura nas bordas.

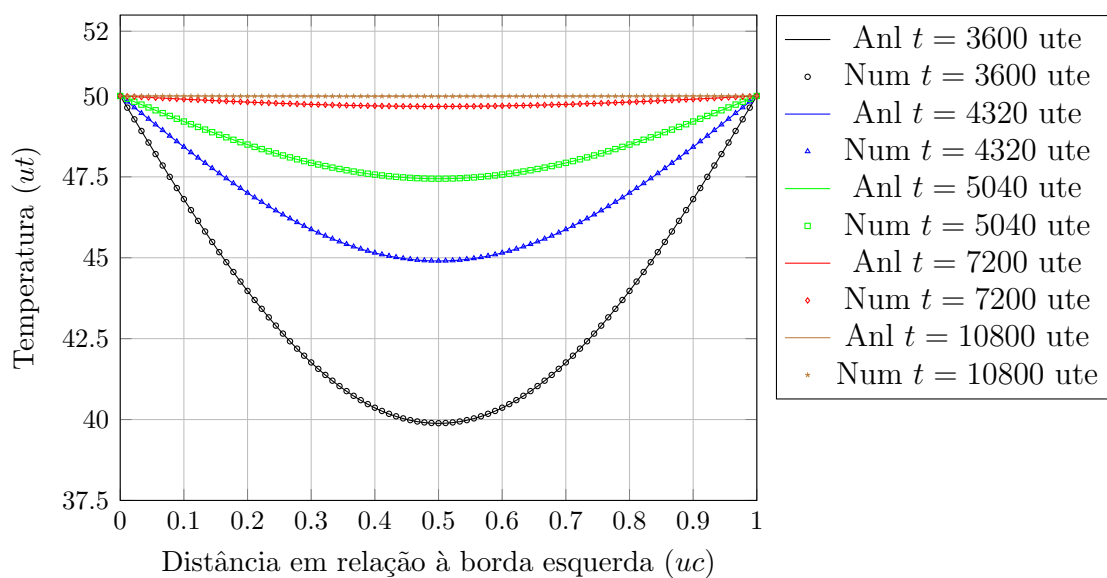


Figura 5.11: Comparativo entre as soluções durante o período de uniformização.

5.1.4 Exemplo 4 - Cilindro longo trocando calor por convecção na superfície

Nesse exemplo é abordado um problema de condução de calor fisicamente não-linear em regime transiente com o qual valida-se principalmente a nova classe `Non-LinearHeatTransferNewmarkBeta`. Para tal, considera-se o cilindro submetido à troca de calor por convecção em sua superfície com um meio à temperatura T_∞ nula, mostrado na Figura 5.12. O coeficiente de troca de calor por convecção é h , a distribuição de temperatura inicial no cilindro é uniforme com valor T_0 . O raio do cilindro é denotado pelo símbolo a e o tempo decorrido após o instante inicial é representado por t . Além disso, considera-se que o cilindro é suficientemente longo de tal forma que o fluxo de calor ocorre somente na direção radial.

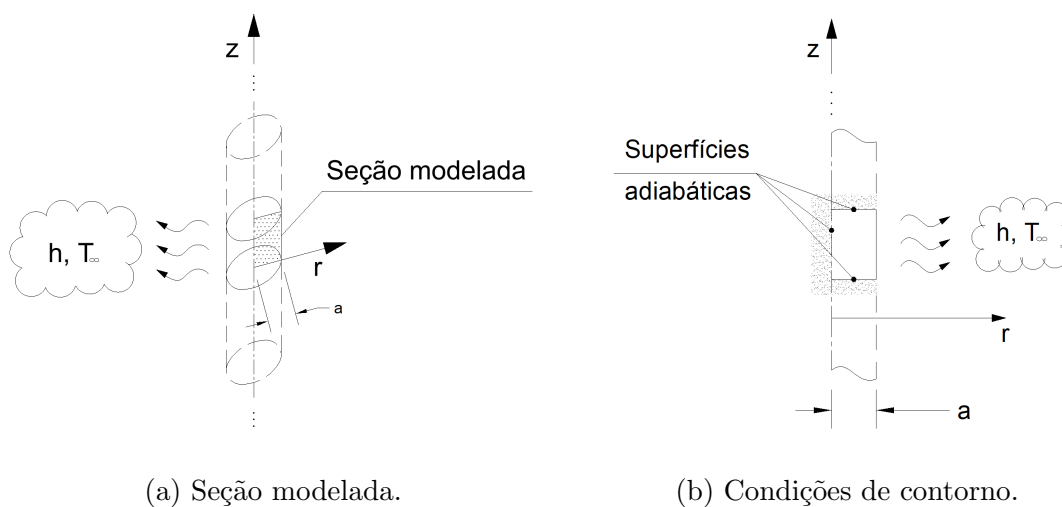


Figura 5.12: Cilindro longo exposto à convecção em sua superfície.

A solução analítica desse problema é tratada em detalhes no Apêndice B. Para obtê-la, segue-se majoritariamente o expediente apresentado por Jahanian (1995), onde se realiza uma transformação de *Kirchhoff* para linearizar a equação diferencial não-linear que rege o problema. Maiores detalhes sobre essa técnica para a solução de problemas de transferência de calor fisicamente não-lineares podem ser encontrados

em Vadasz (2010) e em Carslaw e Jaeger (1959).

Uma abordagem similar à apresentada por Jahanian (1995) pode ser encontrada em Noda (2014), que foi utilizada nesse trabalho como uma referência complementar ao trabalho de Jahanian (1995) durante o detalhamento da formulação apresentada no Apêndice B. Ressalta-se que esses autores tanto realizam a transformação de *Kirchhoff* de maneira diferente, como realizam diferentes considerações durante o detalhamento de um procedimento para a obtenção de uma solução aproximada. Optou-se nesse trabalho por seguir a solução proposta por Jahanian (1995) por sua maior simplicidade e clareza, e deixa-se o desenvolvimento detalhado da formulação analítica proposta por Noda (2014) a trabalhos futuros que dêem continuidade à linha de pesquisa exposta nessa dissertação de mestrado.

Para permitir a comparação entre a solução analítica aproximada e a solução numérica desse problema, foram conferidos os valores descritos na tabela abaixo aos parâmetros pertinentes à análise.

Tabela 5.6: Parâmetros considerados na modelagem do Exemplo 4.

Parâmetro	Símbolo	Valor	Unidade
Raio do cilindro	a	0,05	uc
Temperatura inicial	T_0	300	ut
Temperatura do fluido	T_∞	0	ut
Coefficiente convectivo	h	100	$up/(uc^2 \cdot ut)$
Condutividade térmica	$\lambda(T)$	$85,923 - 0,0437 \cdot T$	$up/(uc \cdot ut)$
Massa específica	ρ	7850	um/uc^3
Calor específico	c_p	435	$ue/(um \cdot ut)$

Para realizar a análise computacional desse problema utilizando o INSANE, elaborou-se um modelo axissimétrico contemplando uma seção do cilindro, como indicado esquematicamente na 5.12. Essa seção foi subdividida ao longo da direção radial em 100 elementos finitos quadrilaterais de nove nós, totalizando 603 graus de liberdade, como ilustrado na Figura 5.13. O comprimento de cilindro modelado foi escolhido de forma que a base e a altura dos elementos finitos fosse igual. Reforça-se

que o calor flui somente na direção radial.

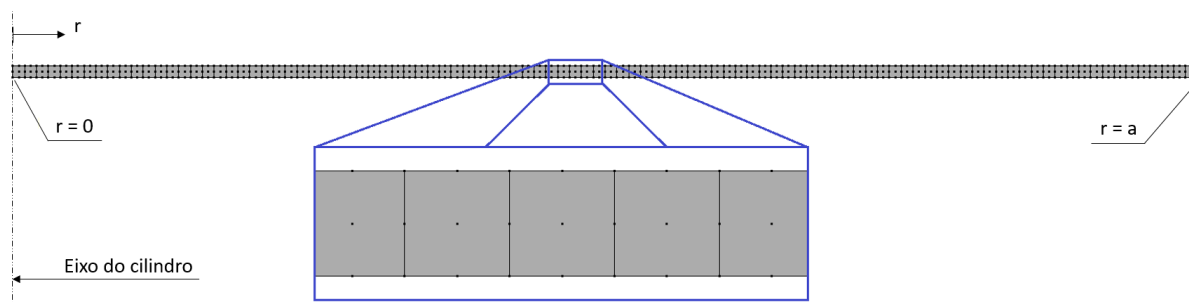


Figura 5.13: Discretização da seção axissimétrica modelada no Exemplo 4.

Quanto às classes empregadas para a definição do comportamento das entidades envolvidas no processo de modelagem e de análise, ressalta-se a utilização da classe de solução `NonLinearHeatTransferNewmarkBeta` e de uma classe que confere dependência das propriedades física do material em relação à temperatura segundo uma função polinomial, `PolynomialTempDepThermalIsotropicMaterial`. A condição inicial do problema foi especificada utilizando a nova metodologia, proposta nesse trabalho.

A análise computacional foi realizada até o instante de tempo 10000 ute em passos de tempo de $0,1 \text{ ute}$. Quanto ao método de integração, utilizou-se o método Newmark- β com o parâmetro $\beta = 0,5$. A tolerância definida para a verificação de convergência ao longo do processo iterativo foi de 1×10^{-12} . Para o cálculo da solução de referência, foram utilizados 5000 termos na série empregada no cálculo da temperatura transformada, Equação B.78 apresentada no Apêndice B.

A evolução temporal da temperatura na superfície do cilindro, calculada tanto analiticamente como numericamente utilizando o INSANE, é ilustrada no gráfico comparativo da Figura 5.14. Observa-se uma boa correspondência entre os resultados, ocorrendo pequenas diferenças atribuídas ao fato da formulação analítica disponível requerer que a difusividade térmica seja constante, mesmo que as propriedades condutividade térmica, calor específico e massa específica variem com a temperatura. Como na implementação atual do INSANE somente é possível variar a

condutividade térmica com a temperatura, tornou-se necessário violar essa premissa. Visto que a não-linearidade introduzida é baixa, as soluções não se distanciaram significativamente.

Todavia, considera-se que os resultados aqui apresentados são suficientes para validarem os algoritmos implementados no INSANE para a solução de problemas de transferência de calor fisicamente não-lineares em regime transiente.

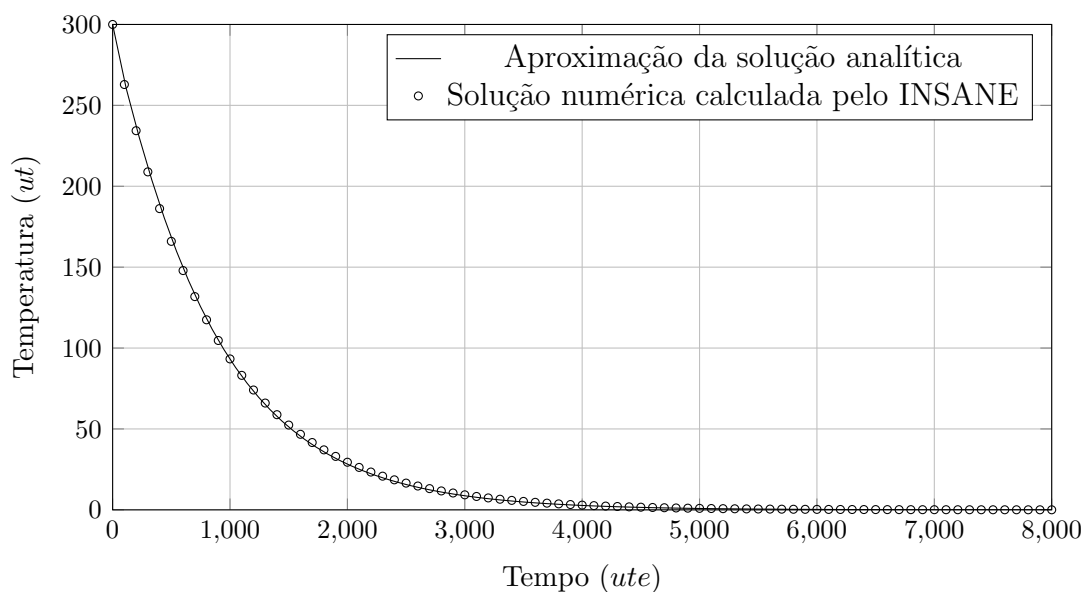


Figura 5.14: Comparativo entre as soluções analítica e numérica do Exemplo 4.

5.2 Problemas Termomecânicos

5.2.1 Exemplo 5 - Viga bi-apoiada com geração de calor

O primeiro exemplo abordado para a validação da implementação dos recursos para a solução de problemas termomecânicos é uma viga bi-apoiada submetida à geração de calor em seu domínio. Essa viga possui comprimento L e seção retangular constante de largura b e altura a . Além do calor \dot{q} gerado uniformemente no domínio da viga, a temperatura é prescrita nas suas faces superior e inferior, sendo

a temperatura na face superior referenciada por T_{sup} e a temperatura na face inferior por T_{inf} . A origem do sistema de coordenadas é disposta no centróide da face esquerda da viga, ilustrada na Figura 5.15.

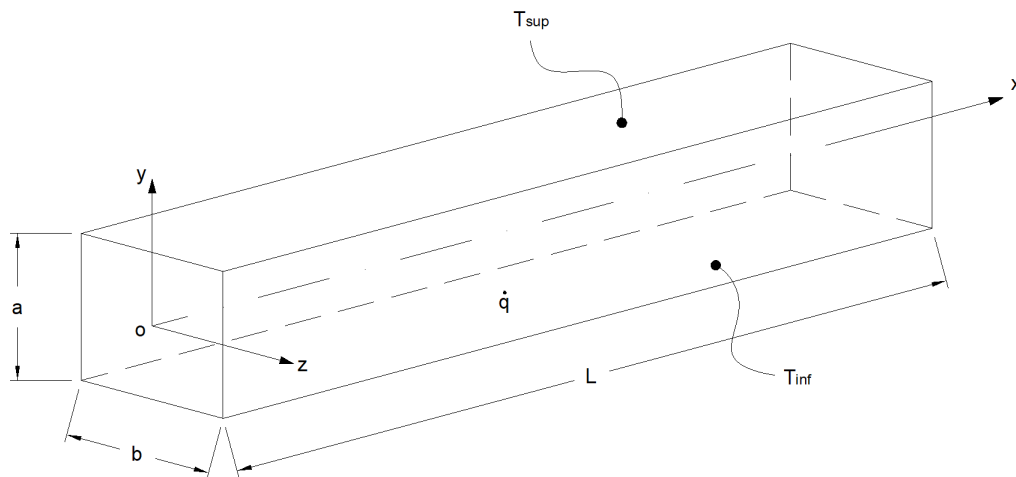


Figura 5.15: Viga submetida a geração de calor e temperaturas prescritas - Ex. 5.

A solução analítica desse problema é apresentada por Hetnarski e Eslami (2009), e seu desenvolvimento é mostrado em detalhes no Apêndice B.

Para as referidas condições de contorno, o fluxo de calor ocorre somente na direção y , e ao solucionar o problema obtém-se a seguinte expressão para o campo de temperaturas:

$$T(y) = -\frac{q}{2k}y^2 + \frac{T_{sup} - T_{inf}}{a}y + \frac{T_{sup} + T_{inf}}{2} + \frac{qa^2}{8k} \quad (5.10)$$

onde k é a condutividade térmica do material, considerada independente da temperatura buscada.

Pode-se notar que o campo de temperaturas descrito pela Equação 5.10 pode ser decomposto em uma parcela antissimétrica em relação ao plano neutro, que varia linearmente ao longo da altura da viga, e uma parcela simétrica em relação a esse mesmo plano, que possui uma variação quadrática ao longo da altura. Considerando a teoria clássica de vigas de Bernoulli-Euler em um regime linear de pequenos

deslocamentos e pequenas deformações, o princípio da superposição pode ser aplicado e os efeitos dessas duas parcelas do campo de temperaturas pode ser estudado separadamente.

A parcela antissimétrica do campo de temperaturas ocasiona flexão pura da viga sem introduzir tensões axiais, já que o problema é isostático e o gradiente de temperaturas é uniforme ao longo da altura. A parcela simétrica, por sua vez, introduz tensões na direção axial visto que ocasiona um gradiente de temperaturas não uniforme.

Por se basear nas hipóteses de Bernoulli-Euler, a solução obtida é uma aproximação da resposta real da estrutura. Nessa teoria, assume-se que as seções transversais permanecem planas e não se deformam. Como resultado dessa aproximação, pode-se representar o comportamento cinemático do corpo pelo campo de deslocamentos da linha elástica, que nesse caso é o eixo paralelo ao comprimento da viga e que passa pelos centróides das seções transversais, reduzindo a análise a um modelo teórico unidimensional.

No problema em discussão nessa seção, as seções transversais desse modelo teórico podem apresentar os movimentos de corpo rígido nas direções axial e vetical, x e y respectivamente, e rotacionarem em torno de eixos paralelos ao eixo z que passem pelo centróide da seção. Além disso, o campo de temperaturas não varia ao longo do comprimento da viga, e sua variação no interior das seções transversais é incorporada na análise pela introdução do esforços generalizados equivalentes P_T e M_{Tz} :

$$P_T = \int_A E\gamma T(y, z) dA \quad (5.11)$$

$$M_{Tz} = \int_A E\gamma T(y, z) y dA \quad (5.12)$$

onde A é a área da seção transversal, γ é o coeficiente de expansão térmica do material e E é o módulo de elasticidade.

Integrando a equação da linha elástica levando em conta o esforço generalizado de flexão M_{Tz} e as condições de contorno de apoios simples, obtém-se a seguinte expressão para a flecha da viga ao longo do comprimento:

$$v(x) = \frac{\gamma(T_{sup} - T_{inf})}{2a} (xL - x^2) \quad (5.13)$$

As tensões normais, decorrentes da interação entre o esforço axial generalizado P_T e do esforço de flexão generalizado M_{Tz} podem ser expressas por:

$$\sigma(y) = E\gamma \frac{\dot{q}a^2}{8k} \left[\left(\frac{2y}{a} \right)^2 - \frac{1}{3} \right] \quad (5.14)$$

Como no INSANE não existem recursos no presente momento para introduzir em um modelo unidimensional os efeitos da variação de temperatura ao longo da seção transversal quando essa variação de temperaturas for arbitrária, torna-se necessário modelar o problema de forma bidimensional ou tridimensional.

Visto que não existem gradientes de temperatura na direção z , opta-se pelo modelo bidimensional elaborado sobre o plano xy devido ao seu menor custo computacional em relação ao modelo tridimensional. O comportamento desse modelo segue as premissas de estado plano de tensões. Para compatibilizar as considerações da modelagem numérica com a formulação analítica, desprezou-se o Efeito de Poisson e as deformações térmicas na direção y , paralela à altura da viga. Para tal, o valor zero foi atribuído ao coeficiente de Poisson, parâmetro do material utilizado na análise mecânica.

Quanto à deformação na direção y associada à variação de temperatura, desconsiderou-se a sua contribuição durante os cálculos numéricos.

Como tanto o campo de deslocamentos como o campo de temperaturas são simétricos em relação ao meio do vão, pode-se discretizar somente metade do domínio e introduzir uma condição de contorno que represente o comportamento no plano de simetria. Nesse plano, os pontos somente podem se deslocar verticalmente, e por isso foram introduzidas restrições ao movimento horizontal.

A figura 5.16 ilustra o domínio discretizado, correspondente à metade esquerda do domínio completo, contendo as condições de contorno levando em consideração a simetria em $x = L/2$. Nessa discretização, a porção modelada do comprimento da viga foi subdividida em 50 partes iguais e a altura da viga foi subdividida em 10 partes iguais. Foram utilizados elementos finitos quadriláteros de nove nós, onde cada nó possui um grau de liberdade no problema térmico e dois graus de liberdade no problema mecânico, que são o deslocamento na direção horizontal e o deslocamento na direção vertical. Dessa forma, o problema discreto possui 2121 graus de liberdade no PTC, onde 1919 são desconhecidos, e 4242 graus de liberdade no PTM, dos quais 22 são conhecidos e os demais são calculados durante a solução do problema.

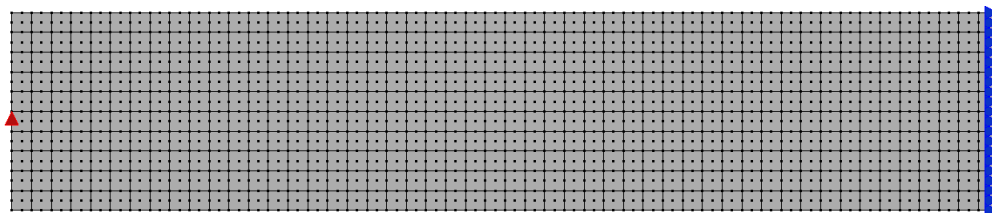


Figura 5.16: Discretização da viga do exemplo 5 contendo as condições de contorno da parcela mecânica.

Para permitir a comparação entre os resultados calculados analiticamente e numericamente, são atribuídos os valores descritos na Tabela 5.7 aos parâmetros que regem o problema.

No que se refere às classes do núcleo numérico utilizadas na análise computacional do exemplo em questão, ressalta-se o emprego das classes com comportamento multifísico para as entidades *problem driver*, *analysis model*, *constitutive model*, *material* e *solution*. As condições de contorno essenciais foram definidas utilizando a nova abordagem, o que é necessário nos problemas termomecânicos, visto que o mesmo modelo discreto é utilizado em ambas as físicas. Em relação ao *simulation manager*, utilizou-se a classe `SteadyStateThermoMecSimManager` trabalhando em

conjunto com um *setuper* da classe `SharedModelSteadyStateTermoMecSetuper`.

Tabela 5.7: Parâmetros considerados na modelagem do Exemplo 5.

Parâmetro	Símbolo	Valor	Unidade
Comprimento da viga	L	10	uc
Altura da seção viga	a	1	uc
Largura da seção viga	b	0,1	uc
Temperatura na face inferior	T_{inf}	100	ut
Temperatura na face superior	T_{sup}	-100	ut
Taxa de geração de calor	\dot{q}	60000	up/uc^3
Condutividade térmica	k	60	$up/(uc \cdot ut)$
Módulo de elasticidade	E	210×10^9	$uf/(ua^2)$
Coefficiente de Poisson	ν	0	adimensional
Coefficiente de expansão térmica	γ	$1,2 \times 10^{-6}$	$uc/(ucut)$

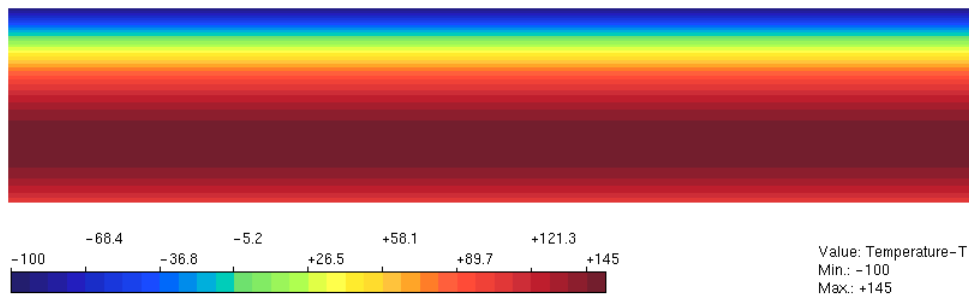


Figura 5.17: Distribuição de temperaturas calculada pelo INSANE - Exemplo 5.

O campo de temperaturas calculado pelo INSANE é mostrado na Figura 5.17, e um gráfico comparativo do perfil de temperaturas ao longo da altura da viga, calculado de forma analítica e de forma numérica pelo INSANE é mostrado na Figura 5.18.

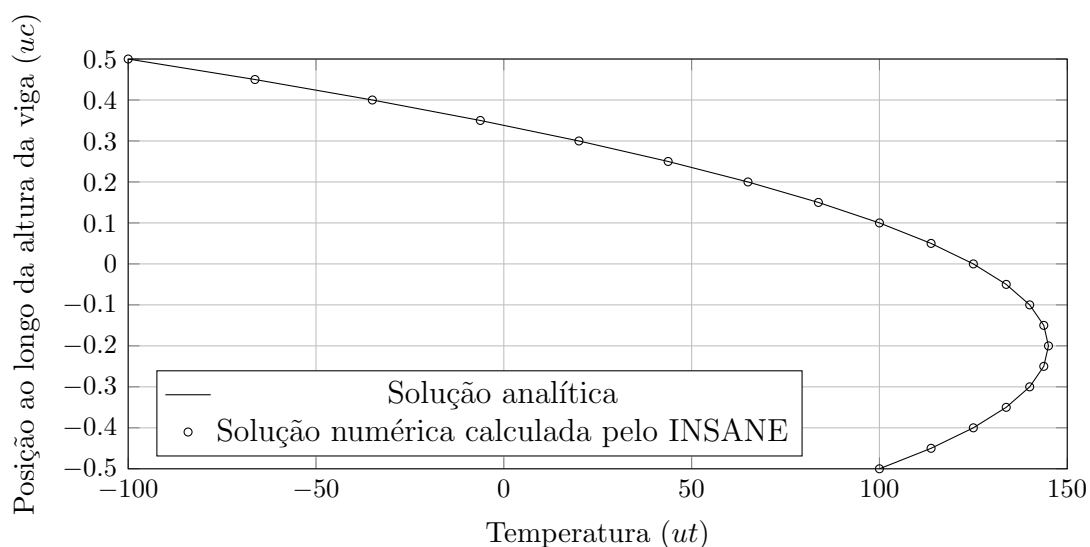


Figura 5.18: Comparativo entre os campos de temperatura analítico e numérico.

O campo de deslocamentos calculado numericamente pelo INSANE e a forma deformada da estrutura são mostrados na Figura 5.19. Observa-se a ocorrência da flexão pura introduzida pela parte antissimétrica do campo de temperaturas e a ocorrência de um aumento de comprimento da viga, dado pela expansão térmica ocasionada pela parte simétrica do campo de temperaturas.

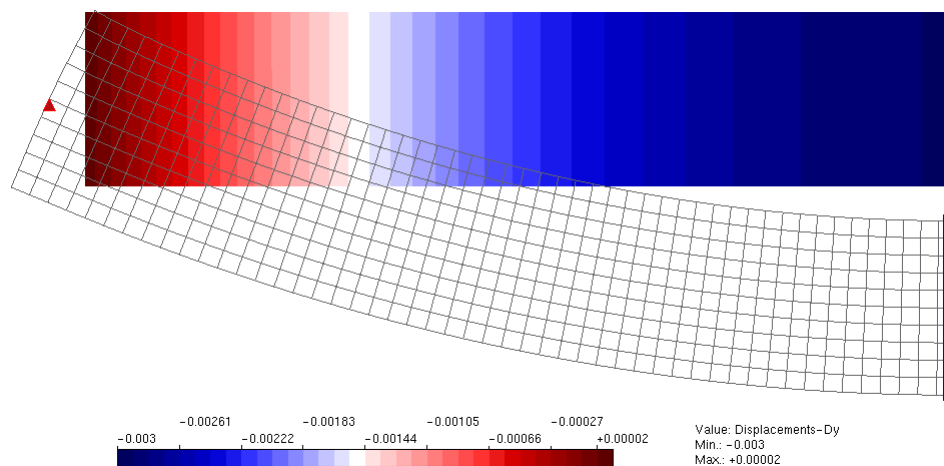


Figura 5.19: Campo de deslocamentos na direção Y calculado pelo INSANE - Ex. 5.

No gráfico da Figura 5.20, realiza-se a comparação entre a flecha ao longo do comprimento da viga calculada analítica e numericamente.

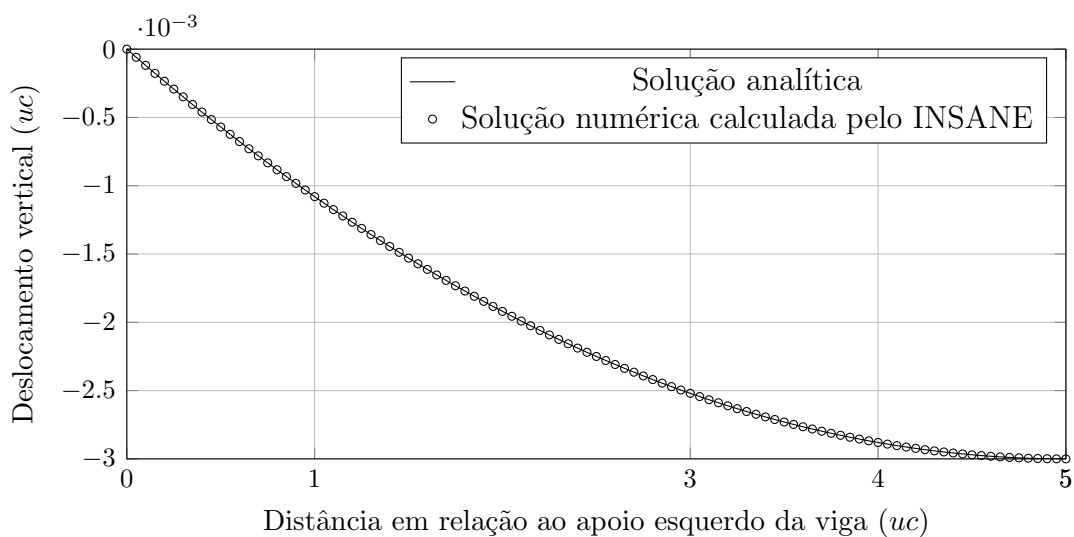


Figura 5.20: Comparativo entre as linhas elásticas analítica e numérica.

Quanto ao campo de tensões na direção axial, o mesmo pode ser visualizado na Figura 5.21. Observa-se que próximo ao apoio há uma região de transição do campo de tensões para atender à condição de contorno de forças de superfícies normais nulas na face esquerda. Esse tipo de comportamento é esperado e não pode ser introduzido na modelagem analítica unidimensional descrita pela teoria de Bernoulli-Euler. Na Figura 5.22, é mostrado um gráfico comparando o valor da tensão normal na direção x ao longo da altura da viga. Nesse gráfico, os valores são calculados na seção $x = 4.95u.c.$, próxima ao centro do vão da viga.

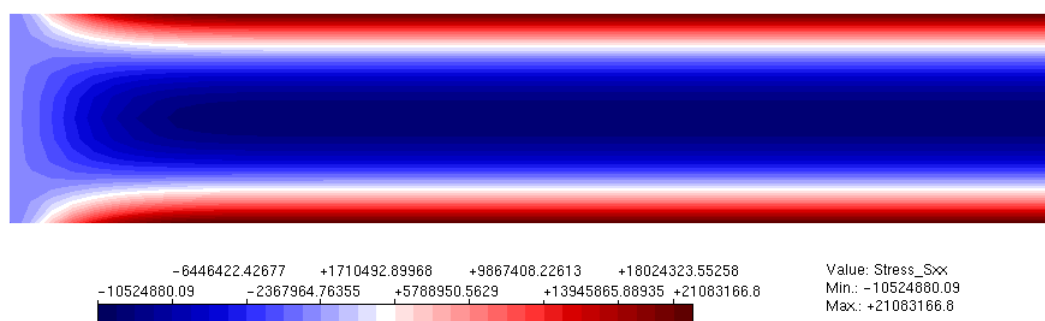


Figura 5.21: Tensões normais na direção X calculadas pelo INSANE - Ex. 5.

Pode-se observar nos resultados obtidos para a distribuição de tensões que distante das bordas os resultados analíticos e os resultados numéricos se igualam, o

que valida a implementação. A divergência nos resultados observada na região próxima ao apoio da viga é esperada e não compromete a validação da implementação realizada.

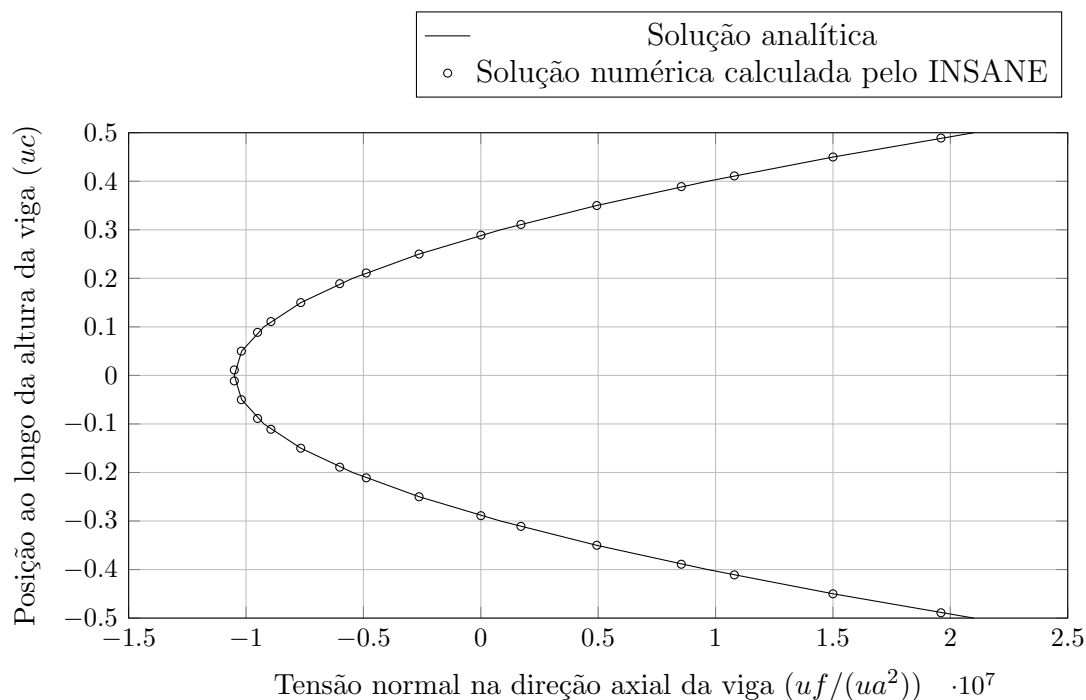


Figura 5.22: Comparativo entre os campos de tensão normal na direção axial.

