



PROJETO DE GRADUAÇÃO

**IMPLEMENTAÇÃO COMPUTACIONAL DA
SOLUÇÃO DE PROBLEMAS TÉRMICOS E
MECÂNICOS PELO MÉTODO DOS
ELEMENTOS FINITOS EM PYTHON**

NASSER SAMIR ALKMIM

Brasília, 14 de dezembro de 2016

UNIVERSIDADE DE BRASÍLIA

**FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL**

**IMPLEMENTAÇÃO COMPUTACIONAL DA SOLUÇÃO
DE PROBLEMAS TÉRMICOS E MECÂNICOS PELO
MÉTODO DOS ELEMENTOS FINITOS EM PYTHON**

NASSER SAMIR ALKMIM

ORIENTADOR: PROF. LINEU JOSÉ PEDROSO

MONOGRAFIA DE PROJETO FINAL II EM ESTRUTURAS

BRASÍLIA/DF: DEZEMBRO DE 2016

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL

**IMPLEMENTAÇÃO COMPUTACIONAL DA SOLUÇÃO DE
PROBLEMAS TÉRMICOS E MECÂNICOS PELO MÉTODO DOS
ELEMENTOS FINITOS EM PYTHON**

NASSER SAMIR ALKMIM

MONOGRAFIA DE PROJETO FINAL II SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM ENGENHARIA CIVIL.

APROVADA POR:

Prof. Lineu José Pedroso, Dr.-Ing (UnB)

(Orientador)

Prof. Márcio Muniz de Farias, Ph.D. (UnB)

(Examinador interno)

Prof. Raul Dario Durand Farfan, D.Sc. (UnB)

(Examinador interno)

BRASÍLIA/DF, 14 DE DEZEMBRO DE 2016

FICHA CATALOGRÁFICA

ALKMIM, NASSER SAMIR

Implementação Computacional da Solução de Problemas Térmicos e Mecânicos pelo Método dos Elementos Finitos em Python[Distrito Federal] 2016.

xvi, 147 p., 210 x 279 mm (ENC/FT/UnB, Bacharel, Engenharia Civil, 2016)

Monografia de Projeto Final. Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Civil e Ambiental.

- | | |
|---------------------------------|---------------------------|
| 1. Método dos Elementos Finitos | 3. Transferência de calor |
| 2. Elasticidade Linear | 4. Python |
| I ENC/FT/UnB | II Título (Bacharel) |

REFERÊNCIA BIBLIOGRÁFICA

ALKMIM, N.S. (2016). Implementação Computacional da Solução de Problemas Térmicos e Mecânicos pelo Método dos Elementos Finitos em Python. Publicação G.PF-001/16, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, xvi, 147 p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Nasser Samir Alkmim

TÍTULO DA MONOGRAFIA DE PROJETO FINAL: Implementação Computacional da Solução de Problemas Térmicos e Mecânicos pelo Método dos Elementos Finitos em Python

GRAU / ANO: Bacharel em Engenharia Civil / 2016

É concedida à Universidade de Brasília a permissão para reproduzir cópias desta monografia de Projeto Final e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta monografia de Projeto Final pode ser reproduzida sem a autorização por escrito do autor.

Nasser Samir Alkmim

SHIS QI 15 Conj. 15 Casa 17 - Lago Sul

71.635-350 - Brasília/DF - Brasil

Resumo

Esse trabalho tem como objetivo a apresentação da metodologia de solução computacional utilizando o método dos elementos finitos para problemas em engenharia relacionados à transferência de calor e à mecânica. As soluções computacionais, ou aproximadas, são obtidas por ferramentas de discretização sendo o Método dos Elementos Finitos e o Método das Diferenças Finitas alguns exemplos destes. O Método dos Elementos Finitos atualmente representa o padrão na indústria padrão solução de modelos em mecânica e em outros ramos da engenharia, isso devido à sua simplicidade de implementação e versatilidade. O modelo matemático para descrever o comportamento elástico de sólidos, teoria de estruturas linear e transferência de calor são apresentados juntamente com os respectivos tratamentos para o procedimento numérico e posterior implementação computacional utilizando a linguagem de programação Python. Resultados dos programas implementados, `elastopy`, `sapy` e `diffuspy`, utilizando elementos planos de quatro nós são verificados pela solução de problemas simples com solução analítica. Alguns exemplos são mostrados para demonstrar as capacidades dos programas desenvolvidos. As rotinas de cálculo e o pós processamento são efetuados na linguagem de programação Python que é uma linguagem de alto nível, de simples sintaxe e de fácil utilização.

Palavras-chaves: Método dos Elementos Finitos. Elasticidade Linear. Transferência de Calor. Python.

Abstract

This work aims to present a well known methodology for computational solutions using the Finite Element Method in engineering problems. Computational solutions, or approximations, are obtained by means of discretization which includes the Finite Element Method and the Finite Differences Method. The Finite Element Method represents the standard tool for solving problems in mechanics and other engineering fields, mainly due its simplicity approach and versatility. The mathematical model for linear elastic solids, linear elastic structures and heat transfer problems are presented together with their treatment for a numerical solution and also computational implementation using Python programming language. Results from the solvers implemented, elastopy, sapy and diffuspy, using plane quadrilateral and two dimensional frame elements are verified through comparisons with analytical solution for simple problems. Examples are presented to demonstrated the libraries capabilities. The computer programs and post processing of results are all made with the Python programming language which is a high level scripting language with a simple syntax and is easy to use.

Key-words: Finite Element Method. Linear Elasticity. Heat Transfer. Python.

Lista de Figuras

Figura 3.1 – Barra engastada sujeira a uma força de corpo distribuída $b(x)$ e uma força R em sua extremidade. (BATHE, 1996)	12
Figura 3.2 – Problema discreto. (GOVINDJEE, 2014)	14
Figura 3.3 – Analogia vetorial do método de Galerkin. O vetor u indica a solução exata, o vetor \tilde{u} indica a solução aproximada e e o erro. A melhor solução, ou seja, menor erro, é obtida quando o erro e a solução aproximada são ortogonais. (ZOHDI, 2014)	18
Figura 3.4 – Funções de forma linear e quadrática para elemento unidimensional.	22
Figura 3.5 – Funções de forma bilinear para elemento plano de quatro lados.	22
Figura 3.6 – Esquema do diagrama de Tonti. (TONTI, 2013)	26
Figura 3.7 – Esquema do diagrama de Tonti para representação de um problema de valor sobre o contorno. (FELIPPA, 2005)	26
Figura 4.1 – Estado de tensões descrito por 9 componentes do tensor tensão de Cauchy em um volume infinitesimal contido no interior do corpo.	32
Figura 4.2 – Elemento infinitesimal com força de corpo b_2 por unidade de volume na direção x_2	33
Figura 4.3 – Equilíbrio interno local com tensões na direção x_2 em cada face.	33
Figura 4.4 – Deformações específicas em um elemento infinitesimal. (SLAUGHTER; VERLAG, 2003)	35
Figura 4.5 – Ilustração do problema da elasticidade num corpo geral. O volume do corpo está sujeita às forças de corpo \mathbf{b} e às forças de superfície dadas pelo vetor tensão $\bar{\mathbf{t}}$. A superfície é dividida em Γ_u e Γ_t . (FELIPPA, 2005)	41
Figura 4.6 – Diagrama de Tonti com esquema gráfico do problema da elasticidade linear em sua forma forte. Nele é possível observar as conexões fortes através das equações cinemática, constituição e equilíbrio. (TONTI, 2013)	41
Figura 4.7 – Ilustração do modelo da elasticidade linear com as conexões fracas, em linha pontilhada. (FELIPPA, 2005)	43

Figura 4.8 – Ilustração do modelo da elasticidade linear com a notação matricial. (FELIPPA, 2005)	46
Figura 4.9 – Ilustração do processod e discretização e de um elemento individual de quatro nós.	47
Figura 4.10–Ilustração do espaço isoparamétrico. Os valores (\hat{x}, \hat{y}) representam a coordenada cartesiana do nó 2 desse elemento.	48
Figura 4.11–Ilustração do processo de mudança de coordenada e formação da matriz Jacobiana.	50
Figura 4.12–Classe pai dos elementos e classe específica que herda os atributos. . . .	60
Figura 4.13–Rotina de cálculo para solução do problema estático.	61
Figura 4.14–Parse class inside the gmsh module	64
Figura 4.15–Verificação do método aplica as condições de contorno de deslocamento, o fator de ampliação do campo de deslocamento é 0.1.	67
Figura 4.16–Verificação do método que cria o vector de carregamento devido às forças de corpo.	68
Figura 4.17–Verificação do método que cria o vector de carregamento devido aos vetores tensão no contorno, o deslocamento possui um fator de ampliação de 100.	70
Figura 4.18–Resultado para deslocamento e tensão normal na direção x no plano x para uma viga com condições de contorno de deslocament na linha neutra e na linha inferior, vetor tensão distribuído uniformemente na linha superior. O deslocamento foi ampliado em 2 vezes.	72
Figura 4.19–Geometria com furo circular.	73
Figura 4.20–Resultado para deslocamento e tensão normal na direção x no plano x numa malha com 208 elementos.	74
Figura 4.21–Resultado para tensão normal na direção x no plano x na seção central que passa pelo furo e acompanhando a borda direta do furo, a linha reta na região do furo delimita o valor da tensão máxima obtida de forma empírica para um $K_t = 2,42 = \sigma_{nom}/\sigma_{max}$, sendo o $\sigma_{nom} = 1,334Pa$	74
Figura 4.22–Geometria da barragem analisada, à esquerda o modelo com a fundação e à direita o modelo sem.	75
Figura 4.23–Resultado do campo de deslocamentos ampliado 2000 vezes.	76
Figura 4.24–Resultado da comparação entre as tensões, σ_x e σ_{xy} , obtidas do modelo com fundação, modelo sem fundação e pelo método analítico em duas alturas: $y = 0m$, na base e $y = 40m$	76
Figura 4.25–Resultados para o campo de tensões, os campos na parte superior são do modelo com fundação e os campo na parte inferior são referentes ao modelo sem fundação.	77

Figura 4.26–Diagrama de classificação para o problema elástico dinâmico. (TONTI, 2013)	78
Figura 5.1 – Graus de liberdade de um nó no plano e no espaço.	80
Figura 5.2 – Discretização de um elemento estrutural.	81
Figura 5.3 – Curva tensão-deformação e área abaixo do gráfico que indica a densidade de energia de deformação.	82
Figura 5.4 – Diagrama de Tonti para o modelo matemático de barra. (FELIPPA, 2005)	84
Figura 5.5 – Graus de liberdade em um elemento de barra e funções de forma linear para aproximação. (FELIPPA, 2004)	86
Figura 5.6 – Interpretação da relação aproximada da deflexão, v , com curvatura, $\kappa = 1/\rho$. Dentre as aproximações estão: $\kappa \approx d^2v/dx^2$, $\theta \approx \tan(\theta)$ e $ds \approx dx$. (FELIPPOU, 2013)	88
Figura 5.7 – Diagrama de Tonti para o modelo matemático de viga. (FELIPPA, 2005)	88
Figura 5.8 – Viga com carregamento uniformemente distribuído $w = -5$	93
Figura 5.9 – Resultado obtido utilizando o programa ftool para viga com carregamento uniformemente distribuído $w = -5$	93
Figura 6.1 – Diagrama de Tonti para o problema da condução térmica em regime permanente. (TONTI, 2013)	97
Figura 6.2 – Campo de temperatura com com $T = 50^\circ C$ na linha 3 e $\bar{q} = -20W/m$ na linha 1, $\lambda = 1W/^\circ C m$	101
Figura 6.3 – Campo de temperatura com com $T = 50^\circ C$ na linha 3, $\bar{q} = -20W/m$ na linha 1, $\lambda = 1W/^\circ C m$ e com geração interna de calor $\sigma_q = 40W/m^2$	102
Figura 6.4 – Campo de temperatura com condutividade constante	103
Figura 6.5 – Campo de temperatura com condutividade descontínua não variável	104
Figura 6.6 – Campo de temperatura com condutividade variavel no espaço	105
Figura 6.7 – Campo de temperatura em geometria com furo	106
Figura 6.8 – Diagrama de Tonti para o problema da condução térmica transiente com uma fonte de calor dependente da temperatura. (TONTI, 2013)	107
Figura 7.1 – Resultado utilizando Gmsh como ferramenta de pós-processamento	111
Figura A.1 – Rotação dos eixos.	118
Figura A.2 – Equilíbrio do corpo livre definido por um nó. A forças de extremidade no elemento, (F_{EE}), dadas pela letra \mathbf{q} equilibram-se com as forças nodais (F_N) representadas pela letra \mathbf{p}	120
Figura A.3 – Forças elementares de extremidade independentes.	121
Figura A.4 – Equilíbrio do corpo livre definido por um elemento.	121

Figura A.5 – Equilíbrio nodal do modelo estrutural. Na figura, q_1, q_2, \dots, q_6 , são as incógnitas $nq = 6$, p_1, p_2, p_3 indicam o valor nodal no grau de liberdade livre, $nf = 3$, e p_4, p_5, \dots, p_9 indicam os graus de liberdade restringidos por apoios.	122
Figura A.6 – Deformação de flexão elementar contínua calculada considerando pequenas rotações, $tg(\theta) \approx \theta$. Em cada trecho dx da estrutura, um δ pode ser calculado. A soma dos δ produz um Λ que representa o cateto oposto no triângulo maior pelo qual é possível calcular a deformação de flexão, v_2 . (FELIPPOU, 2013)	125
Figura A.7 – Relação entre deformação intra-nodal e deslocamentos de extremidade, μ . A figura explicita o movimento de corpo rígido devido aos deslocamentos de extremidade e também as componentes nas referências dos eixos global e local. O ângulo α é a inclinação do elemento no espaço com relação ao eixo x global, o ângulo β é a inclinação da corda do elemento. (FELIPPOU, 2013)	127
Figura A.8 – Deformação elementar de flexão devido ao esforço interno de momento fletor. O diagrama de distribuição da curvatura, na parte central, é obtido por meio do diagrama de momento fletor. Na parte inferior está o elemento em sua forma deformada com a indicação das deformações elementares nas extremidades que foram calculadas por meio da integral em (4.48) e (4.49). (FELIPPOU, 2013)	130
Figura A.9 – Deformação elementar de flexão devido ao carregamento distribuído (esquerda) e devido à campo de temperatura que produz curvatura constante (direita). O elemento em sua forma deformada com a indicação das deformações elementares nas extremidades que foram calculadas por meio da integral em (4.48) e (4.49). (FELIPPOU, 2013)	131

Lista de símbolos

A	[Geometria] Área
b	[Mecânica Contínua] Força de corpo
c	[Termodinâmica] Calor específico
\underline{C}	[Mecânica] Tensor de rigidez elástico
E	[Mecânica] Módulo de elasticidade
F	[Mecânica] Força
g	[Condução Térmica] Gradiente de temperatura
g	[Gravitação Clássica] Aceleração gravitacional
I	[Mecânica] Momento de Inércia
\mathcal{M}	[Mecânica] Momento fletor
\mathcal{N}	[Mecânica] Força axial
n	Vetor normal unitário
q	[Condução Térmica] Densidade de corrente de calor
t	[Mecânica Contínua] Vetor tensão
T	[Termodinâmica] Temperatura
u	[Mecânica] Deslocamento
u	[Termodinâmica] Densidade de energia interna
v	[Mecânica] Velocidade
v	[Mecânica dos Sólidos] Deslocamento transversal
γ	[Mecânica] Deformação de cisalhamento linear

$\underline{\varepsilon}$	[Mecânica] tensor de deformação linear
λ	[Condução Térmica] Condutividade
λ	[Mecânica dos Sólidos] Segunda constante de Lamé
ρ	[Mecânica dos Sólidos] Densidade de massa
$\underline{\sigma}$	[Mecânica Continua] Tensor tensão de Cauchy
σ_q	[Termodinâmica] Densidade da taxa de geração de calor interna
κ	[Mecânica dos Sólidos] Curvatura

Sumário

	1 INTRODUÇÃO	1
1.1	Generalidades	1
1.2	Motivação	1
1.3	Objetivos	2
1.4	Justificativas para o Trabalho	3
1.5	Organização do Trabalho	3
	2 REVISÃO BIBLIOGRÁFICA	5
2.1	Histórico do Método dos Elementos Finitos	5
2.2	Teoria do MEF	8
2.3	Implementação computacional do MEF	8
2.4	Utilização do MEF	9
	3 FUNDAMENTOS	11
3.1	Introdução	11
3.1.1	Forma Forte	11
3.1.2	Forma Fraca	12
3.1.3	Forma Variacional	13
3.2	Discretização	15
3.2.1	Método das Diferenças Finitas	16
3.2.2	Método dos Residuais Ponderados	16
3.2.2.1	Método de Galerkin	17
3.2.2.2	Mínimos Quadrados	18
3.2.3	Método de Rayleigh-Ritz	18
3.3	Método dos Elementos Finitos	19
3.3.1	Aproximação pelo Método dos Elementos Finitos	20
3.3.2	Formulação por Métodos Variacionais	23
3.3.3	Formulação por Métodos de Residuais Ponderados	24

3.4	Diagrama de Tonti	25
3.4.1	Definições para Classificação dos Modelos	27
	4 ELASTICIDADE LINEAR: SÓLIDOS	29
4.1	Introdução	29
4.1.1	Teoria da Elasticidade	30
4.1.2	Equações de Campo	31
4.2	Relação de Equilíbrio	31
4.2.1	Tensão	31
4.2.2	Equilíbrio Interno	33
4.2.3	Forças de Corpo	34
4.2.4	Forças de Superfície	34
4.3	Relação Cinemática	34
4.3.1	Deformação Específica	35
4.3.2	Relação Deformação-Deslocamento	37
4.3.3	Compatibilidade	37
4.4	Relação Constitutiva	38
4.4.1	Elasticidade	38
4.4.2	Linearidade	38
4.4.3	Isotropia	39
4.5	Condições de Contorno	40
4.6	Diagrama de Tonti: Modelo Elástico Linear	40
4.7	Formulação Variacional	41
4.7.1	Construção da Forma Variacional	42
4.7.2	Escolha da Variável Primária	43
4.7.3	Ligações Fracas	43
4.7.4	Forma Fraca	43
4.7.5	Identificando a Função Peso	44
4.7.6	Identificando o Funcional	44
4.7.7	Princípio de Minimização da Energia Potencial	44
4.8	Equações na Forma Matricial	45
4.9	Discretização	46
4.10	Aproximação	47
4.10.1	Mudança de Coordenadas	49
4.10.2	Jacobiano	49
4.11	Construção do Elemento	52
4.11.1	Derivadas das Funções de Forma	53
4.11.2	Matrizes Elementares	55
4.11.3	Matriz de Rigidez \mathbf{K}	55
4.11.4	Vetor de carregamento \mathbf{P}_b	56

4.11.5	Vetor de carregamento P_t	57
4.11.6	Vetor de carregamento P_e	59
4.11.7	Classe do elemento	60
4.12	Solução do problema estático	61
4.13	Malha	62
4.14	Verificação	66
4.14.1	Condições de Contorno de Deslocamento	66
4.14.2	Forças de Corpo	68
4.14.3	Condição de Contorno com Vetor Tensão Aplicado	69
4.15	Exemplos	71
4.15.1	Viga bi Apoiada	72
4.15.2	Geometria com Furo	73
4.15.3	Barragem Gravidade	75
4.16	Problema Dinâmico	78
	 5 ELASTICIDADE LINEAR: ESTRUTURAS	 79
5.1	Introdução	79
5.1.1	Modelo estrutural	80
5.2	Discretização	80
5.3	Formulação por Métodos Variacionais	81
5.3.1	Energia Potencial Total	82
5.3.2	Princípio de Minimização da Energia Potencial Total	83
5.4	Elemento de Barra	83
5.4.1	Esforço Interno Axial	84
5.4.2	Deformação Axial	84
5.4.3	Diagrama de Tonti: Modelo de Barra	84
5.4.4	EPT Elemento de Barra	85
5.5	Elemento de Viga	86
5.5.1	Hipóteses da Teoria de Vigas	87
5.5.2	Diagrama de Tonti: Modelo de Viga	87
5.5.3	EPT Elemento de Viga	88
5.6	Procedimento de Solução	92
5.7	Verificação	92
	 6 CONDUÇÃO TÉRMICA	 94
6.1	Introdução	94
6.2	Variáveis de Interesse	95
6.2.1	Variáveis de Fonte	95
6.2.2	Variáveis de Configuração	96
6.3	Modelo Matemático	96

6.3.1	Diagrama de Tonti: Condução Térmica	97
6.3.2	Modelo na Forma Forte	97
6.3.3	Modelo na Forma Fraca	98
6.4	Discretização	98
6.5	Aproximação	98
6.6	Construção do Elemento	99
6.6.1	Matrizes Elementares	99
6.7	Solução do Problema Estacionário	100
6.8	Verificação	100
6.9	Exemplos	101
6.9.1	Condutividade Constante	103
6.9.2	Condutividade Descontínua Constante	104
6.9.3	Condutividade Variável	105
6.9.4	Geometria Variável	106
6.10	Problema Transiente	107
	7 CONCLUSÃO	108
7.1	Método dos Elementos Finitos	108
7.2	Python	109
7.3	Gmsh	109
7.4	Trabalhos Futuros	110
	REFERÊNCIAS	112
	APÊNDICES	117
	APÊNDICE A – TÓPICOS EM MECÂNICA	118
A.1	Deformação Cisalhante	118
A.2	Formulação por Mecânica do Sólidos	119
A.2.1	Relação de Equilíbrio	120
A.2.1.1	Forças Elementares de Extremidade Independentes	120
A.2.1.2	Forças de Extremidade e Esforços Internos	121
A.2.1.3	Equilíbrio Nodal	122
A.2.2	Relação Cinemática	123
A.2.2.1	Deformações no Ponto Material e na Seção	123
A.2.2.2	Deformações Elementares	124
A.2.2.3	Continuidade Nodal	124
A.2.2.3.1	Passo 1: Relação entre u e μ	126
A.2.2.3.2	Passo 2: Relação entre μ e as dIN	126

A.2.2.3.3	Passo 3: Relação entre dIN e v	127
A.2.3	Relação Constitutiva	129
A.2.3.1	Elemento de Barra	129
A.2.3.2	Elemento de Viga	129
A.2.4	Resumo das Equações	131
A.2.5	Método dos Deslocamentos	132
	APÊNDICE B – PRELIMINARES MATEMÁTICAS	134
B.1	Notação Geral	134
B.1.1	Escalar	134
B.1.2	Vetor	134
B.1.3	Matrix	134
B.1.4	Tensor de Segunda Ordem	134
B.1.5	Tensor de Quarta Ordem	135
B.2	Diferentes Igualdades	135
B.3	Notação Indicial	135
B.4	Cálculo Multivariacional	136
B.4.1	Gradiente de um Escalar	136
B.4.2	Gradiente de um Vetor	136
B.4.3	Divergente de um Vetor	136
B.4.4	Divergente de um Tensor de Segunda Ordem	136
B.5	Classificação das Variáveis	136
B.5.1	Variáveis de Fonte	137
B.5.2	Variáveis de Configuração	137
B.5.3	Variáveis de Energia	138
B.6	Equações Físicas	138
B.7	Teorema da Divergência	139
B.8	Regra do Produto do Cálculo Multivariável	139
B.9	Regra da Cadeia	140
B.10	Produto Duplo de Tensores	140
B.11	Kronecker Delta	140
B.12	Introdução ao Cálculo Variacional	140
B.13	Quadratura Gaussiana	142
	APÊNDICE C – ASPECTOS COMPUTACIONAIS	143
C.1	Python	143
C.2	Softwares em Python	144
C.3	Performance em Python	145
C.4	Programação Orientada Objeto	146

1 Introdução

Nesse capítulo serão tratados os aspectos gerais deste trabalho, sua motivação, os objetivos almejados e sua organização.