5.2.2 Exemplo 6 - Barra bi-engastada em aquecimento uniforme

O exemplo apresentado nessa sub-seção tem como objetivo ilustrar a utilização dos recursos introduzidos no INSANE para a realização de análises termomecânicas fisicamente não-lineares em regime permanente. A não-linearidade abordada nesse exemplo é o comportamento mecânico plástico do material, situação essa que ocorre quando as tensões excedem o limite de escoamento. A análise térmica é linear por simplicidade, mas reforça-se que a mesma poderia ser não-linear.

Escolhe-se como objeto de estudo a barra ilustrada na Figura 5.23, que parte de uma distribuição de temperaturas inicial nula, e após esse instante sofre a imposição de temperaturas iguais em suas extremidades com o valor θ_f . Esse problema de

valor de contorno é complementado pela definição de sua parcela mecânica, na qual as extremidades da barra são impedidas de se deslocarem.

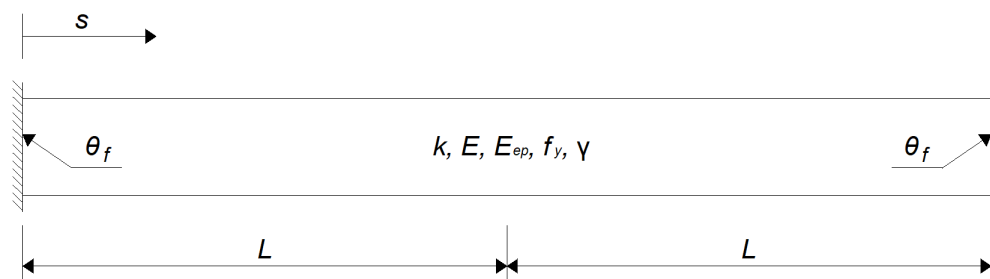


Figura 5.23: Barra bi-engastada sob aquecimento uniforme - Exemplo 6.

O campo de temperaturas na barra pode ser facilmente obtido da integração da Equação 2.19, que corresponde à equação da condução do calor, seguida da imposição das condições de contorno de temperaturas prescritas, o que leva a uma distribuição de temperaturas uniforme com valor θ_f ao longo de toda a barra visto que não há geração de energia e as condições de contorno são iguais em ambas as extremidades.

Como desenvolvido em detalhes no Apêndice B, essas condições de contorno levam a um campo de deslocamentos totais nulo ao longo de todo o domínio. Por sua vez, o campo de deformações mecânicas ϵ_{mec} é dado pela expressão:

$$\epsilon_{mec}(s) = -\gamma\theta_f \quad (5.15)$$

onde γ é o coeficiente de expansão térmica.

Esse campo de deformações, que é constante ao longo de todo o comprimento da barra, causa o aparecimento de tensões de compressão que podem ser calculadas conforme o modelo constitutivo empregado. Nesse exemplo, adota-se o modelo constitutivo de plasticidade unidimensional, detalhadamente discutido em Souza Neto et al. (2009). Adicionalmente, utilizou-se nos cálculos a relação tensão-deformação representada pela curva bi-linear da Figura 5.24, onde a transição do regime elástico para o regime plástico ocorre na tensão de escoamento f_y .

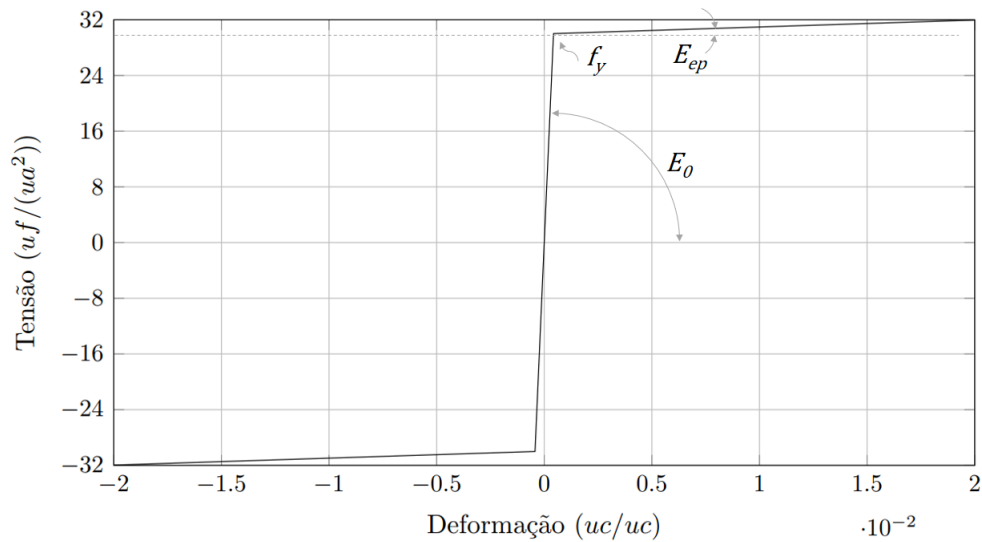


Figura 5.24: Curva tensão-deformação para a barra do Exemplo 6.

Devido à ausência de carregamentos mecânicos externos, o problema é conduzido pela variação da temperatura no corpo, que pode ser considerada previamente conhecida na análise mecânica, o que simplifica o processo de solução. Nesse caso particular, pode-se facilmente determinar se o corpo se encontra em seu regime elástico ou em seu regime plástico. A seguir, são apresentadas de forma resumida as equações utilizadas para o cálculo analítico da tensão atuante nos regimes elástico e plástico.

Primeiro caso: Deformação atuante inferior ou igual, em módulo, à deformação associada ao escoamento, ϵ_y .

$$\epsilon_y = \frac{f_y}{E_0} \quad (5.16)$$

onde f_y é a tensão de escoamento e E_0 é o módulo de elasticidade no regime elástico.

Assim, se $|\epsilon_{mec}| \leq \epsilon_y$, tem-se:

$$\sigma = E_0 \epsilon_{mec} \quad (5.17)$$

sendo σ a tensão causada pelo impedimento à variação da temperatura.

Segundo caso: Deformação atuante superior, em módulo, à deformação associada ao escoamento.

$$\sigma = sig(\epsilon_{mec})[f_y + E_{ep}(|\epsilon_{mec}| - \epsilon_y)] \quad (5.18)$$

onde $sig(\epsilon_{mec})$ é o sinal da deformação mecânica atuante e E_{ep} é o módulo elasto-plástico do material.

Para permitir a comparação entre os resultados analíticos e os calculados numericamente, são adotados os parâmetros descritos na Tabela 5.8.

Tabela 5.8: Parâmetros considerados na modelagem do Exemplo 6.

Parâmetro	Símbolo	Valor	Unidade
Comprimento da barra	$2L$	1	uc
Temperatura inicial	T_0	0	ut
Temperatura prescrita nas bordas	θ	50	ut
Condutividade térmica	k	237	$up/(uc \cdot ut)$
Módulo de elasticidade no regime elástico	E_0	70000	$uf/(ua^2)$
Coefficiente de expansão térmica	γ	$23,1 \times 10^{-6}$	$uc/(uc \cdot ut)$
Tensão de escoamento	f_y	30	$uf/(ua^2)$
Módulo de elasticidade elasto-plástico	E_{ep}	100	$uf/(ua^2)$
Área da seção transversal da barra	A	1	uc^2

Como no regime permanente os campos de temperatura e de deslocamentos são uniformes ao longo de todo o comprimento da barra, pode-se simplificar a discretização espacial. Para a análise discutida nessa seção, dividiu-se a barra em dois elementos finitos unidimensionais de dois nós, resultando em um problema de três graus de liberdade onde dois são prescritos ao se introduzir as condições de contorno.

Quanto às classes empregadas na modelagem do problema, ressalta-se a utilização de entidades multifísicas em uma estrutura de dados compartilhada entre as

físicas térmica e mecânica. Essa estrutura de dados é descrita pela classe `SharedModelThermoMecDataManager`, que é instanciada automaticamente pelo construtor do gerenciador da solução, que é do tipo `SteadyStateThermoMecSimManager`. O problema térmico é resolvido utilizando a classe convencional de solução de problemas em regime permanente, classe `SteadyState`, enquanto a parcela mecânica é solucionada pela nova classe de solução `QuasiStaticEquilibriumPath` trabalhando em conjunto com um objeto do tipo `ThermoMecSNR`, que implementa a interface `Step`.

Nas análises numéricas não-lineares realizadas no INSANE que seguem o procedimento de cálculo incremental-iterativo descrito por Yang e Shieh (1990), o carregamento total é fracionado e inserido de forma gradual a cada passo de carga. Nesse exemplo, o carregamento a ser inserido gradualmente é o carregamento nodal equivalente decorrente da variação de temperatura, calculado numericamente pelo MEF segundo a Equação 3.123. Esse carregamento foi aplicado em 1000 passos de carga, onde a cada passo há um incremento de um milésimo da carga total.

O gráfico na Figura 5.25 ilustra os resultados obtidos, onde a cada passo de carga é introduzido no gráfico um ponto contendo a deformação e a tensão, calculadas numericamente para o passo de carga corrente.

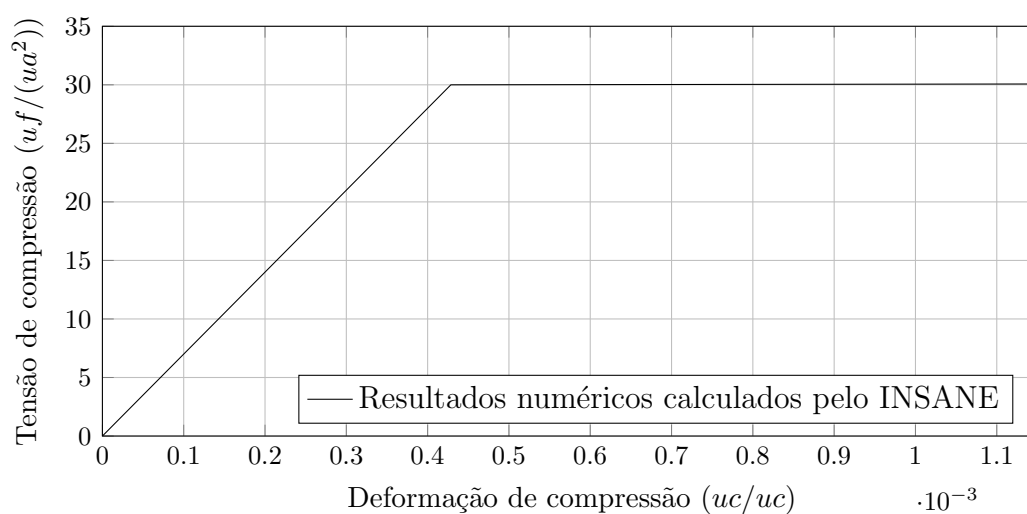


Figura 5.25: Relação entre a tensão e a deformação ao longo da análise do Exemplo 6. Valores em módulo.

A tensão de compressão atuante no regime plástico pode ser melhor visualizada no gráfico da Figura 5.26, que é uma ampliação do gráfico anterior.

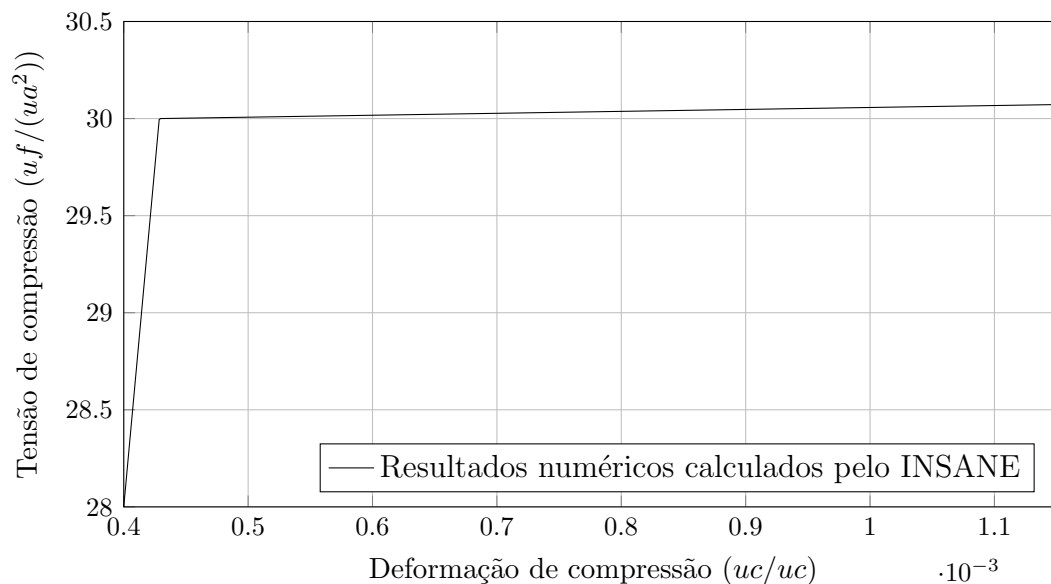


Figura 5.26: Ampliação da região em regime plástico.

Uma análise detalhada indicou que o escoamento ocorre entre os passos de carga 371 e 372. Comparativos entre os valores de deformação e de tensão obtidos pelo INSANE nesses passos de carga e no último passo, correspondente à aplicação completa do carregamento de variação de temperatura, são mostrados nas tabelas a seguir.

Tabela 5.9: Comparação entre as deformações calculadas analítica e numericamente.

Passo	Deformação (uc/uc)	
	Calculada analiticamente	Calculada pelo INSANE
371	-0.000428505	-0.000428505
372	-0.000429660	-0.000429660
1000	-0.001155000	-0.001155000

Tabela 5.10: Comparação entre as tensões calculadas analítica e numericamente.

Passo	Tensão ($uf/(ua^2)$)	
	Calculada analiticamente	Calculada pelo INSANE
371	-29.99535000	-29.99535000
372	-30.00010886	-30.00010886
1000	-30.07264286	-30.07264286

Pode-se observar nas tabelas 5.9 e 5.10 que os resultados numéricos calculados pelo INSANE se igualaram aos resultados calculados analiticamente, o que valida o gerenciador da solução quando o problema é fisicamente não-linear em regime permanente. Esse gerenciador cria automaticamente os carregamentos de variação de temperatura e os introduz na parcela mecânica do problema. Também são validadas com esse exemplo as classes que apresentam comportamento dependente da física corrente e o algoritmo para a solução de problemas mecânicos incrementais-iterativos na presença de variações de temperatura.

Deve ser realizada futuramente uma adequação no pós-processamento do INSANE para que ele seja capaz de reconhecer as classes que possuem comportamento multifísico. No presente momento, os arquivos de saída de dados estão sendo pós-processados para substituir os rótulos das classes multifísicas pelas suas respectivas classes com comportamento térmico ou comportamento mecânico, dependendo da física à qual o arquivo de saída de dados se refere.

5.2.3 Exemplo 7 - Viga submetida a fluxo de calor prescrito em sua face superior

Nesse exemplo também se estuda o comportamento termomecânico de uma viga bi-apoiada, porém as condições de contorno associadas ao PTC diferem das consideradas na Seção 5.2.1, referentes ao Exemplo 5. Além dessa diferença nas condições de contorno, o comportamento é estudado no regime transiente, e não somente em um

instante de tempo suficientemente distante do inicial, como realizado anteriormente no Exemplo 5.

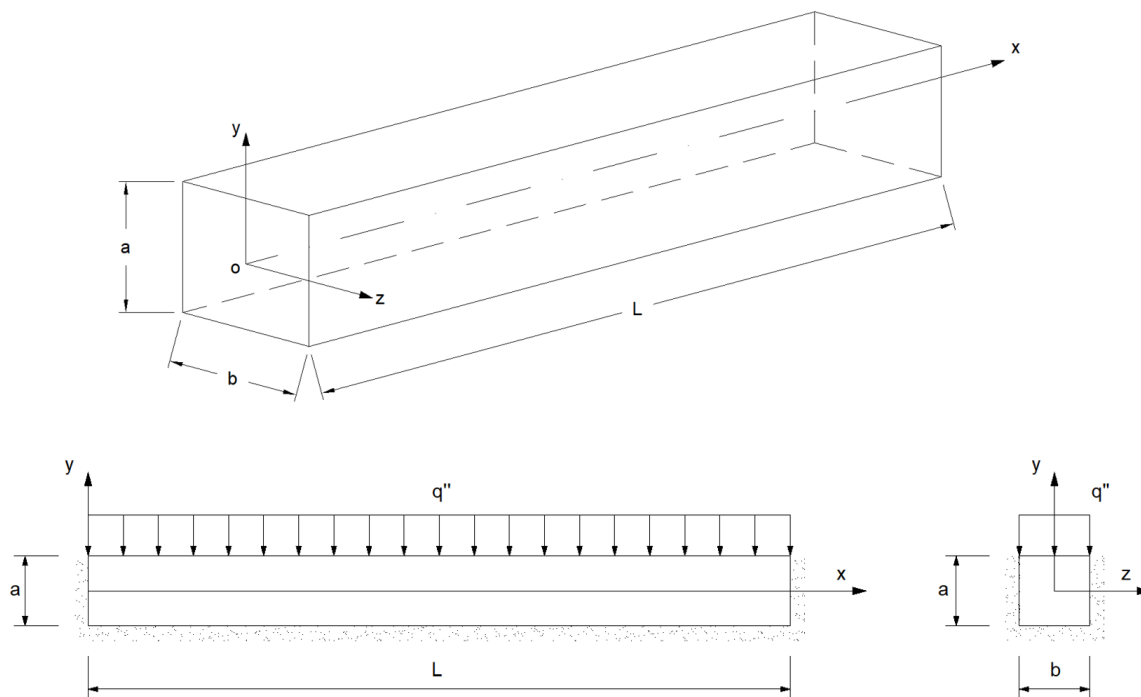


Figura 5.27: Viga submetida a fluxo de calor na face superior - Exemplo 7.

A viga aqui abordada possui comprimento total igual a L , seção transversal de altura a e largura b . A condutividade térmica do material é dada por k , a difusividade térmica α , o módulo de elasticidade E e o coeficiente de expansão térmica γ . A viga está impedida de trocar calor por todas as suas faces, exceto pela sua face superior, que está submetida a um fluxo de calor prescrito q'' . A origem do sistema de coordenadas é posicionada no centro da face esquerda da viga, como ilustrado na Figura 5.27, e a viga parte de uma distribuição de temperaturas uniforme à temperatura $T_0 = 0$.

A solução analítica desse problema foi obtida de Hetnarski e Eslami (2009) e seu desenvolvimento é mostrado em detalhes no Apêndice B dessa dissertação de mestrado.

Do ponto de vista do problema de transferência de calor, o comportamento é

unidimensional, no qual o calor somente flui na direção paralela à altura da viga. A distribuição de temperaturas ao longo do tempo $T(y, t)$ é expressa por:

$$T(y, t) = T_0 + \frac{aq''}{24k} \left[\frac{24\alpha t}{a^2} + \frac{12y^2}{a^2} + \frac{12y}{a} - 1 - \frac{48}{\pi^2} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{(-n^2\pi^2\alpha t/a^2)} \cos \frac{n\pi(2y+a)}{2a} \right] \quad (5.19)$$

Observa-se que a Equação 5.19 possui dois termos que variam ao longo do tempo, sendo que um deles desaparece após um tempo suficiente grande após o instante inicial e o outro não. O termo que contribui somente no início da análise, contido no somatório da Equação 5.19, intensifica o gradiente de temperaturas na direção vertical, enquanto o termo dependente do tempo que permanece ao longo de toda a análise, fora do somatório, ocasiona uma variação uniforme da temperatura em todo o corpo.

Em um instante de tempo suficientemente distante do inicial, a distribuição de temperaturas assume uma variação parabólica ao longo da altura da viga, cujos valores crescem indefinidamente, porém mantendo a mesma função gradiente de temperaturas ao longo da direção y .

Conforme detalhado por Hetnarski e Eslami (2009), o efeito da variação da temperatura ao longo da seção transversal da viga pode ser representado por um esforço normal P_T e um momento fletor M_{Tz} . Esses esforços são calculados a partir da integração, ao longo da superfície da seção transversal, da deformação térmica multiplicada pelo operador constitutivo. Tal operação leva aos esforços:

$$P_T(t) = \frac{E\gamma q'' A\alpha t}{ka} \quad (5.20)$$

$$M_{Tz}(t) = E\gamma \frac{q'' I_z}{2k} \left(1 - \frac{96}{\pi^4} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n^4} e^{(-n^2\pi^2\alpha t/a^2)} \right) \quad (5.21)$$

onde A é a área da seção transversal da seção e I_z é o momento de inércia da seção em relação ao eixo z .

Os esforços equivalentes descritos pelas equações 5.20 e 5.21 ocasionam tanto a flexão da viga como uma expansão na direção axial. O efeito combinado desses esforços introduz tensões normais na direção axial. Ressalta-se que a viga não está submetida a nenhum esforço mecânico externo.

A distribuição de tensões normais na direção axial, $\sigma_{xx}(y, t)$, é dada pela equação a seguir:

$$\sigma_{xx}(y, t) = E\gamma \left[\left(\frac{1}{12} - \frac{y^2}{a^2} \right) + \frac{4}{\pi^2} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{(-n^2\pi^2\alpha t/a^2)} \cos \frac{n\pi(2y+a)}{2a} - \frac{96}{\pi^4} \sum_{n=1,3,5\dots}^{\infty} \frac{e^{(-n^2\pi^2\alpha t/a^2)}}{n^4} \right] \quad (5.22)$$

No que se refere à variação temporal da distribuição de tensões, nota-se que após um tempo suficiente grande ela se torna uma distribuição parabólica que não mais se modifica com o tempo. A distribuição de tensões se estabiliza porquê a variação da temperatura, em instantes de tempo distantes do inicial, ocorre de maneira uniforme no corpo, o que ocasiona apenas o aumento de comprimento da viga sem a introdução de tensões, visto que a mesma encontra-se livre para expandir ou contrair.

Assim como no Exemplo 5, desenvolveu-se um modelo numérico bidimensional no qual a viga é representada no plano xy segundo um modelo de análise de estado plano de tensões. Novamente, o Efeito de Poisson e a expansão térmica na direção paralela à altura da viga foram desprezados a fim de viabilizar a comparação com o modelo analítico, que carrega essas hipóteses na sua formulação.

A simetria do problema em relação ao centro do vão foi utilizada para reduzir o custo computacional da solução numérica. Dessa forma, modelou-se somente a metade esquerda da viga e se impôs uma condição de contorno que impede o deslocamento horizontal dos pontos pertencentes ao plano de simetria.

Essa região modelada foi discretizada em 500 elementos finitos quadriláterais de nove nós, resultando em 2121 nós. Com essa discretização, o problema de transferência de calor possui 2121 graus de liberdade, onde todos são desconhecidos, uma vez

que não existem condições de contorno essenciais associadas a essa física. Em relação ao problema mecânico, existem 4242 graus de liberdade, onde 22 são conhecidos e os demais são calculados durante a análise.

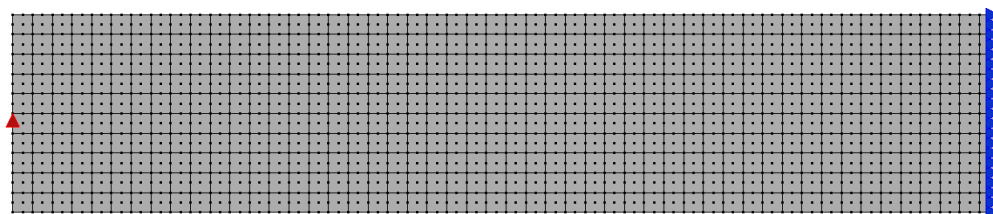


Figura 5.28: Discretização da viga do exemplo 7 contendo as condições de contorno da parcela mecânica.

Para permitir a comparação entre os resultados calculados analiticamente e numericamente, são atribuídos os valores descritos na Tabela 5.11 aos parâmetros que regem o problema.

Tabela 5.11: Parâmetros considerados na modelagem do Exemplo 7.

Parâmetro	Símbolo	Valor	Unidade
Comprimento da viga	L	5	uc
Altura da seção viga	a	0,5	uc
Largura da seção viga	b	0,2	uc
Fluxo de calor na face superior	q''	850	up/uc^2
Condutividade térmica	k	401	$up/(uc \cdot ut)$
Massa específica	ρ	8933	um/uc^3
Calor específico	c_p	385	$ue/(um \cdot ut)$
Difusividade térmica	α	$116,6 \times 10^{-6}$	uc^2/ute
Módulo de elasticidade	E	110×10^9	$uf/(ua^2)$
Coefficiente de Poisson	ν	0	adimensional
Coefficiente de expansão térmica	γ	$16,4 \times 10^{-6}$	$uc/(ucut)$

No que se refere às classes do núcleo numérico utilizadas na análise computacional do exemplo em questão, ressalta-se a utilização de classes multifísicas para definir o modelo discreto de referência, que previamente ao início da análise é dividido em

dois modelos independentes pelo `ClonedModelsThermoMecDataManager`, sendo um modelo utilizado exclusivamente para a análise térmica e o outro exclusivamente para a análise mecânica.

Quanto ao *simulation manager*, utilizou-se a classe `TimeDepThermoMecSimManager` trabalhando com o *setuper* `TimeDepThermalSetuper` para a análise térmica e com o *setuper* `SinglePhysicsFemSetuper` na análise mecânica.

No que se refere às classes de solução, o problema térmico dependente do tempo é linear, sendo solucionado pela classe `LinearHeatTransferNewmarkBeta`. Utilizou-se o parâmetro β igual a 0,5 e um passo de tempo igual a 0,1 *ute*. O problema mecânico linear e em regime quasi-estático é solucionado pela classe `LinearQuasiStatic`.

As distribuições de temperaturas a cada 100 *ute*, calculadas analiticamente pela equação 5.19 e numericamente utilizando o INSANE, estão ilustradas no gráfico da Figura 5.29. Durante o cálculo analítico foram utilizados 512 termos nos somatórios. A distribuição de temperaturas obtida para o instante de tempo $t = 600$ *ute* está mostrada na Figura 5.30.

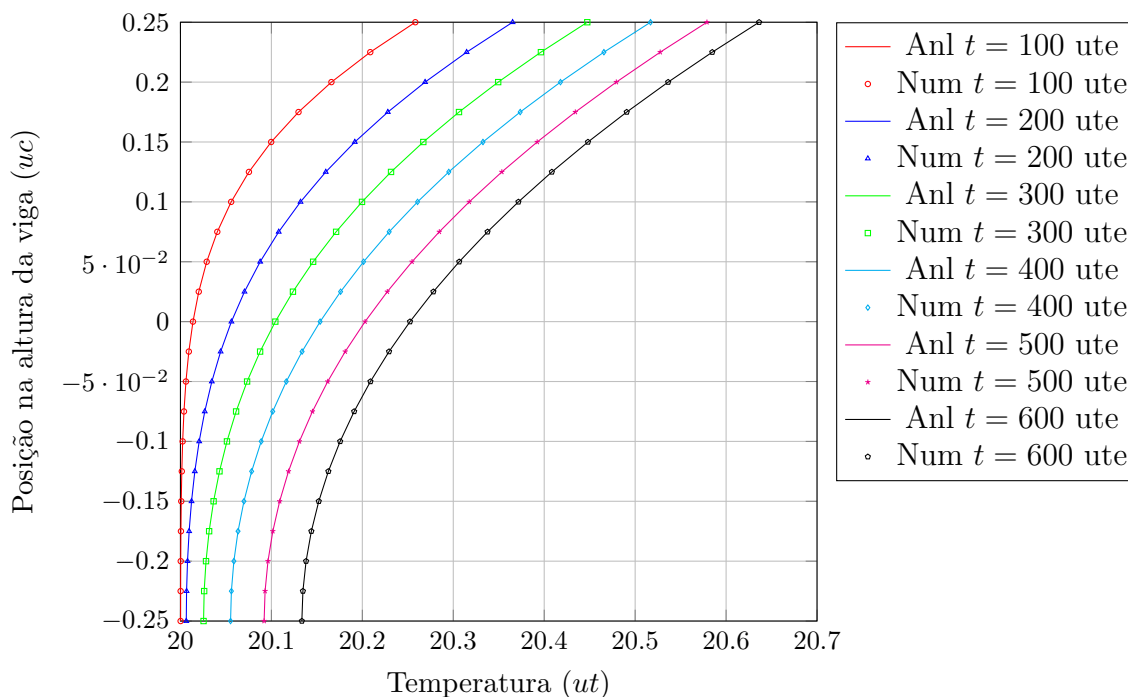


Figura 5.29: Resultados comparativos das distribuições de temperaturas.

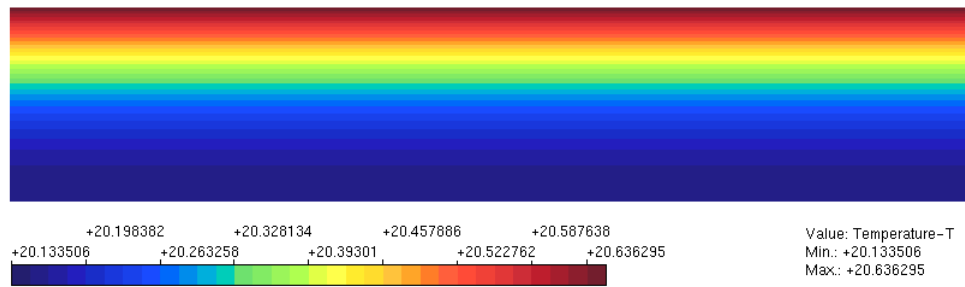


Figura 5.30: Campo de temperaturas calculado pelo INSANE em $t = 600 \text{ ute}$.

Para a distribuição de temperaturas calculada no instante de tempo 600 ute , obtém-se o campo de deslocamentos ilustrado na Figura 5.31, que também contém a forma deformada da viga para esse mesmo instante de tempo. Pode-se observar a ocorrência da flexão e do alongamento da viga ocasionados pelo esforço axial e pelo momento fletor equivalentes associados à variação de temperatura, descritos pelas equações 5.20 e 5.21, respectivamente.

O deslocamento máximo medido na linha de centro da viga ocorre no centro do vão, e pode ser calculado pela expressão:

$$u_{max}(t) = \frac{\gamma q'' L^2}{16k} \left[1 - \frac{96}{\pi^4} \sum_{n=1,3,5\dots}^{\infty} \frac{e^{(-n^2\pi^2\alpha t/a^2)}}{n^4} \right] \quad (5.23)$$

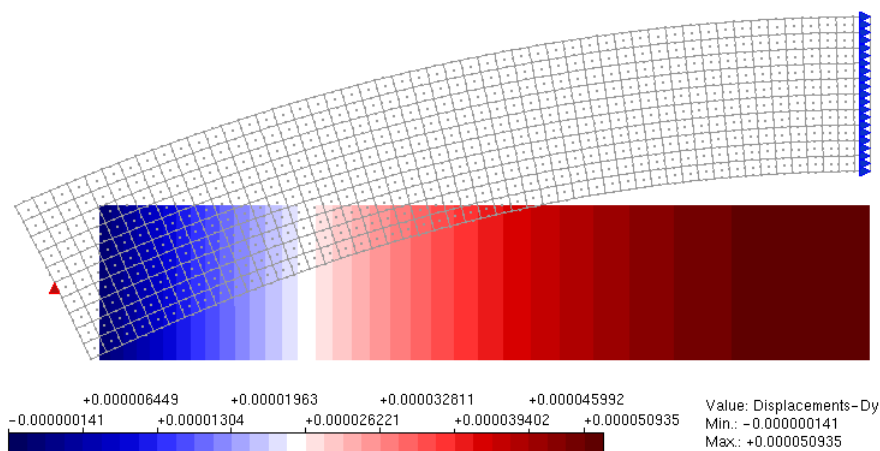


Figura 5.31: Deslocamentos em y calculado pelo INSANE em $t = 600 \text{ ute}$.

Um comparativo entre os deslocamentos máximos medidos na linha de centro, calculados analiticamente pela Equação 5.23 e os deslocamentos máximos calculados numericamente utilizando o INSANE a cada 100 *ute* são mostrados na tabela abaixo.

Tabela 5.12: Comparação entre as flechas máximas calculadas analítica e numericamente.

Instante de tempo (<i>ute</i>)	Flecha máxima (<i>uc</i>)		Erro relativo (%)
	Calculada analiticamente	Calculada pelo INSANE	
100	2.052353533E-05	2.051596160E-05	-0.037
200	3.299683117E-05	3.299199275E-05	-0.015
300	4.086226041E-05	4.085928694E-05	-0.007
400	4.582596031E-05	4.582413641E-05	-0.004
500	4.895850453E-05	4.895738673E-05	-0.002
600	5.093542464E-05	5.093474019E-05	-0.001

As distribuições das tensões normais na direção axial da viga, calculadas analiticamente segundo a Equação 5.22 e numericamente utilizando o INSANE, estão mostrada na Figura 5.32 para os instantes de tempo 100 *ute* e 600 *ute*.

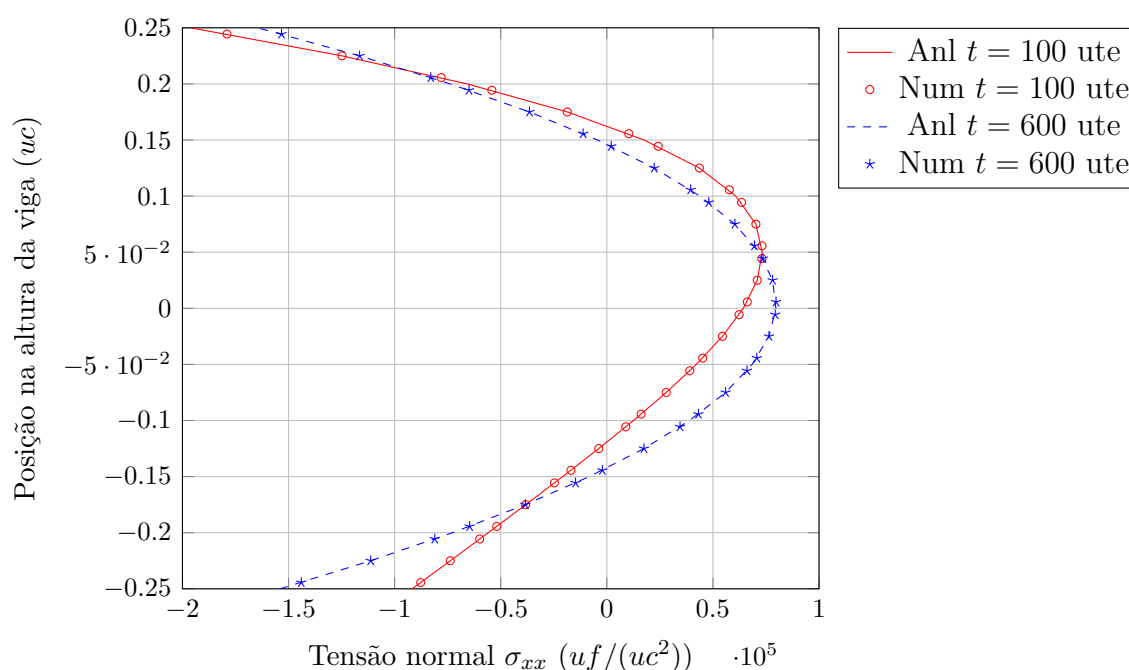


Figura 5.32: Resultados comparativos das distribuições de tensões.

Na Figura 5.33 é mostrado o campo de tensões calculado numericamente pelo INSANE para o instante de tempo 600 *ute*.

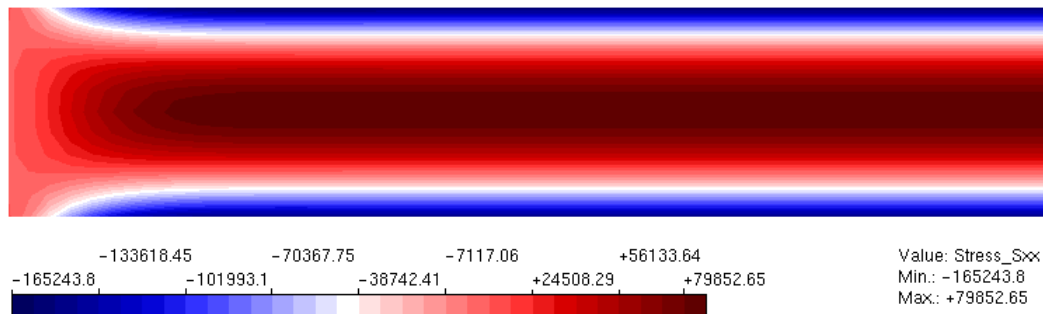


Figura 5.33: Tensões σ_{xx} calculadas pelo INSANE em $t = 600$ *ute*.

Os resultados obtidos validaram a implementação realizada para a solução de problemas termomecânicos lineares em regime transiente. Em especial, nesse exemplo valida-se um caso particular do PTC no qual somente condições de contorno naturais são prescritas sobre o contorno do corpo.

5.2.4 Exemplo 8 - Barra bi-engastada com aquecimento senoidal nas extremidades

O último exemplo discutido nessa dissertação de mestrado se refere à barra cujo comportamento térmico foi abordado no Exemplo 3 e cujas condições de contorno associadas à parcela mecânica do problema foram descritas no Exemplo 6. Essa seção se difere da seção 5.2.2 por estudar a resposta termomecânica no regime transiente, ou seja, no período entre o início da elevação da temperatura nas bordas e um instante de tempo suficientemente distante do momento em que a elevação da temperatura nas bordas se cessa.

A barra ilustrada na Figura 5.34 parte de uma distribuição de temperaturas uniforme e nula, e em seguida é submetida a uma elevação de temperaturas em suas extremidades segundo uma curva senoidal até que a temperatura nesses pontos atinja o valor desejado no instante de tempo denotado por p , indicado na Figura 5.8.

Após atingir o primeiro pico da função senoidal, a temperatura nas bordas é mantida constante até que o regime permanente seja atingido. O restante do contorno da barra está impedido de trocar calor com o ambiente.

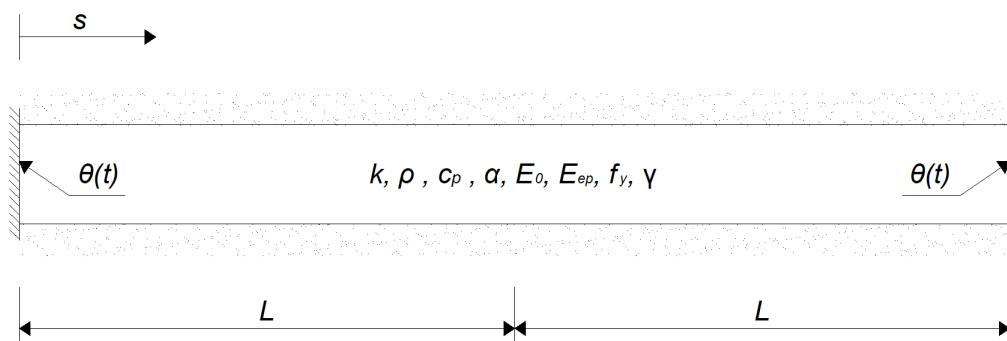


Figura 5.34: Barra bi-engastada sob aquecimento senoidal nas extremidades - Exemplo 8.

A solução da parcela térmica do problema termomecânico abordado nessa seção foi discutida na seção 5.1.3 e a distribuição de temperaturas na barra ao longo do tempo pode ser obtida das equações 5.5, 5.6 e 5.7, onde a primeira se aplica ao período de elevação da temperatura nas bordas e as outras duas ao período em que a temperatura nas bordas é mantida constante.

A atenção nesse momento é direcionada à parcela mecânica. Como as extremidades dessa barra estão impedidas de se deslocar, a variação da temperatura descrita pela solução do PTC ocasiona o surgimento de tensões, que excedem o limite de escoamento do material implicando na sua plastificação, como ilustrado no Exemplo 6 onde foi obtida a resposta desse mesmo problema termoemcânico em regime permanente.

A solução analítica é desenvolvida detalhadamente no Apêndice B. A seguir, serão apresentadas as expressões referentes ao campo de deslocamentos total, às deformações mecânicas e às tensões. As formulações para essas grandezas são apresentadas tanto para o período em que ocorre elevação da temperatura nas extremidades da barra como para o período em que essa temperatura é mantida constante.

O campo de deslocamentos totais é dado pela soma:

$$i_{Tot} = u_T + u_M \quad (5.24)$$

na qual u_T é calculado pela equação abaixo durante o período em que ocorre a elevação das temperaturas, $t \leq p$:

$$u_T(s, t) = s\gamma\theta \operatorname{sen}\left(\frac{\pi t}{2p}\right) + \frac{2\gamma\theta}{p} \sum_{m=1}^{\infty} \left[W(m, t) \left(\frac{\cos(\beta_m L) \operatorname{sen}(\beta_m s)}{\beta_m} \right) \right] + \frac{2\gamma\theta}{p} \sum_{m=1}^{\infty} \left[W(m, t) \left(\frac{\operatorname{sen}(\beta_m L)(1 - \cos(\beta_m s))}{\beta_m} \right) \right] \quad (5.25)$$

sendo

$$W(m, t) = \left[\frac{(-1)^m}{(2m-1)} \right] \cdot \frac{\alpha\beta_m^2 \cos\left(\frac{\pi t}{2p}\right) + \left(\frac{\pi}{2p}\right) \operatorname{sen}\left(\frac{\pi t}{2p}\right) - \alpha\beta_m^2 e^{-\alpha\beta_m^2 t}}{\left[(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2 \right]} \quad (5.26)$$

com o coeficiente β_m apresentado na seção 5.1.3, igual a:

$$\beta_m = (2m-1) \frac{\pi}{2L} \quad (5.27)$$

Quando $p \leq t < \infty$, ou seja, no período de tempo a partir do instante em que a temperatura nas extremidades da barra é mantida constante, o campo u_T é dado pela expressão:

$$u_T(s, t) = s\gamma\theta + \gamma \sum_{n=1}^{\infty} \left(\frac{2L}{n\pi} \right) C_n e^{-n^2\pi^2\alpha(t-p)/4L^2} \left[1 - \cos\left(\frac{n\pi s}{2L}\right) \right] \quad (5.28)$$

onde C_n e A_m também foram definidos na seção 5.1.3 e são rerepresentados a seguir para proporcionar maior clareza:

$$C_n = \sum_{m=1}^{\infty} A_m \frac{\cos(\beta_m L)}{2} \left[\frac{1 - \cos(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) - \beta_m} + \frac{1 - \cos(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) + \beta_m} \right] + \sum_{m=1}^{\infty} A_m \frac{\operatorname{sen}(\beta_m L)}{2} \left[\frac{\operatorname{sen}(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) - \beta_m} - \frac{\operatorname{sen}(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) + \beta_m} \right] \quad (5.29)$$

$$A_m = \left(\frac{2\theta}{pL} \right) \left[\frac{(-1)^m}{(2m-1)} \right] \left[\frac{\left(\frac{\pi}{2P} \right) - \alpha\beta_m^2 e^{-\alpha\beta_m^2 p}}{(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p} \right)^2} \right] \quad (5.30)$$

O campo de deslocamentos associado ao impedimento à expansão da barra é função do deslocamento da extremidade direita :

$$u_M(s, t) = -\frac{s}{2L} \cdot u_T(2L, t) \quad (5.31)$$

A deformação mecânica é o gradiente do campo de deslocamentos u_M :

$$\epsilon_{mec}(s, t) = -\frac{u_T(2L, t)}{2L} \quad (5.32)$$

Nesse exemplo, os campos de deslocamentos e de deformações foram obtidos sem que fosse necessário recorrer à lei constitutiva, o que o torna interessante do ponto de vista de validação visto que todas as grandezas, inclusive as tensões, podem ser calculadas na solução analítica de forma direta sem que seja necessário resolver uma equação diferencial parcial não-linear. Essa é uma característica dos problemas conduzidos pela deformação, no qual não existem ações mecânicas externas e a deformação é previamente conhecida em todo o corpo.

Assim como realizado na seção 5.2.2, adota-se nesse exemplo um modelo constitutivo de plasticidade uniaxial conforme detalhadamente descrito em Souza Neto et al. (2009). A relação entre a tensão e a deformação é representada pela curva bi-linear ilustrada na Figura 5.24. Para obter as tensões em cada instante de tempo deve-se verificar se o módulo da deformação mecânica corrente $|\epsilon_{mec}|$ é inferior ou superior à deformação ϵ_y associada ao escoamento do material, descrita na Equação 5.16.

Quando $|\epsilon_{mec}(t)| < \epsilon_y$, o corpo se encontra em seu regime elástico e a tensão pode ser calculada por:

$$\sigma(t) = E_0 \epsilon_{mec}(t) \quad (5.33)$$

Caso $|\epsilon_{mec}(t)| \geq \epsilon_y$, a barra está em seu regime plástico, no qual a tensão é calculada pela equação a seguir:

$$\sigma(t) = sig(\epsilon_{mec}(t)) \cdot [f_y + E_{ep}(|\epsilon_{mec}(t)| - \epsilon_y)] \quad (5.34)$$

Nas equações acima, E_0 é o módulo de elasticidade no regime elástico, $sig(\epsilon_{mec})$ é o sinal da deformação mecânica atuante e E_{ep} é o módulo elastoplástico do material.

No restante do desenvolvimento desse exemplo será apresentada a modelagem numérica do problema e os resultados obtidos utilizando os novos recursos implementados no INSANE. Para permitir a comparação entre as soluções analítica e numérica, os valores apresentados na Tabela 5.13 foram atribuídos aos parâmetros do problema.

Tabela 5.13: Parâmetros considerados na modelagem do Exemplo 8.

Parâmetro	Símbolo	Valor	Unidade
Comprimento da barra	$2L$	1	uc
Temperatura inicial	T_0	0	ut
Tempo para a elevação da temperatura nas bordas	p	3600	ute
Temperatura final nas bordas	T_f	50	ut
Massa específica	ρ	2702	um/uc^3
Calor específico	c_p	903	$ue/(um \cdot ut)$
Condutividade térmica	k	237	$up/(uc \cdot ut)$
Difusividade térmica	α	$97,1 \times 10^{-6}$	uc^2/ute
Módulo de elasticidade no regime elástico	E_0	70000	$uf/(ua^2)$
Coefficiente de expansão térmica	γ	$23,1 \times 10^{-6}$	$uc/(ucut)$
Tensão de escoamento	f_y	30	$uf/(ua^2)$
Módulo de elasticidade elastoplástico	E_{ep}	100	$uf/(ua^2)$
Área da seção transversal da barra	A	1	uc^2

A discretização da barra segue a mesma subdivisão empregada no Exemplo 3, onde o domínio foi subdividido em 30 elementos finitos paramétricos unidimensionais de quatro nós, o que resulta em 91 graus de liberdade tanto na análise térmica como na análise mecânica.

Na análise térmica, a temperatura é prescrita nos nós das extremidades, enquanto no restante do domínio tem-se como condição de contorno o impedimento à troca de calor com o ambiente. Na análise mecânica, o deslocamento nulo é imposto nos nós da extremidade da barra, enquanto no restante do contorno tem-se como condição a ausência de forças de superfície.

Assim como discutido no exemplo anterior, apresentado na seção 5.2.3, previamente ao início da análise o gerenciador da estrutura de dados cria uma cópia do modelo discreto e reconfigura cada uma das cópias para que contenham informações relativas à física à que se referem. O modelo discreto utilizado de referência para essa cópia é composto por entidades multifísicas, das quais são extraídos os objetos associados a cada uma das físicas.

Para a solução do PTC, são empregadas as mesmas classes discutidas na seção 5.1.3. No que se refere à parcela mecânica, ressalta-se a utilização da classe `QuasiStaticEquilibriumPath` para a solução do problema trabalhando em conjunto com um objeto `step` da classe `ThermoMecSNR`, que introduz as modificações necessárias ao método incremental-iterativo de Newton-Raphson para contemplar as particularidades dos problemas termomecânicos. Um modelo constitutivo de plasticidade segundo o critério de escoamento de von Mises foi utilizado, no qual é considerada uma lei de endurecimento linear.

Quanto ao controle do processo de solução, foi empregado um *simulation manager* do tipo `TimeDepThermoMecSimManager` que opera com dois configuradores do ambiente de solução, um para cada física. Na transferência de calor, utiliza-se o *setuper* `TimeDepThermalSetuper`, enquanto para a mecânica dos sólidos o *setuper* `SinglePhysicsFemSetuper`. O gerenciador da estrutura de dados é automaticamente

criado no contrutor do *simulation manager*, sendo do tipo `ClonedModelsThermoMecDataManager`.

Nas figuras 5.35 e 5.36 são apresentados os comparativos da evolução do campo de deslocamentos ao longo do tempo.

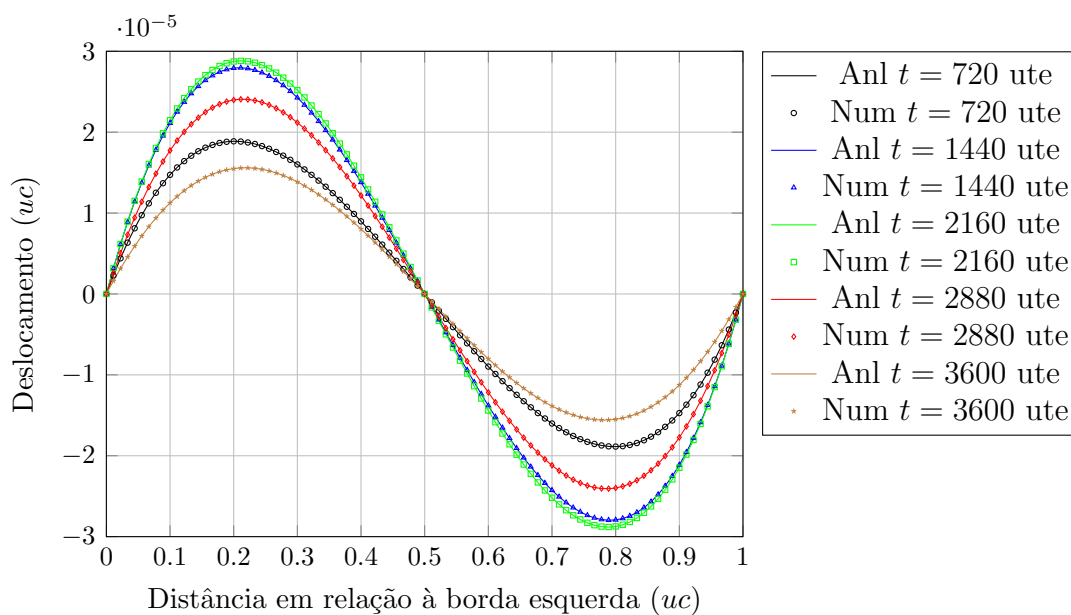


Figura 5.35: Resultados comparativos do campo de deslocamentos durante a elevação da temperatura nas bordas.

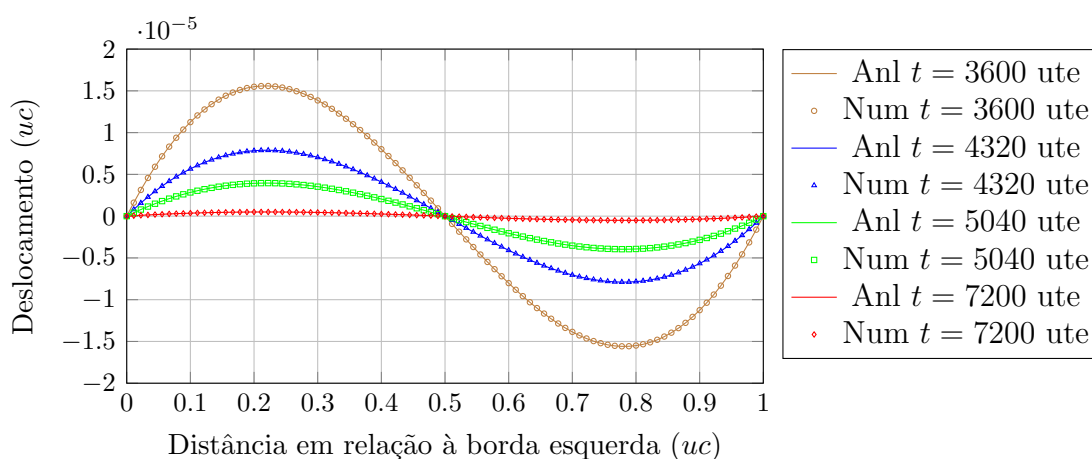


Figura 5.36: Resultados comparativos do campo de deslocamentos após o término da elevação da temperatura nas bordas.

A comparação entre a deformação mecânica calculada analiticamente e a obtida numericamente utilizando o INSANE é mostrada na Figura 5.37. Em seguida, mostra-se na Figura 5.38 esse comparativo para as tenões atuantes na barra.

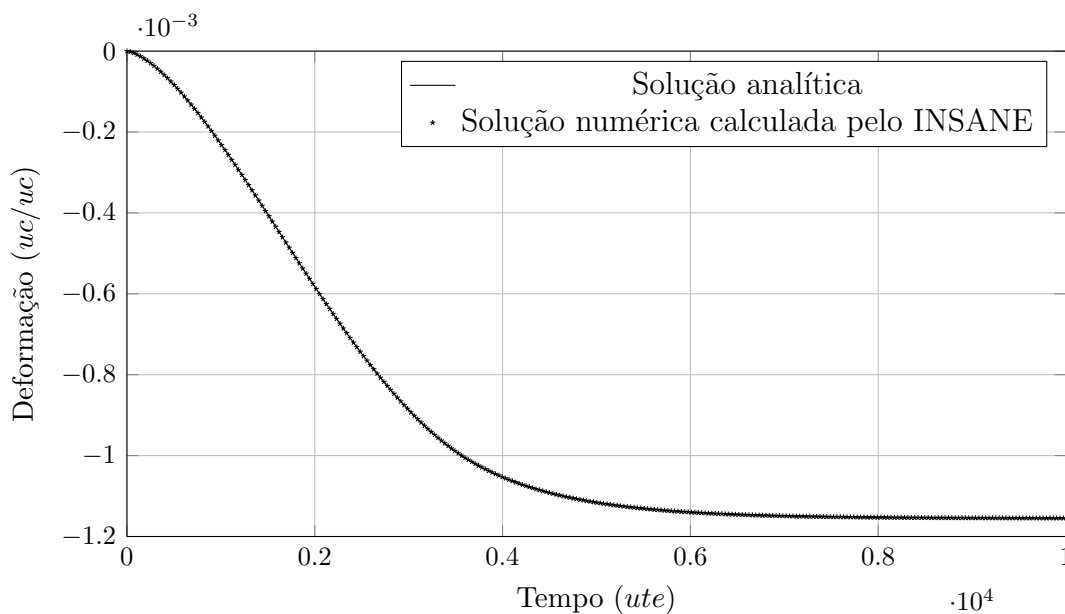


Figura 5.37: Comparativo entre a evolução temporal da deformação calculada analítica e numericamente.

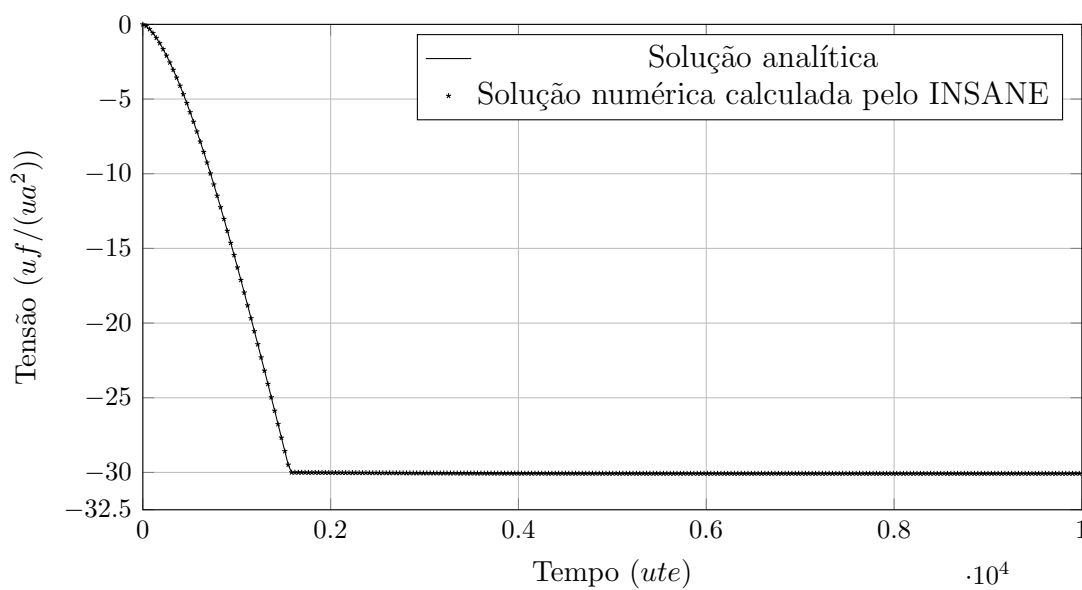


Figura 5.38: Comparativo entre a evolução temporal da tensão calculada analítica e numericamente.

Observa-se que o escoamento do material ocorre próximo a 1580 *ute* após o início da elevação da temperatura nas bordas, instante esse inferior ao tempo em que essa elevação de temperatura se cessa, que é de 3600 *ute*. O modelo numérico foi capaz de reproduzir os resultados calculados analiticamente, como melhor ilustrado na Figura 5.39 onde a janela de tempo apresentada no gráfico foi definida entre 1550 *ute* e 5000 *ute*. Ressalta-se que mesmo após a conclusão da elevação da temperatura nas bordas, tanto a deformação quanto a tensão continuam a crescer, em módulo, até que o regime permanente seja alcançado, o que corresponde a uma distribuição de temperaturas uniforme na barra com valor igual ao da temperatura prescrita nas extremidades.

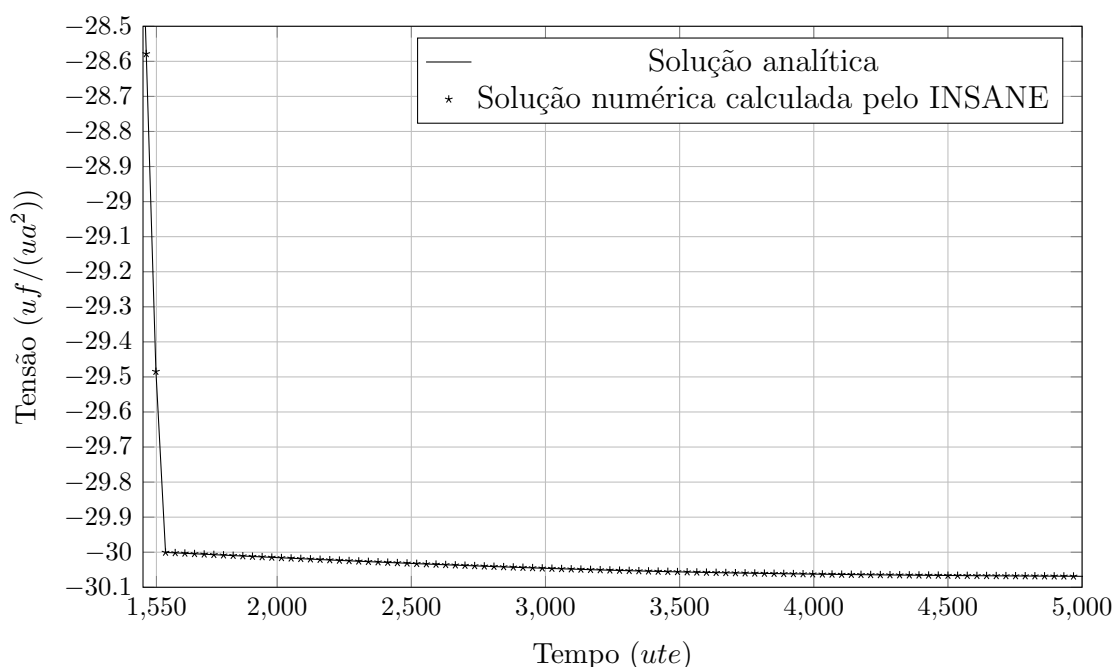


Figura 5.39: Ampliação da região próxima ao escoamento no comparativo das tensões.

Ao consultar os resultados calculados numericamente pelo INSANE no instante de tempo 10224 *ute*, observa-se que a solução em regime transiente converge para a solução em regime permanente, apresentada no Exemplo 6, o que complementa a validação da implementação realizada para o algoritmo em regime transiente.

Capítulo 6

CONSIDERAÇÕES FINAIS

6.1 Contribuições deste trabalho

Este trabalho teve como principal contribuição introduzir recursos no sistema INSANE para o tratamento de problemas termomecânicos desacoplados. A possibilidade do problema analisado possuir não-linearidade física tanto na parcela térmica como na parcela mecânica é contemplada, assim como a possibilidade do problema apresentar dependência temporal. Esses recursos poderão ser utilizados posteriormente tanto como recurso didático em conjunto com as demais ferramentas desse ambiente, como recurso de base para o desenvolvimento de novas pesquisas especializadas na área termomecânica.

A revisão bibliográfica realizada nessa dissertação apresentou, de uma forma simplificada, a evolução da análise termomecânica desde a sua concepção até a consolidação da sua formulação clássica. Nessa revisão também são mencionados alguns dos desdobramentos modernos, cujas futuras expansões por meio de novas pesquisas poderão se apoiar na implementação apresentada nessa dissertação de mestrado.

As condições que devem ser atendidas para que o trabalho aqui apresentado seja aplicado a problemas práticos foram elucidadas durante a apresentação da revisão bibliográfica. Essas condições se referem, especialmente, aos requisitos que devem ser atendidos para que o termo de acoplamento termoelástico possa ser desprezado na equação da condução do calor, o que viabiliza a solução do problema de transferência

de calor de forma independente do problema mecânico. Nesse caso particular, é necessário solucionar a análise térmica previamente à análise mecânica, visto que se deve conhecer a variação de temperatura ao longo do domínio analisado para que o efeito dela seja incluído durante o cálculo do campo de deslocamentos.

Um dos requisitos necessários para desprezar o termo de acoplamento termomecânico na equação da condução do calor é que a variação da temperatura ocorra de forma significativamente mais lenta que a variação das ações mecânicas externas, o que também permite, em geral, desprezar o termo inercial na equação do movimento no problema mecânico, tornando-o quasi-estático.

Para atingir o objetivo principal desta dissertação, tornou-se necessário expandir significativamente o núcleo numérico do INSANE, visto que previamente a esse trabalho somente era possível solucionar problemas de condução de calor lineares em regime permanente. Os algoritmos introduzidos para solucionar o PTC em condições mais abrangentes foram devidamente detalhados, assim como as pequenas modificações realizadas na parcela previamente existente do programa.

Ressalta-se a criação de um novo mecanismo para conferir dependência das propriedades físicas dos materiais previamente existentes no INSANE em relação à temperatura. Com esse mecanismo foi tanto possível solucionar problemas de transferência de calor fisicamente não-lineares, como definir propriedades mecânicas dependentes da temperatura sem comprometer o grau de generalidade do INSANE.

Os recursos introduzidos para a solução de problemas de transferência de calor foram testados em quatro exemplos de validação, que foram escolhidos de forma a cobrir as combinações possíveis entre dependência temporal e não-linearidade física. As soluções analíticas desses exemplos foram obtidas na literatura e serviram de referência para a avaliação da qualidade da solução numérica obtida utilizando o INSANE. Nos três primeiros exemplos, as soluções numérica e analítica se igualaram, e no quarto exemplo observou-se um pequeno distanciamento entre essas soluções, cuja causa é discutida a seguir.

A solução de referência adotada para o Exemplo 4, que é um problema fisicamente não-linear dependente do tempo, permite que a condutividade térmica, o calor específico e a massa específica variem com a temperatura, desde que a difusividade térmica permaneça constante. Tal consideração implica que o produto entre o calor específico e a massa específica varie na mesma proporção que a condutividade térmica varia.

Como o algoritmo implementado no INSANE no presente momento se restringe ao caso onde somente a condutividade térmica pode variar com a temperatura, tornou-se necessário violar o requisito de difusividade térmica constante imposto pela solução analítica adotada para permitir a comparação entre os resultados analíticos e os resultados numéricos nesse exemplo.

Todavia, ressalta-se que mesmo violando o requisito de difusividade térmica constante, os resultados numéricos e os analíticos se aproximaram de forma satisfatória, validando as classes implementadas para a solução numérica desse problema. Tornou-se necessário, portanto, revisitar esse exemplo após modificar o algoritmo da classe `NonLinearHeatTransferNewmarkBeta` para que ele permita a dependência de todas as propriedades físicas do material em relação à temperatura buscada.

No que se refere à forma de aplicar as condições iniciais e as condições de contorno essenciais, esse trabalho proporcionou uma maior flexibilidade ao usuário na forma de defini-las e combiná-las. Essa expansão se encontra no novo projeto denominado `conditions`, essencial à solução termomecânica, mas que também pode ser empregado em problemas convencionais que envolvam apenas uma física.

Para a solução termomecânica, também se tornou necessário adequar as classes de solução para elas representarem as particularidades impostas pela não-linearidade física na presença de variações de temperatura, quando ela ocorre na parcela mecânica, e para conduzirem soluções no regime quasi-estático. Com o intuito de proporcionar uma maior modularidade, optou-se por inserir essas particularidades em novas classes em vez de modificar as classes existentes.

Outros requisitos impostos pela análise termomecânica que foram cumpridos nesse trabalho são a criação de um mecanismo que permita a transmissão automática de informações da análise térmica para a análise mecânica, a criação de uma entidade que proporcione um maior controle do fluxo de execução durante o processo de solução, e um configurador automático do ambiente de solução, especialmente necessário no caso em que o mesmo modelo discreto é compartilhado por ambas as físicas. Esses recursos foram introduzidos em um novo projeto denominado `simulationmanager`.

Os novos projetos `conditions` e `simulationmanager` foram concebidos de forma a garantirem a compatibilidade com todos os trabalhos realizados previamente no INSANE que não os continham.

O projeto `simulationmanager` é subdividido em três pacotes, um que contém os gerenciadores de solução criados, também denominado `simulationmanager`, um para os configuradores do ambiente de solução, denominado `setuper`, e um para os gerenciadores da estrutura de dados, denominados `datamanager`. As classes introduzidas nesses pacotes respeitaram o paradigma de orientação a objetos e foram especializadas à medida que os novos recursos foram introduzidos. Reitera-se que o projeto `simulationmanager` contém classes que não estão associadas exclusivamente a problemas multifísicos, podendo essas serem utilizadas em outras aplicações do INSANE.

Reforça-se também a criação de novas classes que conferem um comportamento multifísico a entidades que possuem o seu comportamento dependente da física em questão, tais como as classes de material, modelo de análise, *problem driver* e modelo constitutivo. Esse comportamento multifísico é proporcionado por classes que armazenam em si dois objetos do mesmo tipo, sendo um aplicável à transferência de calor e o outro à mecânica dos sólidos. Essa estratégia se baseia na aplicação conjunta dos princípios de herança e de composição da programação orientada a objetos.

Assim como realizado para os problemas de transferência de calor, os problemas termomecânicos foram testados nas combinações possíveis entre não-linearidade física e dependência temporal. Novamente, foram escolhidos exemplos que cobrissem da forma mais ampla possível os diversos recursos introduzidos no núcleo numérico. As soluções analíticas desses problemas foram obtidas da literatura. Em todos os casos a solução numérica calculada pelo INSANE convergiu para a respectiva solução analítica, validando a implementação realizada.