1.1 Generalidades

Esse trabalho consiste em uma coleção de projetos desenvolvidos ao longo da graduação. Em todos os projetos, um tema em particular manteve-se constante: a busca por soluções numéricas de modelos matemáticos. Soluções numéricas, ou aproximadas, para problemas de natureza contínua descritos por modelos matemáticos através de equações representam uma classe de solução conveniente para a Engenharia, (MILLMAN; AIVAZIS, 2011). Soluções numéricas são obtidas via cálculos no computador. Optou-se pela utilização da linguagem de programação Python por ser uma linguagem de sintaxe simples, fácil acesso á documentação e ampla disponibilidade de bibliotecas que estendem sua capacidade. Mais informações sobre essa linguagem estão no Apêndice C.

1.2 Motivação

Durante a segunda guerra mundial viu-se a necessidade de se obter respostas a problemas cuja solução era limitada pela grande quantidade de cálculos. O exército americano interessado em calcular trajetórias balísticas desenvolveu o *Electronic Numerical Integrator Analyzer and Computer (ENIAC)*, um dos primeiros computadores eletrônicos. O ENIAC foi construído no período entre 1943 e 1945 na Universidade da Pennsylvania. Suas proporções eram gigantescas, tinha em torno de 30 metros de largura e 2,4 metros de altura com uma massa de 27 toneladas, (DICTIONARY, 2003). Com o advento dos transistores, desenvolvido em 1947 pelos Laboratórios da Bell Telephone, foi possível a construção de processadores capazes de desempenhar operações básicas (aritmética, lógica e controle de entrada e saída), (CRUZ, 2013).

Em 1958, Ed Wilson cria o primeiro código para solução do problema da elasticidade plana linear em FORTRAN (FORMula TRANslator) na Universidade da Califórnia,

Berkeley. Wilson, na época, era aluno de mestrado do Professor Clough. Juntos publicaram primeiro artigo onde o nome “Método dos Elementos Finitos” foi cunhado (CLOUGH; WILSON, 1962).

Em 1968, o software NASTRAN (NASA STRuctural ANalysis) foi desenvolvido para a NASA pela *Computer Sciences Corporation CSC*. O programa foi utilizado para projetar de forma mais eficiente ônibus espaciais. O programa também foi utilizado por outras indústrias gerando economia no desenvolvimento de produtos, (NASTRAN, 2016).

Atualmente, o método dos elementos finitos é o padrão da indústria para discretização de problemas em mecânica dos sólidos. A colocação mais precisa do método no contexto da mecânica será feita em um capítulo posterior. O método também apresenta um aspecto mais geral em seu procedimento o que permite sua aplicação em diversos outros tipos de problemas não mecânicos, (RAO, 2013):

- Condução térmica: distribuição do campo de temperatura em estado permanente e transiente em corpos sólidos ou fluidos;
- Hidráulica: escoamento potencial, escoamento com superfície livre, escoamento em camada limite;
- Eletromagnetismo: análise de máquinas de indução, corrente de Foucault, perdas no maciços em máquinas elétricas.

1.3 Objetivos

Dentre os objetivos do trabalho estão:

1. Apresentação dos modelos matemáticos para descrever o comportamento mecânico dos sólidos, estruturas e do problema de transferência de calor.
2. Apresentação da metodologia de discretização utilizada, o Método dos Elementos Finitos.
3. Apresentação da metodologia de implementação do método para solução das equações diferenciais parciais de forma aproximada, utilizando a linguagem Python.
4. Produção de rotinas em Python para solução do problema da elasticidade plana.
5. Apresentação do pós-processamento dos resultados em forma de gráficos produzidos em Python.

1.4 Justificativas para o Trabalho

Por que fazer um programa do zero? Os principais fins em se fabricar um programa caseiro são educacionais e acadêmicos. Um programa criado do início permite verificar o aprendizado teórico. Utilizando a sintaxe clara e de alto nível do Python, podem-se criar rotinas de simples entendimento que por sua vez possam ser apresentadas num ambiente acadêmico. As rotinas são agrupadas em módulos e sub-módulos cuja nomenclatura foi criteriosamente escolhida para facilitar o entendimento. O código apresenta-se como um pseudo-código pois a sintaxe é clara, muito próxima da linguagem humana. Isso permite o fácil intercâmbio de idéias entre os desenvolvedores do programa. Além disso, a sintaxe clara e modularidade na organização permite uma maior acessibilidade de novos desenvolvedores. O desenvolvimento próprio também facilita a customização. Vale mencionar que a linguagem Python já apresenta pacotes que disponibilizam rotinas matemáticas consagradas para álgebra linear, dessa forma evita-se a disposição de tempo em problemas puramente matemáticos.

Por que utilizar modularidade? A prática de criar módulo é padrão na indústria. As vantagens principais são a organização, arquitetura, lógica e simples, e a manutenção e extensibilidade. A outra opção seria o procedimento ultrapassado de manter tudo em um arquivo com milhares de linhas de código.

Por que não utilizar um software comercial? Dentre os softwares comerciais podem se citar: ANSYS, Abaqus e COMSOL. Os dois primeiros apresentam licenças estudantis. Os softwares comerciais possuem o código fechado, não se tem acesso à implementação. Não apresentam fácil mecanismos para extensões criadas pelo usuário. Novas técnicas e métodos estão a todo momento sendo desenvolvidas. Dentre essas podem-se citar o método das células finitas, técnicas sem malha, otimização topológica, elementos finitos generalizado, dentre outras.

Por que não utilizar um software livre? Dentre os softwares de código livre, que respeitam as liberdades do usuário, estão FEniCS, calculix, gmsh, Fipy, Sfepy, Aster code e outros. Os softwares livre foram escritos em linguagens variadas, geralmente, C, C++ e Python. O software Sfepy, *Simple Finite Elements in Python*, foi criado, como o próprio nome indica, em python. Esse seria o programa ideal para se utilizar, no entanto, encontra-se em um estágio avançado de desenvolvimento e possui uma filosofia diferente da proposta nesse trabalho.

1.5 Organização do Trabalho

No Capítulo 1 são introduzidos aspectos gerais desse trabalho e sua motivação.

No Capítulo 2 é apresentado uma breve revisão bibliográfica sobre a implementação computacional do método dos elementos finitos. Ainda nesses capítulo são apresentados históricos e contextualização sobre modelos matemáticos e o método dos elementos finitos.

O Capítulo 3 é responsável por introduzir conceitos gerais sobre procedimentos de aproximação de equação diferenciais. Nele são tratados as definições de modelos matemáticos, forma forte, fraca e variacional; e os respectivos métodos de discretização para cada uma dessas formas. O método dos elementos finitos é apresentado em sua forma geral. Por fim o diagrama de classificação de Tonti é mostrado juntamente com os componentes que o compõem.

O Capítulo 4 expõe o problema da elasticidade linear para sólidos e a abordagem para sua solução numérica e implementação computacional. A implementação é verificada através de comparações com soluções analíticas e alguns exemplos para demonstrar as capacidades do programa desenvolvido.

O Capítulo 5 trata da elasticidade linear em estruturas, um caso particular de sólidos, e expõe o procedimento para solução numérica.

O Capítulo 6 expõe o modelo matemático para o problema da transferência de calor, em seguida aborda o procedimento para solução numérica e posterior implementação computacional. A implementação é verificada através de comparações com soluções analíticas e alguns exemplos para demonstrar as capacidades do programa desenvolvido.

2 Revisão Bibliográfica

Nesse capítulo será apresentado uma breve revisão bibliográfica sobre o histórico do método dos elementos finitos, sobre sua implementação computacional e sua aplicação em problemas térmicos e mecânicos.

2.1 Histórico do Método dos Elementos Finitos

O Método dos Elementos Finitos (MEF) relaciona-se com a mecânica computacional como um dos métodos de discretização do domínio espacial. Mecânica computacional é um dos ramos presentes no estudo da mecânica clássica. O termo computacional serve para distinguir esse ramo da mecânica teórica que prima a construção de modelos matemáticos para descrição de fenômenos físicos enquanto a área computacional preocupa-se com a solução de problemas específicos através de métodos numéricos de aproximação, (FELIPPA, 2004). Dentre os outros métodos estão as diferenças finitas, elementos de contorno, volumes finitos e outros.

O método dos elementos finitos é um método de discretização utilizado para aproximar a solução de equações diferenciais parciais, (BRENNER; SCOTT, 2007). Atualmente esse é o método padrão utilizado na solução de problemas envolvendo estruturas e mecânica dos sólidos. Dentre os fatores que contribuem para o sucesso do método estão a sua flexibilidade e padronização nos mecanismos de solução. O procedimento do método consiste em resolver de forma aproximada as equações de campo em um sub domínio, denominado elemento, para em seguida por um processo de assemblagem recuperar a solução em todo o domínio mantendo-se a continuidade ou o equilíbrio nodal entre os elementos, (MEEK, 1996).

No contexto do início do desenvolvimento da técnica em sua aplicação computacional no período 1956-1964 o método emergiu com a estrutura do método dos deslocamentos conhecida como *Método Da Rigidez Direta*. Sua popularidade na análise de estruturas deve-se ao fato de ser convenientemente traduzido à um computador através da álgebra linear.

Durante o início do desenvolvimento da técnica podem-se citar 5 importantes

autores que contribuíram para divulgação e aprimoramento da ferramenta: Courant, Argyris, Turner, Clough e Zienkiewicz (MEEK, 1996). Além desses autores, outros nomes como: Ritz, Bubnov, Galerkin são notórios no histórico, (GANDER; WANNER, 2012a).

Walter Ritz (1878-1909) foi um matemático e físico suíço. A idéia de seu método consiste em definir um sub conjunto finito de funções para aproximar o problema por meio da combinação linear entre essas funções. Então, escolhe-se um conjunto de funções conhecidas

$$\psi_i(x), \quad i = 1, 2, \dots \quad (2.1)$$

para então aproximar uma solução w através da combinação linear dessas funções,

$$w(x) = \sum_i a_i \psi_i, \quad (2.2)$$

onde a_i são os coeficientes a serem determinados. As funções ψ_i devem ser escolhidas de forma a satisfazer as condições cinemáticas do problema.

Quando essa aproximação é feita, a energia potencial total Π passa a ser uma função de um conjunto finito de parâmetros $\Pi(a_i)$. Essa pode então ser minimizado para cada um desses parâmetros.

Sua contribuição foi a aplicação da teoria de cálculo variacional, puramente matemática, em uma metodologia para solução de problemas práticos, (GANDER; WANNER, 2012b).

Boris Grigor'evich Galerkin (1871-1945) foi um engenheiro e matemático Russo, antiga União Soviética. Galerkin em 1909 fez uma expedição científica na Europa, por onde passou pela Suíça dentre outros países. Não há evidência de que Galerkin e Ritz se encontraram, no entanto, Galerkin faz referência à Ritz em seu trabalho onde o método desenvolvido leva seu nome, Método de Galerkin, Seção 3.2.2.1. Contudo, nesse artigo, Galerkin nomeia o método como Método de Ritz.

A grande contribuição de Galerkin foi perceber que a minimização de um funcional quadrático não era um caminho único para se obter um conjunto finito de equações algébricas. Ele formaliza o trabalho desenvolvido por Ivan Bubnov (1872-1919), devido à isso o método também é conhecido como Bubnov-Galerkin. A nova abordagem utilizada por Bubnov e Galerkin consiste em multiplicar a equação diferencial (em sua forma residual) por uma função teste e integrá-la no domínio. Essa abordagem hoje é classificada como um *método de residuais ponderados*, (GANDER; WANNER, 2012b).

Richard Courant (1888-1972), matemático alemão-estadunidense, desenvolveu a idéia de minimização de um funcional utilizando funções lineares aproximadas sobre sub regiões do domínio, demarcadas por pontos discretos. O funcional em questão representa a função da energia potencial que quando minimizada reproduz a equação de equilíbrio.

Essa abordagem representa uma aplicação do método de Rayleigh-Ritz, que não define uma função de aproximação específica.

Courant reconhece Ritz e Rayleigh como pioneiros do método que ganha popularidade entre matemáticos teóricos, cálculo variacional. Courant é o responsável também por reconhecer o ponto chave no método que é a escolha adequada das funções de aproximação. Em vez de utilizar autofunções, como Ritz, ou funções trigonométricas, como Bubnov e Galerkin, ele propõe a utilização de *funções chapéu*. A escolha das funções chapéu permitiu grandes avanços do método, principalmente com o advento dos microcomputadores. Essas funções são utilizadas nesse trabalho, (GANDER; WANNER, 2012b).

John Argyris (1913-2004) foi um engenheiro grego pioneiro no desenvolvimento da mecânica computacional. Em seu artigo *Energy Theorems and Structural Analysis* ele desenvolve a teoria da análise estrutural matricial para elementos discretos. No artigo também é demonstrado a construção da matriz de rigidez para o problema da elasticidade plana em um painel retangular, que gerava uma matriz de rigidez elementar 8 por 8, (ARGYRIS; KELSEY, 1960);

Johnathan Turner (1915-1995), matemática e engenheiro estrutural, trabalhava na Boeing analisando as estruturas de aeronaves. Turner em um de seus trabalhos apresenta a matriz de rigidez global para um elemento de treliça juntamente com a construção de um elemento plano triangular chegando à um resultado equivalente ao obtido por Argyris, (CLOUGH, 2004).

Ray William Clough (1920-), engenheiro estrutural e professor emérito da Universidade da Califórnia em Berkeley, foi o primeiro a cunhar a técnica de Elementos Finitos num artigo publicado em 1960. Em 1952, Clough, já como professor de Berkeley, se candidatou para uma posição na empresa Boeing. Trabalhou sob supervisão do engenheiro Turner no tópico de dinâmica das estruturas. O trabalho consistia na análise da vibração de um modelo de asa para aeronaves. O desenvolvimento de modelo proposto para análise da asa feito por Clough foi de uma coleção de elementos de vigas, no entanto, no curto período de tempo em que esteve trabalhando, não produziu resultados satisfatórios. Após despedir-se de Turner, este sugeriu a adoção de elementos de placa bidimensionais para modelar a asa. Durante o próximo ano, 1953, Clough dedicou-se no desenvolvimento da matriz de rigidez para esse elemento. Ele iniciou com elementos retangulares mas posteriormente enfatizou apenas elementos de triangulares, nesse ponto já os chamando de *elementos*. O nome *Método dos Elementos Finitos* foi cunhado para substituir o nome *Método da Rigidez Direta* utilizado pela Boeing. O nome foi escolhido de forma a melhor incorporar o significado da técnica, (CLOUGH, 2004).

2.2 Teoria do MEF

A teoria do método dos elementos finitos foi amplamente discutida e explanada ao longo das últimas décadas. Dezenas de livros foram lançados sobre o assunto, dentre os quais podem ser citados: (BATHE, 1996; ZIENKIEWICZ; TAYLOR, 2000; COOK, 2002; RAO, 2011).

2.3 Implementação computacional do MEF

A seguir são apresentados alguns trabalhos que possuem como tópico principal o desenvolvimento de rotinas computacionais para utilização do método dos elementos finitos.

Ed Wilson, em Clough e Wilson (1962), desenvolveu o primeiro programa de elementos finitos para resolver problemas do estado plano de tensão utilizando a modelagem linear da elasticidade. O programa foi utilizado para resolver o problema de uma barragem gravidade, Norfolk Dam.

Robert Taylor, em Taylor (1987), escreveu um programa de elementos finitos geral, FEAP (Finite Element Analysis Program). O programa foi escrito em FORTRAN e continua em desenvolvimento e expansão. Novos módulos com a implementação de diferentes metodologias de solução e inclusão de novos elementos atualizam o programa para o atual nível de desenvolvimento do método dos elementos finitos. A interface de interação com o programa também foi atualizada para acomodar sua utilização por meio de outras linguagens como Matlab e C.

Bruce Forde, em Forde, Foschi e Stierner (1989), apresenta os problemas de implementação do MEF na forma convencional, programação em procedimento, e descreve as vantagens da utilização do paradigma de programação orientada objeto.

Graham Archer, em Archer (1996), desenvolve uma arquitetura de um programa de elementos finitos utilizando a linguagem C++ e o paradigma de programação orientada objeto. A arquitetura foi desenvolvida com o objetivo de se produzir um código que seja facilmente extensível para novos módulos e inclusão de novos elementos sem que seja necessário repetir código. Essa paradigma é utilizado no presente trabalho pois permite uma organização mais clara da estrutura de dados do programa.

Em Patzák e Bittnar (2001) é apresentado um estrutura geral de uma implementação do método dos elementos finitos com objetivo de facilitar a implementação de novas leis constitutivas. No artigo são feitas críticas aos softwares livres que possuem arquitetura complexa e não clara o que dificulta a utilização. Junta-se a isso o fato de haver pouca

documentação disponível para os softwares livres. O objetivo do trabalho é a construção de uma plataforma aberta, modular, de fácil manutenção e com performance comparada à programas em C ou FORTRAN.

Em Wang e Kolditz (2007) o desenvolvimento de um programa orientado objeto para analisar pelo método dos elementos finitos problemas termo-hidro-mecânicos é feito. O programa é feito em C++ e a escolha pelo paradigma de orientação ao objeto é feito com intuito de encapsular os dados em objetos. O programa é testado em aplicações em geotecnia.

Pooyan Dadvand, em Dadvand, Rossi e Oñate (2010), descreve a arquitetura da implementação do método dos elementos finitos de forma que o programa seja geral, reutilizável, extensível e com boa performance.

Em Phunpeng e Baiz (2015) uma implementação pelo método dos elementos finitos é apresentada utilizando o projeto FEniCS. A implementação é feita para resolver o problema da elasticidade e os resultados são comparados com problemas de *benchmark*.

2.4 Utilização do MEF

Em Gartling (1977) é feito um procedimento numérico para análise em duas dimensões do problema de transferência de calor convectivo com parâmetros materiais não linear que dependem da temperatura.

John Hall, em Hall e Chopra (1982), apresenta uma análise dinâmica de uma barragem gravidade utilizando o MEF para discretização do reservatório e da estrutura. Na análise foi utilizado uma geometria de duas dimensões.

Léger, em Leger e Katsouli (1989) faz um estudo sobre a estabilidade de barragens gravidade sob ação de efeitos sísmicos. Nele, elementos com propriedades constitutivas foram utilizados para modelar a interface da barragem com a fundação. Em Leger, Venturelli e Bhattacharjee (1993), e apresentado uma metodologia para o cálculo de tensões em barragens sob efeitos térmicos e mecânicos. Ele utiliza os modelos matemáticos da transferência de calor e elasticidade linear plana. A solução é obtida por meio do método dos elementos finitos. O problema térmico é resolvido em sua forma transiente, em seguida avalia-se o caso crítico de variação de temperatura. Esse caso é então incluído no problema mecânica estático.

Em Codina (1998) um estudo comparativo entre diferentes tipos de método dos elementos finitos é feito para o problema de difusão e convecção. Os métodos comparadas consistem em extensões da formulação por Galerkin, com a adição de termos de estabilização.

Juliano em Scremin (2011) fez uma análise termo-elástica utilizando o MEF em um modelo de duas dimensões para estimar a probabilidade de falha em barragens de concreto compacto a rolo (CCR).