6.2 Sugestões para trabalhos futuros

Quanto às limitações da abrangência da presente implementação a serem contornadas em trabalhos futuros, tem-se:

1. Implementação restrita a materiais homogêneos e isotrópicos. A implementação de outros modelos materiais requer um estudo mais profundo da bibliografia especializada, que constitui uma das linhas de pesquisa em constante desenvolvimento na área termomecânica;
2. Condições de contorno do problema térmico restritas a fluxos de calor prescritos e a convecções prescritas. Ainda não foram introduzidos recursos para o tratamento das trocas de calor por radiação, visto que ela corresponde a uma condição de contorno não-linear. Além da não-linearidade, a implementação das trocas de calor por radiação requer desenvolvimentos associados a geometria computacional, quando se deseja estudar a troca de calor por esse modo de transferência entre superfícies distintas;
3. Não-linearidade física do problema de transferência de calor restrita à dependência entre a condutividade térmica e a temperatura. Deve-se estender a implementação ao caso que também considere essa dependência para o calor específico e para a massa específica;
4. Implementação do problema termomecânico restrita ao MEF. Atualmente o INSANE possui outras técnicas de discretização implementadas, sendo essas utilizadas em problemas da mecânica dos sólidos. Recomenda-se, portanto, um estudo das modificações necessárias para estender essas técnicas aos problemas de transferência de calor;
5. A implementação atual ainda não contempla modelos de análise de cascas. Um estudo aprofundado pode ser realizado para identificar a forma de inserir

as deformações associadas à variação de temperatura nesse modelo de análise, seguida da expansão do núcleo numérico e devida validação dessa expansão;

6. Implementação termomecânica em regime transiente restrita ao caso quasi-estático, onde os termos inerciais da equação do movimento são desprezados. Conforme previamente discutido ao longo desse trabalho, não é recomendável utilizar a abordagem clássica da teoria termomecânica para o estudo de problemas dinâmicos, visto que neles os critérios para se desprezar o termo de acoplamento termoelástico não são atendidos. Portanto, sugere-se que a pesquisa iniciada nessa dissertação seja continuada incluindo o caso termomecânico acoplado, no qual se obtém simultaneamente os campos de temperatura e de deslocamentos, viabilizando o desenvolvimento de pesquisas contemplando o comportamento termomecânico dinâmico. Sugere-se também a introdução de novos modelos constitutivos aplicáveis à transferência de calor que considerem as suas formulações modernas;
7. Implementação do PTC e do PTM restrita a corpos sólidos. Uma vez que os estudos multifísicos se iniciaram com esse trabalho, sugere-se a realização de pesquisas para a introdução de novas físicas no núcleo numérico, como a mecânica dos fluidos, a fim de viabilizar a expansão em outros ramos multifísicos além do termomecânico;
8. Solução realizada por uma única linha de processamento e sem levar em conta a esparsidade das matrizes envolvidas. Recomenda-se a adequação da implementação discutida nesse trabalho para permitir a utilização das bibliotecas de computação de alto desempenho para a solução de sistemas de equações lineares envolvendo matrizes esparsas. Também sugere-se estudar o processo de solução multifísico aqui proposto para identificar tarefas que possam ser executadas em paralelo, proporcionando assim um ganho em eficiência computacional.

9. Entrada de dados restrita a arquivos de texto. Para aumentar as potencialidades da ferramenta como recurso pedagógico sugere-se a criação de interfaces gráficas ao usuário no pré-processador do INSANE para que os novos recursos introduzidos nesse trabalho possam ser acessados diretamente durante a utilização do programa, proporcionando assim uma alternativa mais amigável à descrição do modelo discreto e configuração dos parâmetros da análise. Também sugere-se aumentar as potencialidades do pós processador para permitir o traçado de curvas da evolução da temperatura ao longo do tempo em pontos escolhidos pelo usuário.
10. Dependência entre as propriedades físicas e a temperatura restrita a relações polinomiais, onde o usuário informa os coeficientes dos polinômios. Sugere-se unificar a metodologia proposta nesse trabalho com a proposta por Teixeira (2017), que consiste em curvas de vários trechos lineares especificadas pelo usuário por meio de tabelas.

As sugestões de trabalhos futuros aqui apresentadas estão dispostas no gráfico a seguir conforme suas respectivas complexidade e prioridade.

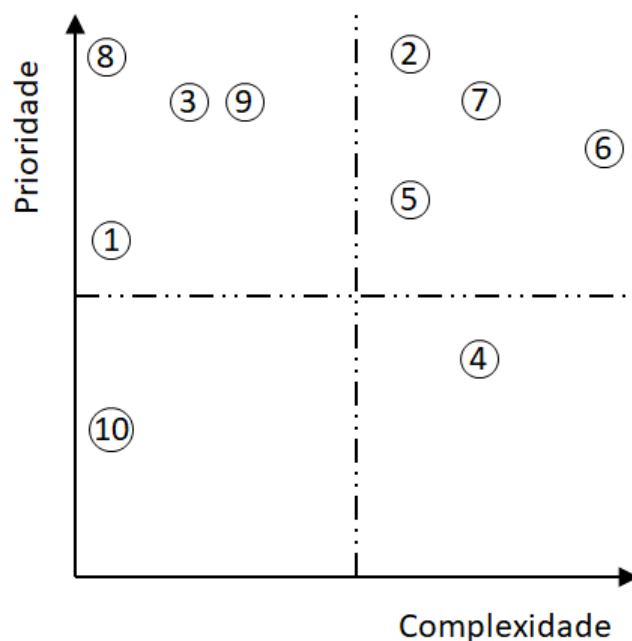


Figura 6.1: Classificação das sugestões para trabalhos futuros.

Apêndice A

ANÁLISE DO NÚCLEO NUMÉRICO DO INSANE

No Capítulo 1 realizou-se uma breve apresentação do sistema computacional INSANE, que foi posteriormente complementada no início do Capítulo 4. Neste apêndice, o estado em que o núcleo numérico do INSANE se encontrava previamente a esse trabalho é abordado de forma detalhada, sendo discutidos os requisitos à implementação computacional dos recursos para a solução do problema termomecânico no âmbito do Método dos Elementos Finitos.

Para ilustrar a organização do núcleo e os principais processos que nele ocorrem, será empregada a UML - *Unified Modelling Language*, que é uma linguagem padronizada pelo OMG - Object Management Group (2019) para a modelagem de sistemas de programas de computador orientados a objetos.

A.1 Pontos de partida do processo de solução

No INSANE, o início do processo de solução de um modelo discreto pode ser disparado de três formas diferentes pelo usuário do programa, a saber:

- Pela interface gráfica, através da chamada ao comando implementado na classe `NewProcessorCommand`. Nesse caso, o modelo discreto construído na etapa de pré-processamento é utilizado pelo solucionador;

- Solicitando a leitura de um arquivo de entrada de dados por meio da interface gráfica. O usuário pode solicitar a leitura do arquivo por meio de um botão no diálogo inicial do INSANE, que dispara o comando implementado na classe `LoadSolverCommand`. Esse comando constroi um diálogo para que o usuário informe o arquivo de entrada de dados, e após a leitura do mesmo pela persistência o processo de solução é continuado na classe `SolverClass`;
- Informando o arquivo de entrada de dados como um parâmetro durante a solicitação à execução do INSANE por linha de comando. Nesse caso, o processo é conduzido pelo método `main()` da classe `StandaloneSolver`.

A análise dos algoritmos para o disparo e condução geral do processo de solução indicou a existência de etapas de configuração do ambiente de solução, e que elas naturalmente foram implementadas para atenderem a uma única física.

No que se refere às condições de contorno, observou-se que há uma limitação na forma de combiná-las. Dentro do âmbito do MEF não existiam entidades específicas para o armazenamento das condições de contorno essenciais, que eram prescritas pelo usuário diretamente sobre a discretização e permaneciam inalteradas durante todo o processo de solução. Por isso, nesses algoritmos o laço iterativo percorria somente as combinações envolvendo as condições de contorno naturais, e para cada combinação um processo de solução era iniciado em uma classe do tipo `Solution`.

A fim de proporcionar uma maior flexibilidade ao usuário, é mais adequado que as combinações a serem percorridas contemplem três entidades: condição inicial, condição de contorno essencial e condição de contorno natural. Estendendo esse conceito ao caso multifísico, considera-se ainda que deve ser informado um conjunto de condições para cada uma das físicas.

Para atender ao caso multifísico, os algoritmos pré-existentes de inicialização e de condução da solução poderiam ser adaptados, entretanto a externalização dessas tarefas para novas classes minimiza a repetição de código, encapsula a funcionalidade

e permite a aplicação do mecanismo de herança, de forma que somente as classes especializadas para a multifísica tenham conhecimento das particularidades desse tipo de problema, o que é recomendável do ponto de vista do paradigma de orientação a objetos.

A.2 Classe abstrata `Solution`

Após a preparação do ambiente computacional, a solução do problema discreto é realizada, para cada uma das combinações, por classes herdeiras da classe abstrata `Solution`, ilustrada abaixo. Nessas classes, a solução se inicia com a chamada ao método `execute()`.

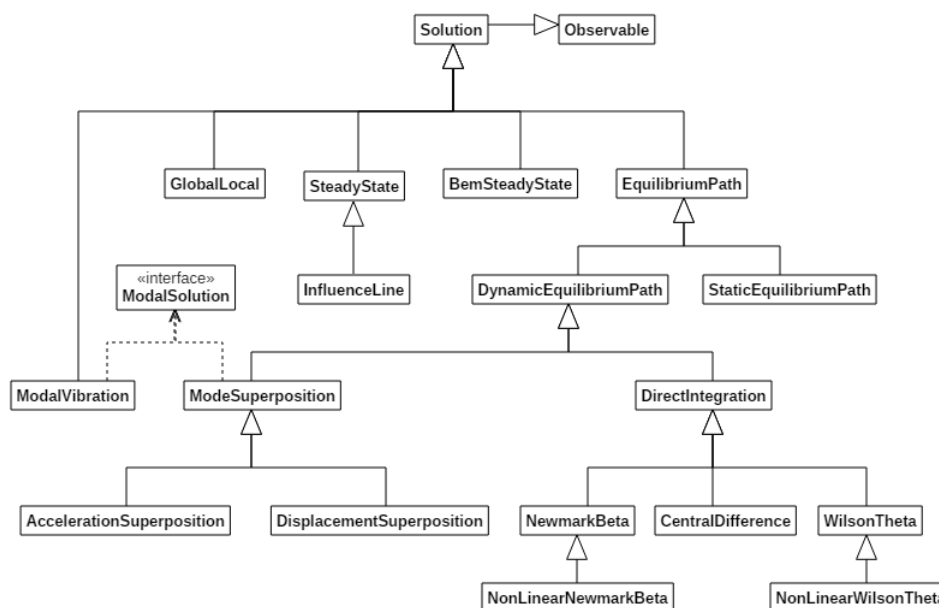


Figura A.1: Diagrama de classe para `Solution`.

Como os trabalhos anteriores realizados no INSANE não contemplavam mais de uma física, os problemas podiam ser resolvidos utilizando apenas uma instância de uma das classes herdeiras de `Solution`. Diversas classes foram implementadas na referida hierarquia, onde cada uma delas atende a um propósito específico. As aplicações práticas de algumas das principais classes do pacote `solution` são mencionadas

abaixo:

- Classe **SteadyState**: empregada em problemas em regime permanente cujo comportamento seja linear;
- Classes derivadas de **StaticEquilibriumPath**: empregadas em problemas em regime permanente cujo comportamento seja não-linear ou com carregamentos quasi-estáticos;
- Classe **ModalVibration**: empregada no cálculo de autovalores e autovetores em análises modais da mecânica dos sólidos;
- Classes derivadas de **DynamicEquilibriumPath**: empregadas em problemas em regime transiente da mecânica dos sólidos.

A.2.1 Processo de solução de problemas não-lineares

Quando os problemas abordados possuem não-linearidade física, é necessário recorrer a métodos iterativos, como o método de Newton detalhado na seção 3.1.3 referente ao problema de transferência de calor. Nesses métodos o sistema parte de uma condição de equilíbrio e, na busca por uma nova condição de equilíbrio, a variável de estado é modificada até que o resíduo se torne inferior à uma tolerância pré-determinada.

Em muitos casos, deseja-se conhecer o comportamento da estrutura em diversas configurações de equilíbrio entre a configuração inicial e a final, correspondente à aplicação completa das ações externas em problemas em regime permanente. Para tal, fraciona-se a aplicação das ações externas em passos, e nesse caso o procedimento de solução é denominado incremental-iterativo.

A curva obtida da conexão entre os pontos de equilíbrio em um gráfico que relaciona a variável de estado e sua variável dual é denominada trajetória de equilíbrio. Nos problemas da mecânica dos sólidos, por exemplo, a variável de estado é o deslocamento e a sua variável dual é a força.

Durante a solução de problemas não-lineares, podem ocorrer pontos limites nas trajetórias de equilíbrio, que correspondem a pontos nos quais o método numérico empregado pode ser incapaz de atingir o próximo ponto de equilíbrio a partir do ponto de equilíbrio corrente.

A fim de contornar situações que impeçam a continuidade da análise nos pontos limites, frequentemente presentes em problemas da mecânica dos sólidos, podem ser empregadas diversas estratégias de controle no processo iterativo, como descrito em Yang e Shieh (1990).

Como mencionado anteriormente, a classe `StaticEquilibriumPath` é utilizada para gerir, de forma genérica, o processo de solução de problemas não-lineares segundo algoritmos incrementais-iterativos, realizando o controle geral sobre os passos da análise. Essa classe possui um objeto do tipo `Step`, que realiza os cálculos durante as iterações, e uma lista de objetos do tipo `IterativeStrategy`, que define os métodos de controle a serem utilizados para a obtenção da trajetória de equilíbrio.

O objeto `Step` possui uma referência a um objeto do tipo `Assembler`, e ao longo do processo iterativo são feitas chamadas em `Assembler` para que sejam fornecidas as matrizes e os vetores necessários.

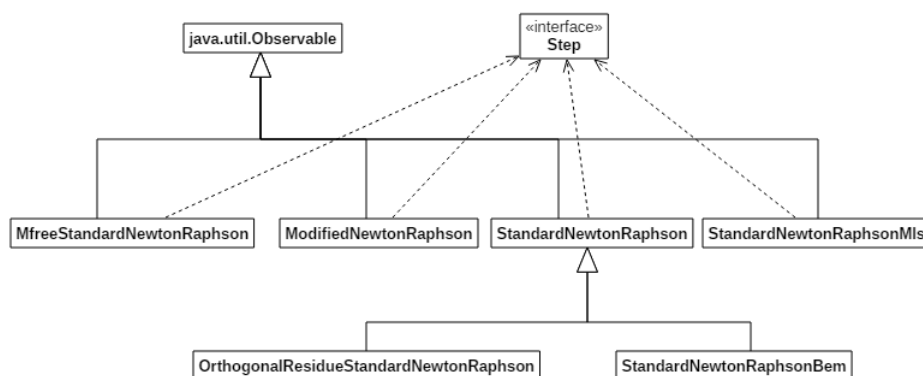


Figura A.2: Diagrama de classe para `Step`.

Por meio do mecanismo de propagação de mudanças Observador-Observado, ao término de cada passo uma notificação é disparada aos observadores do objeto `StaticEquilibriumPath` para que a estrutura de dados seja atualizada e para que

o arquivo de saída referente ao passo seja gravado em disco.

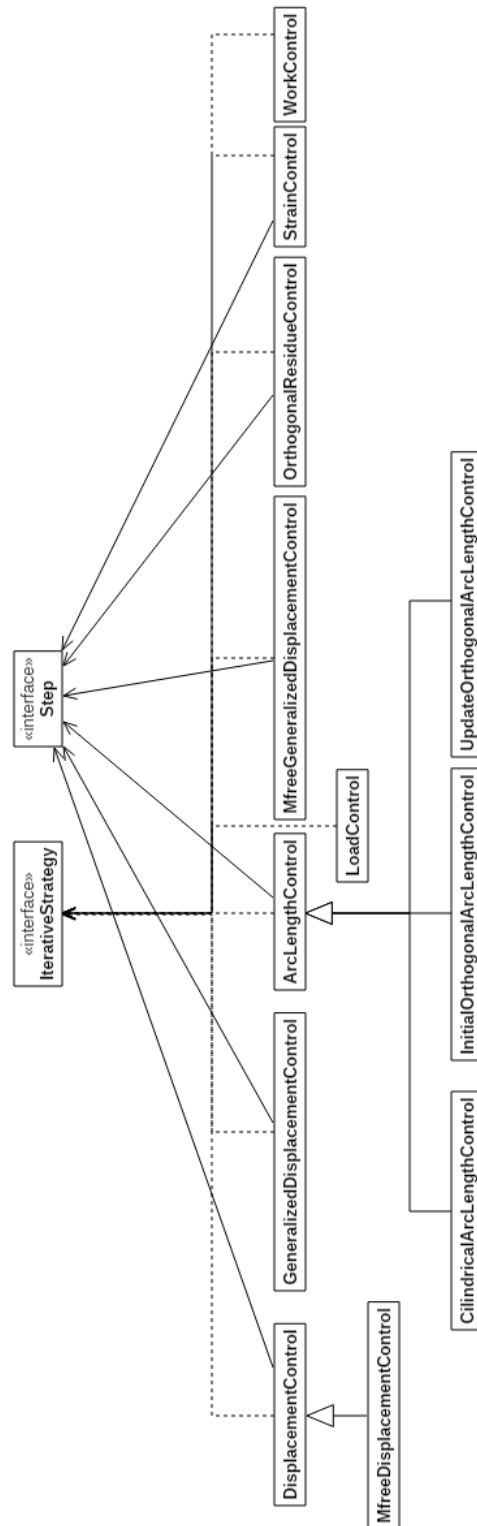


Figura A.3: Diagrama de classe para IterativeStrategy.

A.2.2 Adequações necessárias no pacote `solution`

Durante a análise do projeto orientado a objetos do INSANE observou-se que a classe `SteadyState` pode ser utilizada sem alterações para a solução de problemas de transferência de calor lineares em regime permanente.

Quanto aos problemas de transferência de calor não-lineares em regime permanente, observou-se que para os casos tratados até o presente momento não é usual a ocorrência de pontos limites entre a condição inicial e a condição final. Portanto, não há necessidade de aplicar o carregamento térmico de forma incremental, o que tornaria o processo de obtenção da solução térmica custoso, justificando a implementação do método de Newton em uma nova classe do tipo `Step` específica para essa aplicação.

No âmbito dos problemas de transferência de calor em regime transiente, observa-se que a equação diferencial que modela o problema térmico difere em ordem da equação diferencial que rege o problema dinâmico da mecânica dos sólidos, o que implica na necessidade de novas classes para atender ao PTC nessa situação. Além disso, é necessário implementar uma classe para o caso linear e outra para o caso não-linear.

No que se refere à forma de conduzir o processo de solução multifísico em regime transiente, duas abordagens gerais podem ser empregadas. Uma delas corresponde a realizar a análise completa da primeira física, ou seja, cobrindo todo o intervalo de tempo, e registrando os resultados em um arquivo para em seguida executar a análise na segunda física transferindo as informações necessárias da primeira física a partir da leitura desse arquivo. Essa técnica possui como desvantagem principal a necessidade de realizar toda a análise da primeira física antes de começar a obter resultados da análise na física seguinte, o que pode ser crítico em análises transientes longas nas quais o usuário deseja ter uma indicação preliminar se os parâmetros utilizados tais como tamanho do passo de tempo e o grau de discretização estão adequados para o problema em questão.

Outra abordagem possível para a condução do processo de solução multifísico transiente consiste em avançar paralelamente nas soluções de cada uma das físicas, transferindo as informações entre elas a cada passo de tempo com o auxílio de um gerenciador do processo de solução, sem realizar leituras e escritas em disco para essa troca de dados. Este processo tem como desvantagem um maior consumo de memória, visto que demanda mais de um modelo instanciado simultaneamente em tempo de execução. Sua principal vantagem é disponibilizar ao usuário resultados para todas as físicas envolvidas a cada passo de tempo ao longo da análise, além de provavelmente ser mais veloz, visto que dispensa a leitura do arquivo para troca de dados entre as físicas. Essa segunda abordagem foi a escolhida, e corresponde ao fluxograma da Figura 3.4 , apresentado no capítulo anterior dessa dissertação de mestrado.

Analisando a possibilidade de reutilização das classes previamente implementadas, nota-se que a classe `SteadyState` pode ser utilizada sem alterações para a obtenção da solução mecânica em problemas teromecânicos lineares em regime permanente. Ainda tratando de problemas lineares, porém no regime quasi-estático, torna-se necessário introduzir uma nova classe similar à `SteadyState`, cuja principal diferença seja permitir a solicitação do cálculo da solução para o instante de tempo desejado. No regime quasi-estático em meios lineares, as soluções nos diversos instantes de tempo da análise se referem a soluções em regime permanente calculadas utilizando o carregamento que atua no corpo no instante de tempo desejado.

Como o INSANE já dispunha de modelos constitutivos e classes de solução dedicadas a problemas fisicamente não-lineares da mecânica dos sólidos, fez-se uma análise dessas classes com o objetivo de utilizá-las nesse trabalho, demonstrando assim as potencialidades de um projeto de software baseado no paradigma de programação orientada a objetos. A revisão bibliográfica relativa à análise de materiais em regime plástico não foi realizada nos capítulos anteriores. A fundamentação teórica relativa a esse assunto pode ser encontrada em Souza Neto et al. (2009), e sua

respectiva implementação no INSANE em Penna (2011) e em Oliveira (2016).

Observou-se que o processo de solução incremental-iterativo implementado na classe `StandardNewtonRaphson` no INSANE, que poderia ser utilizado no cálculo da parcela mecânica, carece de adequações para considerar corretamente o efeito da variação de temperatura nas parcelas proporcionais e constante do carregamento. Também é observada a necessidade de incluir novos critérios de convergência quando existem deformações térmicas no corpo, visto que a variação de temperatura ocasiona um carregamento auto-equilibrado.

Portanto, sugere-se a introdução de uma nova classe que implemente a interface `Step` para a solução de problemas termomecânicos fisicamente não-lineares, sejam eles em regime permanente ou em regime quasi-estático, e uma nova classe que estenda de `StaticEquilibriumPath` introduzindo a tarefa de atribuir os carregamentos no objeto `Step` referentes ao instante de tempo desejado, requisito necessário em problemas no regime quasi-estático.

A.3 Interface Assembler

A interface `Assembler`, cujo diagrama UML está mostrado na A.4, é a responsável pela montagem, de forma genérica, de sistemas de equações algébricas de segunda ordem:

$$\mathbf{A}\ddot{\mathbf{X}} + \mathbf{B}\dot{\mathbf{X}} + \mathbf{C}\mathbf{X} = \mathbf{R} - \mathbf{F} \quad (\text{A.1})$$

onde \mathbf{X} é o vetor das variáveis de estado e $\dot{\mathbf{X}}$ e $\ddot{\mathbf{X}}$ são os vetores que representam a primeira e a segunda derivadas temporais da variável de estado. \mathbf{A} , \mathbf{B} e \mathbf{C} são matrizes de coeficientes, que podem ou não depender da variável de estado e de suas derivadas.

O vetor \mathbf{R} é o vetor das cargas nodais, sendo composto de duas componentes: $\mathbf{R} = \mathbf{N} + \mathbf{E}$, onde \mathbf{N} é o vetor que contém as ações externas aplicadas diretamente

sobre os nós e \mathbf{E} é o vetor correspondente às ações aplicadas no contorno ou no domínio do elemento finito.

Quanto ao vetor \mathbf{F} , suas componentes se referem à resposta interna do corpo e, por isso, é utilizado em análises fisicamente não-lineares durante o cálculo do resíduo, no qual é realizada a diferença entre as ações externas \mathbf{R} e a resposta interna \mathbf{F} .

Atualmente, a interface **Assembler** contém implementações específicas para diversas técnicas de discretização como o Método dos Elementos Finitos, o Método dos Elementos de Contorno, o Método dos Elementos Finitos Generalizados, métodos não locais e métodos sem malha. Também existem implementações específicas para a comunicação com bibliotecas externas de processamento de alto desempenho que podem ser utilizadas para a solução dos sistemas de equações algébricas. Como essa dissertação de mestrado se insere no contexto do Método dos Elementos Finitos, os estudos relativos ao processo de montagem se restringiram à classe **FemAssembler**.

Durante a análise da classe **FemAssembler** em conjunto com a classe de solução **StaticEquilibriumPath** observou-se que o carregamento proporcional e o carregamento constante eram calculados somente para a primeira combinação de carregamentos da lista de combinações. Essa limitação é muito restritiva ao caso termomecânico, que contém combinações para a análise térmica e combinações para a análise mecânica. Portanto, torna-se necessário incluir métodos em **FemAssembler** para que essas parcelas de carregamento possam ser calculadas para a combinação de carregamento desejada, no instante de tempo desejado. Ressalta-se que a classe **StaticEquilibriumPath** não foi alterada, e as solicitações à montagem do carregamento proporcional e do carregamento constante para uma combinação de carregamento específica no instante de tempo desejado são feitas por uma classe herdeira de **StaticEquilibriumPath**, que teve essa funcionalidade especializada.

Quanto às parcelas de carregamentos constante e proporcional relacionadas à variação de temperatura, torna-se necessário incluir métodos específicos para calculá-las corretamente, visto que a forma de introduzi-las na análise incremental-iterativa

em problemas termomecânicos com não-linearidade física na parcela mecânica carece de um tratamento especial. Como consequência da necessidade de calcular os carregamentos proporcionais devido à variação de temperatura de uma forma particular, tornou-se necessário separá-los dos demais carregamentos da mecânica dos sólidos. Essa separação implica em introduzir uma nova parcela no vetor \mathbf{R} na Equação 3.90, onde $\mathbf{R} = \mathbf{N} + \mathbf{E} + \mathbf{H}$, sendo \mathbf{H} o vetor contendo os carregamentos decorrentes da variação da temperatura.

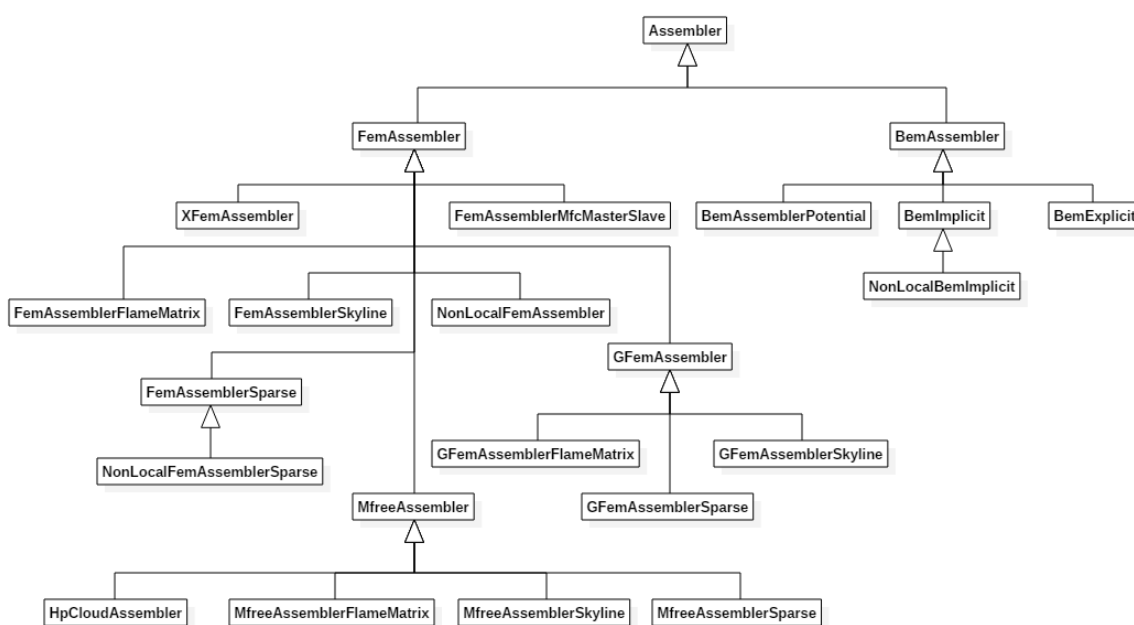


Figura A.4: Diagrama de classe para Assembler.

A.4 Classe abstrata Model

A classe abstrata `Model` encapsula as entidades essenciais para a descrição de um modelo discreto e provê métodos para acesso e manipulação das informações em si armazenadas. Sua implementação aplicável ao método dos elementos finitos, a classe `FemModel` presente no diagrama da Figura A.5 e detalhada na Figura A.6, acrescenta as funcionalidades de: inicialização, reinicialização, atualização e consulta em um modelo de elementos finitos através dos métodos `init()` e `initKeys()`, `cleanResults()`,

update(), *getElementKeys()* e *getNodeKeys()*.

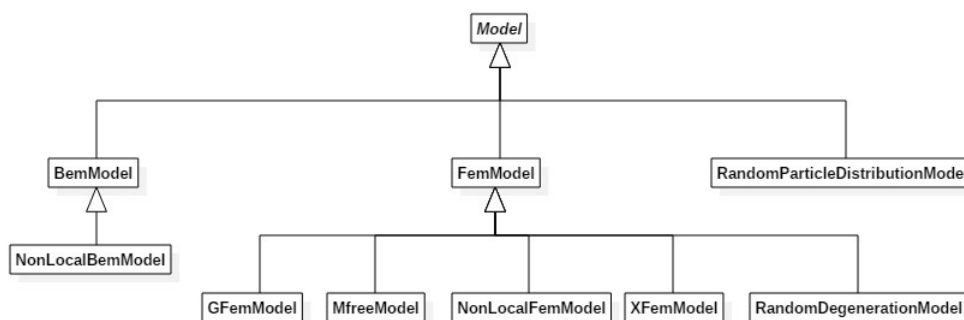


Figura A.5: Diagrama de classe para Model.

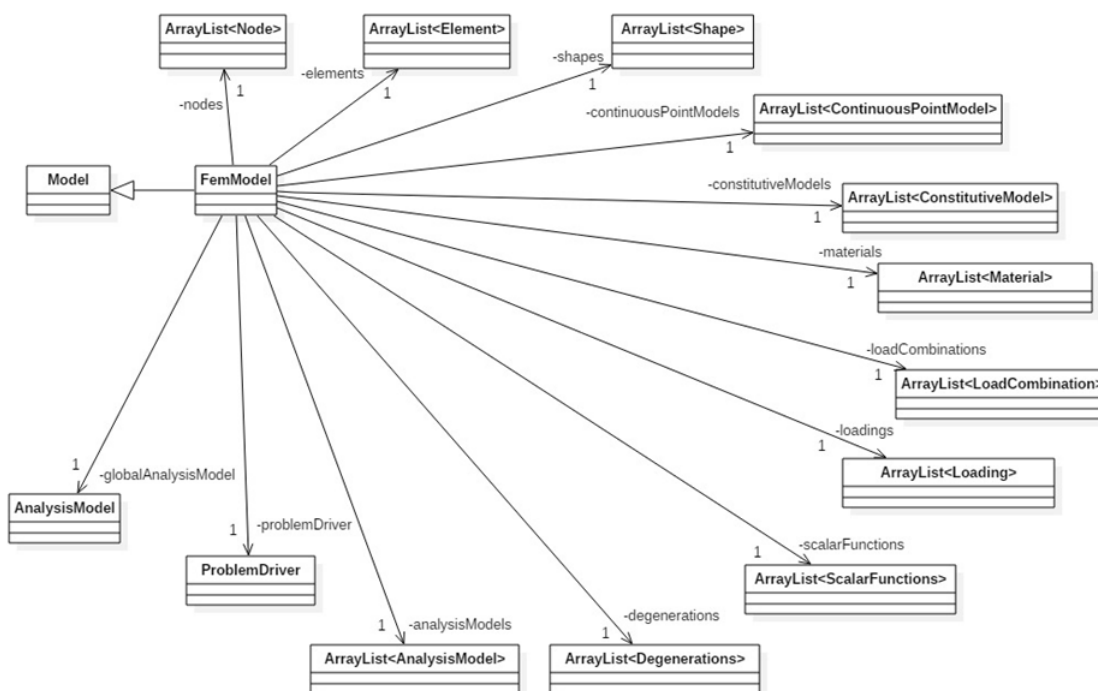


Figura A.6: Diagrama de classe para FemModel.

Como pode ser observado na figura acima, um objeto da classe `FemModel` contém principalmente listas de nós, elementos, modelos de análise, funções de interpolação, carregamentos, combinações de carregamentos, funções escalares, degenerações, materiais, modelos constitutivos e uma referência a um objeto do tipo `ProblemDriver` e uma referência ao modelo de análise global. Essas entidades serão explicadas a seguir.

A.4.1 Classe Node

A classe `Node` representa um nó na discretização espacial do domínio e armazena informações tais como:

- posição espacial;
- valor da variável de estado e de suas derivadas;
- valor da variável dual;
- valores prescritos para a variável de estado e suas derivadas na forma de contorno essencial ou de condição inicial;
- valores prescritos para a variável dual como condição de contorno natural;
- valores desconhecidos para a variável dual;
- identificadores das posições dos valores associados ao nó nas matrizes e vetores que representam o meio discreto;

A classe `Node` possui uma herdeira denominada `ElementNode` como ilustrado na Figura A.7, que representa um nó de um elemento finito, e por isso armazena informações adicionais em um mapa complementar como, por exemplo, as liberações de graus de liberdade nas extremidades de elementos.

A análise do projeto orientado a objetos da classe `Node` identificou a necessidade de incluir mais duas chaves no mapa de valores nodais, a saber: temperatura inicial e temperatura corrente. Esses valores são úteis durante o cálculo dos carregamentos associados à variação da temperatura. A temperatura corrente também pode ser utilizada na análise mecânica quando as propriedades constitutivas possuírem dependência quanto a essa grandeza.

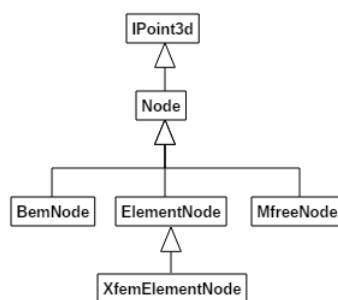


Figura A.7: Diagrama de classe para `Node`.

A.4.2 Classe abstrata `Element`

A classe abstrata `Element` representa os elementos finitos em um modelo discreto e seu diagrama de classes é mostrado a seguir.

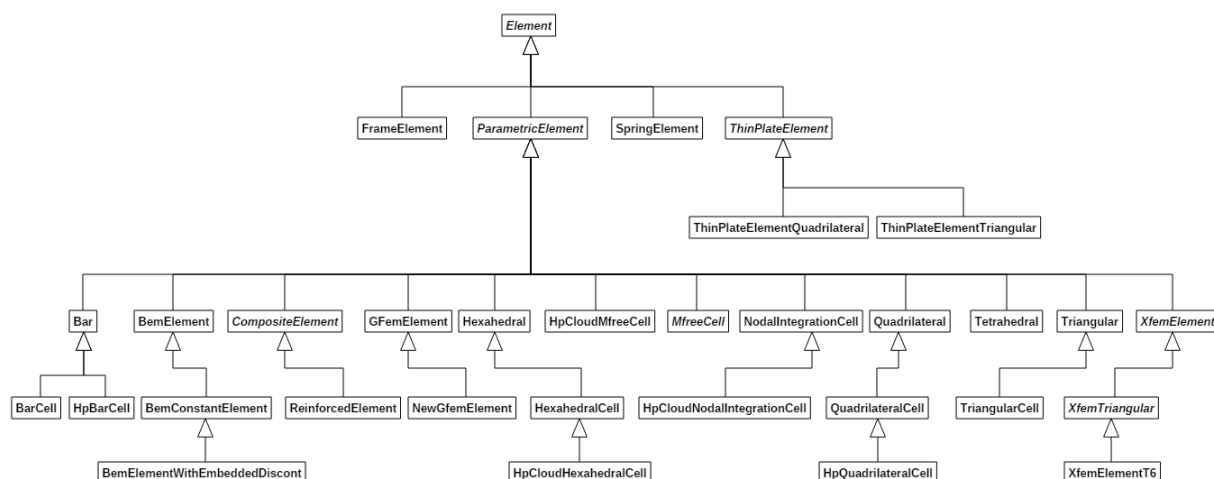


Figura A.8: Diagrama de classe para `Element`.