Constantino, em sua tese de mestrado, Nascimento (2016), fez um estudo comparativo de método analíticos e numéricos, MEF e MDF, de tensões em barragem de concreto gravidade. Emílio em Kavamura (2005) desenvolve uma metodologia para análise termo mecânica de barragens construídas em camadas utilizando o método dos elementos finitos através do *software* ANSYS®.

3 Fundamentos

Nesse capítulo serão discutidos as definições e os fundamentos teóricos para obtenção de soluções aproximadas utilizando métodos de discretização: diferenças finitas, residuais ponderados, Rayleigh-Ritz e elementos finitos.

3.1 Introdução

A metodologia de análise de problemas físicos consiste em *idealizar o problema, discretizá-lo e resolvê-lo*. A idealização, como visto no capítulo anterior, é a etapa na qual o fenômeno físico é traduzido para linguagem matemática através de um modelo matemático. O modelo matemático pode assumir três formas equivalentes: *forma forte, forma fraca e forma variacional*. Pelo fato de serem equivalentes, a solução para um deles representa a solução para o outro. A área da matemática que fornece a base para transição entre essas formas é o Cálculo Variacional.

O problema físico pode ser descrito através de *campos*, ou *variáveis físicas*. No caso da elasticidade os campos de deslocamento u_i , deformação ε_{ij} e tensão σ_{ij} ; ou campos de temperatura θ . Os campos são *conectados* através de equações, ou seja, relacionam-se por meio de expressões matemáticas. Essas conexões podem ser *fortes* ou *fracas*. Uma conexão forte é garantida em todos os pontos onde é definida. Já uma conexão fraca é garantida na média (FELIPPA, 2005).

3.1.1 Forma Forte

A forma forte, ou forma diferencial, consiste na representação do problema através de um sistema de equações diferenciais, parciais ou ordinárias, no espaço ou no tempo. O problema é completo com a especificação de condições de contorno.

Problemas nos quais a variável desconhecida é especificada no contorno são chamados de *problemas de valor de contorno* e os problemas nos quais uma condição inicial é especificada são chamados de *problemas de valor inicial*

Problemas simples podem ser resolvidos analiticamente na forma forte através de técnicas de integração como separação de variáveis, fator integrante e outras. Problemas mais complexos podem não apresentar solução analítica de fácil obtenção. Nesse caso, emprega-se métodos para obtenção de soluções aproximadas. Aproxima-se a solução a partir da substituição do problema contínuo por um discreto no qual equações algébricas podem ser formadas. O processo de discretização será tratado na Seção 3.2, a seguir.

Um exemplo de um problema na forma forte: Uma barra de comprimento L na direção x sujeita a um carregamento de corpo distribuído em seu comprimento $b(x)$; sujeita à condições de contorno essenciais de engaste na extremidade e com uma força de tração, R , na outra.

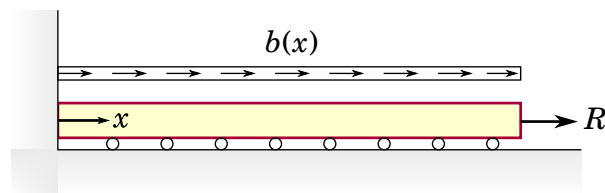


Figura 3.1 – Barra engastada sujeita a uma força de corpo distribuída $b(x)$ e uma força R em sua extremidade. (BATHE, 1996)

A modelagem matemática pode ser representada pelo sistema

$$\text{Forma Forte} \begin{cases} EA \frac{d^2 u}{dx^2} + b = 0, & x \in [0, L], \\ u(0) = 0, \\ EA \frac{du}{dx} = R, & x = L \end{cases} \quad (3.1)$$

onde, $u(x)$ é o campo de deslocamentos e EA é a rigidez axial da barra.

3.1.2 Forma Fraca

A forma fraca consiste na representação do problema através de uma equação integral ponderada ¹. O adjetivo “fraco” diz respeito ao fato de a equação ser resolvida através de uma integral ponderada no domínio, ou seja, em termos de média ao invés de ser solucionada ponto a ponto como na forma forte. Além disso, indica que os requerimentos de continuidade das soluções admissíveis é “enfraquecido”, nesse caso, reduzido a uma ordem inferior. Soluções admissíveis são aquelas que satisfazem condições de compatibilidade e condições de contorno naturais de deslocamento, (COOK, 2002).

Autores como Cook (2002), Bathe (2006), Hughes (2012) incluem dentro de um único grupo a forma fraca e a forma variacional. Já Felippa (2005) faz a distinção entre os dois formatos e a coloca como um passo intermediário entre a forma forte e variacional.

¹ *Weighted integral equation* em inglês.

Utilizando o mesmo exemplo cuja forma forte foi apresentada em (A.12), tem-se a seguinte forma fraca

$$\text{Forma Fraca} \begin{cases} \int_0^L \frac{d\delta u}{dx} EA \frac{du}{dx} dx - \int_0^L b \delta u dx + R\delta u(L) = 0, \\ u(0) = 0, \\ \delta u(0) = 0, \end{cases} \quad (3.2)$$

onde, δu é uma *variação de u* arbitrária, podendo ser, portanto, qualquer função contínua que satisfaz as condições de contorno. A forma fraca do problema é obtida considerando que a igualdade da condição de equilíbrio continua satisfeita para

$$\left(EA \frac{d^2 u}{dx^2} + b \right) \delta u = 0. \quad (3.3)$$

Utilizando a técnica de integração por partes chega-se a (3.2). Em mecânica, esse procedimento é conhecido como princípio dos deslocamentos virtuais, ou princípio dos trabalhos virtuais. Nesse caso, δu representa um deslocamento virtual.

3.1.3 Forma Variacional

A forma variacional consiste na representação do problema como um funcional no qual o mínimo reconstitui a forma forte. O nome variacional é proveniente de um ramo da matemática denominado *Cálculo Variacional* que lida com extremização de funcionais na forma de integrais (RAO, 2013). Em problemas de equilíbrio estático, o funcional é desempenhado pela energia potencial total cujo mínimo reconstrói a condição de equilíbrio.

A energia potencial total Π é um exemplo de funcional. Ela depende do deslocamento, \mathbf{u} , e suas derivadas, $\boldsymbol{\varepsilon}$. O termo funcional indica que a dependência da energia potencial total com o deslocamento não ocorre ponto a ponto mas sim numa região, por integração. Diz-se então que a energia potencial total depende da integral do deslocamento, e de suas derivadas, numa região considerada, (COOK, 2002). Na forma variacional, o problema é expresso através de integrais sobre todo o domínio.

Um funcional que depende de uma variável de campo do tipo $u_i(x_i)$, multivariável vetorial, tem a seguinte forma

$$\Pi = \int_{\Omega} F \left(u_i, \frac{\partial u_i}{\partial x_i}, \frac{\partial^2 u_i}{\partial x_i^2}, \dots \right) d\Omega. \quad (3.4)$$

Dentre as vantagens dessa forma estão:

- **Unificação:** a formulação unifica as propriedades do problema, sendo estas as equações de campo, condições de contorno naturais (derivadas do campo ou condições de Neumann) e leis de conservação;

- Invariância: pelo fato de o funcional considerar valores escalares em vez de quantidades vetoriais;
- Base para aproximação: na forma variacional, existe uma maior gama de funções teste que podem ser empregadas em comparação com a forma diferencial. Isso ocorre pois as funções teste no problema variacional não precisam satisfazer as condições de contorno natural visto que as mesmas encontram-se implícitas na formulação.

Utilizando o mesmo exemplo cujas formas forte e fraca estão em (A.12) e (3.2), respectivamente, a forma variacional é dada por

$$\text{Forma Variacional} \left\{ \begin{array}{l} \Pi = \int_0^L \frac{1}{2} EA \left(\frac{du}{dx} \right)^2 dx - \int_0^L u b dx - R u(L), \\ \delta \Pi = 0, \\ u(0) = 0, \\ \delta u(0) = 0. \end{array} \right. \quad (3.5)$$

Onde a letra grega δ representa “variação em” e δu é uma variação arbitrária em u sujeita a condição de contorno $\delta u(0) = 0$.

Aplicação em Problemas Discretos

O problema mais simples a ser analisado é o de uma massa suspensa por uma mola sob ação da gravidade.

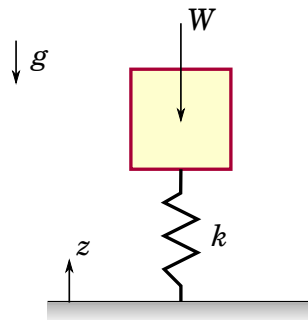


Figura 3.2 – Problema discreto. (GOVINDJEE, 2014)

A energia potencial total é definida pela soma do trabalho da força externa, Π_w , mais a energia potencial devido à deformação elástica, Π_e . O trabalho da força externa, nesse caso, corresponde à energia potencial gravitacional que é definida como o trabalho da força gravitacional para elevar a massa a uma posição a partir de uma referência arbitrária.

$$\Pi_t = \Pi_w + \Pi_e, \quad (3.6)$$

$$= Wz + \frac{1}{2} k(z_0 - z)^2. \quad (3.7)$$

onde, z é a variável que indica a posição do corpo, z_0 é a posição inicial medida a partir de $z = 0$, k é a constante de rigidez da mola.

Invocando o Princípio da Mínima Energia Potencial que diz que as condições de equilíbrio podem ser reconstruídas através da minimização da energia potencial total do sistema. Ou seja,

$$-\frac{\partial \Pi_t}{\partial z} = 0. \quad (3.8)$$

No exemplo da massa suspensa pela mola,

$$-\frac{\partial \Pi_t}{\partial z} = 0, \quad (3.9)$$

$$-W + k(z_0 - z) = 0. \quad (3.10)$$

O que representa a condição de equilíbrio do problema, a força de reação da mola corresponde ao peso da massa em suspensão.

Aplicação em Problemas Contínuos

Ao analisar uma barra engastada sujeita à um carregamento transversal a energia potencial total irá apresentar duas parcelas, como no caso discreto. A primeira parte refere-se a energia armazenada no material, devido a seu comportamento linear elástico, a segunda refere-se a energia proveniente do trabalho da força externa. A expressão da energia potencial total é dada por

$$\Pi_t = \int_0^L \frac{1}{2} EI \kappa^2 dx - P v \quad (3.11)$$

onde, L é o comprimento da barra, EI é a rigidez à flexão, κ a curvatura, P a força externa e v o deslocamento da barra no ponto de aplicação da força.

A curvatura pode ser aproximada pela segunda derivada da deflexão transversal, $\kappa = v''$, portanto, a energia potencial total é função da deflexão $\Pi(v(x))$. A minimização do potencial, nesse caso, trata-se da derivada do potencial com relação a outra função, logo, algo não trivial. Nesse ponto, o Método de Ritz surge como contorno à essa dificuldade, ver a Seção 3.2.3.

3.2 Discretização

O objetivo de se discretizar o domínio do problema consiste em reduzir o número de graus de liberdade do mesmo para um valor finito. Os modelos matemáticos criados para descrever fenômenos físicos são contínuos, portanto, apresentam um número infinito de graus de liberdade. Quando o problema não pode ser resolvido de forma fechada, ou

seja, satisfazer as equações em todos os graus de liberdade, a única alternativa é proceder para uma análise numérica aproximada por meio da discretização.

Os métodos de discretização comumente empregados são: método das diferenças finitas, método dos residuais ponderados, método de Rayleigh-Ritz.

3.2.1 Método das Diferenças Finitas

O método das diferenças finitas consiste em substituir o operador contínuo da derivada por uma diferença discreta. Para, por exemplo, a derivada na direção x da função f no ponto $x = a$,

$$\left. \frac{df}{dx} \right|_{x=a} \approx \frac{f(a + \Delta x) - f(a)}{\Delta x}. \quad (3.12)$$

Esse método apresenta-se como a solução natural para discretizar um problema em sua forma forte. Problemas onde o domínio é retangular ou circular podem ser discretizados pelo método, já problemas onde a geometria é complexo o método apresenta limitações. Uma outra característica é o fato de os resultados serem obtidos apenas nos nós da malha, sem uma óbvia extensão desses resultados para outros pontos, (FELIPPA, 2005).

3.2.2 Método dos Residuais Ponderados

O Método dos residuais ponderados tem como objetivo encontrar uma solução aproximada do problema a partir da discretização do mesmo, sendo esse processo caracterizado pela proposta de uma solução aproximada como combinação linear de funções conhecidas. É o método natural para discretizar problemas na forma fraca, ou melhor, a forma fraca surge da minimização dos residuais ponderados.

Existem diferentes subclasses do método: Galerkin, Petrov-Galerkin, colocação, subdomínio, volume finito, mínimos quadrados, (FELIPPA, 2005). Os métodos citados, com exceção do colocação, são conhecidos como *métodos da função teste*. Essa denominação reflete o fato de que as soluções aproximadas são definidas em todo o domínio ao invés de em apenas um ponto, como ocorre no método das diferenças finitas.

Para um dado problema definido pela forma forte

$$L[u] = r, \quad \text{em } \Omega, \quad (3.13)$$

onde L é um operador diferencial linear e u é a variável de campo de interesse que satisfaz certas condições de contorno. O procedimento básico consiste em assumir uma solução na forma

$$\bar{u} = \sum_{i=1}^n a_i f_i, \quad (3.14)$$

onde a_i são coeficientes constantes desconhecidos e f_i são funções teste definidas à priori. O método dos residuais ponderados considera o residual como sendo a diferença que ocorre na igualdade em (3.13) quando se substituí u por sua aproximação \bar{u}

$$R \stackrel{\text{def}}{=} r - L \left[\sum_{i=1}^n a_i f_i \right], \quad (3.15)$$

portanto, caso a solução assumida seja próxima da solução exata o valor de R será pequeno. Os vários métodos de residuais ponderados diferem-se quanto a técnica para calcular os coeficientes a_i . No entanto, todos os métodos produzem um conjunto de equações algébricas cujas incógnitas são os coeficientes.

A forma geral para se encontrar os coeficientes da solução aproximada, a_i , é dada por

$$\int_{\Omega} w_i F(R) d\Omega = 0, \quad i = 1, 2, \dots, n \quad (3.16)$$

onde, w_i são as *funções peso*, $F(R)$ é uma função qualquer do residual, e n o número de termos na solução aproximada. Em palavras, o resíduo, R , ponderado por um peso, w_i , tem média zero no domínio Ω , (COOK, 2002). Essa *integral de residuais ponderados* possui uma interpretação gráfica mostrada na Seção 3.2.2.1. Essa integral é o primeiro passo para se obter a equação (3.13) em sua *forma fraca*.

No método dos residuais ponderados, opera-se diretamente na equação diferencial do problema. Isso implica que na escolha das funções, estas devem satisfazer ambas condições de contorno, essencial e natural.

3.2.2.1 Método de Galerkin

Dentre os métodos de residuais ponderados utilizados, o método de Galerkin é o mais utilizado e o que produz resultados mais acurados, (ZOHDI, 2014).

Os parâmetros a_i são determinados pela solução de n equações

$$\int_{\Omega} f_i R d\Omega = 0, \quad i = 1, 2, \dots, n; \quad (3.17)$$

onde Ω é o domínio da solução e $f_i = w_i$, são as funções peso dadas por $w_i = \partial \bar{u} / \partial a_i$. A integral pode ser entendida como o produto interno das duas funções. Esse produto interno igual a zero indica que o resíduo R e a solução aproximada ² são ortogonais. Fazendo uma analogia com vetores, Figura 3.3, percebe-se que esse caso indica a melhor aproximação, a que minimiza o erro. No caso, como não se conhece a solução exata o erro não pode ser calculado, portanto, utiliza-se o residual uma vez que este tende a zero quando o erro também tende a zero. O fato de se utilizar as *mesmas funções da aproximação* da solução como *funções de ponderação* caracteriza o método de Galerkin.

² Os coeficientes a_i saem da integral pois são constantes.

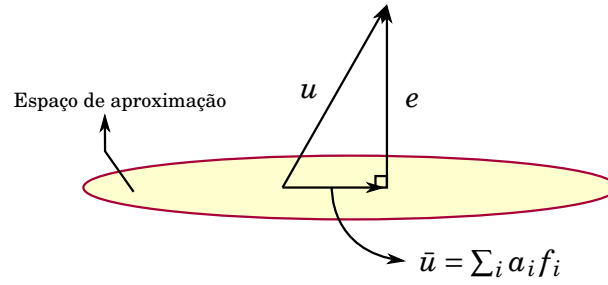


Figura 3.3 – Analogia vetorial do método de Galerkin. O vetor u indica a solução exata, o vetor \tilde{u} indica a solução aproximada e e o erro. A melhor solução, ou seja, menor erro, é obtida quando o erro e a solução aproximada são ortogonais. (ZOHD, 2014)

Fisicamente, no problema mecânico, o método de Galerkin é conhecido como Teorema dos Deslocamentos Virtuais (um corolário do Teorema dos Trabalhos Virtuais).

3.2.2.2 Mínimos Quadrados

O método dos mínimos quadrados considera a minimização dos residuais ao quadrado com respeito aos coeficientes a_i ,

$$\frac{\partial}{\partial a_i} \int_{\Omega} R^2 d\Omega = 0, \quad i = 1, 2, \dots, n. \quad (3.18)$$

Nesse caso, os pesos são dados por $w_i = \partial R / \partial a_i$.

3.2.3 Método de Rayleigh-Ritz

O método de Ritz, ou método de Rayleigh-Ritz, é uma subclasse do método de residual ponderado de Galerkin. Rayleigh, em 1870, estudando vibrações, considerava apenas um termo em sua função de aproximação, ou seja, apenas um grau de liberdade. Walter Ritz (1878-1909) generalizou para a superposição de um conjunto de funções, $\sum_i a_i f_i(x)$, todas as funções satisfaziam as condições cinemáticas e todas possuíam um grau de liberdade, a_i , associado. Diferentemente do método dos residuais ponderados, o método de Ritz é utilizando com a equação na forma variacional.

Como mencionado em 3.2.3, o método considera uma sequência *finita* de funções $f_1(x), f_2(x), \dots, f_n(x)$, *admissíveis*, ou seja, satisfazem condições de contorno essenciais e condições de compatibilidade. Essa idéia parte do objetivo de minimizar o funcional em relação à outra função. O espaço de funções possíveis é limitado apenas pelas condições de contorno, no entanto, há uma infinidade delas. Ritz propõem buscar uma solução aproximada em vez da solução exata. Com a solução aproximada na forma de uma combinação linear de funções conhecidas

$$\sum_{i=1}^n a_i f_i(x)$$

onde as funções $f(x)$ são conhecidas e chamadas de *funções de Ritz* e os coeficientes a_i são *coordenadas generalizados*, pois na forma clássica do método eles não possuem interpretação física. Os coeficientes também podem ser entendidos como *graus de liberdade* (g.l). No método dos elementos finitos as *coordenadas generalizadas* representam o valor do campo aproximado em cada nó.

Com a solução nessa forma aproximada o funcional passa a ser função dos parâmetros constantes, desconhecidos, da combinação linear

$$\Pi(a_1, a_2, \dots, a_n)$$

Nesse caso, o funcional pode ser diferenciado com relação aos coeficientes a_i . Cada derivação irá gerar uma equação, portanto, o sistema final terá n equações.

A questão agora é qual conjunto de funções escolher para aproximar a solução desejada. O método dos elementos finitos, que possui um conjunto característico de funções, pode ser considerado como um método de Ritz. As funções devem ser contínuas até o grau $r - 1$, onde r é o maior grau de derivada no funcional, devem satisfazer *apenas* as condições de contorno essenciais, diferentemente do método dos residuais ponderados, pois o funcional considera implicitamente as condições de contorno naturais; e devem ser linearmente independentes e completas³.

Além da dificuldade em se escolher funções de aproximação, não é sabido qual funções utilizar na aproximação para produzir soluções mais acuradas. Dentre funções comumente utilizadas estão polinômios e funções trigonométricas, ambas podem ser de difícil integração. Os coeficientes generalizados, a_i , não possuem interpretação física. Como as funções f_i precisam ser definidas em todo o domínio o método fica limitado à geometrias regulares.

3.3 Método dos Elementos Finitos

Como visto na Seção 3.2.3, o método dos elementos finitos possui os mesmos passos do método de Ritz. Ambos os métodos são utilizados para resolver, de forma aproximada, o problema. Os dois métodos tem em sua essência a discretização por meio da seleção de um número discreto, finito, de funções para a solução aproximada. Uma diferença entre os métodos é a de que no método dos elementos finitos as funções escolhidas *não são definidas em todo o domínio* e não precisam satisfazer, individualmente, as condições de contorno (RAO, 2013). O problema da geometria mencionado também na Seção 3.2.3 é contornado pelo método dos elementos finitos pois elementos de geometria simples podem ser compostos, de forma aproximada, em formatos complexos.

³ Um conjunto de funções é dito completo quando nenhum termo da série polinomial é omitido, outras condições podem ser encontradas em (COOK, 2002).

O método dos elementos finitos pode ser entendido, então, como uma versão aprimorada do método de Rayleigh e Ritz. Dentre as melhorias estão:

- Utilização de funções de aproximação simples, polinômios de baixa ordem, que são fáceis de serem integrados;
- Melhoria de aproximação por meio de adição de funções em regiões específicas, ou seja, a partir de uma interpretação física do problema pode-se identificar regiões onde as funções devam ser adicionadas;
- A interpretação dos coeficientes generalizados é feita de forma direta em termos dos graus de liberdade. Ou seja, o valor da aproximação no grau de liberdade é justamente o valor do coeficiente;
- Utilização de elementos de geometria simples para gerar geometrias complexas.