Essencialmente, um elemento finito possui como atributos, principalmente, as entidades listadas a seguir:

- uma lista de nós, também denominada incidência, do tipo `ElementNode`. Essa lista de nós define a geometria do elemento e o grau da aproximação numérica;
- um modelo de análise do tipo `AnalysisModel`, que é responsável por fornecer informações específicas do comportamento do elemento finito, como o número

de graus de liberdade por nó, o nome da variável de estado e outras características particulares;

- uma referência ao modelo de análise global, que determina as características gerais da análise a ser realizada no modelo discreto onde o elemento finito em questão está inserido;
- uma referência a um modelo constitutivo, que descreve matematicamente como o meio responde às ações externas. Em geral os modelos constitutivos estabelecem uma relação entre a variável interna e a variável dual interna;
- uma referência a um objeto do tipo **Shape**, que contém as funções matemáticas que descrevem como os valores armazenados nos nós são interpolados no domínio do elemento finito;
- um objeto do tipo **ProblemDriver**, responsável por conduzir os cálculos das matrizes e vetores que representam o comportamento do elemento finito;
- uma lista de degenerações, do tipo **Degeneration**. As degenerações contêm as informações que são perdidas ao descrever o espaço originalmente tridimensional de forma simplificada;
- um mapa de valores armazenados no elemento que contém, por exemplo, as ações externas que podem atuar sobre o contorno do elemento finito.

A classe abstrata **Element** contém uma especificação ampla de métodos, e por isso somente os que têm maior relevância a esse trabalho serão mencionados, sendo agrupados conforme as funcionalidades listadas a seguir:

- Obtenção das matrizes que descrevem o comportamento do elemento finito, tanto no regime linear como no regime não-linear. Destacam-se os métodos *getB()*, *getC()* e *getIncrementalC()*, que basicamente delegam esse cálculo ao

objeto `ProblemDriver` do próprio elemento finito, generalizando assim o processo para qualquer tipo de elemento;

- Obtenção dos carregamentos nodais equivalentes, pelo método `getE()`, e da resposta interna do elemento, pelo método `getF()`;
- Obtenção do valor da variável interna e da variável dual interna em um ponto no interior do elemento. Citam-se os métodos `getPointInternalVariables()` e `getMpInternalVariables()`, e `getPointDualInternalVariables()` e `getMpDualInternalVariables()`;
- Obtenção dos valores da variável de estado nos nós do elemento, que é realizada pelo método `getStateVariables()`;
- Inicialização dos objetos das classes herdeiras de `Element`, efetuada pelo método `init()`;

As classes herdeiras de `Element` podem ser divididas em duas categorias: elementos clássicos e elementos paramétricos. No primeiro grupo, a integração inerente ao cálculo das matrizes e vetores que descrevem o comportamento do elemento finito e das ações que atuam sobre ele é feita previamente à implementação, de forma analítica. As implementações de elementos de pórtico e de placa fina realizadas no `INSANE`, `FrameElement` e `ThinPlateElement` respectivamente, se referem a formulações clássicas. No segundo grupo, que contém os elementos paramétricos, a integração é realizada numericamente em tempo de execução. As classes que implementam formulações paramétricas são organizadas conforme a geometria do elemento e a técnica de discretização em que ele pode ser empregado.

Como esse trabalho se concentra na formulação paramétrica do MEF, um enfoque maior será dado à classe `ParametricElement` e suas herdeiras. Essas classes possuem como atributo adicional a ordem de integração, além de sobrecarregarem alguns métodos da superclasse `Element`.

Em relação à análise das classes `Element` e `ParametricElement`, podem ser ressaltados os pontos:

- Os elementos possuem referências a entidades que têm comportamento dependente da física do problema sendo resolvido, tais como objetos das classes `AnalysisModel`, `ProblemDriver` e `ConstitutiveModel`. No caso da análise multifísica utilizar o mesmo modelo discreto para o cálculo sequencial em cada uma das físicas, deve ser realizada uma reconfiguração dessas referências a fim de que elas apontem para objetos coerentes com a física corrente. Ao final do processo de reconfiguração, o elemento deve ser reinicializado para que as novas referências sejam atualizadas nas degenerações;
- Os métodos de cálculo da variável interna ou da variável dual interna devem ser modificados para contemplarem a presença de deformações oriundas da variação de temperatura. Como descrito na Equação 3.84, as deformações térmicas devem ser abatidas durante o cálculo das tensões, visto que elas são provenientes da parcela de deformação associada exclusivamente às ações mecânicas.

Duas opções são possíveis para mitigar a questão exposta acima, sendo uma delas remover as deformações devidas à variação de temperatura da deformação total nos métodos `getPointInternalVariables()` e `getMpInternalVariables()` e suas respectivas versões implementadas na classe `ParametricElement`, e a outra é remover a deformação térmica da deformação total durante o cálculo das tensões nos métodos `getPointDualInternalVariables()` e `getMpDualInternalVariables()` e suas versões implementadas na classe `ParametricElement`. Reforça-se que a modificação deve ser realizada somente em uma das opções, já que a implementação em ambas causaria a subtração das deformações térmicas duas vezes.

- Como foi discutido na seção 3.3.1, a ordem dos integrandos dos termos das

matrizes que descrevem o comportamento do elemento finito pode aumentar quando houver dependência entre as propriedades do material e a temperatura, tanto na análise térmica como na análise mecânica. Este fenômeno deve ser tratado durante a implementação da análise multifísica.

A.4.3 Classe abstrata `AnalysisModel`

A classe abstrata `AnalysisModel` é implementada por classes representantes dos diversos modelos de análise, conforme mostrado no diagrama de classes da Figura A.9.

De uma forma geral, as classes derivadas `AnalysisModel` não possuem atributos e seus métodos têm caráter informativo para proporcionar uma maior abstração dos algoritmos nas classes de cálculo, como os algoritmos nos métodos de classes derivadas de `ProblemDriver`. Exemplos de métodos informativos são os que retornam o número de graus de liberdade por nó, métodos que informam os nomes dados aos graus de liberdade ou métodos que informam o número de direções espaciais possíveis para o modelo corrente.

Além dos métodos informativos, alguns métodos nas classes herdeiras de `AnalysisModel` realizam operações de cálculo, quando essas são dependentes do modelo de análise em questão. Citam-se como exemplo o cálculo da matriz das derivadas das funções de interpolação no método `getInternalVariablesOperator()` e o cálculo da matriz jacobiana no método `getJacobian()`.

Previamente à realização desse trabalho haviam quatro classes herdeiras de `AnalysisModel` implementados para problemas de transferência de calor, sendo uma delas abstrata e as demais utilizadas para representar o PTC nos espaços uni, bi e tri-dimensional. Todavia, não havia uma classe implementada para a representação de problemas de transferência de calor que possuam axissimetria, sendo portanto recomendada a introdução dessa classe no núcleo numérico.

MEF. Menciona-se, para esse caso, as classes `BemProblemDriver`, `GFemParametric`, `HpCloudParametric` e `MfreeParametric`;

Previamente a esse trabalho já havia sido realizada a implementação de uma herdeira de `ProblemDriver` para problemas de transferência de calor lineares e em regime permanente. Essa implementação foi realizada na classe `HeatTransferPd`.

Para contemplar problemas de transferência de calor fisicamente não-lineares, como proposto nesse trabalho, deve-se estender a classe `HeatTransferPd` criando uma versão especializada que trate da não-linearidade.

Adicionalmente, para lidar com problemas da mecânica dos sólidos nos quais as propriedades físicas do material sejam influenciadas pela temperatura, deve-se estender a classe `Parametric` incluindo essa particularidade nos cálculos.

Ressalta-se que o problema de condução de calor fisicamente não-linear e o problema da mecânica dos sólidos dependente da temperatura devem considerar o aumento da ordem do integrando durante o cálculo da matriz de condutividade térmica e do cálculo do carregamento de variação de temperatura, como enfatizado na seção 3.3.1 dessa dissertação de mestrado.

A.4.5 Classe abstrata `ConstitutiveModel`

A classe abstrata `ConstitutiveModel` e suas herdeiras têm como principal objetivo representar a relação entre a variável interna e a variável dual interna. Seu diagrama de classes é mostrado de forma simplificada na Figura A.11, onde podem ser observadas classes para a descrição constitutiva de meios fissurados, de meios elastoplásticos, de meios contínuos de alta ordem, de meios elastoplásticos, entre outros.

Os principais métodos presentes em `ConstitutiveModel` têm como propósito:

- Retornar o operador constitutivo, seja ele secante ou tangente;
- Retornar o valor da variável dual interna a partir do valor da variável interna.

No caso particular da transferência de calor em corpos sólidos isotrópicos, o operador constitutivo secante é uma matriz diagonal cujos termos não nulos são a condutividade térmica do material. Quando a dependência entre a condutividade térmica e a temperatura pode ser expressa por uma relação polinomial, o operador constitutivo tangente é a matriz diagonal que contém em seus termos não nulos a derivada do referido polinômio avaliada na temperatura desejada, Equação 3.64.

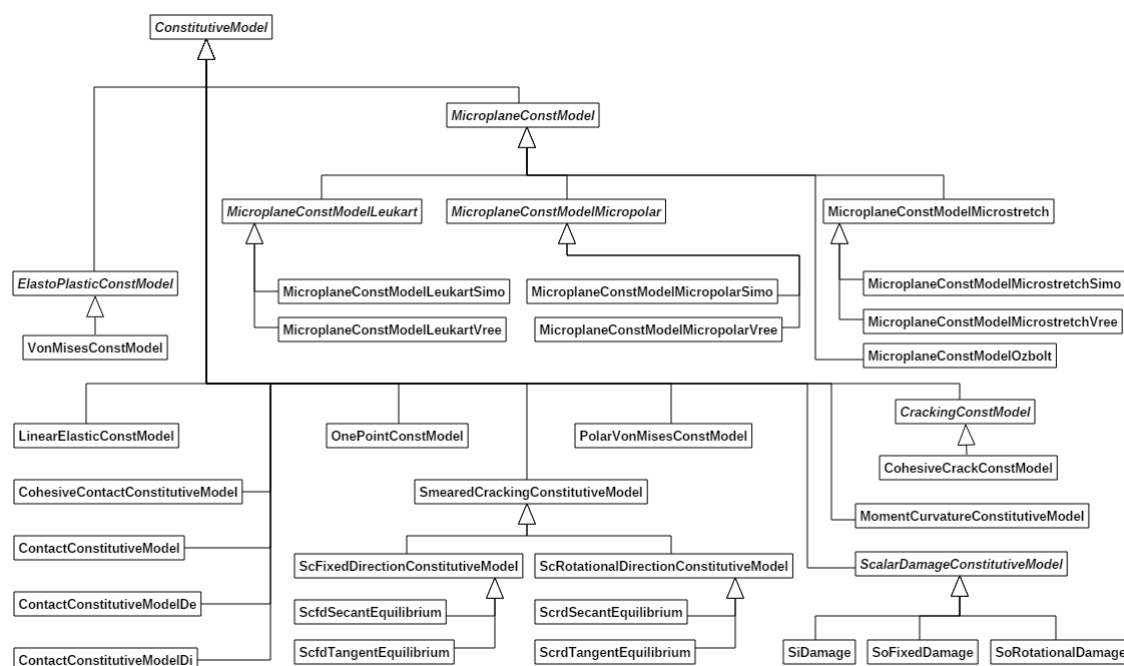


Figura A.11: Diagrama de classe para `ConstitutiveModel`.

Para representar a lei constitutiva em problemas de condução de calor, torna-se necessário implementar um modelo constitutivo dedicado a essa aplicação, visto que a relação constitutiva descrita pela Lei de Fourier contempla um sinal negativo para enfatizar que o fluxo de calor ocorre na direção de decaimento da temperatura. Além disso, torna-se necessário implementar métodos para lidar com problemas de condução de calor não-lineares.

Ressalta-se que as implementações especializadas de `ConstitutiveModel` se relacionam com objetos do tipo `AnalysisModel` e `Material` a fim de assegurar um nível elevado de abstração.

A.4.6 Classe abstrata **Degeneration**

Durante a modelagem, os corpos sólidos que naturalmente são tridimensionais podem ser descritos de formas simplificadas. Estas simplificações podem se referir, por exemplo, à modelagem de uma placa a partir do seu plano médio ou à modelagem de uma barra pela sua linha de centro. Ao realizar esse tipo de simplificação, diz-se que a geometria original é degenerada. Desta forma, um dos objetivos das classes herdeiras de **Degeneration** consiste em armazenar as informações referentes às características geométricas não modeladas, como a espessura de uma placa ou a seção transversal de uma barra, entre outras possibilidades. Um outro objetivo da classe **Degeneration** é conter informações relacionadas à multiplicidade de materiais que podem ocupar um ponto da geometria degenerada do corpo, como ocorre em laminados compostos que possam diversas camadas.

Além dessa primeira aproximação, pode-se ainda subdividir a geometria previamente degenerada durante a modelagem em alguns pontos discretos. Essa subdivisão é realizada, por exemplo, na formulação paramétrica do MEF quando se realiza a integração numérica das matrizes que determinam o comportamento do elemento. Nesse caso, o comportamento do elemento finito é dado pela soma das contribuições de cada ponto de integração multiplicadas pelo respectivo fator-peso.

Como as propriedades físicas do material podem variar no interior do elemento finito e também podem eventualmente sofrer alteração de maneira não uniforme no interior do mesmo ao longo da análise, cada um desses pontos é inicializado como um objeto do tipo **Degeneration** independente. Para conterem as informações referentes à integração numérica em si, essas degenerações possuem um objeto do tipo **Representation**.

Além de armazenarem as coordenadas cartesianas e naturais do ponto de integração e o seu respectivo fator-peso, um objeto do tipo **Representation** também possui uma referência a um modelo de análise, a um modelo constitutivo e possui dois mapas de variáveis constitutivas para acompanhar a evolução histórica do

comportamento físico do material em análises não-lineares.

A análise da hierarquia da classe `Degeneration`, mostrada na Figura A.12, indicou que não é necessário modificar a sua implementação atual nem introduzir novas classes nesse pacote para comportar a solução termomecânica almejada nesse trabalho.

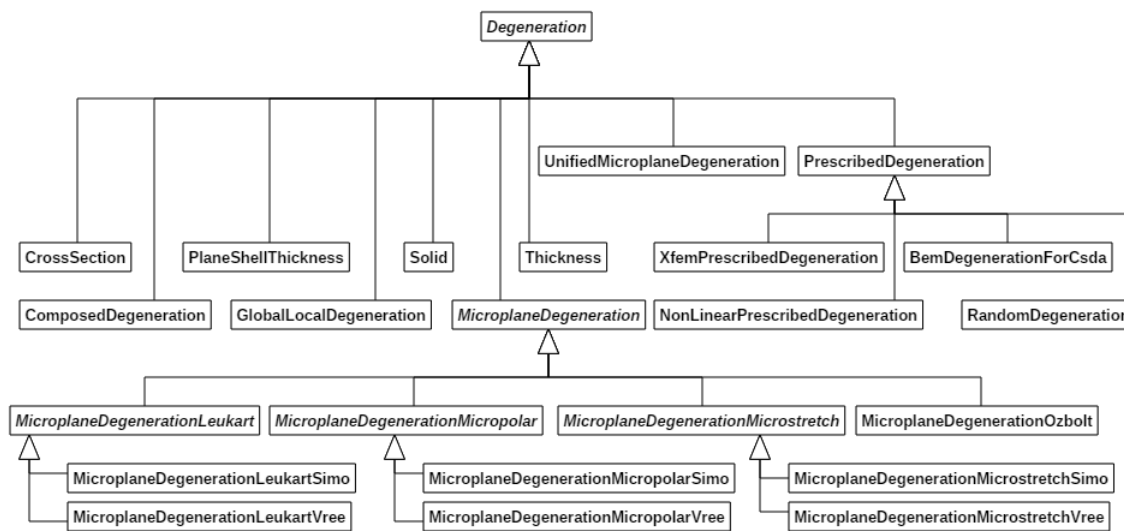


Figura A.12: Diagrama de classe para `Degeneration`.

A.4.7 Classe abstrata `Material`

A representação das propriedades físicas do meio modelado é feita por objetos de classes herdeiras da classe abstrata `Material`, cujo diagrama de classes é mostrado na Figura A.13. Esses objetos possuem um identificador único, ou rótulo, e um mapa de valores cujas chaves são cadeias de caracteres, usualmente denominadas *strings*, pré-definidas para cada uma das propriedades físicas representáveis. Dentre elas, pode-se citar o módulo de elasticidade, o coeficiente de Poisson, a condutividade térmica, entre outras.

Além de armazenarem essas informações, objetos do tipo `Material` dispõem de métodos para retornarem as propriedades físicas na sua forma secante ou tangente em diversos formatos e casos de aplicação.

A análise da estrutura hierárquica de `Material` indicou a necessidade de criar novas chaves para a condutividade térmica e para o calor específico no mapa que contém os valores das propriedades físicas. Também se identificou a necessidade de criar novas entidades para agregar a funcionalidade de possuir dependência em relação à temperatura a objetos do tipo `Material`, necessária em problemas de condução de calor fisicamente não-lineares, onde a condutividade térmica depende da temperatura buscada.

Ressalta-se que a entidade responsável por representar a dependência de propriedades físicas do material em relação à temperatura deve ser genérica o suficiente para ser utilizada no âmbito da mecânica dos sólidos quando as propriedades mecânicas do material possuírem esse comportamento.

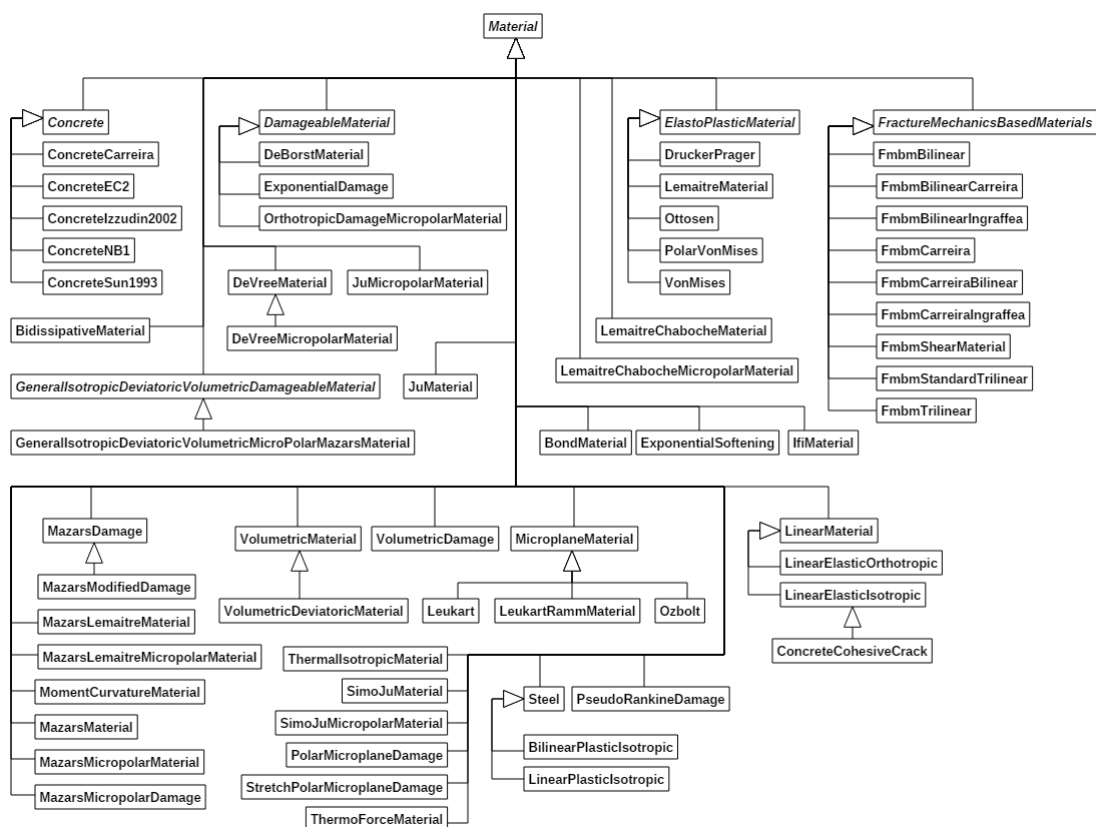


Figura A.13: Diagrama de classe para `Material`.

A.4.8 Classe abstrata Shape

As classes herdeiras da classe abstrata **Shape** mostradas na Figura A.14 podem ser divididas em três grandes grupos: funções de interpolação descritas no sistema de coordenadas cartesiano, funções descritas no sistema de coordenadas naturais e funções de enriquecimento para o caso do método do Método dos Elementos Finitos Generalizados. O diagrama de classes de **Shape** é mostrado a seguir, e como funcionalidades principais tem-se a avaliação, no ponto desejado, dos valores das funções de interpolação, assim como dos valores das primeiras e das segundas derivadas dessas funções. Os objetos do tipo **Shape** também contém métodos que fornecem informações básicas sobre si.

As funções definidas no sistema de coordenadas natural dos elementos finitos são implementadas em diversas classes especializadas conforme a geometria dos elementos, o número de nós que as representam e eventualmente pelos tipos de análise aos quais são aplicáveis.

Durante a análise do pacote **shape** pôde-se notar a existência de um método que informa o número de pontos de integração necessário para calcular com exatidão a integral da derivada da função interpoladora multiplicada por ela mesma, denominado *getDerivedIntegrationOrder()*. Este método é útil, por exemplo, para determinar o número de pontos de integração necessários para o cálculo da matriz de rigidez em problemas da mecânica dos sólidos nos quais a matriz constitutiva não contenha termos polinomiais, situação esta que é usual.

Como neste trabalho a condutividade térmica pode variar com a temperatura de forma polinomial, e a temperatura por sua vez varia no interior do domínio segundo a interpolação dos valores nodais realizada pela própria função de interpolação, os termos da matriz constitutiva podem possuir um formato polinomial. Para integrar a matriz de condutividade com exatidão nos problemas de condução de calor fisicamente não-lineares, é necessário conhecer a ordem polinomial das funções de

interpolação para que o número correto de pontos de integração possa ser calculado automaticamente, o que justifica a introdução de um novo método para tal finalidade.

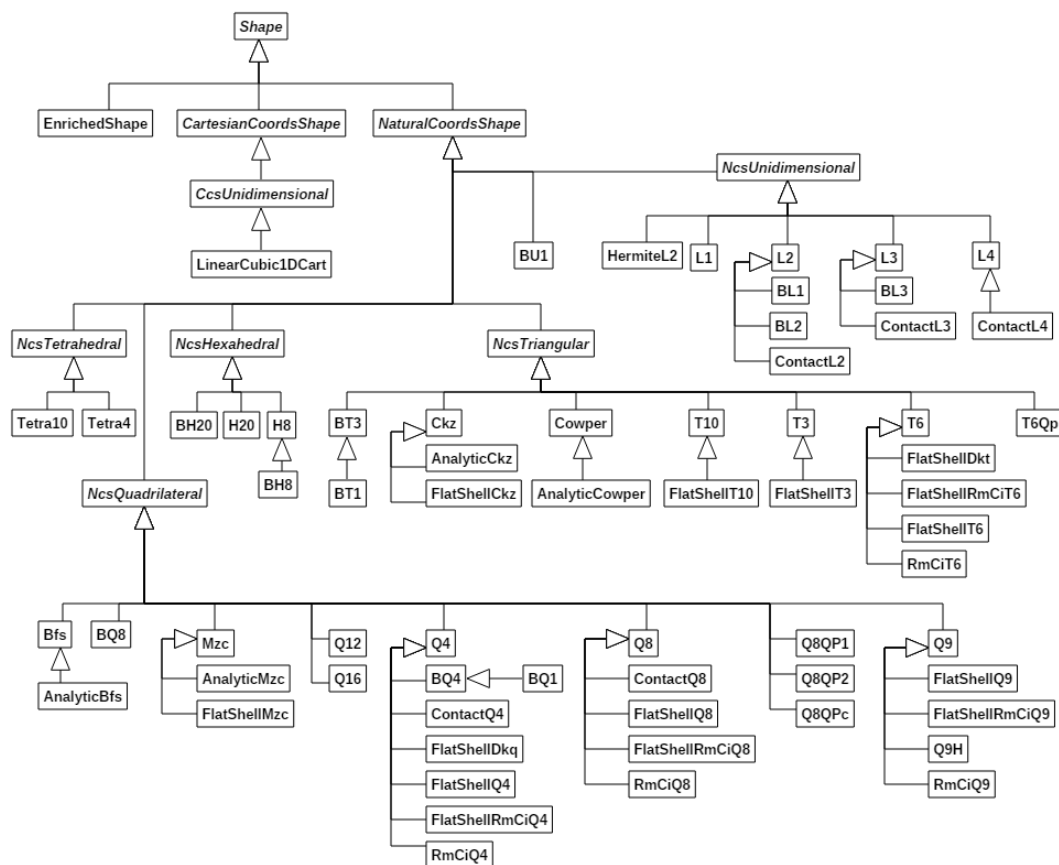


Figura A.14: Diagrama de classe para Shape.

A.4.9 Pacote load

As ações externas que atuam sobre o domínio discretizado são armazenadas em objetos do tipo `FemLoading`, que contém mapas referentes às ações definidas diretamente sobre os nós e às ações que atuam no domínio do elemento, seja ela concentrada em um ponto, representada pela classe `PointLoad`, distribuída sobre uma linha, classe `ElementLineLoad`, sobre uma área, `ElementAreaLoad`, ou uma ação que atue em todo o volume, descrita pela classe `ElementVolumeLoad`. Essas classes estão mostradas na Figura A.15.

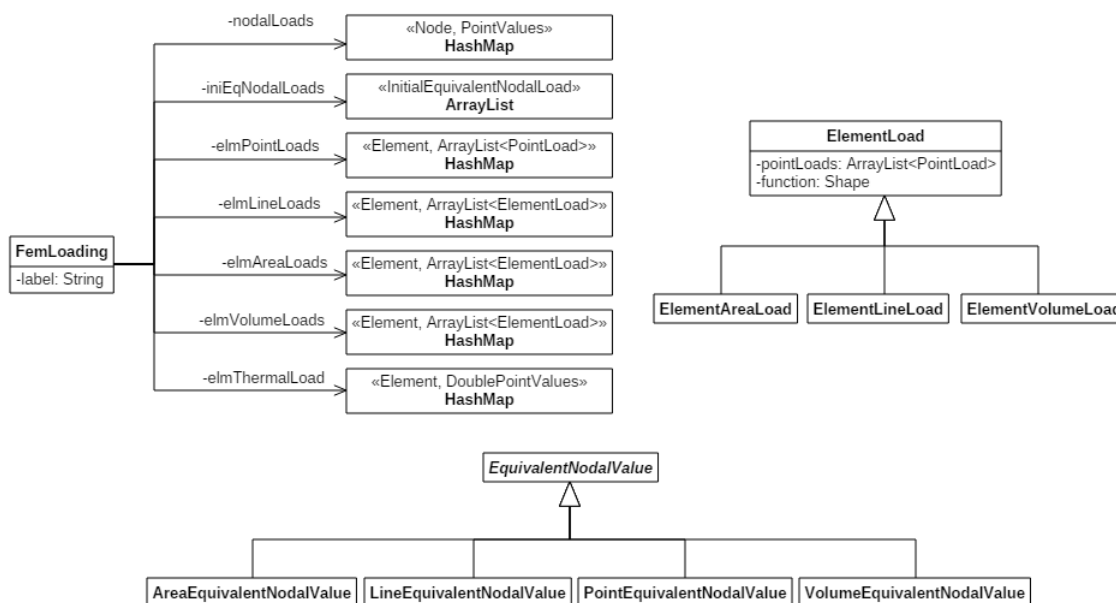


Figura A.15: Diagrama UML das classe contidas no pacote load.

A classe `FemLoading` também continha mapas para representar carregamentos de deformação inicial e carregamentos de variação de temperatura conforme o domínio espacial do elemento finito, incluídos por Botelho et al. (2015) previamente a essa dissertação de mestrado.

Ressalta-se que no INASNE o armazenamento das informações relativas ao carregamento é independente do cálculo do carregamento nodal equivalente. Para este último, são utilizados objetos do tipo `EquivalentNodalValue`, que são implementados nas formas especializadas para os carregamentos sobre pontos, linhas, superfícies ou no volume do elemento finito. No trabalho publicado por Botelho et al. (2015) foram introduzidas classes para o cálculo dos carregamentos nodais equivalentes devidos às deformações iniciais e à variação de temperatura.

Na análise do projeto orientado a objetos do pacote `load` observou-se a necessidade modificar o método de cálculo da deformação ocasionada pela variação da temperatura para que ele receba como parâmetros a temperatura de referência e a nova temperatura, em vez de receber somente a variação da temperatura em um único parâmetro. Essa modificação se mostra necessária quando é considerada a

dependência entre o coeficiente de expansão térmica e a temperatura, visto que o cálculo da deformação térmica contempla diferenças entre potências da nova temperatura e da temperatura de referência, Equação 3.103. Como a implementação prévia era restrita ao regime estacionário em meios com comportamento linear e independente da temperatura, não era necessário utilizar dois parâmetros.

A alteração descrita no parágrafo anterior implica em modificar a forma de cálculo de parcelas proporcionais do carregamento de temperatura. Quando a análise mecânica é fisicamente não-linear, os carregamentos são introduzidos de forma incremental durante a análise. Nesse processo, a variação total da temperatura é introduzida gradativamente juntamente aos demais carregamentos típicos da mecânica dos sólidos, e para isso torna-se necessário obter suas parcelas proporcionais. Para calculá-las corretamente, deve-se sobrecarregar o método *scale()* na classe especializada que descreve esse tipo de carregamento. Na versão sobrecarregada, a temperatura de referência é mantida inalterada e a temperatura final é modificada de tal forma que se obtenha uma variação de temperatura dada pelo seu valor original multiplicada pelo fator de escala.

Outro ponto observado se refere à forma como os carregamentos de variação de temperatura são utilizados após a automação do cálculo termomecânico, proposta nessa dissertação. Como esses carregamentos serão gerados automaticamente a partir dos resultados da análise térmica, eles sempre atuarão sobre toda a geometria do elemento. Essa consideração dispensa a implementação de uma classe diferente para cada domínio espacial, como constava na implementação detalhada em Botelho et al. (2015), visto que a geometria do carregamento é dada pela geometria do elemento finito ao qual ele é atribuído.

A.4.10 Classe abstrata `ScalarFunction`

As classes herdeiras de `ScalarFunction`, mostradas no diagrama de classes da Figura A.16, são utilizadas no contexto desse trabalho para descreverem a variação

temporal das condições de contorno. Na implementação atual, as funções escalares são associadas aos casos de carregamento, descritos por objetos do tipo `LoadCase`, que podem ser combinados em objetos do tipo `LoadCombination`.

Percebe-se na análise dessa metodologia a necessidade de também viabilizar a atribuição de uma função escalar à valores prescritos da variável de estado, o que implica em criar classes para representar os casos de condição de contorno essencial e classes para representar as combinações de casos desse tipo de condição de contorno.

A expansão acima descrita permite, por exemplo, obter as distribuições de temperaturas em um corpo para diferentes valores de temperaturas conhecidos em diferentes regiões. Outro caso de aplicação prática seria prescrever valores de temperatura conhecidos que se modifiquem ao longo do tempo em certas regiões do corpo para calcular como o campo de temperatura em todo o domínio se comporta ao longo do tempo.

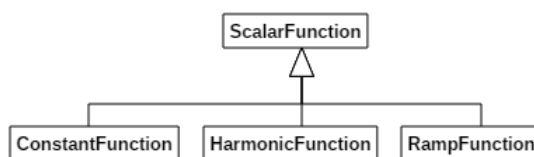


Figura A.16: Diagrama de classe para `ScalarFunction`.

No que se refere à análise das classes do tipo `ScalarFunction`, não são necessárias modificações ou expansões.

A.5 Interface Persistence

A interface `Persistence` basicamente prescreve duas funcionalidades: a de preenchimento da estrutura de dados do INSANE a partir de um arquivo de entrada, e a geração de um arquivo de saída a partir dos dados armazenados nessa estrutura de dados.

Neste trabalho, utilizou-se a classe `PersistenceAsXml`, que implementa a interface `Persistence` realizando as etapas de leitura e escrita em arquivos de texto que seguem os moldes da linguagem XML - eXtensible Markaup Language.

A linguagem XML é uma linguagem padronizada pela W3C (2019), amplamente utilizada para a representação de dados. Sua estrutura é bastante flexível tanto em relação à maneira de representar as informações, por permitir que o usuário crie seus próprios tipos de dados, como em relação à sua portabilidade, já que é um formato que não depende das plataformas de hardware ou de software. Contudo, a linguagem estabelece uma sintaxe a ser seguida pelos arquivos XML o que permite sua utilização como padrão para arquivos de entrada e saída de dados em programas de computador.

As informações em um arquivo XML são demarcadas através de marcas, ou tags, sendo o conteúdo demarcado por uma tag chamado elemento. O arquivo XML é estruturado na forma de uma árvore, na qual existe um elemento raiz que contém todos os elementos do arquivo. Um elemento em um arquivo XML, além de ser filho do elemento raiz, pode conter outros elementos.

A padronização do arquivo XML para a entrada e a saída de dados do INSANE é definida em um arquivo de esquema, denominado *insane.xsd*, que no momento da redação dessa dissertação é armazenado no projeto *ui.rich.full* do ambiente de desenvolvimento do INSANE.

Devido a essa estruturação em árvore do arquivo XML, a leitura e a escrita são implementadas de forma segmentada em vários métodos na classe `PersistenceAsXml`, onde cada método é responsável ou pela leitura ou pela escrita referente a um dos itens da estrutura de dados do INSANE.

Quanto à análise da classe `PersistenceAsXml`, nota-se a existência de uma referência a um objeto do tipo `Model` e a um objeto do tipo `Solution`. Essas referências podem ser manipuladas ao longo da análise para a utilização de mais de uma classe de solução, bem como a leitura e escrita de informações em mais de um modelo de

elementos finitos.

Também se observa a presença de um contador inteiro para representar a combinação de carregamento corrente, e que o uso desse contador pressupõe que as combinações de carregamento são analisadas na ordem sequencial em que estão armazenadas na lista de combinações presente no objeto `Model`. Além disso, a cada vez que um arquivo de saída é gerado para uma combinação de carregamento, esse contador é automaticamente incrementado.

A implementação da persistência necessita de modificação para comportar o caso multifísico, que requer uma maior flexibilidade por lidar com mais de uma classe de solução, por ter combinações de carregamento em diferentes físicas e por lidar com diferentes modelos de elementos finitos, um para cada física. Sugere-se então uma melhoria do mecanismo de propagação de mudanças atualmente implementado para a geração do arquivo de saída. Nessa melhoria, propõe-se que esse mecanismo carregue um conjunto de informações complementar ao notificar a persistência por meio da chamada ao método `update()`. Dentre essas informações, devem constar ao mínimo a referência à classe de solução que gerou os resultados, a combinação de carregamento utilizada durante os cálculos e uma referência ao modelo ao qual os resultados se referem.

Como este trabalho propõe a introdução de novas entidades no núcleo numérico do INSANE, conforme discutido ao longo desse capítulo, também torna-se necessária a implementação de novos métodos na classe `PersistenceAsXml` e a devida adequação do arquivo de esquema `insane.xsd`.

Apêndice B

SOLUÇÕES ANALÍTICAS DOS EXEMPLOS DE VALIDAÇÃO

Neste apêndice serão apresentados os desenvolvimentos detalhados das soluções analíticas dos exemplos de validação apresentados no Capítulo 5 dessa dissertação de mestrado. No total, são discutidos oito problemas, quatro exclusivamente associados à transmissão de calor e quatro problemas termomecânicos. A Tabela 5.1 resume o conteúdo abordado em cada um desses testes de validação, e a apresentação detalhada de cada um desses exemplos pode ser encontrada nas respectivas seções do Capítulo 5;

B.1 Solução analítica do Exemplo 1 - Placa quadrada com temperaturas prescritas nas arestas

O primeiro exemplo desenvolvido nesse apêndice corresponde ao cálculo da distribuição de temperaturas em regime permanente na placa retangular de largura a e altura b , que é objeto de estudo da seção 5.1.1, e está mostrada nas figuras 5.1 e 5.2. Nessa placa, as faces superior e inferior são adiabáticas, bem como suas faces laterais. Uma das outras faces laterais se encontra a uma temperatura prescrita T_{AB} e a face

restante está exposta à troca de calor por convecção com um ambiente à temperatura T_∞ nula. O coeficiente convectivo é denotado por h_c . A solução analítica desse exemplo foi obtida de Carslaw e Jaeger (1959), e será mostrada detalhadamente a seguir.

Como o calor somente flui no plano da placa e não há geração de calor, pode-se reduzir o problema ao domínio bidimensional ABCD indicado na Figura 5.2. Além disso, a análise se resume ao regime permanente, reduzindo a equação da condução do calor ao formato:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, \quad 0 < x < a, \quad 0 < y < b \quad (\text{B.1})$$

complementada pelas condições de contorno:

- Superfícies adiabáticas nas faces laterais que contêm as arestas AD e CD;

$$\frac{\partial T}{\partial x} = 0, \quad x = 0, \quad 0 < y < b \quad (\text{B.2})$$

$$\frac{\partial T}{\partial y} = 0, \quad y = b, \quad 0 < x < a \quad (\text{B.3})$$

- Temperatura prescrita na face lateral que contém a aresta AB dada pela função $f(x) = T_{AB}$;

$$T = f(x), \quad y = 0, \quad 0 < x < a \quad (\text{B.4})$$

- Troca de calor por convecção na face lateral que contém a aresta BC. A condutividade térmica é k , o coeficiente de troca de calor por convecção é h_c e a temperatura do fluido é $T_\infty = 0$.

$$-k \frac{\partial T}{\partial x} = h_c T, \quad y = 0, \quad 0 < x < a \quad (\text{B.5})$$

Observa-se que a equação diferencial B.1 e as condições de contorno B.2 e B.3 são atendidas pela equação transcendental a seguir:

$$\cos(\alpha_n x) \cosh[\alpha_n (b - y)] \quad (\text{B.6})$$

Adicionalmente, a condição de contorno B.5 é atendida em $x = a$ se as constantes α_n na equação B.6 forem raízes da seguinte equação:

$$\alpha_n \tan(\alpha_n a) = h \quad (\text{B.7})$$

onde $h = h_c/k$. Para visualizar esse último requisito, basta substituir a solução geral B.6 na condição de contorno B.5, em $x = a$.

Escreve-se a solução completa do problema como uma combinação linear das infinitas funções $\cos(\alpha_n x) \cosh[\alpha_n (b - y)]$ que atendem à Equação B.1 e às condições de contorno do problema:

$$T(x, y) = \sum_{n=1}^{\infty} A_n X_n \quad (\text{B.8})$$

onde X_n são as funções:

$$X_n = \cos(\alpha_n x) \cosh[\alpha_n (b - y)] \quad (\text{B.9})$$

e os fatores A_n são determinados de forma a garantir que a distribuição de temperatura se reduza à função $f(x)$ em $y = 0$ e $0 < x < a$.

A forma de obter o valor de cada A_n se baseia em multiplicar todos os termos $T(x, y)$ por X_n e integrar, termo a termo, de $x = 0$ até $x = a$. Isso permite aproveitar as propriedades que são provadas em Carslaw e Jaeger (1959):

$$\int_0^a X_m X_n dx = 0, m \neq n \quad (\text{B.10})$$

$$\int_0^a X_m X_n dx \neq 0, m = n \quad (\text{B.11})$$

assim,

$$A_n \int_0^a X_n^2 dx = \int_0^a f(x) X_n dx$$

portanto:

$$A_n = \frac{\int_0^a f(x) X_n dx}{\int_0^a X_n^2 dx} \quad (\text{B.12})$$

Ressalta-se que durante o cálculo de A_n deve-se fazer $y = 0$ em X_n .

Desenvolvendo o denominador da Equação B.12, tem-se:

$$\int_0^a X_n^2 dx = \int_0^a \cos^2(\alpha_n x) \cosh^2(\alpha_n b) dx = \cosh^2(\alpha_n b) \left[\frac{a}{2} + \frac{\text{sen}(\alpha_n a) \cos(\alpha_n a)}{2\alpha_n} \right]$$

A equação acima pode ser simplificada utilizando a relação $\alpha_n \tan(\alpha_n a) = h$:

$$\int_0^a X_n^2 dx = \cosh^2(\alpha_n b) \left[\frac{(\alpha_n^2 + h^2)a + h}{2(h^2 + \alpha_n^2)} \right] \quad (\text{B.13})$$

Substituindo a Equação B.13 na Equação B.12:

$$A_n = \frac{2(h^2 + \alpha_n^2)}{[(\alpha_n^2 + h^2)a + h] \cosh(\alpha_n b)} \int_0^a f(x) \cos(\alpha_n x) dx \quad (\text{B.14})$$

Retornando a expressão para A_n à Equação B.8, obtém-se a solução completa:

$$T = \sum_{n=1}^{\infty} \frac{2(h^2 + \alpha_n^2) \cos(\alpha_n x) \cosh[\alpha_n(b - y)]}{[(\alpha_n^2 + h^2)a + h] \cosh(\alpha_n b)} \int_0^a f(x) \cos(\alpha_n x) dx \quad (\text{B.15})$$

Quando a distribuição de temperaturas prescritas sobre a face AB é constante com valor T_{AB} , tem-se:

$$T(x, y) = 2 h T_{AB} \sum_{n=1}^{\infty} \frac{\cos(\alpha_n x) \cosh[\alpha_n(b - y)]}{[(\alpha_n^2 + h^2)a + h] \cos(\alpha_n a) \cosh(\alpha_n b)} \quad (\text{B.16})$$

B.2 Solução analítica do Exemplo 2 - Esfera oca com temperaturas prescritas nas superfícies

Esse exemplo aborda a obtenção do campo de temperaturas no regime permanente em uma esfera oca na qual a condutividade térmica possui uma dependência em relação à temperatura segundo um polinômio de primeiro grau. Nessa esfera, a temperatura T_a é prescrita na superfície interna, cujo raio é R_a , e a temperatura T_b é prescrita na face externa da esfera, cujo raio é R_b . Ademais, à temperatura T_a a condutividade térmica vale k_a , e à temperatura T_b , a condutividade térmica é k_b . A esfera em questão está ilustrada na Figura 5.5.

Dessa forma, a função que descreve a variação da condutividade térmica com a temperatura pode ser expressa como:

$$k(T) = k_a + (k_b - k_a) \frac{T - T_a}{T_b - T_a} \quad (\text{B.17})$$

A equação diferencial que rege o problema é a equação da condução do calor, que no regime permanente e em um sistema de coordenadas esférico se reduz ao formato mostrado abaixo devido à ausência de termos fonte e devido à simetria do problema:

$$\frac{1}{r^2} \frac{d}{dr} \left(k(T) r^2 \frac{dT}{dr} \right) = 0 \quad (\text{B.18})$$

Para obter a solução analítica, adapta-se a metodologia proposta por (Danish et al., 2011), onde no referido trabalho o autor apresenta um método de solução para a obtenção da distribuição de temperaturas em barras nas quais a temperatura é prescrita nas extremidades e a condutividade térmica varia linearmente com a temperatura.

Como o fluxo de calor em uma esfera oca nas condições previamente mencionadas é unidimensional, quando analisado em um sistema de coordenadas esféricos, pode-se adotar a mesma metodologia proposta por (Danish et al., 2011) apenas modificando a maneira de adimensionalizar a coordenada espacial, que no caso corresponde à

transformação da coordenada dimensional r para a coordenada adimensional ξ , como detalhado a seguir.

Com o objetivo de simplificar as condições de contorno, escolhe-se uma transformação de coordenadas tal que no raio interno R_a a coordenada adimensional valha 0 e no raio externo R_b essa coordenada adimensional tenha o valor 1. Essa transformação é dada pela equação:

$$\xi = \frac{r - R_a}{R_b - R_a} \quad (\text{B.19})$$

Assim como feito para a posição, adimensionaliza-se a temperatura de forma que à temperatura T_a a temperatura adimensional θ seja zero e à temperatura T_b a temperatura adimensional tenha o valor 1. Essa adimensionalização pode ser expressa por:

$$\theta = \frac{T - T_a}{T_b - T_a} \quad (\text{B.20})$$

Com o objetivo de simplificação, também é introduzido o parâmetro adimensional β , que fornece um indicativo da intensidade da não-linearidade imposta pela dependência da condutividade térmica com a temperatura:

$$\beta = \frac{k_b}{k_a} - 1 \quad (\text{B.21})$$

A Equação B.17 pode então ser reescrita utilizando o parâmetro β :

$$k(T) = k_a \left(1 + \beta \frac{T - T_a}{T_b - T_a} \right) \quad (\text{B.22})$$

Substituindo a temperatura adimensional θ na equação anterior, obtém-se:

$$k(T) = k_a(1 + \beta\theta) \quad (\text{B.23})$$

A seguir, são apresentados os passos para transformar a Equação B.18 das variáveis dimensionais r e T para as variáveis adimensionais ξ e θ .

Da equação B.19 obtém-se o diferencial dr em função de ξ :

$$dr = (R_b - R_a)d\xi \quad (\text{B.24})$$

O termo $\frac{dT}{dr}$ pode ser expandido pela regra da cadeia em:

$$\frac{dT}{dr} = \frac{dT}{d\theta} \frac{d\theta}{d\xi} \frac{d\xi}{dr} \quad (\text{B.25})$$

Da Equação B.20, obtém-se a relação:

$$\frac{dT}{d\theta} = T_b - T_a \quad (\text{B.26})$$

Portanto, o termo $\frac{dT}{dr}$ é dado por:

$$\frac{dT}{dr} = \frac{T_b - T_a}{R_b - R_a} \frac{d\theta}{d\xi} \quad (\text{B.27})$$

Substituindo as equações B.24, B.23 e B.27 na Equação B.18, obtém-se após simplificações:

$$\frac{1}{r^2} \frac{d}{d\xi} \left[r^2 (1 + \beta\theta) \frac{d\theta}{d\xi} \right] = 0 \quad (\text{B.28})$$

Como r é um número real e positivo, para que a Equação B.28 seja atendida para qualquer valor de r , tem-se que:

$$\frac{d}{d\xi} \left[r^2 (1 + \beta\theta) \frac{d\theta}{d\xi} \right] = 0 \quad (\text{B.29})$$

Integrando a equação anterior em relação a ξ , obtém-se:

$$[(R_b - R_a)^2 \xi^2 + 2\xi(R_b - R_a)R_a + R_a^2] [(1 + \beta\theta)\theta'] = C_1 \quad (\text{B.30})$$

onde C_1 é uma constante de integração e θ' representa a derivada de θ em relação a ξ .

Expandindo o produto e dividindo ambos os lados da equação por R_a^2 , obtém-se:

$$(1 + \beta\theta)\theta' = \frac{C_1}{\left(\frac{R_b - R_a}{R_a}\right)^2 \xi^2 + 2\left(\frac{R_b - R_a}{R_a}\right)\xi + 1} \quad (\text{B.31})$$

A equação acima pode ser simplificada como mostrado a seguir:

$$(1 + \beta\theta)\theta' = \frac{C_1}{\left(\frac{R_b - R_a}{R_a}\xi + 1\right)^2} \quad (\text{B.32})$$

Integrando a Equação B.32 em relação a ξ :

$$\theta + \beta\frac{\theta^2}{2} = \frac{C_1}{\left(\frac{R_b - R_a}{R_a}\xi + 1\right)} + C_2 \quad (\text{B.33})$$

As constantes C_1 e C_2 são obtidas substituindo as condições de contorno em $\xi = 0$ e $\xi = 1$:

$$\begin{aligned} \theta(\xi = 0) = 0 \quad \rightarrow \quad 0 &= C_1 + C_2 \\ \therefore C_2 &= -C_1 \end{aligned} \quad (\text{B.34})$$

$$\begin{aligned} \theta(\xi = 1) = 1 \quad \rightarrow \quad \left(1 + \frac{\beta}{2}\right) &= \frac{C_1}{\frac{R_b - R_a}{R_a} + 1} - C_1 \\ \therefore C_1 &= -\frac{R_b}{R_b - R_a} \left(1 + \frac{\beta}{2}\right) \end{aligned} \quad (\text{B.35})$$

Substituindo a Equação B.35 na Equação B.34:

$$C_2 = \frac{R_b}{R_b - R_a} \left(1 + \frac{\beta}{2}\right) \quad (\text{B.36})$$

Retornando os resultados das equações B.36 e B.35 à Equação B.33:

$$\theta + \beta\frac{\theta^2}{2} = \frac{R_b}{R_b - R_a} \left(1 + \frac{\beta}{2}\right) \left[1 - \frac{1}{\frac{R_b - R_a}{R_a}\xi + 1}\right] = c \quad (\text{B.37})$$

Como a função à direita da igualdade não contém termos em θ , ela será representada pelo símbolo c . A distribuição de temperaturas é obtida a partir da solução da equação polinomial:

$$\theta + \beta \frac{\theta^2}{2} - c = 0 \quad (\text{B.38})$$

cujas raízes são:

$$\theta = \frac{-1 \pm \sqrt{1 + 2\beta c}}{\beta} \quad (\text{B.39})$$

Na equação anterior o sinal negativo não é aplicável, visto que ele não atende à condição de contorno em $\xi = 0$. Nessa situação, o termo c é nulo e somente o sinal positivo leva ao resultado desejado $\theta = 0$ em $\xi = 0$. Substituindo a função c na Equação B.39, desprezando a solução associada ao sinal negativo anterior à raiz nessa mesma equação e transformando de volta das variáveis adimensionais ξ e θ para as variáveis dimensionais r e T , obtém-se:

$$T(r) = T_a + k_a (T_b - T_a) \left(\frac{-1 + \sqrt{1 + \left(\frac{k_b^2 - k_a^2}{k_a^2} \right) \left(\frac{R_b}{R_b - R_a} \right) \left(\frac{r - R_a}{r} \right)}}{k_b - k_a} \right) \quad (\text{B.40})$$

B.3 Solução analítica do Exemplo 3 - Barra com aquecimento senoidal nas extremidades

A solução analítica desenvolvida nessa seção se refere à barra mostrada na Figura 5.9, na qual prescrevem-se valores de temperaturas em suas extremidades. O restante da barra está impedido de trocar calor com a vizinhança e a temperatura nas extremidades cresce de maneira simétrica segundo uma função senoidal até atingir o primeiro pico dessa função, em um instante de tempo denotado por p . Após esse instante, a temperatura nas bordas é mantida constante e aguarda-se a estabilização do sistema. O comportamento descrito para a condição de contorno de temperatura prescrita é ilustrado na Figura 5.8. A distribuição de temperaturas inicial ao longo da barra é uniforme e nula.

Divide-se a solução do problema em duas etapas distintas, sendo a primeira etapa a obtenção do campo de temperaturas na barra ao longo do tempo durante o período em que as temperaturas nas bordas se elevam, e a segunda etapa o período de estabilização no qual as temperaturas nas bordas são mantidas constantes.

B.3.1 Período de elevação da temperatura nas bordas

Como a geometria, o comportamento material e as condições de contorno são simétricos, sabe-se de antemão que o gradiente de temperaturas, e conseqüentemente o fluxo de calor no centro da barra, são nulos. Dessa forma, pode-se transformar o problema original em um problema de buscar a distribuição de temperaturas somente na metade direita da barra sob a condição de contorno de fluxo de calor nulo na face esquerda desse problema modificado e sob a condição de temperatura variando segundo uma função senoidal na extremidade direita. Essa transformação é mostrada na figura a seguir.

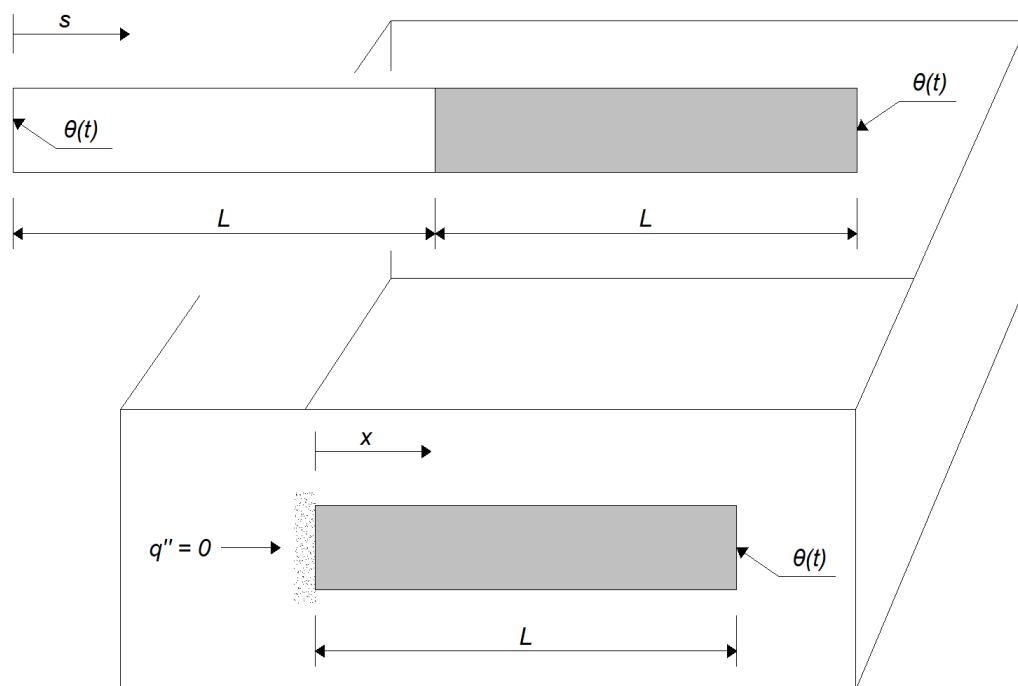


Figura B.1: Problema modificado com base na simetria do problema original.

A solução analítica desse problema segue a metodologia proposta por Groulx (2007), que é uma extensão dos trabalhos apresentados nos livros textos de Carslaw e Jaeger (1959) e de Özisk (1993). A solução é obtida a partir da aplicação do teorma de Duhamel e em seguida do método de separação de variáveis, como será detalhado a seguir.

O comprimento da barra no problema modificado é L , o fluxo e calor na face esquerda é q'' , a condutividade térmica é k , a massa específica é ρ , o calor específico é c_p e a difusividade térmica é denotada por α . A posição ao longo da barra, medida em relação à face esquerda, é indicada por x .

Para aplicar o Teorema de Duhamel, primeiramente obtém-se a solução do problema de valor de contorno contorno:

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{1}{\alpha} \frac{\partial \phi}{\partial t} \quad (\text{B.41})$$

$$x = 0, t \geq 0 \quad \left. \frac{\partial \phi}{\partial x} \right|_{x=0} = 0$$

$$x = L, t \geq 0 \quad \phi(t) = 1$$

$$t = 0, 0 \leq x \leq L \quad \phi(x) = 0$$

O problema previamente posto é solucionado primeiramente obtendo a resposta homogênea e em seguida a resposta particular:

$$\phi = \phi_h + \phi_p \quad (\text{B.42})$$

A solução homogênea desse problema é:

$$\phi_h = 1 \quad (\text{B.43})$$

A solução particular é (Groulx, 2007):

$$\phi_p = \frac{4}{\pi} \sum_{m=1}^{\infty} \frac{(-1)^m}{2m-1} \cos(\beta_m x) e^{-\alpha \beta_m t} \quad (\text{B.44})$$

onde β_m é calculado pela relação:

$$\beta_m = (2m-1) \frac{\pi}{2L} \quad (\text{B.45})$$

Em seguida, aplica-se o Teorema de Duhamel para considerar a variação temporal da condição de contorno de temperatura prescrita na extremidade direita da barra. Essa condição de contorno no período de tempo $0 \leq t \leq p$ é representada pela função harmônica:

$$T(x=L, t) = \theta \operatorname{sen} \left(\frac{\pi t}{2p} \right) \quad (\text{B.46})$$

onde θ é a temperatura a ser atingida durante o período de elevação da temperatura nas bordas da barra e p é o período de tempo para que ocorra essa elevação da temperatura.

Aplicando o teorema de Duhamel, obtém-se:

$$T(x, t) = \int_0^t \left[1 + \frac{4}{\pi} \sum_{m=1}^{\infty} \frac{(-1)^m}{2m-1} \cos(\beta_m x) e^{-\alpha \beta_m \tau} \right] \left(\frac{\pi}{2p} \right) \cos \left(\frac{\pi \tau}{2p} \right) d\tau \quad (\text{B.47})$$

Integrando, chega-se à distribuição de temperaturas ao longo do período de subida, referente ao problema modificado:

$$T(x, t) = \theta \operatorname{sen} \left(\frac{\pi t}{2p} \right) + \frac{2\theta}{p} \sum_{m=1}^{\infty} \left\{ \frac{(-1)^m}{(2m-1)} \cdot \frac{\cos(\beta_m x)}{(\alpha \beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2} \left[\alpha \beta_m^2 \cos \left(\frac{\pi t}{2p} \right) + \left(\frac{\pi}{2p} \right) \operatorname{sen} \left(\frac{\pi t}{2p} \right) - \alpha \beta_m^2 e^{(-\alpha \beta_m^2 t)} \right] \right\} \quad (\text{B.48})$$

Para referir à geometria original, na qual o sistema de coordenadas está posicionado na extremidade esquerda da barra, que possui comprimento total de $2L$ conforme mostrado na Figura 5.9, realiza-se a transformação de coordenadas:

$$x = |L - s| \quad 0 \leq s \leq 2L \quad (\text{B.49})$$

Substituindo a Equação B.49 na Equação B.48, obtém-se a distribuição de temperaturas ao longo de toda a barra:

$$T(s, t) = \theta \operatorname{sen} \left(\frac{\pi t}{2p} \right) + \frac{2\theta}{p} \sum_{m=1}^{\infty} \left\{ \frac{(-1)^m}{(2m-1)} \cdot \frac{\cos(\beta_m |L - s|)}{(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2} \left[\alpha\beta_m^2 \cos \left(\frac{\pi t}{2p} \right) + \left(\frac{\pi}{2p} \right) \operatorname{sen} \left(\frac{\pi t}{2p} \right) - \alpha\beta_m^2 e^{(-\alpha\beta_m^2 t)} \right] \right\} \quad (\text{B.50})$$

B.3.2 Período de uniformização da temperatura na barra

Na segunda parte da solução, realiza-se uma transformação de variáveis na variável temporal, de forma que o instante em que a elevação da temperatura se cessa nas extremidades da barra, $t = p$, corresponda a um instante inicial no problema reformulado.

A nova variável temporal η é então determinada pela relação:

$$\eta = t - p \quad t > p \quad (\text{B.51})$$

No instante inicial do problema reformulado, a distribuição de temperaturas é determinada pela Equação B.50 fazendo $t = p$:

$$T(s, t = p) = \theta + \frac{2\theta}{p} \sum_{m=1}^{\infty} \left\{ \frac{(-1)^m}{(2m-1)} \cdot \frac{\cos(\beta_m |L - s|)}{(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2} \left[\left(\frac{\pi}{2p} \right) - \alpha\beta_m^2 e^{(-\alpha\beta_m^2 p)} \right] \right\} \quad (\text{B.52})$$

A solução nessa situação foi obtida seguindo os procedimentos descritos em Boyce e DiPrima (2006). Reforça-se que no exemplo discutido nessa seção o comprimento da barra é $2L$. Para obtê-la, primeiramente é realizada uma modificação no problema para tornar as condições de contorno homogêneas. Isso é feito subtraindo de todo o

domínio a solução em regime permanente, que é uma distribuição uniforme no valor da temperatura θ prescrita nas extremidades.

Após as adequações para tornar as condições de contorno homogêneas, pode-se facilmente resolver o problema pelo método de separação de variáveis. Aplicando o método é obtida a solução a seguir, na qual a distribuição uniforme que havia sido subtraída já foi readicionada:

$$T(s, \eta) = \theta + \sum_{n=1}^{\infty} C_n e^{-n^2 \pi \alpha \eta / (4L^2)} \operatorname{sen} \left(\frac{n\pi s}{2L} \right) \quad (\text{B.53})$$

onde os coeficientes C_n são calculados pela relação:

$$C_n = \frac{1}{L} \int_0^{2L} T(s, \eta = 0) \operatorname{sen} \left(\frac{n\pi s}{2L} \right) \quad (\text{B.54})$$

na qual $T(s, \eta = 0)$ é dado pela Equação B.52.

Realizando a integração presente na Equação B.54 chega-se à seguinte expressão:

$$C_n = \sum_{m=1}^{\infty} A_m \frac{\cos(\beta_m L)}{2} \left[\frac{1 - \cos(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L} - \beta_m\right)} + \frac{1 - \cos(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L} + \beta_m\right)} \right] + \sum_{m=1}^{\infty} A_m \frac{\operatorname{sen}(\beta_m L)}{2} \left[\frac{\operatorname{sen}(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L} - \beta_m\right)} - \frac{\operatorname{sen}(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L} + \beta_m\right)} \right] \quad (\text{B.55})$$

onde A_m é:

$$A_m = \left(\frac{2\theta}{pL} \right) \left[\frac{(-1)^m}{(2m-1)} \right] \left[\frac{\left(\frac{\pi}{2p}\right) - \alpha\beta_m^2 e^{-\alpha\beta_m^2 p}}{(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2} \right] \quad (\text{B.56})$$

A solução final para o trecho de uniformização da temperatura na barra, correspondente ao período de tempo onde $t > p$ é dado pela Equação B.53, substituindo na mesma as equações B.51 e B.55. Ressalta-se a ocorrência de indeterminações nos termos das séries cujos limites são definidos conforme discutido na seção 5.1.3 dessa dissertação de mestrado.

B.4 Solução analítica do Exemplo 4 - Cilindro longo com convecção na superfície

Considere o cilindro mostrado na Figura 5.12, submetido à troca de calor por convecção em sua superfície com um meio à temperatura T_∞ nula. O coeficiente de troca de calor por convecção é h , a distribuição de temperatura inicial no cilindro é uniforme com valor T_0 e o cilindro é suficientemente longo para o fluxo de calor ocorra somente na direção radial. O raio do cilindro é denotado pelo símbolo a e o tempo decorrido após o instante inicial é representado por t . A condutividade térmica varia com a temperatura, porém essa dependência não deve ser expressiva para que as considerações apresentadas posteriormente nessa seção tenham validade.

A solução analítica aqui apresentada segue o expediente apresentado por Jahanian (1995). Todavia, como no trabalho de Jahanian (1995) o desenvolvimento teórico é apresentado de forma sucinta, utiliza-se como bibliografia complementar a formulação exposta por Noda (2014). Apesar de suas similaridades, esses trabalhos empregam diferentes funções durante a transformação de *Kirchhoff*, passo necessário para linearizar a equação diferencial da condução do calor. Além disso, essas referências também realizam considerações diferentes durante o processo de linearização do problema. Devido à sua maior simplicidade, optou-se pela solução apresentada por Jahanian (1995) como referência para o desenvolvimento ao longo dessa seção.

Quando a condutividade térmica possui dependência em relação à temperatura, a equação diferencial da condução do calor é não-linear e possui a forma:

$$\frac{1}{r} \frac{\partial}{\partial r} \left(\lambda(T) r \frac{\partial T(r, t)}{\partial r} \right) = \rho c_p \frac{\partial T(r, t)}{\partial t} \quad (\text{B.57})$$

A equação B.57 é uma versão reduzida da equação completa visto que, devido às características do problema, pode-se desprezar os gradientes de temperatura nas direções axial e circunferencial. Nessa equação, r é a posição radial, $\lambda(T)$ é a condutividade térmica dependente da temperatura, $T(r, t)$ é a temperatura na posição

r no instante de tempo t , ρ é a massa específica e c_p o calor específico do material.

Para viabilizar a solução, aplica-se a transformação de Kirchhoff (Jahanian, 1995) onde se mapeia a temperatura segundo a equação a seguir:

$$\phi = \frac{1}{K_0} \int_0^\theta K(\theta) d\theta \quad (\text{B.58})$$

Na equação acima, a função $K(\theta)$ representa a condutividade térmica da seguinte forma:

$$K(\theta) = K_0 K^*(\theta) \quad (\text{B.59})$$

onde K_0 é a parcela dimensional da função $K(\theta)$ e $K^*(\theta)$ a parcela adimensional dessa mesma função, dada pela equação:

$$K^*(\theta) = 1 - K_1 \theta \quad (\text{B.60})$$

na qual K_1 é um parâmetro adimensional.

O parâmetro θ é a temperatura adimensional, nesse problema definida como:

$$\theta = \frac{T_f - T}{T_f - T_0} \quad (\text{B.61})$$

sendo T_f a temperatura final atingida pelo corpo no regime permanente, e T_0 a temperatura inicial do corpo. Nesse problema a temperatura final é nula, visto que não há geração de energia e a troca de calor na superfície ocorre com um fluido à temperatura zero. Quanto à temperatura adimensional, também se observa que ela se encontra na faixa entre 1 e 0, onde 1 corresponde à distribuição de temperaturas uniforme no instante inicial e zero à distribuição de temperaturas uniforme no instante final.

Sabendo o valor de T_f , pode-se simplificar a expressão B.61 para a forma:

$$\theta = \frac{T}{T_0} \quad (\text{B.62})$$

A dependência da condutividade térmica em relação à temperatura T é descrita pela função linear:

$$\lambda(T) = \lambda_0 + \lambda_1 T \quad (\text{B.63})$$

onde λ_0 e λ_1 são constantes.

Substituindo B.62 em B.63 e igualando à B.59, obtém-se:

$$\lambda_0 + \lambda_1 T_0 \theta = K_0(1 - K_1 \theta) \quad (\text{B.64})$$

Uma análise comparativa dos termos que multiplicam θ e dos termos que não realizam essa multiplicação leva a:

$$K_0 = \lambda_0 \quad e \quad K_1 = -\frac{\lambda_1 T_0}{\lambda_0} \quad (\text{B.65})$$

Aplicando a regra da cadeia após a transformação de *Kirchoff*, a equação da condução do calor na variável transformada adquire o formato:

$$\frac{K(\theta)}{\rho c_p} \frac{1}{r} \frac{\partial}{\partial r} \left(\lambda(T) r \frac{\partial T(r, t)}{\partial r} \right) = \frac{\partial T(r, t)}{\partial t} \quad (\text{B.66})$$

Uma solução aproximada pode ser obtida considerando que a propriedade difusividade térmica, ao ser avaliada em relação à variável de temperatura transformada, apresentará um comportamento constante. Essa consideração é empregada na formulação apresentada nesse trabalho, e pode-se dessa forma tratar a Equação B.66 como uma forma linearizada. Uma solução mais sofisticada, onde esse comportamento da difusividade é tratado como uma função linear em relação à temperatura transformada é apresentado em Noda (2014).

O problema linearizado descrito pela Equação B.66 é complementado pela condição inicial e pelas condições de contorno transformadas:

$$\begin{cases} \phi = \phi_0 & \text{em } t = 0 \\ \frac{\partial \phi}{\partial r} + \frac{h}{K_0} \phi = 0 & \text{em } r = a \end{cases}$$

Para resolver a Equação B.66 considerando as condições de contorno acima descritas, aplica-se o método de separação de variáveis, conforme descrito por Noda et al. (2003) para um problema fisicamente linear dentro das mesmas condições.

$$\phi(r, t) = f(r)g(t) \quad (\text{B.67})$$

o que reduz a Equação B.66 a:

$$\frac{1}{\kappa g(t)} \frac{dg(t)}{dt} = \frac{1}{f(r)} \left[\frac{d^2 f(r)}{dr^2} + \frac{1}{r} \frac{df(r)}{dr} \right] = -\beta^2 \quad (\text{B.68})$$

onde κ é a difusividade térmica considerada constante em relação à temperatura transformada, mas dependente de θ . Assim, $\kappa = K(\theta)/(\rho c_p)$.