O método também pode ser utilizado como um tipo de residuais ponderados.

Matematicamente o método dos elementos finitos é entendido como uma ferramenta para encontrar solução aproximada em problemas de valor sobre o contorno através da *caracterização de uma solução como o mínimo de um funcional quadrático adequado*, (OLVER, 2008). No contexto mecânico, esse funcional é a energia potencial total. O método de Rayleigh-Ritz também propõem uma caracterização da solução, portanto, essa definição confunde os dois métodos.

3.3.1 Aproximação pelo Método dos Elementos Finitos

A aproximação é feita através da *escolha adequada* das funções de aproximação. A escolha das funções define regiões no domínio denominadas *elementos*. As funções empregadas no método correspondem a cada nó, ou seja, subdomínio definido.

Funções de Aproximação

As funções de aproximação⁴ utilizadas no método dos elementos finitos são uma das diferenças básicas entre o método e o clássico método de Ritz. Estas funções possuem um conjunto de propriedades particulares:

- O valor de cada função em seu respectivo nó vale 1 (unidade);
- O valor de cada funções nos outros nós é 0 (zero);

⁴ Ou função de forma, funções interpoladoras.

- A função entre nós pode ser linear (caso mais simples e prático), quadrática ou graus maiores (não empregadas normalmente);
- O conjunto das funções de aproximação, em sua coleção, satisfazem as condições de contorno de deslocamento;
- As funções apresentam continuidade no interior dos elemento, mas não nos nós. Ou seja, possuem derivada definida.

Solução Aproximada

A aproximação da solução para o campo $u_i(x_i)$ pode ser escrita como

$$\mathbf{u}(\mathbf{x}) = u_i(x_i) \approx \bar{u}_i = \sum_{a=1}^n \mu_{ia}^{(e)} N_a^{(e)}(x_i) = \begin{bmatrix} \sum_{a=1}^n \mu_{xa}^{(e)} N_a^{(e)}(x_i) \\ \sum_{a=1}^n \mu_{ya}^{(e)} N_a^{(e)}(x_i) \\ \sum_{a=1}^n \mu_{za}^{(e)} N_a^{(e)}(x_i) \end{bmatrix}, \quad (3.19)$$

onde, $N_a^{(e)}(x_i)$ são as funções de aproximação e $\mu_{ia}^{(e)}$ as coordenadas generalizadas⁵, ou coeficientes, para o elemento e ⁶. Cada elemento terá portanto um número de coeficientes igual ao produto entre o número de graus de liberdade por nó vezes o número de componentes de aproximação para cada elemento. Para implementação computacional essa aproximação é empregada em sua forma matricial

$$\bar{\mathbf{u}} = \bar{u}_i = \mathbf{N}\boldsymbol{\mu} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots \end{bmatrix} \begin{bmatrix} \mu_{x1} \\ \mu_{y1} \\ \mu_{z1} \\ \mu_{x2} \\ \mu_{y2} \\ \mu_{z2} \\ \vdots \end{bmatrix}, \quad (3.20)$$

onde, \mathbf{N} é a matriz com as funções de forma como função das variáveis espaciais de dimensões $(3 \times 3n)$, o vetor $\boldsymbol{\mu}$, de dimensões $(3n \times 1)$, é o vetor com os coeficientes para cada nó para cada coordenada espacial $i = 1, 2, 3$. O valor de n é definido através do grau de função de forma empregado. No caso de um elemento unidimensional com função de forma linear $n = 2$, caso as funções sejam quadráticas $n = 3$, Figura 3.4. Para um elemento plano com funções de forma bilinear $n = 4$, Figura 3.5. As funções são comumente dadas em função das coordenadas isoparamétricas, (ξ, η, ζ) , que vão de $[-1, 1]$.

⁵ O índice (e) que indica o elemento ao qual os coeficientes se referem será suprimido para evitar excesso de notação.

⁶ Os índices referentes ao número de fatores na aproximação, $\bar{u} = \mu_1 N_1 + \mu_2 N_2 + \dots$ foram colocadas na posição superior, sobrescritos, para não confundi-los com as coordenadas espaciais.

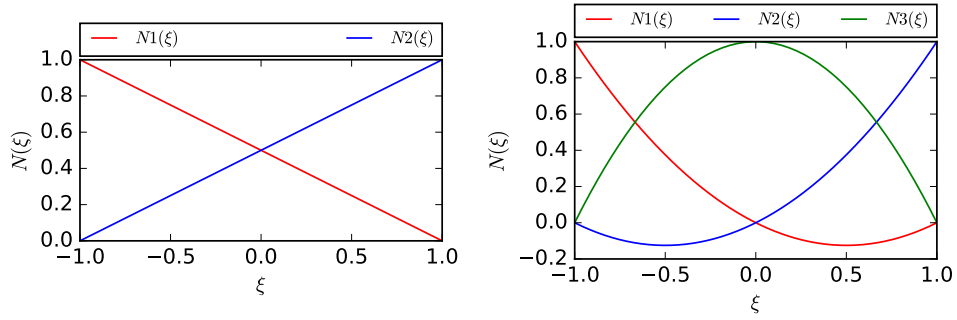


Figura 3.4 – Funções de forma linear e quadrática para elemento unidimensional.

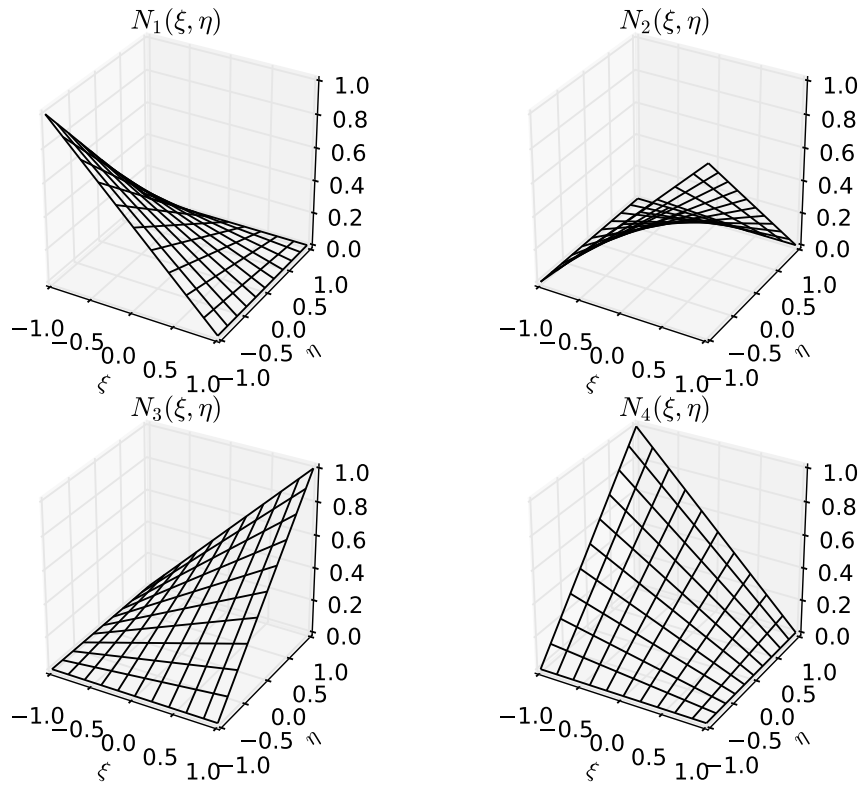


Figura 3.5 – Funções de forma bilinear para elemento plano de quatro lados.

Para problemas onde a variável de campo primária de interesse é escalar, caso do problema do calor, a aproximação tem apenas uma componente escalar, portanto, pode ser dada por

$$u(\mathbf{x}) = u(x_i) \approx \bar{u} = \sum_{a=1}^n \mu_a^{(e)} N_a^{(e)}(x_i), \quad (3.21)$$

e em forma matricial,

$$\bar{u} = \mathbf{N}\boldsymbol{\mu} = \begin{bmatrix} N_1 & N_2 & N_3 & \dots \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \vdots \end{bmatrix} \quad (3.22)$$

A solução aproximada então pode ser utilizada no problema em sua forma fraca ou

variacional, ou seja, Método de Galerkin ou Método Rayleigh-Ritz.

3.3.2 Formulação por Métodos Variacionais

Para um problema geral, que pode ter natureza física ou puramente matemática, em sua forma variacional a solução requer a minimização do funcional Π no domínio Ω cujo contorno superficial é dado por Γ . A forma geral do funcional é dada por

$$\Pi(u_i) = \int_{\Omega} F\left(u_i, \frac{\partial u_i}{\partial x_i}, \frac{\partial^2 u_i}{\partial x_i^2}, \dots\right) d\Omega + \int_{\Gamma} q\left(u_i, \frac{\partial u_i}{\partial x_i}, \frac{\partial^2 u_i}{\partial x_i^2}, \dots\right) d\Gamma. \quad (3.23)$$

A variável de campo primária, $u_i(x_i)$, é uma função multivariável vetorial. Utilizando a aproximação proposta na Seção 3.3.1, equação (3.19), o funcional passa a ser função das coordenadas generalizadas contidas no vetor $\boldsymbol{\mu}$, ou seja, $\Pi(\boldsymbol{\mu})$. A minimização do funcional é feita através

$$\frac{\partial \Pi}{\partial \boldsymbol{\mu}} = \begin{bmatrix} \frac{\partial \Pi}{\partial \boldsymbol{\mu}_1} \\ \frac{\partial \Pi}{\partial \boldsymbol{\mu}_2} \\ \vdots \\ \frac{\partial \Pi}{\partial \boldsymbol{\mu}_{nn}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \Pi}{\partial \mu_{x1}} \\ \frac{\partial \Pi}{\partial \mu_{y1}} \\ \frac{\partial \Pi}{\partial \mu_{z1}} \\ \frac{\partial \Pi}{\partial \mu_2} \\ \vdots \\ \frac{\partial \Pi}{\partial \mu_{nn}} \end{bmatrix} = \mathbf{0}, \quad (3.24)$$

onde, nn é o número total de nós, o vetor $\boldsymbol{\mu}_j = [\mu_{xj}, \mu_{yj}, \mu_{zj}]$ contém os coeficientes para o nó j referentes aos graus de liberdade. Caso o funcional Π seja quadrático, o que ocorre para o problema da elasticidade linear, cada derivada irá gerar uma equação algébrica onde os coeficientes, ou coordenadas generalizadas, serão as incógnitas.

Como as funções são definidas para cada elemento e de acordo com as propriedades apresentadas na Seção 3.3.1, propriedade na qual a função de forma tem o valor máximo no nó do elemento e valor nulo nos outros, o funcional pode ser calculado individualmente para cada elemento e em seguida somado

$$\Pi = \sum_{e=1}^E \Pi^{(e)} \quad (3.25)$$

onde E é o número total de elementos e $^{(e)}$ é o índice que indica o elemento. A minimização local, em cada elemento, é dada por

$$\frac{\partial \Pi}{\partial \boldsymbol{\mu}} = \sum_{e=1}^E \frac{\partial \Pi^{(e)}}{\partial \boldsymbol{\mu}_j} = 0, \quad j = 1, 2, \dots, nn. \quad (3.26)$$

a derivada do funcional com relação às coordenadas generalizadas gera um conjunto de equações algébricas para cada elemento, essas equações podem ser organizadas em forma matricial,

$$\frac{\partial \Pi^{(e)}}{\partial \boldsymbol{\mu}^{(e)}} = \mathbf{K}^{(e)} \boldsymbol{\mu}^{(e)} - \mathbf{p}^{(e)} = 0 \quad (3.27)$$

nesse caso, j deve percorrer todas as funções de forma em cada elemento. Caso o elemento seja unidimensional e as funções lineares, $j = 1, 2$. Caso trate-se de um elemento plano com funções bilineares, $j = 1, 2, 3, 4$.

3.3.3 Formulação por Métodos de Residuais Ponderados

Uma equação de balanço, ou equilíbrio, geral pode ser expressa por

$$H(u_i) = G(u_i), \quad \text{em } \Omega, \quad (3.28)$$

onde, H e G são funções do campo u_i . Utilizando a solução aproximada, da mesma forma proposta na Seção 3.2.2, que satisfazem as condições de contorno essenciais e naturais,

$$u_i(x) = \bar{u}_i(x) = \sum_{a=1}^n \mu_{ia} N_a(x_i). \quad (3.29)$$

A função residual pode ser definida a partir da substituição da solução aproximada na Equação (3.28)

$$R = G(\bar{u}_i) - H(\bar{u}_i). \quad (3.30)$$

Com o objetivo de fazer o residual mínimo, o que corresponde a ter um erro mínimo, utiliza-se um dos métodos descritos na Seção 3.2.2. De forma geral, minimiza-se uma ponderação de um função do potencial, $w_i F(R)$, onde w_i são as funções peso. Dessa forma, a minimização é obtida através de

$$\int_{\Omega} w_i F(R) dV = 0. \quad (3.31)$$

A função $F(R)$ é escolhida de forma a $F(R=0) = 0$, ou seja, quando a solução aproximada é igual a solução exata (RAO, 2013). Utilizando o método de Galerkin, onde as funções peso são as mesmas funções utilizadas na aproximação, $w = N(x)$, e a função do residual é o próprio residual $f(R) = R$. Portanto,

$$\int_{\Omega} N^a R dV = 0, \quad a = 1, 2, \dots, n. \quad (3.32)$$

Como a equação de campo, proposta em (3.28), mantém-se válida em todo do domínio Ω , pode-se considerar o subdomínio contido em Ω referente à um elemento individual $\Omega^{(e)}$. Nesse caso, uma aproximação da solução é proposta para cada elemento,

$$u_i^{(e)}(x_i) = \bar{u}_i^{(e)}(x_i) = \sum_{a=1}^n \mu_{ia}^{(e)} N_a^{(e)}(x_i). \quad (3.33)$$

Em um elemento, a minimização do residual fica

$$\int_{\Omega^{(e)}} N_a^{(e)} \left(G \left(\bar{u}_i^{(e)} \right) - F \left(\bar{u}_i^{(e)} \right) \right) d\Omega^{(e)} = 0, \quad a = 1, 2, 3, \dots, n. \quad (3.34)$$

Substituindo o valor da função aproximada em (3.34), gera-se um conjunto de integrais onde as variáveis estão apenas nas funções N^a , portanto, devido ao fato de serem polinômios, a integração é simplificada. Cada elemento portanto terá um conjunto de equações resultantes dessa integral que pode ser arranjada em forma matricial,

$$\mathbf{K}^{(e)} \boldsymbol{\mu}^{(e)} - \mathbf{p}^{(e)} = 0 \quad (3.35)$$

Uma observação a ser feita é a de que no método dos residuais ponderados, as funções de aproximação devem satisfazer ambas as condições de contorno, essencial (de deslocamento) e natural (tensão de superfície). Portanto, torna o método dos residuais ponderados de Galerkin não prático. O método é utilizando normalmente na equação já em sua forma fraca. Portanto, multiplica-se a equação na forma forte (residual) por um peso geral e integra-se no domínio. Após operações matemáticas, a equação é transformada em sua forma fraca, ou seja, os requerimentos para a solução são enfraquecidos, ou reduzidos, e a condição de contorno natural é considerada diretamente na equação. Nesse ponto utiliza-se o método de Galerkin, ou seja, assume-se que as funções peso são iguais as funções consideradas na solução aproximada. Logo, as mesmas funções de aproximação utilizadas no método de Ritz podem ser utilizadas, (BATHE, 1996).

3.4 Diagrama de Tonti

Criado pelo físico matemático Enzo Tonti, (TONTI, 2013), o Diagrama de Classificação é uma representação gráfica das relações entre as variáveis com seus respectivos elementos espaciais. A classificação surge a partir da busca por algo que une as variáveis físicas. O ingrediente encontrado responsável por unir as variáveis é o *espaço*. A Figura 3.6, mostra um esquema básico do diagrama. As variáveis de configuração, definida posteriormente na seção 3.4.1, são dispostas no lado esquerdo do diagrama enquanto as variáveis de fonte do lado direito. Essa disposição decorre do fato de que as variáveis de configuração de todas as teorias físicas são dotadas de *orientação interna* enquanto as variáveis de fonte possuem *orientação externa*, (TONTI, 2013).

Na Figura 3.6, as variáveis de campo $A, \mathbf{b}, d, \mathbf{c}$ representam a densidade espacial das variáveis globais correspondentes. A variável A , nesse exemplo, é associada a um ponto e portanto sua densidade é ela própria. A variável \mathbf{b} está associada a um elemento de linha, é um vetor cuja integral de linha retorna a variável global, B . A variável \mathbf{c} está associada a uma superfície e é um vetor cuja integral de área retorna a variável global

associada, C . A variável d está associada a um volume e é uma escalar cuja integral de volume retorna a variável global D .

A primeira linha do diagrama contém as variáveis associadas aos elementos espaciais de 1 dimensão interna e externa.

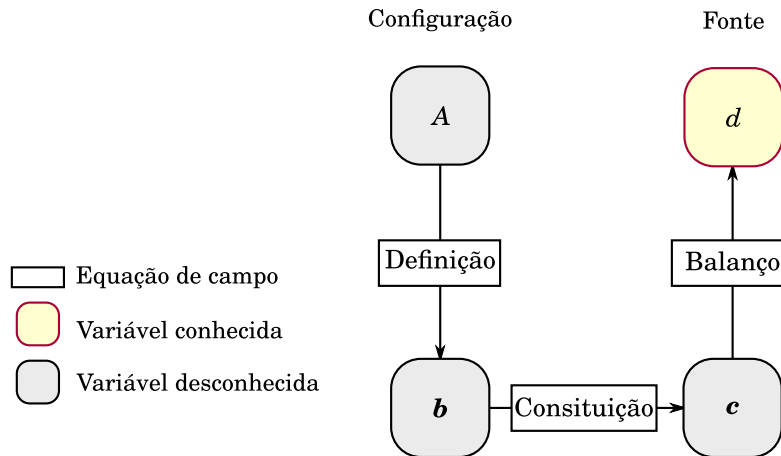


Figura 3.6 – Esquema do diagrama de Tonti. (TONTI, 2013)

Em Felippa (2005), o diagrama proposto é baseado no apresentado por Tonti, no entanto, não possui o rigor de critérios para sua construção. Nele são incluídas as condições de contorno como elementos extra no digrama e o rigor de elemento espacial associado é dispensado. As variáveis de campo são classificadas em primárias, intermediárias ou de fluxo. As variáveis primárias são aquelas que definem a resposta e caracterizam o fenômeno físico estudado. As variáveis intermediárias nem sempre apresentam interpretação física, por exemplo, para a equação do calor a variável intermediária é o gradiente da temperatura. O diagrama pode ser empregado de forma geral para problemas de valor sobre o contorno e sua estrutura básica está representada na Figura 3.7.

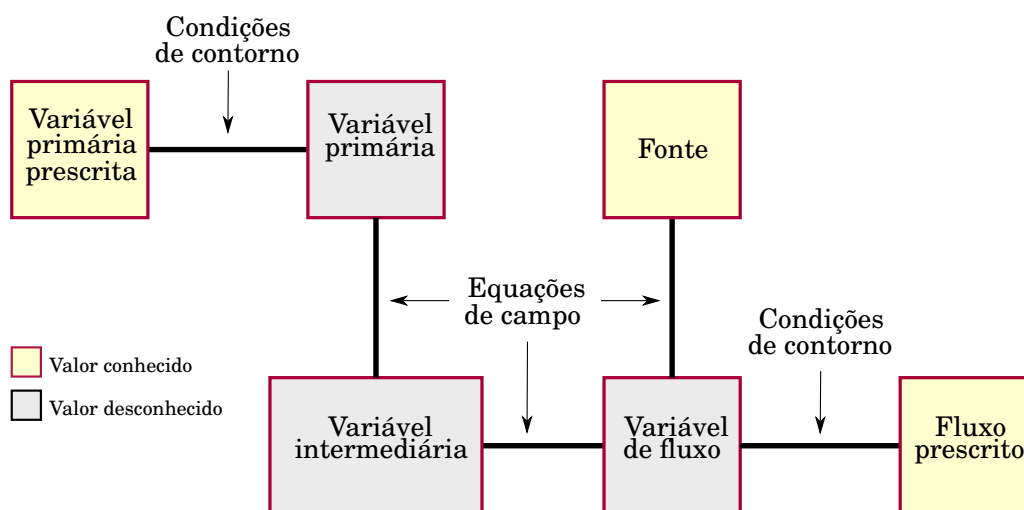


Figura 3.7 – Esquema do diagrama de Tonti para representação de um problema de valor sobre o contorno. (FELIPPA, 2005)

3.4.1 Definições para Classificação dos Modelos

Variáveis Globais e Variáveis de Campo

A motivação para distinção entre variáveis globais e de campo surge da descrição matemática da física, (TONTI, 2014). A *formulação diferencial* da física utiliza a densidade ou taxa das variáveis globais. Densidade refere-se a razão entre a variável por um elemento espacial, ou seja, uma linha, uma superfície ou um volume.

Uma *variável global* no espaço é uma variável que não é uma densidade. A formação de densidades de variáveis globais esconde o conteúdo geométrico e físico da variável. Por exemplo, a densidade de massa é obtida pela razão entre massa e o volume, a pressão é a razão entre força e área. Essa ligação entre variáveis globais e o elemento espacial e temporal nos permite obter uma formulação discreta por meio de argumentos físicos. Essa formulação pode ser utilizado na física computacional.