A Equação B.68 leva a duas equações diferenciais desacopladas:

$$\frac{dg(t)}{dt} + \beta^2 \kappa g(t) = 0 \quad (\text{B.69})$$

$$\frac{d^2}{dr^2} + \frac{1}{r} \frac{df(r)}{dr} + \beta^2 f(r) = 0 \quad (\text{B.70})$$

Observa-se que a Equação B.70 é a equação diferencial de Bessel de ordem zero (Watson, 1995) e que ela possui duas soluções particulares:

$$J_0(\beta r) \text{ e } Y_0(\beta r) \quad (\text{B.71})$$

A função $J_0(\beta r)$ é uma função Bessel de primeira espécie e de ordem zero, e a função $Y_0(\beta r)$ é uma função Bessel de segunda espécie e de ordem zero.

As soluções gerais das equações B.69 e B.70 são:

$$g(t) = E, \quad f(r) = F + G \ln(r) \text{ para } \beta = 0 \quad (\text{B.72})$$

$$g(t) = H e^{-\kappa \beta^2 t}, \quad f(r) = I J_0(\beta r) + J Y_0(\beta r) \text{ para } \beta \neq 0 \quad (\text{B.73})$$

Como as funções devem existir para valores arbitrários de s , escreve-se uma solução geral utilizando as soluções das equações B.72 e B.73:

$$\phi(r, t) = A + B \ln(r) + [C J_0(\beta r) + D Y_0(\beta r)] e^{-\kappa \beta^2 t} \quad (\text{B.74})$$

onde

$$A = EF, \quad B = EG, \quad C = HI, \quad D = HJ \quad (\text{B.75})$$

Como a função Bessel $Y_0(\beta r)$ e $\ln(r)$ devem ser finitas em $r = 0$, B e D devem ser nulos. A distribuição de temperaturas possui então a forma geral:

$$\phi^{(0)} = A + C J_0(\beta a) = 0 \quad (\text{B.76})$$

A substituição da Equação B.76 nas condições de contorno transformadas leva a $A = 0$ e a:

$$h J_0(\beta_k a) - \beta_k J_1(\beta_k a) = 0 \quad (\text{B.77})$$

Como a equação anterior possui um número infinito de soluções, a temperatura transformada pode ser desenvolvida como uma série infinita, denominada série de Bessel:

$$\phi = \sum_{k=1}^{\infty} A_k J_0(\beta_k r) e^{-\kappa \beta_k^2 t} \quad (\text{B.78})$$

Multiplicando ambos os lados da Equação B.78 por $r J_0(\beta_k r)$, integrando de 0 até a e utilizando as propriedades das integrações das funções Bessel, tem-se:

$$A_k = 2\phi_0 \left(\frac{ah}{K_0} \right) \left\{ \frac{1}{\left[\beta_k^2 a^2 + \left(\frac{ah}{K_0} \right)^2 \right] J_0(\beta_k a)} \right\} \quad (\text{B.79})$$

Para calcular o valor de ϕ_0 aplica-se a transformada de Kirchhoff à distribuição de temperaturas dada pela condição inicial. Como na condição inicial o valor da temperatura é T_0 , o valor correspondente da temperatura adimensional é $\theta = 1$

$$\phi_0 = \frac{1}{K_0} \int_0^1 [K_0(1 - K_1\theta)]d\theta = 1 - \frac{K_1}{2} \quad (\text{B.80})$$

Substituindo a Equação B.80 em B.79, e o resultado em B.78, obtém-se uma expressão para o campo de temperaturas transformado:

$$\phi = \frac{2ah}{K_0} \left(1 - \frac{1}{2}\right) \sum_{k=1}^{\infty} \frac{J_0(\beta_k r) e^{-\beta_k^2 \tau}}{\left[\beta_k^2 a^2 + \left(\frac{ah}{K_0}\right)^2\right] J_0(\beta_k a)} \quad (\text{B.81})$$

Pode-se então resolver a equação acima para valores específicos da constante τ , realizar a transformação de *Kirchhoff* inversa para obter a temperatura adimensional θ , e em seguida utilizar o valor de temperatura adimensional para calcular o instante de tempo que corresponde ao valor de temperatura calculado. Ao final, multiplica-se o valor de temperatura adimensional pela temperatura inicial para obter o valor de temperatura dimensional desejado. Esse procedimento é detalhado a seguir.

A transformação de *Kirchhoff* inversa desse problema é dada por:

$$\theta(r, \tau) = \frac{1}{K_1} \left(1 - \sqrt{1 - 2K_1\phi(r, \tau)}\right) \quad (\text{B.82})$$

Conhecido o valor de $\theta(r, \tau)$, pode-se calcular a parte adimensional da condutividade térmica dada pela equação B.60. O instante de tempo correspondente é então calculado como mostrado a seguir:

$$t = \tau \frac{\rho c_p}{K_0(1 - K_1\theta)} \quad (\text{B.83})$$

e a temperatura dimensional, resposta final do problema para a posição e instante de tempo calculado pela Equação B.83, é dada por $T(r, t) = \theta(r, t) \cdot T_0$.

B.5 Solução analítica do Exemplo 5 - Viga bi-apoiada sob gradiente térmico ao longo da altura

Nessa seção é apresentada a solução analítica de um problema termomecânico em regime permanente correspondente a uma viga bi-apoiada submetida à geração de calor em seu domínio. Essa solução foi obtida de Hetnarski e Eslami (2009).

O comprimento da viga em questão é denotado por L e sua seção transversal é retangular, constante, de largura b e altura a . Além do calor \dot{q} gerado uniformemente no domínio da viga, a temperatura é prescrita nas suas faces superior e inferior, sendo a temperatura na face superior referenciada por T_{sup} e a temperatura na face inferior por T_{inf} . As demais faces da viga estão impedidas de trocar calor com a vizinhança. A origem do sistema de coordenadas é disposta no centróide da face esquerda da viga, como indicado na Figura 5.15, e a viga parte de uma distribuição de temperaturas uniforme e nula.

Como as faces anterior e posterior, e as faces esquerda e direita são adiabáticas, o calor somente flui na direção vertical, paralela ao eixo y mostrado na Figura 5.15. A equação da condução do calor em regime permanente, para o caso particular de fluxo de calor unidirecional em regime permanente, se resume a:

$$\frac{\partial^2 T}{\partial y^2} = -\frac{\dot{q}}{k} \quad (\text{B.84})$$

sob as condições de contorno:

$$\begin{aligned} T(y = a/2) &= T_{sup} \\ T(y = -a/2) &= T_{inf} \end{aligned} \quad (\text{B.85})$$

Integrando duas vezes a Equação B.84 e substituindo as condições de contorno dadas pela Equação B.85, obtém-se a distribuição de temperaturas ao longo da altura da viga:

$$T(y) = -\frac{\dot{q}}{2k}y^2 + \frac{T_{sup} - T_{inf}}{a}y + \frac{T_{sup} + T_{inf}}{2} + \frac{qa^2}{8k} \quad (\text{B.86})$$

A parcela mecânica desse problema é resolvida como descrito em Hetnarski e Eslami (2009), onde se aplica a teoria de vigas de Bernoulli-Euler. Nessa teoria, para flexão ocorrendo no plano xy em torno do eixo z , tem-se o campo de deformações:

$$\epsilon_{xx}(y) = \epsilon_0 + \frac{y}{r_y} \quad (\text{B.87})$$

onde ϵ_0 é a deformação na direção x medida sobre o eixo neutro, $\epsilon(y)$ é a deformação na direção x medida fora do eixo neutro e r_y é o raio de giração da viga, medido na coordenada $y = 0$.

Considera-se uma lei constitutiva linear elástica na presença de deformações associadas à variação de temperatura:

$$\sigma_{xx} = E(\epsilon_{xx} - \gamma T) \quad (\text{B.88})$$

na qual E é o módulo de elasticidade, σ_{xx} é a tensão normal na direção axial da viga, γ é o coeficiente de expansão térmica e T a variação de temperatura ocorrida no ponto onde as deformações e as tensões são calculadas.

Substituindo a Equação B.87 na Equação B.88:

$$\sigma_{xx} = E\left(\epsilon_0 + \frac{y}{r_y} - \gamma T\right) \quad (\text{B.89})$$

Como a viga não está submetida a ações mecânicas externas, por questões de equilíbrio os esforços generalizados na seção transversal devem se anular:

$$\int_A \sigma_{xx} dA = 0 \quad (\text{B.90})$$

$$\int_A \sigma_{xx} y dA = 0 \quad (\text{B.91})$$

onde A é a área da seção transversal da viga.

Substituindo a Equação B.89 nas equações B.90 e B.91:

$$\epsilon_0 \int_A dA + \frac{1}{r_y} \int_A y dA = \int_A \gamma T dA \quad (\text{B.92})$$

$$\epsilon_0 \int_A y dA + \frac{1}{r_y} \int_A y^2 dA = \int_A \gamma T y dA \quad (\text{B.93})$$

Sabendo que $\int_A y dA = 0$ e que $\int_A y^2 dA$ é o momento de inércia I_z da seção transversal em relação ao eixo z , e introduzindo a definição dos esforços generalizados P_T e M_{Tz} associados à variação de temperatura:

$$P_T = \int_A E \gamma T dA \quad (\text{B.94})$$

$$M_{Tz} = \int_A E \gamma T y dA \quad (\text{B.95})$$

Tem-se das equações B.92 e B.93:

$$\epsilon_0 = \frac{P_T}{EA} \quad (\text{B.96})$$

$$r_y = \frac{EI_z}{M_{Tz}} \quad (\text{B.97})$$

Substituindo as equações B.96 e B.97 na Equação B.89, obtém-se:

$$\sigma_{xx} = -E \gamma T + \frac{P_T}{A} + \frac{M_{Tz} y}{I_z} \quad (\text{B.98})$$

Substituindo o campo de temperaturas da Equação B.86 nas equações B.94 e B.95 chega-se às expressões para P_T e M_{Tz}

$$P_T = E \gamma A \left(\frac{1}{12} \dot{q} a^2 + \frac{T_1 + T_2}{2} \right) \quad (\text{B.99})$$

$$M_{Tz} = \frac{E \gamma (T_1 - T_2) I_z}{\gamma} \quad (\text{B.100})$$

Substituindo as equações B.99 e B.100 na Equação B.98 obtém-se o campo de tensões ao longo da altura da viga:

$$\sigma_{xx} = E\gamma \frac{qa^2}{8k} \left[\left(\frac{2y}{a} \right)^2 - \frac{1}{3} \right] \quad (\text{B.101})$$

Segundo as hipóteses cinemáticas da teoria de Bernoulli-Euler em um regime de pequenos deslocamentos, tem-se a equação diferencial para a deflexão da viga:

$$-\frac{d^2y}{dx^2} = \frac{1}{r_y} \quad (\text{B.102})$$

Substituindo a expressão de r_y da Equação B.97 na Equação B.102,

$$\frac{d^2y}{dx^2} = -\frac{M_{Tz}}{EI_z} \quad (\text{B.103})$$

Integrando a equação anterior duas vezes e substituindo as condições de apoio simples nas extremidades da viga, obtém-se a flecha ao longo do comprimento da viga:

$$v(x) = \frac{\gamma(T_{sup} - T_{inf})}{2a} (xL - x^2) \quad (\text{B.104})$$

Na Equação B.104 a flecha é considerada positiva na direção negativa do eixo y do sistema de coordenadas no qual o problema é definido.

B.6 Solução analítica do Exemplo 6 - Barra bi-engastada sob variação de temperatura uniforme

Nesse exemplo é estudado o comportamento termomecânico da barra mostrada na 5.23, que parte de uma distribuição de temperaturas inicial nula, e após esse instante sofre a imposição de temperaturas iguais em suas extremidades com o valor

θ . Esse problema de valor de contorno é complementado pela definição de sua parcela mecânica, na qual as extremidades da barra são impedidas de se deslocarem.

A equação da condução do calor para um PTC linear, unidimensional, em regime permanente e sem geração de energia é:

$$\frac{d^2T}{ds^2} = 0 \quad (\text{B.105})$$

Integrando a Equação B.105 duas vezes, obtém-se:

$$T(s) = C_1s + C_2 \quad (\text{B.106})$$

Substituindo as condições de contorno $T(s = 0) = \theta$ e $T(s = 2L) = \theta$, obtém-se uma distribuição de temperaturas uniforme com valor θ , visto que $C_1 = 0$ e $C_2 = \theta$.

Inicia-se a solução da parcela mecânica estudando o comportamento da barra em seu regime mecânico linear. Para tal, divide-se o problema original em dois sub-problemas, cuja superposição reconstitui as condições de contorno do problema completo.

O primeiro sub-problema corresponde à barra submetida à variação de temperatura, impedida de se deslocar em sua extremidade esquerda e livre em sua extremidade direita, e o segundo corresponde à mesma barra, impedida de se deslocar em sua extremidade esquerda e submetida a um deslocamento na extremidade direita igual, em módulo, ao deslocamento na extremidade direita calculado no primeiro sub-problema, porém em no sentido oposto. Esses problemas são postos matematicamente a seguir:

Primeiro sub-problema:

$$\frac{d^2u_1}{ds^2} = 0 \quad (\text{B.107})$$

sob as condições de contorno:

$$u_1 = 0 \quad \text{para} \quad s = 0 \quad (\text{B.108})$$

$$\frac{du_1}{ds} - \epsilon_T = 0 \quad \text{para } s = 2L \quad (\text{B.109})$$

onde $\epsilon_T = \gamma\theta$ é a deformação associada à variação da temperatura, uniforme ao longo de toda a barra. A parcela à esquerda da igualdade na Equação B.109 corresponde à deformação mecânica ϵ_{mec} , uma vez que $\epsilon_1 = du_1/ds$ e $\epsilon_1 = \epsilon_T + \epsilon_{mec}$

Segundo sub-problema:

$$\frac{d^2u_2}{ds^2} = 0 \quad (\text{B.110})$$

sob a condição de contorno:

$$u_2 = 0 \quad \text{para } s = 0 \quad (\text{B.111})$$

$$u_2 = -u_1 \quad \text{para } s = 2L \quad (\text{B.112})$$

Integrando a Equação B.107 duas vezes e aplicando as condições de contorno das equações B.108 e B.109 obtém-se:

$$u_1 = \gamma\theta \left(\frac{s}{2L} \right) \quad (\text{B.113})$$

Seguindo o mesmo procedimento para o segundo problema, obtém-se o campo de deslocamentos associado a ele:

$$u_2 = -\gamma\theta \left(\frac{s}{2L} \right) \quad (\text{B.114})$$

Portanto, nesse exemplo o campo de deslocamentos do problema original é nulo em toda a barra:

$$u = u_1 + u_2 = 0 \quad (\text{B.115})$$

Por consequência, o campo de deformações totais também é nulo. Todavia, a parcela do tensor de deformações associada aos esforços mecânicos ϵ_{mec} é diferente de zero:

$$\epsilon_{mec} = \frac{du}{dx} - \gamma\theta = -\gamma\theta \quad (\text{B.116})$$

Como o campo de deformações mecânicas é calculado diretamente a partir da variação da temperatura nesse caso particular no qual não existem ações mecânicas externas, pode-se dizer que na análise mecânica ele é uma grandeza previamente conhecida. Naturalmente, o campo de deslocamentos também é previamente conhecido. Visto que a geometria do corpo é simples e o carregamento é monotônico, a tensão pode ser calculada diretamente a partir da deformação por meio da consulta ao modelo constitutivo, mesmo no regime fisicamente não-linear.

No presente exemplo, o modelo constitutivo de plasticidade uniaxial detalhadamente descrito em Souza Neto et al. (2009) é empregado. Esse modelo é utilizado em conjunto com a curva tensão-deformação bi-linear apresentada na Figura 5.24. Para obter o valor da tensão, basta avaliar se o módulo da deformação mecânica atuante excede a deformação na qual se inicia o escoamento, e de acordo com essa avaliação realizar o cálculo como descrito a seguir. A deformação associada ao escoamento é dada por:

$$\epsilon_y = \frac{f_y}{E_0} \quad (\text{B.117})$$

onde f_y é o limite de escoamento, tensão que caracteriza a transição entre o regime elástico e o início do regime plástico no caso de um carregamento monotônico, e E_0 é o módulo de elasticidade no regime elástico.

Primeiro caso: Deformação atuante inferior ou igual, em módulo, à deformação associada ao escoamento. A tensão é dada por:

$$\sigma = E_0\epsilon_{mec} \text{ para } |\epsilon_{mec}| \leq \epsilon_y \quad (\text{B.118})$$

sendo ϵ_{mec} a parcela do tensor de deformações totais associadas aos esforços mecânicos e σ a tensão causada pelo impedimento à variação da temperatura.

Segundo caso: Deformação atuante superior, em módulo, à deformação associada ao escoamento. A expressão para a tensão se torna:

$$\sigma = sig(\epsilon_{mec})[f_y + E_{ep}(|\epsilon_{mec}| - \epsilon_y)] \text{ para } |\epsilon_{mec}| > \epsilon_y \quad (\text{B.119})$$

onde $sig(\epsilon_{mec})$ é o sinal da deformação mecânica atuante e E_{ep} é o módulo elasto-plástico do material.

B.7 Solução analítica do Exemplo 7 - Viga submetida a fluxo de calor na face superior

Ao longo dessa seção, é apresentada a solução analítica obtida de Hetnarski e Eslami (2009) para o comportamento de uma viga bi-apoiada onde todas as faces são impedidas de trocar calor com o ambiente exceto a sua face superior que está submetida a um fluxo de calor conhecido q'' . O comprimento da viga é L , a seção transversal possui altura a e largura b , a condutividade térmica do material é dada por k , a difusividade térmica por α , o módulo de elasticidade por E e o coeficiente de expansão térmica é γ . A origem do sistema de coordenadas é posicionada no centro da face esquerda da viga, como ilustrado na Figura 5.27 e a viga parte de uma distribuição de temperaturas uniforme à temperatura T_0 .

Segundo Hetnarski e Eslami (2009), a solução desse problema pode ser obtida a partir de uma superposição de três sub-problemas, já que o comportamento é linear. Além dessa subdivisão em três problemas, o autor move a posição da origem do sistema de coordenadas para a fibra inferior da viga, e após obter o perfil de temperaturas é realizada uma transformação de coordenadas para referir esse perfil ao sistema indicado na Figura 5.27 a fim de dar continuidade na análise mecânica sob os efeitos da variação de temperatura.

O problema de transferência de calor é unidimensional devido à simetria e ao impedimento à troca de calor em todas as faces, exceto na superior. Desta forma, o calor flui da face superior da viga para a sua face inferior. Para um sistema de coordenadas similar ao ilustrado na Figura 5.27, porém posicionado na fibra inferior da viga, tem-se o problema de valor inicial e de contorno:

$$k \frac{\partial^2 T}{\partial y^2} = \rho c_p \frac{\partial T}{\partial t} \quad (\text{B.120})$$

sob as condições de contorno:

$$k \frac{\partial T}{\partial y} = q'' \quad y = a \quad t > 0 \quad (\text{B.121})$$

$$\frac{\partial T}{\partial y} = 0 \quad y = 0 \quad t > 0 \quad (\text{B.122})$$

e sob a condição inicial:

$$T = T_0 \quad t = 0 \quad 0 \leq y \leq a \quad (\text{B.123})$$

onde T é a temperatura buscada, y é a posição ao longo da altura da seção transversal da viga, c_p é o calor específico, e ρ a massa específica.

O problema é então subdividido em três sub-problemas, onde o primeiro somente depende do tempo, o segundo somente da posição e o terceiro depende da posição e do tempo como descrito na equação a seguir:

$$T(y, t) = T_1(t) + T_2(y) + T_3(y, t) \quad (\text{B.124})$$

Como o primeiro problema não depende da posição, o mesmo corresponde ao problema de valor inicial:

$$k \frac{\partial T_1}{\partial t} = \lambda \quad t > 0 \quad (\text{B.125})$$

onde λ é uma constante a ser determinada, e onde a condição inicial é:

$$T_1(0) = T_0 \quad (\text{B.126})$$

O segundo problema, que depende somente da posição, leva ao problema de valor de contorno:

$$k \frac{\partial^2 T_2}{\partial y^2} = \lambda \quad (\text{B.127})$$

sob as condições de contorno:

$$k \frac{\partial T_2}{\partial y} = q'' \quad y = a \quad (\text{B.128})$$

$$\frac{\partial T_2}{\partial y} = 0 \quad y = 0 \quad (\text{B.129})$$

O terceiro problema, que depende tanto do tempo como da posição é similar ao problema original:

$$k \frac{\partial^2 T_3}{\partial y^2} = \rho c_p \frac{\partial T_3}{\partial t} \quad (\text{B.130})$$

porém sob as condições de contorno homogêneas:

$$k \frac{\partial T_3}{\partial y} = 0 \quad y = a \quad t > 0 \quad (\text{B.131})$$

$$\frac{\partial T_3}{\partial y} = 0 \quad y = 0 \quad t > 0 \quad (\text{B.132})$$

A condição inicial do terceiro problema é escolhida de maneira a garantir que a soma dos três problemas em $t = 0$ resulte na condição inicial do problema original, logo:

$$T_3(y, 0) = -T_2(y) \quad (\text{B.133})$$

Integrando a Equação B.125, referente ao primeiro problema, e substituindo a condição inicial, obtém-se:

$$T_1 = \frac{\lambda}{\rho c_p} t + T_0 \quad (\text{B.134})$$

Integrando a Equação B.127 e substituindo as condições de contorno, determina-se $\lambda = q''/a$ e chega-se ao perfil de temperaturas:

$$T_2 = \frac{q''}{ak} \frac{y^2}{2} + C_1 \quad (\text{B.135})$$

onde C_1 é uma constante de integração a ser determinada posteriormente.

Para resolver terceiro problema, dado pela Equação B.130, aplica-se o método de separação de variáveis:

$$T_3(y, t) = Y(y)\tau(t) \quad (\text{B.136})$$

o que leva às infinitas soluções, a seguir, ao se considerar as condições de contorno:

$$Y_n(y) = C_2 \cos(\beta_n y) \quad (\text{B.137})$$

$$\tau_n(t) = C_3 e^{-\beta_n^2 \alpha t} \quad (\text{B.138})$$

com $\beta_0 = 0$ e $\beta_n = n\pi/a$ para $n = 1, 2, \dots$

Como o problema é linear, a solução do terceiro problema é escrita como uma superposição das infinitas funções $Y_n(y)$ e $\tau_n(t)$ que satisfazem a Equação B.130:

$$T_3(y, t) = B_0 + \sum_{n=1}^{\infty} B_n e^{-\beta_n^2 \alpha t} \cos(\beta_n y) \quad (\text{B.139})$$

onde as constantes C_2 e C_3 estão incluídas em B_0 e B_n .

A contribuição de cada uma das soluções $Y_n(y)$ e $\tau_n(t)$ à solução total é definida pela condição inicial da Equação B.133. Fazendo $t = 0$ na Equação B.139 e igualando a expressão obtida ao lado direito da Equação B.135 multiplicado por -1 , obtém-se:

$$A_0 + \sum_{n=1}^{\infty} B_n \cos(\beta_n y) = -\frac{q''}{ak} \frac{y^2}{2} \quad (\text{B.140})$$

sendo $A_0 = B_0 + C_1$.

Para permitir a determinação de A_0 e de cada um dos termos B_n , expande-se o lado direito da Equação B.140 em uma série de cossenos no intervalo $0 < y < a$, o que resulta em:

$$A_0 = -\frac{q''a}{6k} \quad (\text{B.141})$$

$$B_n = -\frac{2aq''}{kn^2\pi^2} \cdot (-1)^n \quad (\text{B.142})$$

Substituindo as soluções dos três sub-problemas na Equação B.124, obtém-se:

$$T = T_0 + \frac{q''a}{k} \left[\frac{\alpha t}{a^2} + \frac{y^2}{2a^2} - \frac{1}{6} - \frac{2}{\pi^2} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{-\left(\frac{n^2\pi^2}{a^2}\alpha t\right)} \cos\left(\frac{n\pi y}{a}\right) \right] \quad (\text{B.143})$$

Realizando uma transformação de coordenadas, movendo a origem do sistema de coordenadas da fibra inferior para a meia-altura da seção transversal da viga, $y = a/2$, tem-se:

$$T(y, t) = T_0 + \frac{aq''}{24k} \left[\frac{24\alpha t}{a^2} + \frac{12y^2}{a^2} + \frac{12y}{a} - 1 - \frac{48}{\pi^2} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{(-n^2\pi^2\alpha t/a^2)} \cos\frac{n\pi(2y+a)}{2a} \right] \quad (\text{B.144})$$

Para obter o campo de deslocamentos e o campo de tensões em função do tempo, segue-se a metodologia apresentada por Hetnarski e Eslami (2009) assim como realizado durante o desenvolvimento da solução analítica do Exemplo 5, mostrada em detalhes na seção B.5. Nessa metodologia, o efeito da variação de temperatura é introduzido na teoria de flexão de vigas de Bernoulli-Euler na forma dos esforços generalizados P_t e M_{T_z} conforme as equações B.94 e B.95, respectivamente.

Como no presente exemplo a viga também parte de uma distribuição de temperaturas inicial uniforme com $T_0 = 0$, substituindo a distribuição de temperaturas da Equação B.144 nas equações B.94 e B.95, e realizando a integração, obtém-se:

$$P_T(t) = \frac{E\gamma q'' A\alpha t}{ka} \quad (\text{B.145})$$

$$M_{Tz}(t) = E\gamma \frac{q'' I_z}{2k} \left(1 - \frac{96}{\pi^4} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n^4} e^{(-n^2\pi^2\alpha t/a^2)} \right) \quad (\text{B.146})$$

Substituindo as expressões obtidas para P_t e M_{Tz} , dadas pelas equações B.145 e B.146 na Equação B.98, chega-se ao campo de tensões ao longo do tempo e da altura da seção:

$$\sigma_{xx}(y, t) = E\gamma \left[\left(\frac{1}{12} - \frac{y^2}{a^2} \right) + \frac{4}{\pi^2} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} e^{(-n^2\pi^2\alpha t/a^2)} \cos \frac{n\pi(2y+a)}{2a} - \frac{96}{\pi^4} \sum_{n=1,3,5,\dots}^{\infty} \frac{e^{(-n^2\pi^2\alpha t/a^2)}}{n^4} \right] \quad (\text{B.147})$$

A equação da linha elástica, na ausência de esforços mecânicos externos, pode ser obtida solucionando a equação diferencial:

$$\frac{d^2v}{dx^2} = -\frac{M_{Tz}}{EI_z} = \frac{\gamma q''}{2k} \left[1 - \frac{96}{\pi^4} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n^4} e^{-n^2\pi^2\alpha t/a^2} \right] \quad (\text{B.148})$$

complementada pelas condições de contorno:

$$v(0) = 0 \quad (\text{B.149})$$

$$v(L) = 0 \quad (\text{B.150})$$

o que leva à expressão para a flecha:

$$v(x) = \frac{\gamma q''}{4k} \left[1 - \frac{96}{\pi^4} \sum_{n=1,3,5,\dots}^{\infty} \frac{e^{-n^2\pi^2\alpha t/a^2}}{n^4} \right] (xL - x^2) \quad (\text{B.151})$$

A flecha máxima ocorre no centro do vão, em $x = L/2$, com o valor:

$$v_{max} = -\frac{\gamma q'' L^2}{16k} \left[1 - \frac{96}{\pi^4} \sum_{n=1,3,5,\dots}^{\infty} \frac{e^{(-n^2\pi^2\alpha t/a^2)}}{n^4} \right] \quad (\text{B.152})$$

Ressalta-se que a flecha é positiva quando o deslocamento medido na linha de centro ocorre no sentido negativo do eixo coordenado y indicado na Figura 5.27.

B.8 Solução analítica do Exemplo 8 - Barra bi-engastada sob aquecimento senoidal nas bordas

A última solução analítica desenvolvida nessa dissertação corresponde ao comportamento termomecânico em regime transiente de uma barra bi-engastada, mostrada na Figura 5.34, sob aquecimento senoidal em suas extremidades. Maiores detalhes sobre esse problema encontram-se disponíveis na seção 5.2.4 do Capítulo 5 dessa dissertação de mestrado.

O desenvolvimento detalhado do cálculo analítico da distribuição de temperaturas nessa barra foi apresentado na seção B.3 desse apêndice. Para a parcela mecânica, segue-se nessa seção um procedimento similar ao apresentado na seção B.6, porém estendendo o estudo ao comportamento no regime quasi-estático.

No presente exemplo, a análise no regime elástico consiste em solucionar o problema regido pela equação de Navier, descrita no Capítulo 2 pela Equação 2.64, porém desprezando os efeitos inerciais. Como na modelagem unidimensional uma das hipóteses adotadas é considerar o coeficiente de Poisson igual a zero, a Equação 2.64 se reduz a:

$$\frac{d^2 u}{ds^2} = \gamma \frac{dT}{dx} \quad (\text{B.153})$$

sob as condições de contorno:

$$u = 0 \quad \text{para} \quad s = 0, \quad t > 0 \quad (\text{B.154})$$

$$u = 0 \text{ para } s = 2L, t > 0 \quad (\text{B.155})$$

e sob a condição inicial:

$$u = 0 \text{ para } 0 \leq s \leq 2L, t = 0 \quad (\text{B.156})$$

O problema colocado acima pode ser decomposto em dois sub-problemas, assim como realizado no Exemplo 6, e a solução final é obtida a partir da superposição da solução desses dois sub-problemas.

O primeiro sub-problema consiste na barra submetida à variação da temperatura, fixada na extremidade esquerda e livre para se deslocar na extremidade direita. O segundo sub-problema consiste na barra impedida de se deslocar em sua extremidade esquerda e submetida a um deslocamento prescrito em sua extremidade direita igual, em módulo, ao deslocamento calculado no primeiro sub-problema nessa extremidade, porém em sentido oposto. Esses problemas, ao serem superpostos, reconstituem as condições de contorno naturais e essenciais do problema original. Matematicamente, esses problemas podem ser postulados como descrito a seguir.

Primeiro sub-problema:

$$\frac{d^2 u_1}{ds^2} = \gamma \frac{dT}{dx} \quad (\text{B.157})$$

sob as condições de contorno:

$$u_1 = 0 \text{ para } s = 0, t > 0 \quad (\text{B.158})$$

$$\frac{du_1(s)}{ds} - \epsilon_T(s, t) = 0 \text{ para } s = 2L, t > 0 \quad (\text{B.159})$$

sob a condição inicial:

$$u_1 = 0 \text{ para } 0 \leq s \leq 2L, t = 0 \quad (\text{B.160})$$

onde $\epsilon_T = \gamma T(s, t)$ é a deformação associada à variação da temperatura. A parcela à esquerda da igualdade na Equação B.159 corresponde à deformação mecânica ϵ_{mec1} , uma vez que $\epsilon_1 = du_1/ds$ e $\epsilon_1 = \epsilon_T + \epsilon_{mec1}$

Segundo sub-problema:

$$\frac{d^2 u_2}{ds^2} = 0 \quad (\text{B.161})$$

sob as condições de contorno:

$$u_2 = 0 \quad \text{para } s = 0, \quad t > 0 \quad (\text{B.162})$$

$$u_2 = -u_1 \quad \text{para } s = 2L, \quad t > 0 \quad (\text{B.163})$$

e sob a condição inicial:

$$u_2 = 0 \quad \text{para } 0 \leq s \leq 2L, \quad t = 0 \quad (\text{B.164})$$

Integrando a Equação B.157 uma vez, obtém-se:

$$\frac{du_1}{ds} = \gamma T(s, t) + C_1 = \epsilon_T(s, t) + C_1 \quad (\text{B.165})$$

Integrando a Equação B.165 mais uma vez:

$$u_1(s) = \int \gamma T(s, t) ds + C_1 x + C_2 \quad (\text{B.166})$$

onde C_1 e C_2 são constantes de integração.

Introduzindo a condição de contorno da Equação B.159 na Equação B.165:

$$\epsilon_T(s = 2L) + C_1 - \epsilon_T(s = 2L) = 0$$

o que implica em:

$$C_1 = 0 \quad (\text{B.167})$$

Aplicando a condição de contorno da Equação B.158 na Equação B.157:

$$C_2 = - \left[\int \gamma T(s, t) ds \right]_{s=0} \quad (\text{B.168})$$

A seguir, será desenvolvida a integração da deformação térmica na Equação B.166. No presente problema há um período em que ocorre a elevação da temperatura nas bordas da barra, no qual a distribuição de temperaturas é regida pela Equação B.50, e um período no qual a temperatura nas bordas é mantida constante, para o qual a distribuição de temperaturas é dada pela Equação B.53. Portanto, torna-se necessário desenvolver uma solução para cada um desses regimes.

Para o período de elevação da temperatura nas bordas $0 \leq t \leq p$:

$$\int \gamma T(s, t) ds = s\gamma\theta \operatorname{sen} \left(\frac{\pi t}{2p} \right) + \frac{2\gamma\theta}{p} \sum_{m=1}^{\infty} \left[W(m, t) \left(\frac{\cos(\beta_m L) \operatorname{sen}(\beta_m s)}{\beta_m} - \frac{\operatorname{sen}(\beta_m L) \cos(\beta_m s)}{\beta_m} \right) \right] \quad (\text{B.169})$$

onde $W(m, t)$ é dado pela expressão:

$$W(m, t) = \left[\frac{(-1)^m}{(2m-1)} \right] \cdot \frac{\alpha\beta_m^2 \cos \left(\frac{\pi t}{2p} \right) + \left(\frac{\pi}{2p} \right) \operatorname{sen} \left(\frac{\pi t}{2p} \right) - \alpha\beta_m^2 e^{-\alpha\beta_m^2 t}}{\left[(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p} \right)^2 \right]} \quad (\text{B.170})$$

com o coeficiente β_m igual a:

$$\beta_m = (2m-1) \frac{\pi}{2L} \quad (\text{B.171})$$

Fazendo $s = 0$ na Equação B.169 e substituindo o resultado na Equação B.168, determina-se o valor de C_2 :

$$C_2 = \frac{2\gamma\theta}{p} \sum_{m=1}^{\infty} W(m, t) \left(\frac{\operatorname{sen}(\beta_m L)}{\beta_m} \right) \quad (\text{B.172})$$

o que leva ao campo de deslocamentos referente ao primeiro sub-problema durante o período de elevação da temperatura nas bordas:

$$u_T(s, t) = s\gamma\theta \operatorname{sen}\left(\frac{\pi t}{2p}\right) + \frac{2\gamma\theta}{p} \sum_{m=1}^{\infty} \left[W(m, t) \left(\frac{\cos(\beta_m L) \operatorname{sen}(\beta_m s)}{\beta_m} \right) \right] + \frac{2\gamma\theta}{p} \sum_{m=1}^{\infty} \left[W(m, t) \left(\frac{\operatorname{sen}(\beta_m L) (1 - \cos(\beta_m s))}{\beta_m} \right) \right] \quad (\text{B.173})$$

Para o período em que a temperatura nas bordas é mantida constante, correspondente a $t > p$, segue-se o mesmo procedimento e obtém-se:

$$u_T(s, t) = s\gamma\theta + \gamma \sum_{n=1}^{\infty} \left(\frac{2L}{n\pi} \right) C_n e^{-n^2\pi^2\alpha(t-p)/(4L^2)} \left[1 - \cos\left(\frac{n\pi s}{2L}\right) \right] \quad (\text{B.174})$$

onde C_n e A_m também foram definidos na seção 5.1.3 e são rerepresentados a seguir para proporcionar maior clareza:

$$C_n = \sum_{m=1}^{\infty} A_m \frac{\cos(\beta_m L)}{2} \left[\frac{1 - \cos(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) - \beta_m} + \frac{1 - \cos(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) + \beta_m} \right] + \sum_{m=1}^{\infty} A_m \frac{\operatorname{sen}(\beta_m L)}{2} \left[\frac{\operatorname{sen}(n\pi - 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) - \beta_m} - \frac{\operatorname{sen}(n\pi + 2\beta_m L)}{\left(\frac{n\pi}{2L}\right) + \beta_m} \right]$$

$$A_m = \left(\frac{2\theta}{pL} \right) \left[\frac{(-1)^m}{(2m-1)} \right] \left[\frac{\left(\frac{\pi}{2p}\right) - \alpha\beta_m^2 e^{-\alpha\beta_m^2 p}}{(\alpha\beta_m^2)^2 + \left(\frac{\pi}{2p}\right)^2} \right]$$

No que se refere ao segundo sub-problema, a solução é obtida a partir da integração da Equação B.161. Realizando esse procedimento de integração duas vezes e aplicando as condições de contorno descritas pelas equações B.162 e B.163, obtém-se:

$$u_2(s, t) = -\frac{s}{2L} \cdot u_1(2L, t) \quad (\text{B.175})$$

O campo de deslocamentos totais é o resultado da superposição:

$$u = u_1 + u_2 \quad (\text{B.176})$$

A parcela do tensor de deformações associada aos esforços mecânicos ϵ_{mecTot} , correspondente à soma das parcelas mecânicas relativas aos dois sub-problemas, é dada por:

$$\epsilon_{mecTot} = \frac{du_2}{ds} = -\frac{u_1(2L, t)}{2L} \quad (\text{B.177})$$

Esse campo de deformações constante ao longo do comprimento da barra também leva a um campo de tensões constante em relação à coordenada espacial, porém ainda dependente do tempo.

Observa-se no desenvolvimento da solução analítica para o regime elástico que tanto o campo de deslocamentos como o campo de deformações mecânicas depende exclusivamente do campo de temperaturas, devido à ausência de ações mecânicas externas. Essa é uma característica particular da classe de problemas em que se insere esse exemplo, que corresponde aos problemas conduzidos pela deformação, que é uma grandeza previamente conhecida em todo o corpo. Tal característica permite que o estudo realizado até o presente momento em regime elástico seja estendido ao regime plástico.

Para obter as tensões, segue-se a mesma metodologia descrita na seção B.6, na qual se emprega um modelo de plasticidade uniaxial conforme detalhadamente descrito em Souza Neto et al. (2009). Como o regime é quasi-estático e o carregamento é monotônico, a avaliação das tensões é realizada, em cada instante de tempo, utilizando o valor de deformação no respectivo instante e substituindo essa deformação no modelo constitutivo.

Para tal, primeiramente determina-se qual é o valor da deformação que corresponde à transição entre o regime elástico e o regime plástico, denominada deformação de escoamento:

$$\epsilon_y = \frac{f_y}{E_0} \quad (\text{B.178})$$

onde f_y é o limite de escoamento, tensão que caracteriza a transição entre o regime

elástico e o início do regime plástico no caso de um carregamento monotônico, e E_0 é o módulo de elasticidade no regime elástico.

Em seguida, é realizada uma verificação do valor do módulo da deformação mecânica atuante. Se ela não exceder a deformação na qual se inicia o escoamento, o corpo se encontra no regime elástico e a tensão é calculada como mostrado a seguir:

$$\sigma = E_0 \epsilon_{mec} \text{ para } |\epsilon_{mec}| \leq \epsilon_y \quad (\text{B.179})$$

sendo ϵ_{mec} a parcela do tensor de deformações totais associadas aos esforços mecânicos e σ a tensão causada pelo impedimento à variação da temperatura.

Caso o módulo da deformação mecânica atuante no instante de tempo corrente exceda a deformação de escoamento, a tensão no regime plástico é calculada como:

$$\sigma = sig(\epsilon_{mec}) [f_y + E_{ep} (|\epsilon_{mec}| - \epsilon_y)] \text{ para } |\epsilon_{mec}| > \epsilon_y \quad (\text{B.180})$$

onde $sig(\epsilon_{mec})$ é o sinal da deformação mecânica atuante e E_{ep} é o módulo elasto-plástico do material.

Apêndice C

CLASSES EMPREGADAS NOS EXEMPLOS DE VALIDAÇÃO

Nesse apêndice são relacionadas as classes utilizadas para a modelagem e análise dos exemplos apresentados no Capítulo 5 dessa dissertação de mestrado. Maiores detalhes sobre o conteúdo abordado nesses exemplos podem ser encontrados no referido capítulo.

C.1 Exemplo 1

Tabela C.1: Classes utilizadas na modelagem e análise do Exemplo 1.

Entidade	Classe utilizada
<i>Solução</i>	SteadyState
<i>Model</i>	FemModel
<i>Problem Driver</i>	HeatTransferPd
<i>Global Analysis Model</i>	HeatTransfer2D
<i>Material</i>	ThermalIsotropicMaterial
<i>Degeneration</i>	PrescribedDegeneration
<i>Element</i>	Quadrilateral paramétrico
<i>Shape</i>	Q4
Modelo de análise dos elementos finitos	HeatTransfer2D
Modelo constitutivo dos elementos finitos	HeatTransferConstitutiveModel

C.2 Exemplo 2

Tabela C.2: Classes utilizadas na modelagem e análise do Exemplo 2.

Entidade	Classe utilizada
<i>Solução</i>	NonIncrementalEqPath
<i>Step</i>	NewtonIterative
<i>Model</i>	FemModel
<i>Problem Driver</i>	NonLinearHeatTransfer
<i>Global Analysis Model</i>	HeatTransfer3D
<i>Material</i>	PolynomialTempDepThermalIsotropicMaterial
Material que recebe a dependência térmica	ThermalIsotropicMaterial
<i>Degeneration</i>	PrescribedDegeneration
<i>Element</i>	Hexahedral paramétrico
<i>Shape</i>	H8
Modelo de análise dos elementos finitos	HeatTransfer3D
Modelo constitutivo dos elementos finitos	HeatTransferConstitutiveModel
<i>SimulationManager</i>	SinglePhysicsFemSimManager
<i>Setuper</i>	SinglePhysicsFemSetuper

C.3 Exemplo 3

Tabela C.3: Classes utilizadas na modelagem e análise do Exemplo 3.

Entidade	Classe utilizada
<i>Solução</i>	LinearHeatTransferNewmarkBeta
<i>Model</i>	FemModel
<i>Problem Driver</i>	HeatTransferPd
<i>Global Analysis Model</i>	HeatTransfer1D
<i>Material</i>	ThermalIsotropicMaterial
<i>Degeneration</i>	PrescribedDegeneration
<i>Element</i>	Bar paramétrico
<i>Shape</i>	L4
Modelo de análise dos elementos finitos	HeatTransfer1D
Modelo constitutivo dos elementos finitos	HeatTransferConstitutiveModel
<i>SimulationManager</i>	SinglePhysicsFemSimManager
<i>Setuper</i>	TimeDepThermalSetuper

C.4 Exemplo 4

Tabela C.4: Classes utilizadas na modelagem e análise do Exemplo 4.

Entidade	Classe utilizada
<i>Solução</i>	NonLinearHeatTransferNewmarkBeta
<i>Model</i>	FemModel
<i>Problem Driver</i>	NonLinearHeatTransfer
<i>Global Analysis Model</i>	HeatTransferAxisymmetric
<i>Material</i>	PolynomialTempDepThermalIsotropicMaterial
Material que recebe a dependência térmica	ThermalIsotropicMaterial
<i>Degeneration</i>	PrescribedDegeneration
<i>Element</i>	Quadrilateral
<i>Shape</i>	Q9
Modelo de análise dos elementos finitos	HeatTransferAxisymmetric
Modelo constitutivo dos elementos finitos	HeatTransferConstitutiveModel
<i>SimulationManager</i>	SinglePhysicsFemSimManager
<i>Setuper</i>	TimeDepThermalSetuper

C.5 Exemplo 5

Tabela C.5: Classes utilizadas na modelagem e análise do Exemplo 5.

Entidade	Classe utilizada
<i>Solução térmica</i>	SteadyState
<i>Solução mecânica</i>	SteadyState
<i>Model</i>	FemModel
<i>Problem Driver térmico</i>	HeatTransferPd
<i>Problem Driver mecânico</i>	Parametric
<i>Global Analysis Model térmico</i>	HeatTransfer2D
<i>Global Analysis Model mecânico</i>	PlaneStress
<i>Material térmico</i>	ThermalIsotropicMaterial
<i>Material mecânico</i>	LinearElasticIsotropic
<i>Degeneration</i>	PrescribedDegeneration
<i>Element</i>	Quadrilateral paramétrico
<i>Shape</i>	Q9
Modelo de análise térmico dos elementos finitos	HeatTransfer2D
Modelo de análise mecânico dos elementos finitos	PlaneStress
Modelo constitutivo térmico dos elementos finitos	HeatTransferConstitutiveModel
Modelo constitutivo mecânico dos elementos finitos	LinearElasticConstModel
<i>SimulationManager</i>	SteadyStateThermoMecSimManager
<i>Setuper</i>	SharedModelSteadyStateTermoMecSetuper

C.6 Exemplo 6

Tabela C.6: Classes utilizadas na modelagem e análise do Exemplo 6.

Entidade	Classe utilizada
<i>Solução térmica</i>	NonIncrementalEqPath
<i>Solução mecânica</i>	QuasiStaticEquilibriumPath
<i>Step</i>	ThermoMecSNR
<i>IterativeStrategy</i>	LoadControl
<i>Model</i>	FemModel
<i>Problem Driver térmico</i>	HeatTransferPd
<i>Problem Driver mecânico</i>	PhysicallyNonLinear
<i>Global Analysis Model térmico</i>	HeatTransfer1D
<i>Global Analysis Model mecânico</i>	Line1D
<i>Material térmico</i>	ThermalIsotropicMaterial
<i>Material mecânico</i>	VonMises
<i>Lei de endurecimento do material mecânico</i>	HardeningLawLinear
<i>Degeneration</i>	PrescribedDegeneration
<i>Element</i>	Bar paramétrico
<i>Shape</i>	L2
Modelo de análise térmico dos elementos finitos	HeatTransfer1D
Modelo de análise mecânico dos elementos finitos	Line1D
Modelo constitutivo térmico dos elementos finitos	HeatTransferConstitutiveModel
Modelo constitutivo mecânico dos elementos finitos	SLFOCM_Plasticity_VonMises
<i>SimulationManager</i>	SteadyStateThermoMecSimManager
<i>Setuper</i>	SharedModelSteadyStateTermoMecSetuper

C.7 Exemplo 7

Tabela C.7: Classes utilizadas na modelagem e análise do Exemplo 7.

Entidade	Classe utilizada
<i>Solução térmica</i>	<code>LinearHeatTransferNewmarkBeta</code>
<i>Solução mecânica</i>	<code>LinearQuasiStatic</code>
<i>Model</i>	<code>FemModel</code>
<i>Problem Driver térmico</i>	<code>HeatTransferPd</code>
<i>Problem Driver mecânico</i>	<code>Parametric</code>
<i>Global Analysis Model térmico</i>	<code>HeatTransfer2D</code>
<i>Global Analysis Model mecânico</i>	<code>PlaneStress</code>
<i>Material térmico</i>	<code>ThermalIsotropicMaterial</code>
<i>Material mecânico</i>	<code>LinearElasticIsotropic</code>
<i>Degeneration</i>	<code>PrescribedDegeneration</code>
<i>Element</i>	<code>Quadrilateral paramétrico</code>
<i>Shape</i>	<code>Q9</code>
Modelo de análise térmico dos elementos finitos	<code>HeatTransfer2D</code>
Modelo de análise mecânico dos elementos finitos	<code>PlaneStress</code>
Modelo constitutivo térmico dos elementos finitos	<code>HeatTransferConstitutiveModel</code>
Modelo constitutivo mecânico dos elementos finitos	<code>LinearElasticConstModel</code>
<i>SimulationManager</i>	<code>TimeDepThermoMecSimManager</code>
<i>Setuper térmico</i>	<code>TimeDepThermalSetuper</code>
<i>Setuper mecânico</i>	<code>SinglePhysicsFemSetuper</code>

C.8 Exemplo 8

Tabela C.8: Classes utilizadas na modelagem e análise do Exemplo 8.

Entidade	Classe utilizada
<i>Solução térmica</i>	LinearHeatTransferNewmarkBeta
<i>Solução mecânica</i>	QuasiStaticEquilibriumPath
<i>Step</i>	ThermoMecSNR
<i>IterativeStrategy</i>	LoadControl
<i>Model</i>	FemModel
<i>Problem Driver térmico</i>	HeatTransferPd
<i>Problem Driver mecânico</i>	PhysicallyNonLinear
<i>Global Analysis Model térmico</i>	HeatTransfer1D
<i>Global Analysis Model mecânico</i>	Line1D
<i>Material térmico</i>	ThermalIsotropicMaterial
<i>Material mecânico</i>	VonMises
<i>Lei de endurecimento do material mecânico</i>	HardeningLawLinear
<i>Degeneration</i>	PrescribedDegeneration
<i>Element</i>	Bar paramétrico
<i>Shape</i>	L4
Modelo de análise térmico dos elementos finitos	HeatTransfer1D
Modelo de análise mecânico dos elementos finitos	Line1D
Modelo constitutivo térmico dos elementos finitos	HeatTransferConstitutiveModel
Modelo constitutivo mecânico dos elementos finitos	SLFOCM_Plasticity_VonMises
<i>SimulationManager</i>	TimeDepThermoMecSimManager
<i>Setuper térmico</i>	TimeDepThermalSetuper
<i>Setuper mecânico</i>	SinglePhysicsFemSetuper

Referências Bibliográficas

- Almansi, E., 1897. *Use of the Stress Function in Thermoelasticity*. Vol. 47 of 2, Memorie della Reale accademia delle scienze di Torino.
- Alves, P. D., Barros, F. B. e Pitangueira, R. L. S., 2013. ‘An object-oriented approach to the generalized finite element method’. *Advances in Engineering Software*, vol. 59, pp. 1–18.
- Badreddine, H., Labergère, C. e Saanouni, K., 2016. ‘Ductile damage prediction in sheet and bulk metal forming’. *Comptes Rendus Mécanique*, vol. 344, pp. 296–318.
- Barron, R. F. e Barron, B. R., 2012. *Design for Thermal Stresses*. 1 edn, John Wiley & Sons.
- Belytschko, T., Liu, W., Moran, B. e Elkhodary, K. I., 2014. *Nonlinear Finite Elements for Continua and Structures*. 2 edn, John Wiley & Sons.
- Bergheau, J.-M. e Fortunier, R., 2008. *Finite Element Simulation of Heat Transfer*. Wiley-ISTE.
- Bergman, T. L., Lavine, A. S., Incropera, F. P. e Dewitt, D. P., 2011. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons.
- Biot, M. A., 1956. ‘Thermoelasticity and irreversible thermodynamics’. *Journal of Applied Physics*, vol. 27, pp. 240–253.
- Boley, B. A., (1980), Thermal stresses: a survey, *in* ‘Thermal stresses in severe thermal environments - Proceedings of the International Conference on Thermal Stresses in Materials and Structures in Severe Thermal Environments’, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

- Booley, B. A. e Weiner, J. H., 1997. *Theory of Thermal Stresses*. Dover Publications.
- Boresi, A. P., Chong, K. P. e Lee, J. D., 2011. *Elasticity in Engineering Mechanics*. 3 edn, John Wiley & Sons.
- Botelho, G. G., Fonseca, F. T. e Pitangueira, R. L. S., (2015), Sistema computacional orientado a objetos para análises acopladas termo-estruturais pelo método dos elementos finitos, in ‘Proceedings of the XXXVI Ibero-Latin American Congress on Computational Methods in Engineering - CILAMCE 2015’, Rio de Janeiro, Brazil.
- Bouery, C., 2012. Contribution to algorithmic strategies for solving coupled thermo-mechanical problems by an energy-consistent variational approach. Tese de Doutorado, École Centrale Nantes, Nantes, França.
- Boyce, W. E. e DiPrima, R., 2006. *Equações Diferenciais Elementares e Problemas de Valores de Contorno*. LTC.
- Bridgman, P. W., 1943. *The Nature of Thermodynamics*. Harvard University Press.
- Bridgman, P. W., 1950. ‘The thermodynamics of plastic deformation and generalized entropy’. *Review of Modern Physics*, vol. 22, pp. 56–63.
- Caratheodory, C., 1909. ‘Untersuchungen über die grundlagen der thermodynamik’. *Mathematische Annalen*, vol. 67, pp. 335–386.
- Carnot, S., 1986. *Reflections on the Motive Power of Fire and on Machines Fitted to Develop that Power*. Manchester University Press.
- Carslaw, H. S. e Jaeger, J. C., 1959. *Conduction of Heat in Solids*. 2 edn, Oxford University Press.
- Cerit, M., 2011. ‘Thermo-mechanical analysis of a partially ceramic coated piston used in an si engine’. *Surface and Coatings Technology*, vol. 205, pp. 3499–3505.
- Clapeyron, E., 1834. ‘Mémoire sur la puissance motrice de la chaleur’. *Journal de l’École polytechnique*, vol. 14, pp. 153–191.

- Clausius, R., 1851. ‘On the moving force of heat, and the laws regarding the nature of heat itself which are deducible therefrom’. *Philosophical Magazine and Journal of Science*, vol. 2, pp. 1–21.
- da Fonseca, F. T., 2008. Sistema computacional para análise dinâmica geometricamente não-linear através do método dos elementos finitos. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Danish, M., Kumar, S. e Kumar, S., (2011), Exact analytical solutions of three nonlinear heat transfer models, *in* ‘Proceedings of the World Congress on Engineering’, London, UK.
- de Donder, T. e van Rysselberghe, P., 1936. *Thermodynamic theory of affinity*. Stanford university press.
- Dhondt, G., 2004. *The Finite Element Method for Three-dimensional Thermomechanical Applications*. 1 edn, Wiley.
- Dippel, B., Johlitz, M. e Lion, A., 2014. ‘Thermo-mechanical couplings in elastomers - experiments and modelling’. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, .
- Duhamel, J. M. C., 1837. ‘Second mémoire sur les phénomènes thermomécaniques’. *Journal de l’École polytechnique*, vol. 15, pp. 1–57.
- Duhem, P., 1911. *Traité d’énergétique ou de thermodynamique générale*. Gauthier-Villars.
- Eraslan, A. N. e Orcan, Y., 2002. ‘Computation of transient thermal stresses in elastic-plastic tubes: Effect of coupling and temperature-dependent physical properties’. *Journal of Thermal Stresses*, vol. 25, pp. 559–572.
- Esmati, V., Nazari, M. B. e Rokhi, M. M., 2018. ‘Implementation of xfem for dynamic thermoelastic crack analysis under non-classic thermal shock’. *Theoretical and Applied Fracture Mechanics*, vol. 95, pp. 42–58.

- Feng, S., Cui, X. e Li, G., 2013. ‘Transient thermal mechanical analyses using a face-based smoothed finite element method (fs-fem)’. *International Journal of Thermal Sciences*, vol. 74.
- Fish, J. e Belytschko, T., 2007. *A First course in Finite Elements*. 1 edn, John Wiley & Sons.
- Fourier, J.-B.-J., 1822. *Théorie analytique de la chaleur*. Firmin Didot, Paris.
- Gibbs, J. W., 1928. *The Collected Works of J. Willard Gibbs*. Vol. I, London : Longmans, Green.
- Gori, L., Penna, S. S. e Pitangueira, R. L. d. S., 2017. ‘A computational framework for constitutive modelling’. *Computers & Structures*, vol. 187, pp. 1–23.
- Groulx, D., (2007), Analytical solution for the time-dependent one-dimensional conduction problem with time-dependent boundary conditions: Duhamel’s theorem and separation of variables, in ‘CSME Forum’, Ottawa, Canada.
- Hasselman, D. P. H. e Heller, R. A., 1980. *Thermal stresses in severe thermal environments*. Plenum Press.
- Hecht, F., 2012. ‘New development in freefem++’. *Journal of Numerical Mathematics*, vol. 20, pp. 251–266.
- Hetnarski, R. B. e Achenbach, J. D., 1996. *Mechanics and Mathematical Methods*. Vol. IV of *Thermal Stresses*, Elsevier Science B.V.
- Hetnarski, R. B. e Ignaczak, J., 1999. ‘Generalized thermoelasticity’. *Journal of Thermal Stresses*, vol. 22, pp. 451–476.
- Hetnarski, R. e Eslami, M., 2009. *Thermal Stresses - Advanced Theory and Applications*. Springer.
- Hosseini, S. M., 2013. ‘Shock-induced thermoelastic wave propagation analysis in a thick hollow cylinder without energy dissipation using mesh-free generalized finite difference (gfd) method’. *Acta Mechanica*, vol. 224, pp. 465–478.

- Ignaczak, J. e Ostoja-Starzewski, M., 2009. *Thermoelasticity with Finite Wave Speeds*. 1 edn, Oxford University Press.
- Jahanian, S., 1995. 'Thermoelastoplastic and residual stress analysis during induction hardening of steel'. *Journal of Materials Engineering and Performance*, vol. 4, pp. 737–744.
- Jakubowska, M., 1982. 'Kirchhoff's formula for thermoelastic solid'. *Journal of Thermal Stresses*, vol. 5, pp. 127–144.
- Jian, Q. e Shui, Y., 2017. 'Numerical and experimental analysis of transient temperature field of ventilated disc brake under the condition of hard braking'. *International Journal of Thermal Sciences*, vol. 122, pp. 115–123.
- Johns, D. J. e Neal, B. G., 1965. *Thermal Stress Analyses*. Pergamon.
- Jussila, P. e Ruokolainen, J., 2007. 'Thermomechanics of porous media - II: thermo-hydro-mechanical model for compacted bentonite'. *Transport in Porous Media*, vol. 67, pp. 275–296.
- Kang, S.-H. e Im, Y.-T., 2007. 'Three-dimensional thermo-elastic-plastic finite element modeling of quenching process of plain-carbon steel in couple with phase transformation'. *International Journal of Mechanical Sciences*, vol. 49, pp. 423–439.
- Karban, P., Mach, F., Kůs, P., Pánek, D. e Doležal, I., 2013. 'Numerical solution of coupled problems using code Agros2D'. *Computing*, vol. 95, pp. 381–408.
- Kim, J., Im, S. e Kim, H.-G., 2005. 'Numerical implementation of a thermo-elastic-plastic constitutive equation in consideration of transformation plasticity in welding'. *International Journal of Plasticity*, vol. 21, pp. 1383–1408.
- Kirk, B. S., Peterson, J. W., Stogner, R. H. e Carey, G. F., 2006. 'libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations'. *Engineering with Computers*, vol. 22(3-4), pp. 237–254.

- Kolkwitz, B., Foeckerer, T., Heinzl, C., Zaeh, M. e Brinksmeier, E., 2011. ‘Experimental and numerical analysis of the surface integrity resulting from outer-diameter grind-hardening’. *Procedia Engineering*, vol. 19, pp. 222–227.
- Kumar, P. e Kodur, V., 2017. ‘Modeling the behavior of load bearing concrete walls under fire exposure’. *Construction and Building Materials*, vol. 154, pp. 993–1003.
- Lei, S., Shin, Y. e Incropera, F., 1999. ‘Thermo-mechanical modeling of orthogonal machining process by finite element analysis’. *International Journal of Machine Tools and Manufacture*, vol. 39, pp. 731–750.
- Logan, D. L., 2011. *A First Course in the Finite Element Method*. CL Engineering.
- Logg, A., 2007. ‘Automating the finite element method’. *Archives of Computational Methods in Engineering*, vol. 14(2), pp. 93–138.
- Logg, A., Mardal, K., Wells, G. N. et al., 2012. *Automated Solution of Differential Equations by the Finite Element Method*. Springer.
- Matsumoto, T., Guzik, A. e Tanaka, M., 2005. ‘A boundary element method for analysis of thermoelastic deformations in materials with temperature dependent properties’. *International Journal for Numerical Methods in Engineering*, vol. 64, pp. 1432–1458.
- Maugin, G. A., 1999. *The Thermomechanics of Nonlinear Irreversible Behaviours: An Introduction*. World Scientific Series on Nonlinear Science Series A Vol. 27, World Scientific.
- Maxwell, J. C., 1871. *Theory of Heat*. Cambridge University Press.
- Miehe, C., Hofacker, M., Schänzel, L. e Aldakheel, F., 2015. ‘Phase field modeling of fracture in multi-physics problems. part ii. coupled brittle-to-ductile failure criteria and crack propagation in thermo-elastic-plastic solids’. *Computer Methods in Applied Mechanics and Engineering*, vol. 294, pp. 486–522.
- Miehe, C., Schänzel, L. e Ulmer, H., 2015. ‘Phase field modeling of fracture in multi-physics problems. part i. balance of crack surface and failure criteria for

- brittle crack propagation in thermo-elastic solids’. *Computer Methods in Applied Mechanics and Engineering*, vol. 294, pp. 449–485.
- Navarra, E. O. I., 1995. *Calculo de Estructuras por el Metodo de Elementos Finitos*. Centro Internacional de Métodos Numéricos en Ingeniería.
- Navier, C. L. M. H., 1821. ‘Mémoire sur les lois de l’équilibre et du mouvement des corps solides élastiques’. *Mémoire lu À l’Académie royale des sciences de l’Institut de France*, vol. 7, pp. 375–393.
- Neumann, F., 1885. *Vorlesungen über die Theorie der Elasticität der festen Körper und des Lichtäthers*. B. G. Teubner, Leipzig.
- Nicholson, D. W., 2003. *Finite Element Analysis - Thermomechanics of Solids*. 1 edn, CRC Press.
- Noda, N., 2014. *Thermal Stresses in bodies with Temperature-Dependent Properties, Encyclopedia of Thermal Stresses*, editado por Richard B. Hetnarski. Springer Netherlands.
- Noda, N., Hetnarski, R. B. e Tanigawa, Y., 2003. *Thermal Stresses*. 2 edn, Taylor & Francis.
- Nowacki, W., 1970. ‘Problems of thermoelasticity’. *Progress in Aerospace Sciences*, vol. 10, pp. 1–63.
- Oliveira, D. B., 2016. Implementação computacional de modelos elastoplásticos para análise fisicamente não linear. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- OMG - Object Management Group, 2019, ‘Unified modeling language TM(uml [®])’, <http://uml.org>.
- Özisk, M. N., 1993. *Heat Conduction*. Wiley InterScience.
- Parkus, H., 1976. *Thermoelasticity*. 2 edn, Springer-Verlag Wien.
- Peixoto, R. G., Ribeiro, G. O. e Pitangueira, R. L. S., 2018. ‘A progressive cells activation algorithm for physically non-linear bem analysis’. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 40.

- Penna, S. S., 2011. Formulação Multipotencial para Modelos de Degradação Elástica - Unificação Teórica, Proposta de Novo Modelo, Implementação Computacional e Modelagem de Estruturas de Concreto. Tese de Doutorado, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Pitangueira, R. L. S., Fonseca, F. T., Fuina, J. S., Camara, L. S., Fonseca, M. T., Ferreira, R. L., Modeira, R. N., Saliba, S. S. e Penna, S. S., (2008), Insane - versão 2.0, in 'Proceedings of the XXIX Ibero-Latin American Congress on Computational Methods in Engineering - CILAMCE 2008', Maceió, Brazil.
- Planck, M., 1903. *Treatise on Thermodynamics*. London : Longmans, Green.
- Plews, J. A. e Duarte, C. A., 2016. 'A two-scale generalized finite element approach for modeling localized thermoplasticity'. *International Journal for Numerical Methods in Engineering*, vol. 108, pp. 1125–1158.
- Putanowicz, R., Jaskowiec, J., Plucinski, P. e Stankiewicz, A., 2013. 'Finite element implementation of nonlinear thermo-elasticity as typical coupling of diffusion and momentum balance'. *Mechanics and Control*, vol. 32, pp. 152.
- Rattez, H., 2017. Thermo-hydro-mechanical couplings and strain localization in Cosserat continua : application to stability analysis of rapid shear in faults. Tese de Doutorado, Université Paris-Est, Champs sur Marne, França.
- Rayleigh, L., 1899. *Scientific papers, vol.1-6*. Cambridge University Press.
- Reddy, J. N., 2005. *An Introduction to Nonlinear Finite Element Analysis*. Oxford University Press.
- Riedlbauer, D., Steinmann, P. e Mergheim, J., 2014. 'Thermomechanical finite element simulations of selective electron beam melting processes: performance considerations'. *Computational Mechanics*, vol. 54, pp. 109–122.
- Schacht, C. A., 1993. 'Improved mechanical material property definition for predicting the thermomechanical behavior of refractory linings of teeming ladles'. *Journal of the American Ceramic Society*, vol. 76, pp. 202–206.

- Sharp, K. e Matschinsky, F., 2015. ‘Translation of ludwig boltzmann’s paper ”on the relationship between the second fundamental theorem of the mechanical theory of heat and probability calculations regarding the conditions for thermal equilibrium”sitzungsberichte der kaiserlichen akademie der wissenschaften. mathematisch-naturwissen classe. abt. ii, lxxvi 1877, pp 373-435 (wien. ber. 1877, 76:373-435). reprinted in wiss. abhandlungen, vol. ii, reprint 42, p. 164-223, barth, leipzig, 1909’. *Entropy*, vol. 17, pp. 1971–2009.
- Shih, A. J. e Yang, H. T. Y., 1993. ‘Experimental and finite element predictions of residual stresses due to orthogonal metal cutting’. *International Journal for Numerical Methods in Engineering*, vol. 36, pp. 1487–1507.
- Sladek, J., Sladek, V., Sulek, P., Tan, C. e Zhang, C., 2009. ‘Two- and three-dimensional transient thermoelastic analysis by the mlp method’. *Computer Modeling in Engineering & Sciences*, vol. 47, pp. 61–96.
- Sluzalec, A., 2005. *Theory of Thermomechanical Processes in Welding*. 1 edn, Springer Netherlands.
- Souza Neto, E. A., Perić, D. e Owen, D. R. J., 2009. *Computational Methods for Plasticity*. 1 edn, Wiley.
- Stallman, R. M. e Gay, J., 2015. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. 3 edn, Free Software Foundation Inc.
- Tedone, O., 1906. ‘Allgemeine theoreme der matematischen elastizitätslehre (integrationstheorie)’. *Encyklopädie der matematischen Wissenschaften*, vol. 4, pp. 55–124 e 125–214 (Segundo artigo escrito com A. Timpe).
- Teixeira, L. B., 2017. Modelagem fisicamente não linear de materiais refratários a altas temperaturas considerando efeitos de dano. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Thomson, W. e Tait, P. G., 1867. *Treatise on Natural Philosophy*. Clarendon Press.
- Thornton, E. A., 1996. *Thermal Structures for Aerospace Applications*. AIAA Education Series, American Institute of Aeronautics and Astronautics.

- Vadasz, P., 2010. ‘Analytical solution to nonlinear thermal diffusion: Kirchhoff versus cole-hopf transformations’. *Journal of Heat Transfer*, vol. 132, pp. 121302–1–121302–6.
- Voigt, W., 1910. *Lehrbuch der Kristallphysik*. B. G. Teubner, Berlin.
- von Mayer, R., 1842. ‘Bemerkungen über die kräfte der unbelebten natur’. *Annalen der Chemie und Pharmacie*, vol. 42, pp. 233–240.
- W3C, 2019, ‘Extensible markup language (xml)’, <http://www.w3.org/XML>.
- Watson, G. N., 1995. *A Treatise on the Theory of Bessel Functions*. 2 edn, Cambridge University Press.
- Weaver, W. e Johnston, P. R., 1984. *Finite Elements for Structural Analysis*. Prentice-Hall Inc.
- Wu, L., Wen, Z., Li, W. e Jin, X., 2011. ‘Thermo-elastic-plastic finite element analysis of wheel/rail sliding contact’. *Wear*, vol. 271, pp. 437–443.
- Yang, Y. e Shieh, M., 1990. ‘Solution method for nonlinear problems with multiple critical points’. *AIAA Journal*, vol. 28, pp. 2110–2116.
- Zachmanoglou, E. C., 1971. ‘Caratheodory’s theorem on the second law of thermodynamics’. *SIAM Journal on Applied Mathematics*, vol. 25, pp. 592–596.
- Zienkiewicz, O. C. e Taylor, R. L., 2000. *The Finite Element Method*. Vol. 1: The Basis, 5 edn, Butterworth-Heinemann.