Variáveis de Fonte, Configuração e Energia

No começo da formulação de uma teoria física, observa-se a distinção entre a causa do fenômeno e os efeitos no sistema. A causa pode ser originada pela fonte do campo e o efeito pela configuração ou potencial do campo. Pode-se então dividir variáveis físicas em três grupos relacionados com o seu papel na teoria física: *variáveis de fonte, configuração e energia*.

Variáveis de Fonte são a fonte de campos. Por exemplo, forças são a causa de deslocamentos; calor é a causa do campo de temperatura.

Variáveis de Configuração são responsáveis por definir a configuração do sistema. Por exemplo, os deslocamentos descrevem a configuração de um sólido; a temperatura descreve a configuração do campo de temperatura.

Variáveis de Energia são variáveis obtidas pelo produto de variáveis de fonte e variáveis de configuração.

Problema Fundamental de Teorias Físicas

O problema fundamental de teorias física consiste em encontrar a configuração de um sistema ou campo dado a distribuição temporal e espacial de fontes.

Elementos Espaciais e suas Orientações

Elementos espaciais são:

- Ponto **P**;
- Linhas **L**;
- Superfícies **S**;
- Volumes **V**.

Cada elemento pode ter orientação *interna* ou *externa*. Orientação interna está relacionada com a ordem de pontos internos na superfície. Um triângulo, por exemplo, possui orientação interna quando os seus vértices são ordenados em sequência. Enquanto orientação externa leva em conta os dois lados da superfície. Por exemplo, quando uma quantidade física atravessa uma superfície para fora.

Associação de Variáveis Globais com Elementos Espaciais

As variáveis globais estão intrinsecamente associadas a um elemento espacial. Por exemplo, o conteúdo de massa está associado com o volume com orientação externa. Uma regra empírica sugere a adoção de espaços dotados de orientação interna a variáveis de configuração e espaços dotados de orientação externa a variáveis de fonte, (TONTI, 2013).

4 Elasticidade Linear: Sólidos

Nesse capítulo será discutido o modelo matemático para o problema da mecânica dos sólidos, o procedimento para obtenção da solução aproximada, o procedimento para implementação computacional e resultados obtidos dos programas desenvolvidos.

4.1 Introdução

A mecânica dos sólidos é o ramo da física interessado em estudar o efeito de forças e energia em corpos sólidos. Também conhecida como resistência dos materiais, a mecânica dos sólidos pode ser entendida como a teoria da elasticidade com um tratamento de menor rigor matemático. As bases da teoria foram desenvolvidas durante a primeira metade do século XIX por matemáticos como Leonhard Euler (1707-1783), Augustin Luis Cauchy (1789-1857), George Green (1793-1841), Siméon Denis Poisson (1781-1840), (CHANDRASEKHARAIH; DEBNATH, 1994, p. 155).

O problema de interesse consiste em entender a relação entre os esforços internos e ações externas, qual o efeito dos esforços internos em deformações ou qual o efeito de uma deformação nos esforços internos, qual a relação entre os deslocamentos e as deformações internas. Todos esses *fenômenos físicos* podem ser representados através de *modelos matemáticos*.

O *problema fundamental*, (TONTI, 2013, p. 325), pode ser posto como:

- Dado um corpo sólido;
- Dado a forma e a natureza do material que forma esse corpo;
- Dado as restrições agindo no corpo;
- Dado um intervalo de tempo;
- Dado a posição inicial e a velocidade inicial de todos os pontos do corpo;
- Dado as forças de volume e as forças de superfície no contorno;

- **Encontrar** a posição de todos os pontos do corpo em todos os instantes subsequentes.

Os modelos matemático descrevem *leis fundamentais de conservação* como conservação de massa, momento linear e energia, (CHANDRASEKHARAI AH; DEBNATH, 1994, p. 325). Essas equações são válidas em todos os pontos de um meio contínuo e não fazem distinção quanto a estrutura física da matéria, ou seja, são gerais. No entanto, não são suficientes para determinar a variável desconhecida. Portanto, uma outra relação fundamental é necessária: essa denominada *relação constitutiva*. Essa relação descreve propriedades intrínsecas do material, com isso distingue-se um material do outro.

4.1.1 Teoria da Elasticidade

A teoria da elasticidade é um ramo da mecânica dos meios contínuos interessada em sólidos de comportamento elásticos, (CHANDRASEKHARAI AH; DEBNATH, 1994). Sólidos elásticos são aqueles capazes de retornar à configuração inicial após remoção das forças que causaram a deformação. Particularmente, um sólido elástico é dito de comportamento *linear* quando a *lei material*, que estabelece a relação entre esforço e deformação, é linear. Diversos materiais práticos possuem esse comportamento quando sujeitos à *pequenas deformações*.

A teoria da elasticidade estende os conceitos da mecânica dos sólidos dando a ela um maior formalismo matemático. O formalismo provém de ferramentas matemáticas como tensores e equações diferenciais parciais. As hipóteses utilizadas são matemáticas em vez de físicas. Uma hipótese típica é a de que seções planas permanecem planas após flexão. Na teoria da elasticidade essa hipótese não é considerada diretamente e uma solução é proposta com base em princípios básicos como a mecânica Newtoniana (equilíbrio), geometria Euclidiana (cinemática) e lei de Hooke (constituição) (BARBER, 2002).

A teoria consiste num modelo matemático composto por equações de campo que descrevem o comportamento de corpos físicos. As equações de campo são equações diferenciais parciais cujas variáveis primárias utilizadas para definir a resposta do corpo são o campo de tensões e campo de deslocamentos. O modelo contém três equações acopladas que estabelecem as relações entre as variáveis. Todos os problemas que envolvem corpos deformáveis podem ser descritos através dessas equações. A diferença entre os problemas é estabelecida por meio de condições de contorno que são particulares a cada problema. A teoria da elasticidade, finalmente, pode ser entendida como um problema de valor de contorno (SLAUGHTER; VERLAG, 2003).

4.1.2 Equações de Campo

As três equações de campo que modelam o comportamento mecânico dos sólidos são: equilíbrio interno, compatibilidade (ou cinemática) e relação constitutiva (ou relação tensão-deformação).

4.2 Relação de Equilíbrio

O equilíbrio é um caso particular de uma lei fundamental da física, *conservação de momento linear*.

O caso estático da conservação de momento linear resulta no balanço de forças que agem num determinado ponto material, ou volume infinitesimal, dentro de um corpo. Qualquer volume material pode ser visto com um corpo livre. Uma força pode ser caracterizada em *força de corpo* ou *força de superfície*, ambas serão discutidas posteriormente. As forças também podem ser caracterizadas em *interna* ou *externa*. Forças internas agem dentro de um mesmo *corpo livre* definido a priori em um volume material aleatório e forças externas são aquelas provenientes de um volume material fora do corpo livre, (SLAUGHTER; VERLAG, 2003, p. 157).

Um corpo sólido pode ser pensado como uma estrutura de alta hiperestaticidade, ou seja, cada ponto material está intimamente conectado com o seu vizinho. Uma estrutura hiperestática é caracterizada por apresentar redundância em suportes e com isso um menor número de graus de liberdade, devido à isso, existe um número maior de incógnitas, forças internas, do que equações. A diferença é compensada pela introdução de equações de compatibilidade, cinemática, introduzida na seção subsequente, (BARBER, 2002).

4.2.1 Tensão

Forças podem agir em um corpo de duas maneiras, a primeira distribuída no volume e a segunda distribuída na superfície. Na superfície a força é a integral de área da densidade de força, denominada *vetor tensão*, \mathbf{t} . Essa densidade é um vetor definido através do limite no qual um elemento de área tende a zero.

$$\mathbf{t} \stackrel{\text{def}}{=} \lim_{A \rightarrow 0} \frac{\mathbf{T}}{A} \quad (4.1)$$

onde \mathbf{T} é o vetor força de superfície resultante em uma área A .

As tensões surgem como uma resposta do corpo às solicitações de carregamento. As tensões são o meio pelo qual as forças externas são transmitidas de ponto a ponto no interior do corpo, (SLAUGHTER; VERLAG, 2003). O estado de tensões de um ponto

material pode ser descrito por meio de nove componentes de tensão arranjadas em um tensor σ_{ij} onde os índices indicam componente na face i na direção j . A face é definida pelo vetor normal à ela, portanto a face i apresenta vetor normal nessa direção. A Figura 4.1 mostra a representação das tensões num elemento volumétrico infinitesimal.

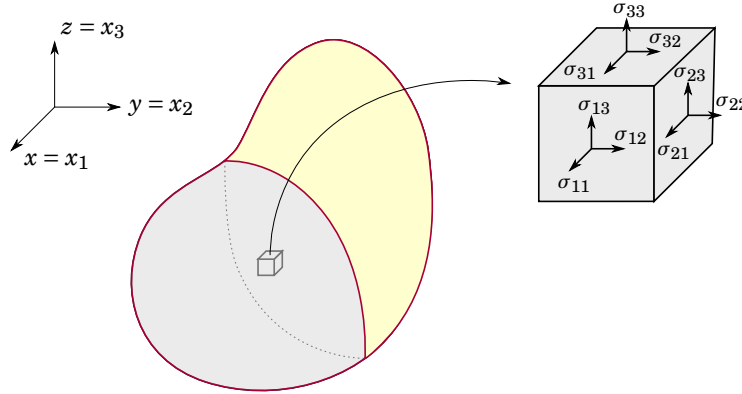


Figura 4.1 – Estado de tensões descrito por 9 componentes do tensor tensão de Cauchy em um volume infinitesimal contido no interior do corpo.

A forma completa das tensões em um ponto material é conhecida como o *tensor de tensão de Cauchy* e é expressa por

$$\underline{\sigma} \equiv \sigma_{ij} \equiv \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \equiv \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}. \quad (4.2)$$

Através do teorema de Cauchy, $\mathbf{t} = \underline{\sigma}\mathbf{n}$, o vetor tensão num ponto pode ser determinado para qualquer plano definido pelo vetor normal \mathbf{n} por meio do tensor tensão de Cauchy. A Figura 4.5 mostra a distribuição de vetores tensão, \mathbf{t} , que surgem do seccionamento do volume material. O seccionamento do volume é feito por meio de um plano, a distribuição de vetor tensão é característica específica desse plano pela hipótese de Cauchy, i.e., $\mathbf{t} \stackrel{\text{def}}{=} \mathbf{t}(\mathbf{x}, \mathbf{n})$, (GURTIN, 1981, p. 97).

O equilíbrio de momento em um dos pontos do elemento infinitesimal gera as seguintes relações

$$\sigma_{xy} = \sigma_{yx}, \quad \sigma_{yz} = \sigma_{zy}, \quad \sigma_{xz} = \sigma_{zx}. \quad (4.3)$$

Com isso, o número de componentes independentes no tensor de tensões passa de nove para seis, podendo ser convenientemente representado por um vetor

$$\boldsymbol{\sigma} = \left[\sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{12} \quad \sigma_{13} \quad \sigma_{23} \right]^T. \quad (4.4)$$

Essa notação é útil para implementação computacional.

4.2.2 Equilíbrio Interno

Num elemento infinitesimal, o equilíbrio entre as forças agindo nas áreas gera, no limite, o equilíbrio do ponto material. Esse equilíbrio é expresso através do divergente da tensão. O divergente é uma ferramenta matemática que expressa o saldo líquido de um campo em um ponto, ver apêndice B.7. A Figura 4.2, mostra o elemento infinitesimal juntamente com suas dimensões e força de corpo.

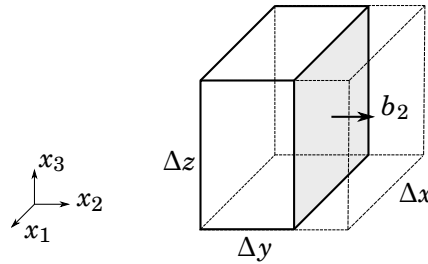


Figura 4.2 – Elemento infinitesimal com força de corpo b_2 por unidade de volume na direção x_2 .

Na direção x_2 , por exemplo, o equilíbrio é composto pela tensão normal e por componentes da tensão cisalhante,

$$\left(\frac{\partial\sigma_{22}}{\partial x_2}\Delta y\right)\Delta x\Delta z + \left(\frac{\partial\sigma_{12}}{\partial x_1}\Delta x\right)\Delta y\Delta z + \left(\frac{\partial\sigma_{32}}{\partial x_3}\Delta z\right)\Delta x\Delta y + b_2\Delta x\Delta y\Delta z = 0. \quad (4.5)$$

As parcelas com derivada representam o primeira termo da série de Taylor, ou seja, o campo de tensões varia dentro do elemento infinitesimal uma quantidade proporcional à taxa de variação do campo nessa direção vezes o tamanho do intervalo. A Figura 4.3 mostra as componentes da tensão que compõem o equilíbrio. Em notação indicial esse balanço de forças fica

$$\sigma_{12,1} + \sigma_{22,2} + \sigma_{32,3} + b_2 = 0, \quad \text{ou} \quad \sigma_{j2,j} + b_2 = 0. \quad (4.6)$$

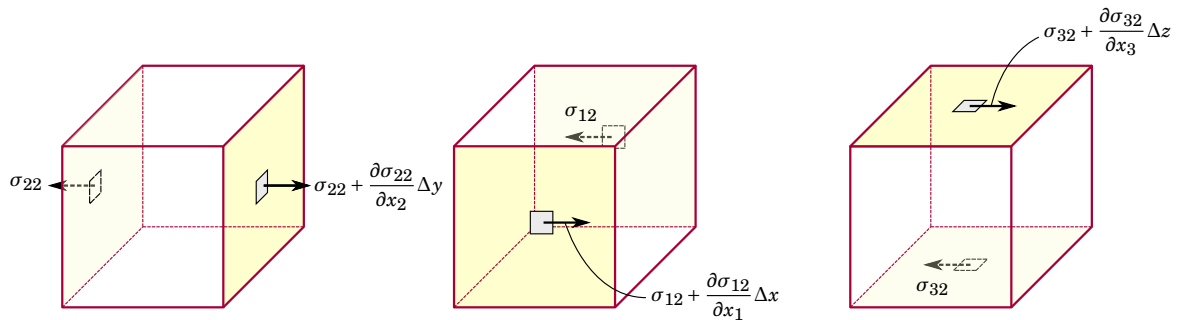


Figura 4.3 – Equilíbrio interno local com tensões na direção x_2 em cada face.

De forma análoga para o balanço de forças nas direções $x_1 = x$ e $x_3 = z$, chega-se ao seguinte conjunto de equações

$$\sigma_{j1,j} + b_1 = 0, \quad (4.7)$$

$$\sigma_{j3,j} + b_3 = 0. \quad (4.8)$$

As três equações podem ser, de forma resumida, expressas por

$$\nabla \cdot \underline{\sigma} + \underline{b} \stackrel{\text{law}}{=} 0, \quad \sigma_{ji,j} + b_i \stackrel{\text{law}}{=} 0. \quad (4.9)$$

Essas equações também são obtidas mais formalmente através do balanço de momentos lineares. A operação $\nabla \cdot \underline{\sigma}$ está explicitada no Apêndice B.4.4.

4.2.3 Forças de Corpo

Nas equações apresentadas na seção anterior, 4.2.2, b_i representa as forças de corpo que agem distribuídas na massa do corpo por unidade de volume. A força gravitacional é um exemplo de força de corpo.

$$b_i = \rho g_i, \quad (4.10)$$

onde $g_i = [0, 0, -g]$.

4.2.4 Forças de Superfície

Forças de superfície são aquelas que agem nos pontos materiais localizados no contorno do corpo livre. O vetor tensão, $\bar{\mathbf{t}}$, é especificado nos pontos da superfície do corpo. Esse tipo de força entra como condição de contorno na equação diferencial como será visto adiante.

4.3 Relação Cinemática

Cinemática é o termo dado a descrição matemática das deformações e movimento de um volume material, Marsden e Hughes (1994), ou o estudo da geometria do movimento, Gonzalez e Stuart (2008, p. 112), Chandrasekharaiah e Debnath (1994, p. 167).

Um corpo é composto por matéria discreta em forma de átomos, moléculas, cadeias poliméricas dentre outras formas de aglomeração de matéria. Uma premissa central da mecânica dos meios contínuos é a hipótese de que essa natureza discreta da matéria pode ser desconsiderada no estudo das deformações. O corpo é modelado como contínuo, (SLAUGHTER; VERLAG, 2003).

As relações cinemáticas, ou relações de compatibilidade, estabelecem a conexão entre as deformações específicas do corpo, ε_{ij} , e os deslocamentos do ponto material, u_i . Para utilizar essas relações a fim de resolver o problema mecânico é necessário uma ligação entre essas variáveis e as variáveis de equilíbrio. Essa ligação é feita a partir da constituição do material, discutido na seção subsequente.

A Figura 4.4 mostra para um elemento infinitesimal as componentes de um elemento plano sujeito ao deslocamento $\mathbf{u} = [u_x, u_y]$. O elemento possui inicialmente comprimento Δx e Δy , diz-se que esta é a configuração *não deformada*. A configuração dita *deformada* considera que o elemento sofreu deformações.

O efeito do campo de deslocamentos u_i nas deformações é admitido em sua aproximação linear, i.e., os termos de segunda ordem são desprezados. Essa hipótese possui validade para pequenos deslocamentos, pois $\partial_{x_i}^2 u_i \ll 1$. A hipótese de pequenos deslocamentos também permite ao modelo a análise da estrutura em sua configuração não deformada, (MOSALAM, 2011).

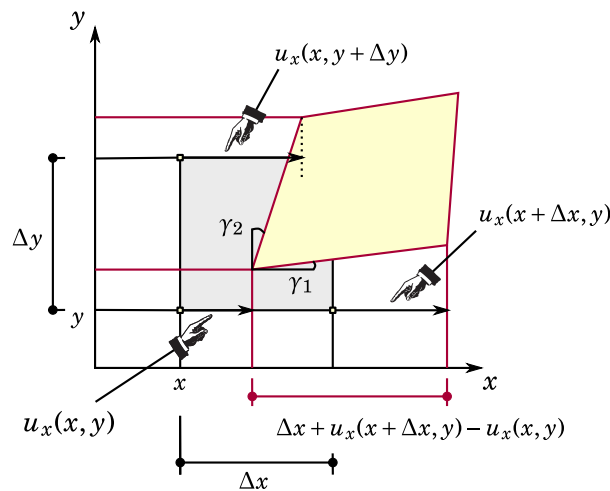


Figura 4.4 – Deformações específicas em um elemento infinitesimal. (SLAUGHTER; VERLAG, 2003)

4.3.1 Deformação Específica

Existem diversas medidas de deformação, por exemplo, tensor deformação de Cauchy-Green, Green-St. Venant e o tensor de deformação linear. Essas medidas são baseadas no mapeamento de vetores quando o corpo deforma. Quando o problema apresenta pequenas deformações, o tensor de deformação linear apresenta bons resultados que são verificados por meio de experimentos. Nesse trabalho, o tensor de deformação linear será descrito para formulação matemática do problema da elasticidade. A deformação específica, ou simplesmente deformação, por definição, é a razão entre a deformação e o comprimento inicial. Essa medida é útil pois pode-se representa um medida de deformação que independe do comprimento inicial do corpo. Utilizando a Figura 4.4, na direção x tem-se

$$\begin{aligned} \varepsilon_{11} &\stackrel{\text{def}}{=} \lim_{\Delta x \rightarrow 0} \frac{u_x(x + \Delta x, y) - u_x(x, y)}{\Delta x}, \\ &\stackrel{\text{def}}{=} \frac{\partial u_x}{\partial x} \equiv \frac{\partial u_1}{\partial x_1}. \end{aligned} \quad (4.11)$$

De forma análoga para as outras direções e planos,

$$\varepsilon_{22} = \frac{\partial u_2}{\partial x_2}, \quad \varepsilon_{33} = \frac{\partial u_3}{\partial x_3}. \quad (4.12)$$

Essas três quantidades indicam o quanto de deformação ocorreu em uma direção quando se percorre Δx nessa mesma direção. Ou seja, para a direção x , o quanto o corpo deformou na direção x ao longo dessa direção.

As componentes do tensor deformação específica ε_{ij} cujos índices $i = j$ representam um alongamento das fibras nessa direção, já as componentes com índices $i \neq j$ representam a deformação cisalhante, que indica o quanto o corpo distorce.

Pela Figura 4.4, a deformação específica cisalhante pode ser entendida como $\gamma_{12} = \gamma_1 + \gamma_2$. Da maneira semelhante a deformação de alongamento, a deformação cisalhante é obtida analisando o quanto o corpo deforma numa direção quando se percorre um intervalo numa direção ortogonal. Por exemplo, percorrendo Δy , o corpo distorce na direção x uma quantidade igual a $u_x(x, y + \Delta y) - u_x(x, y)$. Essa distorção é dividida pela distância Δy para fornecer a taxa de distorção por quantidade de comprimento. Somando esse valor com o análogo para a outra direção, tem-se a deformação cisalhante,

$$\begin{aligned} \gamma_{xy} &\stackrel{\text{def}}{=} \lim_{\Delta x, \Delta y \rightarrow 0} \frac{u_x(x, y + \Delta y) - u_x(x, y)}{\Delta y} + \frac{u_y(x + \Delta x, y) - u_y(x, y)}{\Delta x} \\ &\stackrel{\text{def}}{=} \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \equiv \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \end{aligned} \quad (4.13)$$

Para *pequenas deformações* pode-se dizer que $\partial_{x_1} u_1 \ll 1$ e $\partial_{x_2} u_1 \ll 1$, logo, as tangentes dos ângulos γ_1 e γ_2 se confundem com os próprios ângulos $\tan(\gamma) \approx \gamma$, portanto a deformação cisalhante pode ser vista como a soma desses ângulos.

Essa notação, $\gamma_{xy} \equiv \gamma_{12}$ é conhecida como *deformação cisalhante de engenharia* e ela se relaciona com a deformação cisalhante com $\varepsilon_{ij} = \frac{1}{2}\gamma_{ij}$ para $i \neq j$. O fator $\frac{1}{2}$ surge da definição de *tensor deformação infinitesimal* que por sua vez surge quando os termos de alta ordem são desprezados do tensor de deformação de Green-St. Venant. De forma análoga para as outras direções

$$\gamma_{13} = u_{1,3} + u_{3,1}, \quad \text{e} \quad \gamma_{23} = u_{2,3} + u_{3,2}. \quad (4.14)$$

O tensor deformação linear com todas suas componentes é dado por,

$$\underline{\varepsilon} = \varepsilon_{ij} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} & \frac{1}{2}\gamma_{12} & \frac{1}{2}\gamma_{13} \\ \frac{1}{2}\gamma_{21} & \varepsilon_{22} & \frac{1}{2}\gamma_{23} \\ \frac{1}{2}\gamma_{31} & \frac{1}{2}\gamma_{32} & \varepsilon_{33} \end{bmatrix}. \quad (4.15)$$

4.3.2 Relação Deformação-Deslocamento

A relação entre as deformações, ε_{ij} , e deslocamentos, u_i , pode ser expressa por meio da definição do tensor de deformação linear,

$$\underline{\varepsilon} \stackrel{\text{def}}{=} \frac{1}{2} \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right] \quad \text{ou} \quad \varepsilon_{ij} \stackrel{\text{def}}{=} \frac{1}{2} (u_{i,j} + u_{j,i}). \quad (4.16)$$

Existem outras definições para o tensor de deformação, dentre as quais podem se citar *Green-St. Venant*, que leva em conta um sistema de referências Lagrangiana¹, e a definição do tensor de deformação de *Almansi-Hamel*, que leva em conta um sistema de referências Euleriana². Mais informações podem ser encontradas em Slaughter e Verlag (2003).

4.3.3 Compatibilidade

Deformações são consideradas compatíveis matematicamente quando puderem ser definidas em termos de campos de deslocamentos continuamente diferenciáveis. Fisicamente, pode-se pensar a compatibilidade quando um corpo sofre uma deformação e os seus membros combinam-se de forma coesa, sem se sobrepor ou deixar espaços vazios. Um corpo contínuo deve permanecer contínuo após sofrer deformações.

O tensor deformação possui 6 componentes independentes. No entanto, os deslocamento apresentam apenas 3 componentes independentes. Devido a isso, existem três equações de compatibilidade, (FELIPPA, 2005). Em três dimensões escreve-se

$$\varepsilon_{ij,kl} + \varepsilon_{kl,ij} = \varepsilon_{ik,jl} + \varepsilon_{jl,ik}. \quad (4.17)$$

Em duas dimensões, apenas uma relação surge

$$\varepsilon_{11,22} + \varepsilon_{22,11} = 2\varepsilon_{12,12}, \quad \frac{\partial^2 \varepsilon_{11}}{\partial x_2^2} + \frac{\partial^2 \varepsilon_{22}}{\partial x_1^2} = \frac{\partial^2 \varepsilon_{12}}{\partial x_2 \partial x_1} \quad (4.18)$$

Existem diferentes métodos para se proceder com o solucionamento do problema da elasticidade linear sendo eles o método dos deslocamentos, método das forças e método do deslocamento-força (mixo).

No método dos deslocamentos, as tensões no equilíbrio são substituídas através das relações constitutivas pelas deformações e essas por meio das relações cinemáticas pelos deslocamentos. O resultado é um conjunto de expressões em termos dos deslocamentos, ou seja, as incógnitas são primariamente os deslocamentos.

¹ Também conhecida como descrição material, diz respeito a observação do sistema físico que pode ser decomposto de corpos e esse em partículas, (TONTI, 2013).

² Também conhecida como descrição espacial, diz respeito a observação de uma região no espaço definida como volume de controle, (TONTI, 2013).

No método das forças, as relações de compatibilidade para as deformações descritas nessa seção são utilizadas para escrever componentes independentes de deformação (aleatórios) em termos de termos livres. A nomenclatura livre e independente decorre da álgebra linear e diz respeito a sistemas lineares que possuem mais incógnitas do que equações, nesse caso, pode-se rearranjar o sistema de forma a escrever a solução em termos de variáveis livres, portanto, o sistema passa a ter infinitas soluções. Utilizando as deformações independentes nas relações constitutivas, o número de componentes de tensões incógnitas é reduzido para o número de equações de equilíbrio, logo, possibilita a solução do problema.

4.4 Relação Constitutiva

As tensões estão intimamente ligadas as deformações no corpo através de relações constitutivas características a cada material. As relações constitutivas são modelos matemáticos que descrevem o comportamento mecânico dos materiais.

4.4.1 Elasticidade

Um material é elástico quando a variação de tensões em um ponto material independe do tempo e do caminho espacial tomado dentre todas as configurações possíveis. Uma configuração espacial é caracterizada pela posição única de cada ponto material no espaço. Portanto, o estado de tensões pode ser determinado, em um determinado instante, unicamente pelo atual estado de deformações do ponto material, (SLAUGHTER; VERLAG, 2003).

Pode-se dizer também que um material é elástico quando este retorna a sua configuração inicial após a remoção das causas que provocaram a deformação inicialmente.

4.4.2 Linearidade

Um material é linear quando as componentes do tensor tensão são linearmente proporcionais às componentes do tensor deformação. Como o tensor tensão e o tensor deformação são de segunda ordem esses se relacionam por meio de um tensor de quarta ordem,

$$\underline{\sigma} = \underline{C} : \underline{\epsilon} \quad \text{ou} \quad \sigma_{ij} = C_{ijkl} \epsilon_{kl}. \quad (4.19)$$

onde, C_{ijkl} é o tensor de rigidez elástica de quarta ordem. Detalhes do produto interno duplo estão no Apêndice B.10. Essa equação também é conhecida como lei de Hooke generalizada. Deformações iniciais, devido a algum gradiente de temperatura, ou tensões iniciais, devido a pré-tensionamento, podem ser superpostos na Equação (4.19).

Explicitamente, tem-se

$$\begin{aligned}
\sigma_{11} &= C_{1111}\varepsilon_{11} + C_{1112}\varepsilon_{12} + \cdots + C_{1133}\varepsilon_{33} \\
\sigma_{12} &= C_{1211}\varepsilon_{11} + C_{1212}\varepsilon_{12} + \cdots + C_{1233}\varepsilon_{33} \\
&\vdots \\
\sigma_{33} &= C_{3311}\varepsilon_{11} + C_{3312}\varepsilon_{12} + \cdots + C_{3333}\varepsilon_{33}
\end{aligned} \tag{4.20}$$

4.4.3 Isotropia

O tensor de rigidez C_{ijkl} apresenta no total $3^4 = 81$ componentes, no entanto, devido à simetrias e outras condições discutidas em Slaughter e Verlag (2003), esse número reduz para apenas 2 parâmetros quando o material é considerado *isotrópico*. Os parâmetros são: E módulo de elasticidade, ou Young, e ν coeficiente de Poisson. Materiais isotrópicos possuem as mesmas propriedades independente da direção. Caso a Equação (4.19) seja reescrita com relação a outro sistema de coordenadas, o tensor de elasticidade manteria-se o mesmo em um material isotrópico, (CHANDRASEKHARIAH; DEBNATH, 1994). A mudança de sistema de coordenadas é a forma matemática de dizer que a propriedade mantém-se constante independente da orientação no espaço.

O tensor de rigidez isotrópico pode ser dado por

$$C_{ijkl} \stackrel{\text{def}}{=} \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}), \tag{4.21}$$

onde, δ_{ij} é o delta de Kronecker, ver Apêndice B.11, e λ e μ são constante de Lamé dados por

$$\lambda \equiv \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu \equiv \frac{E}{2(1+\nu)}. \tag{4.22}$$

Essa forma surge da avaliação do tensor de quarta ordem quando sujeito à mudanças de coordenadas específicas. A primeira mudança corresponde a uma mudança de base na qual o primeiro vetor da base ortonormal cartesiana rotaciona 90 graus com relação ao eixo z , o segundo rotaciona 90 graus com relação ao eixo x e o terceiro rotaciona 90 graus com relação ao eixo y . Essa mudança de base apresenta como matriz de transformação linear α_{ij} que leva um vetor nas coordenadas cartesianas para o novo sistema,

$$\alpha_{ij} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.23}$$

cujas linhas são as coordenadas dos novos vetores da base nas coordenadas da base antiga. Ou seja, o primeiro vetor $[1, 0, 0]$ vai para $[0, 1, 0]$, que significa uma rotação em z . A matriz que leva um vetor nas coordenadas da nova base para a base cartesiana é a inversa de α_{ij} ,

como a base é ortonormal, conseqüentemente α_{ij} é ortogonal, implicando que a inversa é igual a transposta,

$$\alpha^T \alpha = \mathbf{I} \quad \implies \quad \alpha^T \equiv \alpha^{-1} \quad (4.24)$$

Essa mudança de coordenada resulta na igualdade entre alguns termos do tensor de 4 ordem C_{ijkl} . Outras mudanças resultam em outras igualdades de outros termos e a nulidade de outros. Os termos iguais são igualados a constantes, α, β, γ , ou λ, μ , que são invariantes aos sistemas de coordenadas.

Em seguida, definido um tensor de 4 ordem como,

$$A_{ijkl} = C_{ijkl} - (\alpha \delta_{ij} \delta_{kl} + \beta \delta_{ik} \delta_{jl} + \gamma \delta_{il} \delta_{jk}) \quad (4.25)$$

utilizando as igualdades obtidas com as mudanças de coordenadas e fazendo outra mudança de coordenadas chega-se a equação (4.21). Mais detalhes dessa demonstração podem ser encontradas em Chandrasekharaiah e Debnath (1994, capítulo 2)

4.5 Condições de Contorno

As condições de contorno são condições impostas às variáveis do problema no contorno do domínio de interesse. Em problemas mecânicos as condições de contorno podem assumir duas formas com interpretação física, a primeira os deslocamentos são especificados no contorno (condição de Dirichlet) e a segunda tensões são aplicadas no contorno (condição de Neumann).

Condições de contorno de deslocamento podem ser representadas por

$$u_i(x_i) = \bar{u}_i, \quad x_i \in \Gamma_u, \quad (4.26)$$

e condições de tensão de superfície por

$$\sigma_{ij} n_j = \bar{t}_i, \quad x_i \in \Gamma_t \quad (4.27)$$

onde n_j é o vetor unitário normal à superfície do corpo, Γ_u representa parte da fronteira do domínio onde o deslocamento é imposto, Γ_t representa parte da fronteira onde tensões são impostas na superfície. Cada parte do contorno deve estar sujeita à um tipo de condição, ou seja, a superfície do corpo $\Gamma = \Gamma_u \cup \Gamma_t$. A Figura 4.5 ilustra a aplicação das condições de contorno e a distinção entre as superfícies.

4.6 Diagrama de Tonti: Modelo Elástico Linear

O diagrama de Tonti para representação gráfica do problema da elasticidade linear com material isotrópico é mostrado na Figura 4.6. Nessa representação o problema encontra-se em sua forma forte, ou seja, com conexões fortes entre as variáveis.

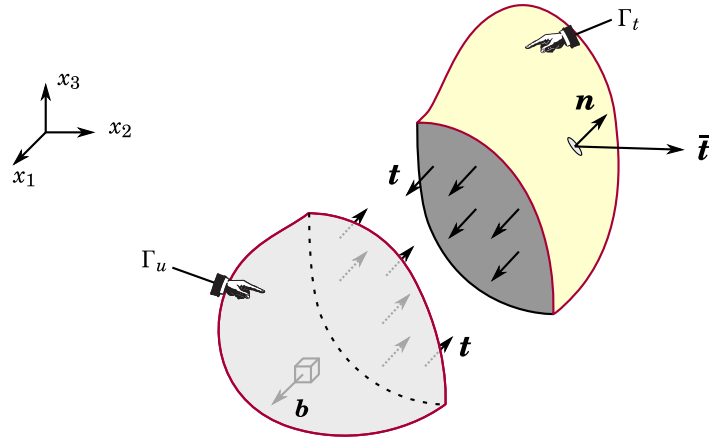


Figura 4.5 – Ilustração do problema da elasticidade num corpo geral. O volume do corpo está sujeita às forças de corpo \mathbf{b} e às forças de superfície dadas pelo vetor tensão $\bar{\mathbf{t}}$. A superfície é dividida em Γ_u e Γ_t . (FELIPPA, 2005)

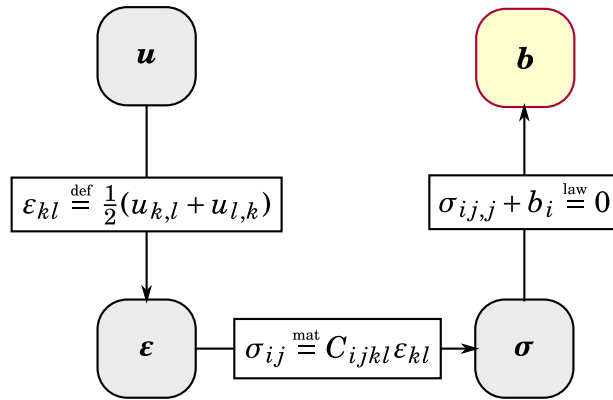


Figura 4.6 – Diagrama de Tonti com esquema gráfico do problema da elasticidade linear em sua forma forte. Nele é possível observar as conexões fortes através das equações cinemática, constituição e equilíbrio. (TONTI, 2013)

4.7 Formulação Variacional

Com o objetivo de se resolver o problema da elasticidade utilizando o método de discretização dos elementos finitos deve-se primeiro reescrever uma das equações do modelo em sua forma integral, “enfraquecendo”, portanto, os requerimentos de continuidade da solução. Detalhes sobre a forma estão explicitos na Seção 3.1.2. A equação passa a valer em sentido *de média* sobre os subdomínios, devido a inserção de uma função peso na integral, (FELIPPA, 2005).

A forma integral da equação pode ser obtida por meio do cálculo variacional. A construção da forma variacional a partir da forma forte começa pela escolha de um dos campos, ver diagrama na Figura 4.6, e enfraquecer uma ou mais ligações. No caso, as ligações representam equações diferenciais que ligam as variáveis de campo. Ao enfraquecer as ligações, gera-se um conjunto de novas equações em sua *forma fraca*, o que pode

ser entendido como uma etapa intermediária na passagem da *forma forte* para a *forma variacional*, (FELIPPA, 2005).

A *forma variacional* é materializada em termos de um funcional, Π , que contém relações integrais dos campos de interesse do problema. Associando a forma variacional à um princípio variacional, por exemplo, fazer a primeira variação ser igual a zero, $\delta\Pi = 0$, retorna-se a *forma forte* do modelo. Em mecânica, o funcional é chamado de Energia Potencial Total e o princípio variacional de Princípio de Minimização da Energia Potencial.

A construção da forma variacional a partir da forma forte do modelo é descrita a seguir.

4.7.1 Construção da Forma Variacional

A construção do funcional para o problema tem início na forma forte e por meio de manipulações matemáticas, e.g., integração por partes e teorema da divergência. Termina-se checando se a minimização do funcional retorna de fato a equação diferencial governante. Notar que, para um problema específico, não existe apenas um mas vários funcionais podem ser aplicados, (BATHE, 1996, p. 116). As vantagens na utilização das equações na forma variacional foram explicitadas na Seção 3.1.3.

Em Rao (2011, p. 295), a expressão variacional é obtida por meio da definição de energia potencial total como sendo a energia potencial da deformação elástica e uma parcela negativa do trabalho realizado por forças externas.

Em Felippa (2005), a construção da forma variacional consiste na seleção dos campos de interesse primários para em seguida enfraquecer um ou mais conexões, i.e., equações que ligam as variáveis de campos. O processo de enfraquecimento produz a forma fraca, equivalente a obtida por um método de residuais ponderados. A forma fraca é vista como um meio termo entre a forma forte, diferencial, e a forma variacional, integral. Os passos para produção da forma variacional são:

1. Seleção dos campos de interesse primário, *master field*. No caso, o campo de deslocamento, u_i , é escolhido como primário; e os de tensão e de deformação, σ_{ij} e ε_{ij} como secundários.
2. Seleção das ligações fracas, *weak links*. Ligações são equações que relacionam os campos, podendo ser *fortes* ou *fracas*. As ligações fortes garantem a relação *ponto-a-ponto* enquanto as ligações fracas garantem a relação em termos de média, por meio de uma relação integral, i.e., soma no domínio, com um peso.
3. Construção da forma fraca. A forma fraca é construída escolhendo *pesos* para as ligações fracas e integrando-os sobre o domínio;

4. Identificar na forma fraca o peso como sendo a variação de alguma variável física.
5. Identificar o funcional e verificar se a sua minimização retorna a equação diferencial.

4.7.2 Escolha da Variável Primária

O primeiro passo na construção da forma variacional é definir qual será a variável primária e conseqüentemente as secundárias. A escolha da variável primária define o caminho para discretização das equações. Os deslocamentos serão escolhidos como variáveis de interesse primário.

4.7.3 Ligações Fracas

As ligações fracas escolhidas são as equações de balanço, ou equilíbrio, e as condições de contorno de fluxo, ver Figura 4.7,

$$\text{Ligações Fracas} \quad \begin{cases} \sigma_{ij,j} + b_i = 0, & \text{em } \Omega \\ \sigma_{ij}n_j = \bar{t}_i & \text{em } \Gamma_t \end{cases} \quad (4.28)$$

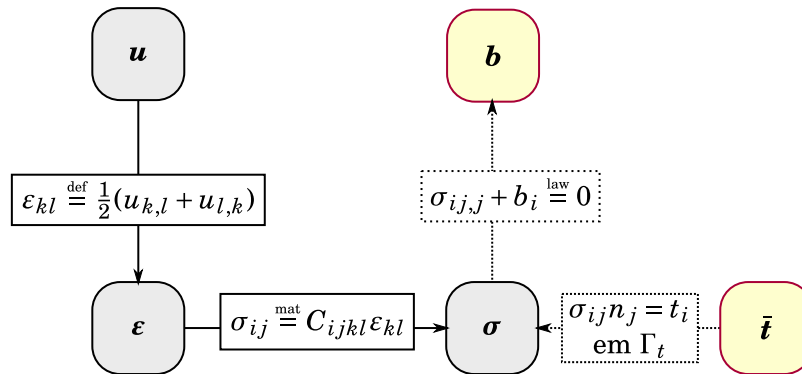


Figura 4.7 – Ilustração do modelo da elasticidade linear com as conexões fracas, em linha pontilhada. (FELIPPA, 2005)

4.7.4 Forma Fraca

A geração da forma fraca do modelo matemático representa um passo intermediário para construção da forma variacional. Para transformar uma equação da sua forma forte para a fraca utiliza-se o mesmo princípio estabelecido na construção do método dos residuais ponderados. Primeiro multiplica-se a equação por um campo vetorial diferenciável qualquer, v_i , e em seguida calcula-se a integral no domínio:

$$\int_{\Omega} (\sigma_{ij,j} + b_i) v_i \, d\Omega = 0. \quad (4.29)$$

Aplicando a regra do produto do cálculo, $(\sigma_{ij}v_i)_{,j} = \sigma_{ij,j}v_i + \sigma_{ij}v_{i,j}$, ver Apêndice B.8, e o teorema da divergência, Apêndice B.7, no primeiro termo, chega-se a

$$\int_{\Omega} \sigma_{ij,j}v_i \, d\Omega = - \int_{\Omega} \sigma_{ij}v_{i,j} \, d\Omega + \int_{\Gamma} \sigma_{ij}n_jv_i \, d\Gamma. \quad (4.30)$$

4.7.5 Identificando a Função Peso

A função peso v_i , utilizada na construção da forma fraca, pode ser assumida como sendo a variação do campo de deslocamentos, δu_i . Essa escolha surge pois reconhece-se a expressão $\int_{\Omega} \sigma_{ij}v_i \, d\Omega$ como medida da energia interna de deformação elástica. Essa escolha é justificada de forma rigorosa *a posteriori*.

Alguns autores, partem, na construção da forma fraca, diretamente com uma variação dos deslocamento. Em mecânica esse procedimento é chamado de Teorema dos Deslocamentos Virtuais. Identificando o campo vetorial v_i como sendo uma função peso, chega-se a exatamente a mesma construção por meio do Método de Residuais Ponderados.

4.7.6 Identificano o Funcional

Substituindo (4.30) em (4.29), e utilizando $v_i = \delta u_i$, tem-se

$$\begin{aligned} \int_{\Omega} \sigma_{ij,j}v_i \, d\Omega + \int_{\Omega} b_iv_i \, d\Omega &= 0 \\ - \int_{\Omega} \sigma_{ij}v_{i,j} \, d\Omega + \int_{\Gamma} \sigma_{ij}n_jv_i \, d\Gamma + \int_{\Omega} b_iv_i \, d\Omega &= 0 \\ \int_{\Omega} \sigma_{ij}\delta u_{i,j} \, d\Omega - \int_{\Gamma} \sigma_{ij}n_j\delta u_i \, d\Gamma - \int_{\Omega} b_i\delta u_i \, d\Omega &= 0. \end{aligned} \quad (4.31)$$

Identificando $\delta u_{i,j} = \delta \varepsilon_{ij}$ e substituindo o valor da tensão tração no contorno Γ , $\sigma_{ij}n_j = \bar{t}_i$, chega-se a

$$\delta \Pi = \int_{\Omega} \sigma_{ij}\delta \varepsilon_{ij} \, d\Omega - \int_{\Omega} b_i\delta u_i \, d\Omega - \int_{\Gamma} \bar{t}_i\delta u_i \, d\Gamma = 0, \quad (4.32)$$

que corresponde aplicar a primeira variação da Energia Potencial Total igual a zero. Dessa forma, para materiais lineares elásticos é possível identificar o funcional por meio da definição,

$$\Pi[u_i] \stackrel{\text{def}}{=} \frac{1}{2} \int_{\Omega} \sigma_{ij}\varepsilon_{ij} \, d\Omega - \int_{\Omega} b_iu_i \, d\Omega - \int_{\Gamma} \bar{t}_iu_i \, d\Gamma. \quad (4.33)$$

4.7.7 Princípio de Minimização da Energia Potencial

A energia potencial de um corpo elástico é definida como

$$\Pi \stackrel{\text{def}}{=} U - W \quad (4.34)$$

onde, U representa a energia potencial da deformação elástica e W é o trabalho das forças externas, de corpo e de superfície. O sinal negativo acompanhando o trabalho indica que ao realizar trabalho, que representa uma das forças de transferir energia, perde-se energia potencial, (COOK, 2002, p. 139).

O Princípio de Minimização da Energia Potencial Total diz: *De todos os possíveis estados que um corpo pode assumir o estado que satisfaz as condições de equilíbrio faz a Energia Potencial Total ser mínima.* Matematicamente esse princípio pode ser expresso com a primeira variação do funcional igualada a zero, $\delta\Pi = 0$.

A energia potencial de deformação elástica é definida como,

$$U \stackrel{\text{def}}{=} \int_{\Omega} \mathcal{U} \, d\Omega, \quad (4.35)$$

onde, \mathcal{U} é a densidade de energia de deformação elástica dada por, (COOK, 2002, p. 142) e (BUCALEM; BATHE, 2011, p. 391)

$$\mathcal{U} \stackrel{\text{def}}{=} \int_0^{\sigma_x} \sigma_x \, d\varepsilon_x + \int_0^{\sigma_y} \sigma_y \, d\varepsilon_y + \int_0^{\sigma_z} \sigma_z \, d\varepsilon_z + \int_0^{\sigma_{xy}} \sigma_{xy} \, d\gamma_{xy} + \int_0^{\sigma_{xz}} \sigma_{xz} \, d\gamma_{xz} + \int_0^{\sigma_{yz}} \sigma_{yz} \, d\gamma_{yz} \quad (4.36)$$

Caso o problema seja linear, essa expressão é reduzida ao seguinte resultado com as componentes expressas nas Equações (4.2) e (4.15).

$$\mathcal{U} \stackrel{\text{def}}{=} \frac{1}{2} \sigma_{ij} \varepsilon_{ij} \equiv \frac{1}{2} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}. \quad (4.37)$$

4.8 Equações na Forma Matricial

A implementação computacional do modelo matemático para o problema da elasticidade linear será feita utilizando a linguagem de programação Python. Optou-se por fazer primeiramente o caso da elasticidade linear plana, simplificando o pós processamento.

Primeiramente serão apresentados as equações do modelo da elasticidade linear plana em sua forma matricial. obtidas após os procedimentos de *discretização e aproximação*. Em seguida as rotinas serão apresentadas. Por fim, os resultados obtidos através de implementação.

As equações de campo descritas no diagrama apresentado na Figura 4.6 podem ser convenientemente expressas em uma forma matricial. Para elasticidade linear plana, o tensor tensão de Cauchy e o tensor de deformação são simétricos, ou seja, podem ser descritos por meio de 3 parâmetros cada. Utilizando esse fato, os tensores podem ser reescritos como vetores de 3 coordenadas,

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{bmatrix}. \quad (4.38)$$

A Figura 4.8 ilustra o diagrama do modelo da elasticidade linear em notação matricial. Nesse caso, para elasticidade plana as equações explicitamente são

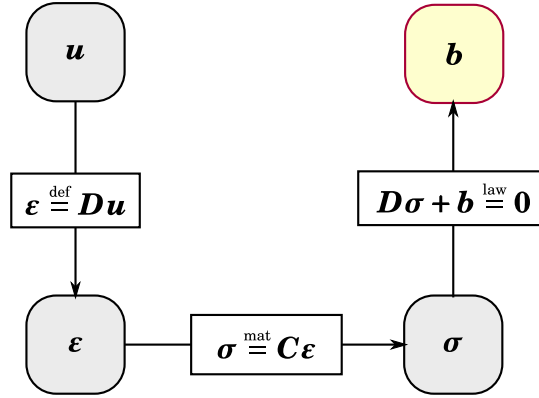


Figura 4.8 – Ilustração do modelo da elasticidade linear com a notação matricial. (FELIPPA, 2005)

$$D^T \sigma + b^{\text{law}} = 0, \quad \begin{bmatrix} \partial_x & 0 & \partial_y \\ 0 & \partial_y & \partial_x \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \end{bmatrix}^{\text{law}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (4.39)$$

$$\epsilon \stackrel{\text{def}}{=} D u, \quad \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} \partial_x & 0 \\ 0 & \partial_y \\ \partial_y & \partial_x \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \quad (4.40)$$

$$\sigma^{\text{mat}} \stackrel{\text{mat}}{=} C \epsilon, \quad \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \stackrel{\text{mat}}{=} \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix} \quad (4.41)$$

A forma fraca do modelo da elasticidade linear, expressa na Equação (4.32), também pode ser expressa utilizando a notação matricial,

$$\int_{\Omega} \delta \epsilon^T \sigma \, d\Omega - \int_{\Omega} \delta u^T b \, d\Omega - \int_{\Gamma} \delta u^T \bar{t} \, d\Gamma = 0, \quad (4.42)$$

que corresponde ao Teorema Dos Deslocamentos Virtuais, (BATHE, 1996, p. 156).

4.9 Discretização

O processo de discretização consiste reduzir o número de graus de liberdade para um número finito, como discutido na Seção 3.2. Matematicamente, essa etapa é efetuada fazendo a partição do domínio em subdomínio denominados elementos finitos, $\Omega^{(e)}$, ver Figura 4.9. A forma fraca pode ser expressa para cada elemento individualmente, visto que o operador de integração é linear,

$$\sum_{e=1}^{ne} \left(\int_{\Omega^e} \delta \epsilon^T \sigma \, d\Omega^e - \int_{\Omega^e} \delta u^T b \, d\Omega^e - \int_{\Gamma^e} \delta u^T \bar{t} \, d\Gamma^{(e)} \right) = 0 \quad (4.43)$$

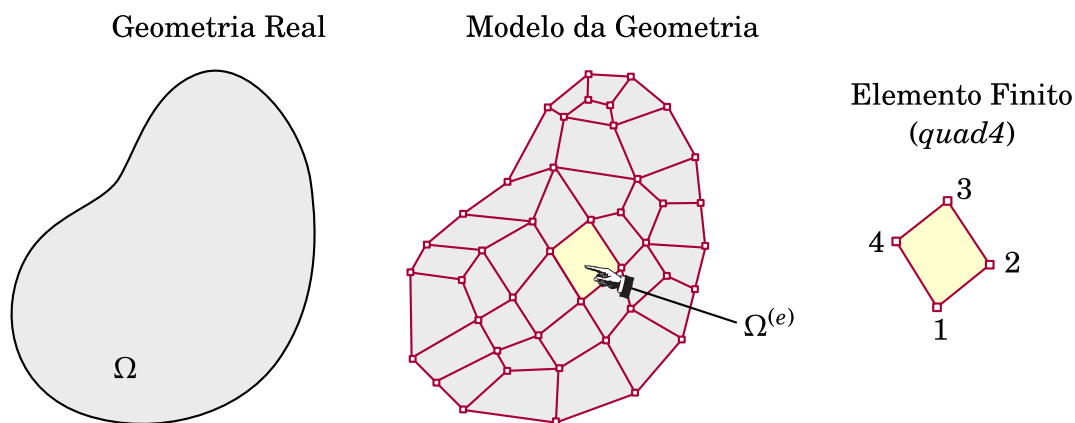


Figura 4.9 – Ilustração do processod e discretização e de um elemento individual de quatro nós.

4.10 Aproximação

O processo de aproximação consiste em assumir uma solução aproximada para o campo de deslocamentos. A motivação para essa etapa pode ser encontrada nas Seções 3.2.3 e 3.3.1.

Como nessa etapa são escolhidas funções interpoladoras para a solução aproximada, é definido então o *tipo de elemento* utilizado. Nesse trabalho decidiu-se, para uma primeira implementação, utilizar o elemento plano de quatro nós (*quad4*).

A solução aproximada é definida para cada elemento individualmente,

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} u_x(\mathbf{x}) \\ u_y(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} N_1(\mathbf{x})\mu_{x1} + N_2(\mathbf{x})\mu_{x2} + N_3(\mathbf{x})\mu_{x3} + N_4(\mathbf{x})\mu_{x4} \\ N_1(\mathbf{x})\mu_{y1} + N_2(\mathbf{x})\mu_{y2} + N_3(\mathbf{x})\mu_{y3} + N_4(\mathbf{x})\mu_{y4} \end{bmatrix} \quad (4.44)$$

onde, μ_{xi} e μ_{yi} são as coordenadas generalizadas, ou coeficientes da combinação linear, incógnitas. O índice $i = 1, 2, 3, 4$ corresponde a cada um dos nós. Os oito coeficientes compõem os graus de liberdade (dof) do elemento, portanto, cada elemento desse tipo possui dois graus de liberdade por nó, e no total oito dof. Como esses coeficientes referem-se a apenas um elemento, uma indicação por superscrito ^(e) traria mais clareza, no entanto, aumentaria a complexidade da notação, portanto, optou-se por suprimi-la. As funções $N_i(\mathbf{x})$ são definidas a priori e definem o tipo do elemento, nesse caso existem apenas 4 para o elemento *quad4*. Esse conjunto de funções são da família Lagrangiana linear, ou seja, possuem a característica de valer 1 num ponto específico e zero nos outros e a transição entre os pontos é linear. De fato, são bilineares, pois são lineares nas duas direções.

A aproximação pode ser convenientemente expressa em forma matricial por,

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} N_1(\mathbf{x}) & 0 & N_2(\mathbf{x}) & 0 & N_3(\mathbf{x}) & 0 & N_4(\mathbf{x}) & 0 \\ 0 & N_1(\mathbf{x}) & 0 & N_2(\mathbf{x}) & 0 & N_3(\mathbf{x}) & 0 & N_4(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \mu_{x1} \\ \mu_{y1} \\ \mu_{x2} \\ \mu_{y2} \\ \mu_{x3} \\ \mu_{y3} \\ \mu_{x4} \\ \mu_{y4} \end{bmatrix} \equiv \mathbf{N}\boldsymbol{\mu} \quad (4.45)$$

As funções de forma, ou funções interpolares, $N_i(\mathbf{x})$, são dadas apenas em função das coordenadas isoparamétricas $\boldsymbol{\xi} \equiv (\xi, \eta)$. A utilização do espaço isoparamétrico como sistema de coordenadas apresenta vantagens com relação a implementação computacional e será discutida na Seção 4.10.1, adiante. Explicitamente, as funções de forma em função das variáveis que definem o espaço isoparamétrico são

$$\begin{aligned} N_1(\boldsymbol{\xi}) &= \frac{1}{4}(1-\xi)(1-\eta) \\ N_2(\boldsymbol{\xi}) &= \frac{1}{4}(1+\xi)(1-\eta) \\ N_3(\boldsymbol{\xi}) &= \frac{1}{4}(1+\xi)(1+\eta) \\ N_4(\boldsymbol{\xi}) &= \frac{1}{4}(1-\xi)(1+\eta) \end{aligned} \quad (4.46)$$

Essas funções foram plotadas na Figura 3.5. Elas podem ser convenientemente escritas como,

$$N_a(\boldsymbol{\xi}) = \frac{1}{4}(1 + \xi_a \xi)(1 + \eta_a \eta) \quad (4.47)$$

onde, (ξ_a, η_a) são as coordenadas dos nós no espaço isoparamétrico.

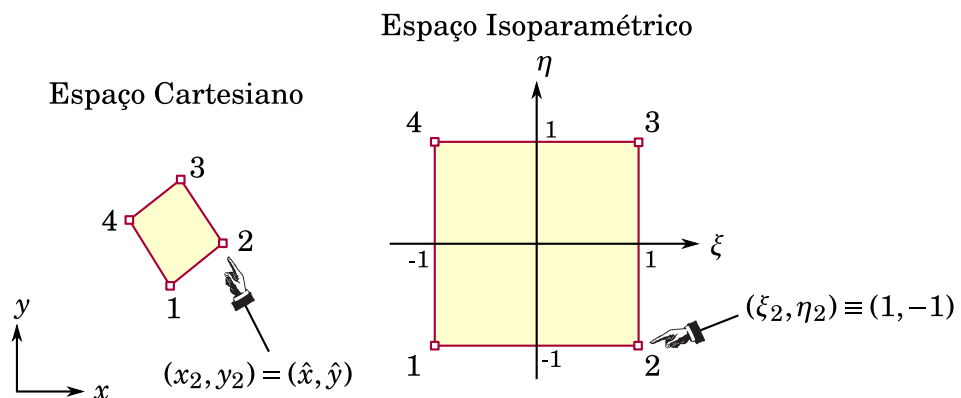


Figura 4.10 – Ilustração do espaço isoparamétrico. Os valores (\hat{x}, \hat{y}) representam a coordenada cartesiana do nó 2 desse elemento.

4.10.1 Mudança de Coordenadas

A utilização do espaço isoparamétrico foi inicialmente introduzida por Bruce Irons em 1968, (WILSON, 1998, p. 5.1). A introdução do novo sistema de coordenadas permitiu o rápido desenvolvimento da técnica pois elementos de alta ordem, e de diferentes geometria, puderam ser desenvolvidos com certa facilidade, uma vez que o desenvolvimento parte do mesmo princípio.

A mudança de coordenadas entre o espaço cartesiano e o espaço isoparamétrico, também chamado de espaço natural, é feita através da transformação, $\mathbf{G}(\xi)$,

$$\mathbf{x} = \mathbf{G}(\xi) \equiv \mathbf{x}(\xi) \quad (4.48)$$

por meio do qual é possível expressar as coordenadas cartesianas em função das coordenadas isoparamétricas, (ξ, η) . Explicitamente essa transformação é dada por,

$$\mathbf{x}(\xi) = \begin{bmatrix} x(\xi, \eta) \\ y(\xi, \eta) \end{bmatrix} = \begin{bmatrix} x_1 N_1(\xi, \eta) + x_2 N_2(\xi, \eta) + x_3 N_3(\xi, \eta) + x_4 N_4(\xi, \eta) \\ y_1 N_1(\xi, \eta) + y_2 N_2(\xi, \eta) + y_3 N_3(\xi, \eta) + y_4 N_4(\xi, \eta) \end{bmatrix} \equiv \begin{bmatrix} \sum_{a=1}^4 x_a N_a \\ \sum_{a=1}^4 y_a N_a \end{bmatrix} \quad (4.49)$$

onde, (x_i, y_i) são as coordenadas dos nós do elementos no espaço cartesiano. Esse mapeamento é facilmente verificado com a inserção das coordenadas de um dos nós nas coordenadas isoparamétricas,

$$\mathbf{x}(-1, -1) = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (4.50)$$

pois nesse ponto, $(-1, -1)$, as funções de forma são nulas com exceção de $N_1(-1, -1) = 1$. Essa transformação pode ser também expressa em forma matricial,

$$\mathbf{x}(\xi) = \begin{bmatrix} N_1(\xi) & 0 & N_2(\xi) & 0 & N_3(\xi) & 0 & N_4(\xi) & 0 \\ 0 & N_1(\xi) & 0 & N_2(\xi) & 0 & N_3(\xi) & 0 & N_4(\xi) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix} \quad (4.51)$$

O espaço isoparamétrico leva esse nome pois os deslocamentos e coordenadas são dadas parametricamente, parâmetros ξ e η , e através das mesmas funções de forma, N , (MOSALAM, 2011).

4.10.2 Jacobiano

A utilização da técnica de substituição de variáveis para resolver integrais causa alterações na própria função, $N(x, y) \equiv N(x(\xi, \eta), y(\xi, \eta)) \equiv N(\xi, \eta)$, nos limites de integração,

de $\int_{x_1}^{x_2} \int_{y_1}^{y_2}$ para $\int_{-1}^1 \int_{-1}^1$, e no diferencial das variáveis de integração, $d\Omega^e \equiv dx \wedge dy \equiv \mathcal{J} d\xi \wedge d\eta$. O operador \wedge indica *produto externo* e possui as seguintes propriedades: $d\rho \wedge d\rho \equiv 0$, $d\rho \wedge d\theta \equiv -d\theta \wedge d\rho$, (TONTI, 2013, p. 24). O termo \mathcal{J} representa o determinante da matriz Jacobiana e pode ser obtido como,

$$\begin{aligned} dx \wedge dy &\equiv \left(\frac{\partial x}{\partial \xi} d\xi + \frac{\partial x}{\partial \eta} d\eta \right) \wedge \left(\frac{\partial y}{\partial \xi} d\xi + \frac{\partial y}{\partial \eta} d\eta \right) \\ &\equiv \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} \right) d\xi \wedge d\eta + \left(\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \right) d\eta \wedge d\xi \\ &\equiv \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \right) d\xi \wedge d\eta \end{aligned} \quad (4.52)$$

onde,

$$\mathcal{J} \stackrel{\text{def}}{=} \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \right) \equiv \det \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \quad (4.53)$$

O Jacobiano possui uma interpretação geométrica: é o fator de escala entre elementos de área em cada sistema de coordenadas. A Figura 4.11 mostra um elemento no sistema de coordenadas cartesiano e um no sistema de coordenadas isoparamétricas. A partir dos vetores,

$$\mathbf{p}_1 = \begin{bmatrix} x(\xi_1, \eta_1) \\ y(\xi_1, \eta_1) \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} x(\xi_1 + \Delta\xi, \eta_1) \\ y(\xi_1 + \Delta\xi, \eta_1) \end{bmatrix}, \quad \mathbf{p}_3 = \begin{bmatrix} x(\xi_1, \eta_1 + \Delta\eta) \\ y(\xi_1, \eta_1 + \Delta\eta) \end{bmatrix} \quad (4.54)$$

e as seguintes diferenças,

$$\mathbf{v} = \mathbf{p}_2 - \mathbf{p}_1 = \begin{bmatrix} x(\xi_1 + \Delta\xi, \eta_1) - x(\xi_1, \eta_1) \\ y(\xi_1 + \Delta\xi, \eta_1) - y(\xi_1, \eta_1) \end{bmatrix}, \quad \mathbf{w} = \mathbf{p}_3 - \mathbf{p}_1 = \begin{bmatrix} x(\xi_1, \eta_1 + \Delta\eta) - x(\xi_1, \eta_1) \\ y(\xi_1, \eta_1 + \Delta\eta) - y(\xi_1, \eta_1) \end{bmatrix} \quad (4.55)$$

podemos calcular a área ΔA .

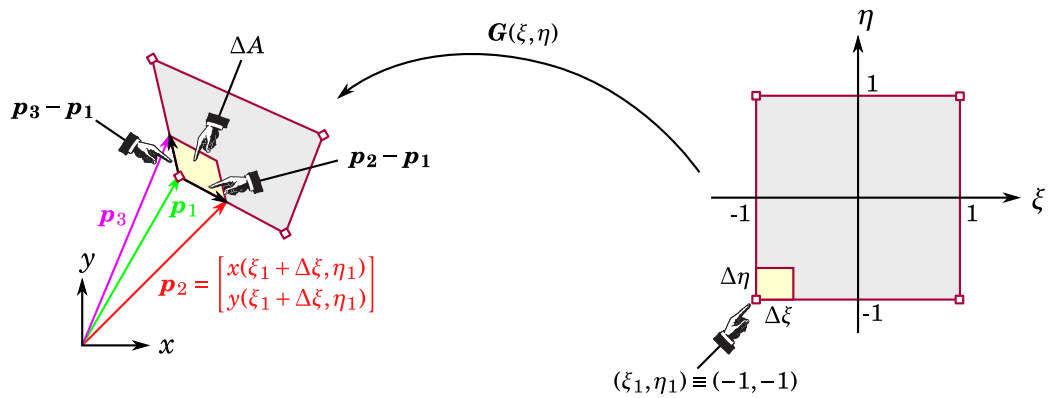


Figura 4.11 – Ilustração do processo de mudança de coordenada e formação da matriz Jacobiana.

Reconhecendo os termos nos vetores como os presentes numa expansão de Taylor truncada no termo de primeira derivada, os vetores podem ser reescritos como,

$$\mathbf{v} = \begin{bmatrix} \frac{\partial x}{\partial \xi}(\xi_1, \eta_1) \\ \frac{\partial y}{\partial \xi}(\xi_1, \eta_1) \end{bmatrix} \Delta \xi, \quad \mathbf{w} = \begin{bmatrix} \frac{\partial x}{\partial \eta}(\xi_1, \eta_1) \\ \frac{\partial y}{\partial \eta}(\xi_1, \eta_1) \end{bmatrix} \Delta \eta, \quad (4.56)$$

finalmente, fazendo o produto vetorial entre esses dois vetores chega-se a área ΔA , mostrada na Fig. 4.11,

$$\|\mathbf{v} \times \mathbf{w}\| = \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \right) \Delta \xi \Delta \eta \quad (4.57)$$

conclui-se portanto que a área no espaço cartesiano relaciona-se com a área do espaço isoparamétrico por meio do termo em parêntesis, chamado de Jacobiano. O Jacobiano corresponde ao determinante de uma matrix, chamada de matriz Jacobiana,

$$\mathbf{J} \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (4.58)$$

O código abaixo mostra a função `jacobian(self, xyz, dN_ei)` que cria a matriz jacobiana, calcula a inversa, o determinante e o comprimento de arco para a integral de linha.

```

1 def jacobian(self, xyz, dN_ei):
2     """Creates the Jacobian matrix of the mapping between an element
3
4     """
5     # Jac = [ x1_e1 x2_e1
6             x1_e2 x2_e2 ]
7     Jac = np.dot(dN_ei, xyz)
8
9     det_jac = ((Jac[0, 0]*Jac[1, 1] -
10                Jac[0, 1]*Jac[1, 0]))
11
12     jac_inv = np.linalg.inv(Jac)
13
14     # Using Chain rule,
15     # N_xi = N_eI * eI_xi (2x8 array)
16     dN_xi = np.zeros((2, 4))
17     dN_xi[0, :] = (dN_ei[0, :]*jac_inv[0, 0] +
18                   dN_ei[1, :]*jac_inv[0, 1])
19
20     dN_xi[1, :] = (dN_ei[0, :]*jac_inv[1, 0] +
21                   dN_ei[1, :]*jac_inv[1, 1])
22
23     # Length of the transformation arch
24     # Jacobian for line integral-2.
25     arch_length = np.array([
26         (Jac[0, 0]**2. + Jac[0, 1]**2.)**(1./2.),
27         (Jac[1, 0]**2. + Jac[1, 1]**2.)**(1./2.),
28         (Jac[0, 0]**2. + Jac[0, 1]**2.)**(1./2.),
29         (Jac[1, 0]**2. + Jac[1, 1]**2.)**(1./2.)])
30     return det_jac, dN_xi, arch_length

```

4.11 Construção do Elemento

Substituindo a solução aproximada na forma fraca, Eq. (5.6). Nesse caso, a inserção ocorre por meio da utilização das relações constitutivas e cinemáticas para expressar as tensões em termos dos deslocamentos,

$$\int_{\Omega^e} \delta \boldsymbol{\varepsilon}^T \mathbf{C} \mathbf{D} \mathbf{N} \boldsymbol{\mu} d\Omega^e - \int_{\Omega^e} \delta \mathbf{u}^T \mathbf{b} d\Omega^e - \int_{\Gamma^e} \delta \mathbf{u}^T \bar{\mathbf{t}} d\Gamma^e = 0, \quad (4.59)$$

como foi utilizado uma função qualquer $\mathbf{v} = \delta \mathbf{u}$, ou um deslocamento virtual qualquer, na construção da forma fraca, pode-se, agora, defini-la utilizando o mesmo conjunto de funções utilizado para os deslocamentos,

$$\mathbf{v} = \mathbf{N}(\boldsymbol{\xi}) \mathbf{v} \quad (4.60)$$

onde, \mathbf{v} são os coeficientes constantes da combinação linear.

O procedimento de escolher o mesmo conjunto de funções de aproximação, \mathbf{N} , para a solução aproximada e para a função peso é conhecido como *Método de Galerkin*, ver Seção 3.2.2.1, nesse caso, a matriz de rigidez, definida adiante, será simétrica, (BATHE, 1996, p. 163). A função teste deve satisfazer os mesmos requerimentos de continuidade para a função de aproximação e ainda ser nula nos pontos onde o deslocamento é especificado, (BATHE, 1996).

Caso tivéssemos substituído a solução aproximada na expressão do funcional da energia potencial total, expresso na Eq. (4.33). E em seguida aplicar o Teorema de Minimização da Energia Potencial Total para agora minimizar o funcional com respeito a um número finito de parâmetros, $\boldsymbol{\mu}$, chegaríamos no mesmo resultado que os dois anteriores, ver Seção 3.3.2.

Substituindo a solução aproximada para o deslocamento virtual, ou função teste,

$$\int_{\Omega^e} (\mathbf{D} \mathbf{N} \mathbf{v})^T \mathbf{C} \mathbf{D} \mathbf{N} \boldsymbol{\mu} d\Omega^e - \int_{\Omega^e} \mathbf{N} \mathbf{v}^T \mathbf{b} d\Omega^e - \int_{\Gamma^e} \mathbf{N} \mathbf{v}^T \bar{\mathbf{t}} d\Gamma^e = 0, \quad (4.61)$$

notando que $\delta \boldsymbol{\varepsilon} = \mathbf{D} \delta \mathbf{u}$, da mesma forma que na Eq. (5.4). Chamando $\mathbf{B} = \mathbf{D} \mathbf{N}$ uma matriz 3×8 contendo as derivadas das funções de forma com respeito as variáveis cartesianas, explicitamente

$$\mathbf{B} = \begin{bmatrix} \partial_x N_1(\xi_i) & 0 & \partial_x N_2(\xi_i) & 0 & \partial_x N_3(\xi_i) & 0 & \partial_x N_4(\xi_i) & 0 \\ 0 & \partial_y N_1(\xi_i) & 0 & \partial_y N_2(\xi_i) & 0 & \partial_y N_3(\xi_i) & 0 & \partial_y N_4(\xi_i) \\ \partial_y N_1(\xi_i) & \partial_x N_1(\xi_i) & \partial_y N_2(\xi_i) & \partial_x N_2(\xi_i) & \partial_y N_3(\xi_i) & \partial_x N_3(\xi_i) & \partial_y N_4(\xi_i) & \partial_x N_4(\xi_i) \end{bmatrix}. \quad (4.62)$$

Novamente utilizou-se $\partial_x \equiv \partial/\partial x$. A equação na forma fraca em notação matricial fica,

$$\int_{\Omega^e} (\mathbf{B} \mathbf{v})^T \mathbf{C} \mathbf{B} \mathbf{u} d\Omega^e - \int_{\Omega^e} (\mathbf{N} \mathbf{v})^T \mathbf{b} d\Omega^e - \int_{\Gamma^e} (\mathbf{N} \mathbf{v})^T \bar{\mathbf{t}} d\Gamma^e = 0. \quad (4.63)$$

A primeira integral gera a matriz de rigidez, a segunda gera um carregamento devido as forças de corpo e a terceira gera um carregamento devido ao vetor tensão no contorno do elemento, se existir.

Com o objetivo de se fatorar os coeficientes constantes da função teste,

$$\int_{\Omega^e} \mathbf{v}^T \mathbf{B}^T \mathbf{C} \mathbf{B} \mathbf{u} \, d\Omega^e - \int_{\Omega^e} \mathbf{v}^T \mathbf{N}^T \mathbf{b} \, d\Omega^e - \int_{\Gamma^e} \mathbf{v}^T \mathbf{N}^T \bar{\mathbf{t}} \, d\Gamma^e = 0, \quad (4.64)$$

argumenta-se que como foram escolhidos de forma arbitrária podemos fazê-los iguais a qualquer valor. Em particular, fazendo o vetor de coeficientes da função teste igual ao vetor unitário cujo único componente não nulo ocupa a posição i , \mathbf{e}_i , resulta na operação da primeira equação do sistema linear. Repetindo esse procedimento para outros \mathbf{e}_i , chega-se ao sistema linear cujas incógnitas são os coeficientes da solução aproximada. Esse procedimento possui interpretação física e corresponde a assumir deslocamentos virtuais unitários, (BATHE, 1996, p. 164).

$$\int_{\Omega^e} \mathbf{B}^T \mathbf{C} \mathbf{B} \mathbf{u} \, d\Omega^e - \int_{\Omega^e} \mathbf{N}^T \mathbf{b} \, d\Omega^e - \int_{\Gamma^e} \mathbf{N}^T \bar{\mathbf{t}} \, d\Gamma^e = 0. \quad (4.65)$$

Para proceder com o cálculo das matrizes elementares é necessário algumas etapas preliminares para efetuar as operações de integração utilizando substituição de variáveis.

4.11.1 Derivadas das Funções de Forma

A derivação ocorre com respeito a variável cartesiana, no entanto, as funções de forma são definidas apenas no espaço isoparamétrico. Isso ocorre pois evita-se a necessidade de se criar funções de forma específicas para cada elementos. Para obter a derivada das funções de forma com relação as variáveis cartesianas, seria necessário utilizar a transformação inversa a apresentada na Eq. (4.48). Essa inversão requer um esforço algébrico e não é facilmente extensível para o caso de se utilizar funções de forma de outro grau. Utiliza-se então um outro procedimento, descrito a seguir.

A derivada das funções de forma com respeito as variáveis do espaço isoparamétrico pode ser expressa por

$$\frac{\partial N_a(x_i)}{\partial \xi_j} \equiv \frac{\partial N_a(\mathbf{x}(\boldsymbol{\xi}))}{\partial \xi_j}, \quad a = 1, 2, 3, 4 \quad (4.66)$$

nesse caso, para indicar a variável da função é conveniente utilizar o negrito, $\boldsymbol{\xi} = (\xi, \eta)$ e $\mathbf{x} = (x, y)$, para não confundir com os índices da notação indicial e não subentender somas implícitas. Por meio da regra da cadeia, Apêndice B.9, essa derivada é dada por

$$\begin{aligned} \frac{\partial N_a(\mathbf{x}(\boldsymbol{\xi}))}{\partial \xi} &= \frac{\partial N_a}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_a}{\partial y} \frac{\partial y}{\partial \xi}, \\ \frac{\partial N_a(\mathbf{x}(\boldsymbol{\xi}))}{\partial \eta} &= \frac{\partial N_a}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_a}{\partial y} \frac{\partial y}{\partial \eta}, \end{aligned} \quad (4.67)$$

que em forma matricial fica

$$\begin{bmatrix} \frac{\partial N_a}{\partial \xi} \\ \frac{\partial N_a}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_a}{\partial x} \\ \frac{\partial N_a}{\partial y} \end{bmatrix} \quad (4.68)$$

nota-se que a matriz com as derivadas parciais das variáveis cartesianas com respeito as variáveis isoparamétrica é a matriz Jacobiana, apresentada na Eq. (4.58). Portanto, para se obter as derivadas das funções de forma com respeito as variáveis cartesianas, basta

$$\begin{bmatrix} \frac{\partial N_a}{\partial x} \\ \frac{\partial N_a}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial N_a}{\partial \xi} \\ \frac{\partial N_a}{\partial \eta} \end{bmatrix}. \quad (4.69)$$

As derivadas das funções de forma com respeito as variáveis isoparamétricas são obtidas utilizando Eq. (4.47),

$$\begin{aligned} \frac{\partial N_a(\xi, \eta)}{\partial \xi} &= \frac{1}{4} \xi_a (1 + \eta_a \eta), \\ \frac{\partial N_a(\xi, \eta)}{\partial \eta} &= \frac{1}{4} \eta_a (1 + \xi_a \xi). \end{aligned} \quad (4.70)$$

A inversa da matriz Jacobiana, no caso plano, pode ser obtida diretamente por,

$$\mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \quad (4.71)$$

as derivadas das variáveis cartesianas, nesse caso funções, com respeito as variáveis isoparamétricas podem ser calculadas diretamente utilizando o mapeamento descrito na Eq. (4.48) e Eq. (4.49), e as derivadas das funções de forma na Eq. (4.70),

$$\begin{aligned} \frac{\partial x}{\partial \xi} &= \frac{\partial Q_1(\xi, \eta)}{\partial \xi} \\ &= \frac{\partial}{\partial \xi} \sum_{a=1}^4 x_a N_a(\xi, \eta) \\ &= \sum_{a=1}^4 x_a \frac{\partial N_a(\xi, \eta)}{\partial \xi} \\ &= \sum_{a=1}^4 x_a \frac{1}{4} \xi_a (1 + \eta_a \eta) \end{aligned} \quad (4.72)$$

utilizando o mesmo procedimento para as outras derivadas chega-se a

$$\begin{aligned} \frac{\partial x}{\partial \eta} &= \sum_{a=1}^4 x_a \frac{1}{4} \eta_a (1 + \xi_a \xi), \\ \frac{\partial y}{\partial \xi} &= \sum_{a=1}^4 y_a \frac{1}{4} \xi_a (1 + \eta_a \eta), \\ \frac{\partial y}{\partial \eta} &= \sum_{a=1}^4 y_a \frac{1}{4} \eta_a (1 + \xi_a \xi). \end{aligned} \quad (4.73)$$

O código a seguir mostra a função, `shape_funcao(self, xez)`, responsável por criar as funções de forma e calcular o seu valor nos pontos definidos pela coordenada $x_{ez}=(\xi,\eta)$.

```

1 def shape_funcao(self, xez):
2     """Create the basis function and evaluate them at xez coordinates
3
4     """
5     # variables in the natural (iso-parametric) domain
6     xi = xez[0]
7     eta = xez[1]
8
9     # Terms of the shape function
10    xi_term = 1/2 *(1 + self.XEZ[:, 0] * xi)
11    eta_term = 1/2 *(1 + self.XEZ[:, 1] * eta)
12
13    # Basis functions
14    # N = [ N_1 N_2 N_3 N_4 ]
15    N = xi_term * eta_term
16    self.N = np.array(N)
17
18    # Derivative of the shape functions
19    # dN_ei = [ dN1_e1 dN2_e1 ...
20              dN1_e2 dN2_e2 ... ]
21    self.dN_ei = np.zeros((2, 4))
22    self.dN_ei[0, :] = 0.5 * self.XEZ[:, 0] * eta_term
23    self.dN_ei[1, :] = 0.5 * self.XEZ[:, 1] * xi_term
24
25    return self.N, self.dN_ei

```

4.11.2 Matrizes Elementares

Para formar as matrizes elementares a partir da Eq. (4.65) basta retirar os termos constantes da integração,

$$\mathbf{K} \mathbf{u} = \mathbf{P} \quad (4.74)$$

onde,

$$\begin{aligned} \mathbf{K} &\stackrel{\text{def}}{=} \int_{\Omega^e} \mathbf{B}^T \mathbf{C} \mathbf{B} \, d\Omega^e \\ \mathbf{P} &\stackrel{\text{def}}{=} \int_{\Omega^e} \mathbf{N}^T \mathbf{b} \, d\Omega^e + \int_{\Gamma^e} \mathbf{N}^T \bar{\mathbf{t}} \, d\Gamma^e \equiv \mathbf{P}_b + \mathbf{P}_t \end{aligned} \quad (4.75)$$

4.11.3 Matriz de Rigidez \mathbf{K}

O primeiro termo na Eq. (4.65) gera a matriz de rigidez do elemento. A matriz de rigidez é dada por,

$$\begin{aligned} \mathbf{K} &\stackrel{\text{def}}{=} \int_{\Omega^e} \mathbf{B}^T \mathbf{C} \mathbf{B} \, d\Omega^e \\ &= \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T \mathbf{C} \mathbf{B} \, \mathcal{J} \, d\xi \wedge d\eta \end{aligned} \quad (4.76)$$

A integral é calculada nas variáveis (ξ,η) , nesse caso os limites de integração $[-1,1]$ permitem a utilização do procedimento numérico Quadratura Gaussiana para sua solução,

Apêndice B.13. Nesse caso, utilizando Quadratura com dois pontos a solução da integral fica,

$$\mathbf{K} = \sum_r^2 \sum_s^2 \mathbf{B}^T(\xi_r, \eta_s) \mathbf{C} \mathbf{B}(\xi_r, \eta_s) \mathcal{J}(\xi_r, \eta_s). \quad (4.77)$$

Nesse caso, as funções de forma e as derivadas contidas no jacobiano são calculadas nos pontos de Gauss, (ξ_r, η_s) .

Em um *loop* para cada elemento, as matrizes \mathbf{B} , \mathbf{C} e \mathbf{J} devem ser formadas. A matriz com as derivadas das funções de forma, \mathbf{B} pode ser formada após calculado a matriz Jacobiana. A matriz Jacobiana, Eq. (4.58), é formada com a derivada da função mapeamento, Eq. (4.72) e Eq. (4.73), que depende apenas das coordenadas nodais no sistema de coordenadas global, (x_a, y_a) .

```

1 def stiffness_matrix(self, t=1):
2     """Build the element stiffness matrix
3
4     """
5     k = np.zeros((8, 8))
6
7     C = self.c_matrix(t) # Build the element material matrix
8
9     gauss_points = self.XEZ / np.sqrt(3.0)
10
11    for gp in gauss_points:
12        _, dN_ei = self.shape_function(xez=gp)
13        dJ, dN_xi, _ = self.jacobian(self.xyz, dN_ei)
14
15        B = np.array([
16            [dN_xi[0, 0], 0, dN_xi[0, 1], 0, dN_xi[0, 2], 0,
17             dN_xi[0, 3], 0],
18            [0, dN_xi[1, 0], 0, dN_xi[1, 1], 0, dN_xi[1, 2], 0,
19             dN_xi[1, 3]],
20            [dN_xi[1, 0], dN_xi[0, 0], dN_xi[1, 1], dN_xi[0, 1],
21             dN_xi[1, 2], dN_xi[0, 2], dN_xi[1, 3], dN_xi[0, 3]])]
22
23        k += (B.T @ C @ B)*dJ
24
25    return k

```

4.11.4 Vetor de carregamento \mathbf{P}_b

O vetor de carregamento devido às forças de corpo, \mathbf{P}_b , é dado por

$$\begin{aligned} \mathbf{P}_b &\stackrel{\text{def}}{=} \int_{\Omega^e} \mathbf{N}^T \mathbf{b} d\Omega^e \\ &= \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T \mathbf{b}(\xi, \eta) \mathcal{J} d\xi \wedge d\eta. \end{aligned} \quad (4.78)$$

Nesse caso, a função que define a força de corpo em $[N/m^3]$ deve ser calculada utilizando as variáveis no espaço isoparamétrico, no entanto ela é especificada com as variáveis cartesianas, portanto, deve-se primeiro converter as variáveis das coordenadas isoparamétricas para o espaço cartesiano e em seguida utilizar esses valores para calcular a função no ponto correto, nesse ponto o valor é uma constante e pode sair da expressão integral (linearidade). A função responsável por gerar o mapeamento utiliza a Eq. (4.49).

Utilizando Quadratura Gaussiana para o cálculo da integral, o vetor pode ser obtido por

$$\mathbf{P}_b = \sum_r^2 \sum_s^2 \mathbf{N}^T(\xi_r, \eta_s) \mathbf{b} \mathcal{J}(\xi_r, \eta_s). \quad (4.79)$$

O código abaixo mostra esse procedimento

```

1  def load_body_vector(self, b_force, t=1):
2      """Build the element vector due body forces b_force
3
4      """
5      gauss_points = self.XEZ / np.sqrt(3.0)
6
7      pb = np.zeros(8)
8      for gp in gauss_points:
9          N, dN_ei = self.shape_function(xez=gp)
10         dJ, dN_xi, _ = self.jacobian(self.xyz, dN_ei)
11
12         x1, x2 = self.mapping(self.xyz)
13
14         pb[0] += N[0]*b_force(x1, x2, t)[0]*dJ
15         pb[1] += N[0]*b_force(x1, x2, t)[1]*dJ
16         pb[2] += N[1]*b_force(x1, x2, t)[0]*dJ
17         pb[3] += N[1]*b_force(x1, x2, t)[1]*dJ
18         pb[4] += N[2]*b_force(x1, x2, t)[0]*dJ
19         pb[5] += N[2]*b_force(x1, x2, t)[1]*dJ
20         pb[6] += N[3]*b_force(x1, x2, t)[0]*dJ
21         pb[7] += N[3]*b_force(x1, x2, t)[1]*dJ
22
23     return pb

```

4.11.5 Vetor de carregamento \mathbf{P}_t

O vetor carregamento devido ao vetor tensão na superfície é dado por,

$$\begin{aligned}
 \mathbf{P}_t &\stackrel{\text{def}}{=} \int_{\Gamma^e} \mathbf{N}^T \bar{\mathbf{t}} d\Gamma^e \\
 &= \sum_{\ell=1}^4 \int_{\Gamma_\ell} \mathbf{N}^T \bar{\mathbf{t}} d\Gamma_\ell^e \\
 &= \sum_{\ell=1}^4 \int_{-1}^1 \mathbf{N}^T(\xi, \eta) \bar{\mathbf{t}}(\xi, \eta) \mathcal{J}_\ell d\Gamma_\ell^e \\
 &= \int_{-1}^1 \mathbf{N}^T(1, \eta) \bar{\mathbf{t}}(1, \eta) \sqrt{\left(\frac{\partial x}{\partial \eta}(1, \eta)\right)^2 + \left(\frac{\partial y}{\partial \eta}(1, \eta)\right)^2} d\eta \\
 &\quad + \int_{-1}^1 \mathbf{N}^T(\xi, 1) \bar{\mathbf{t}}(\xi, 1) \sqrt{\left(\frac{\partial x}{\partial \xi}(\xi, 1)\right)^2 + \left(\frac{\partial y}{\partial \xi}(\xi, 1)\right)^2} d\xi \\
 &\quad + \int_{-1}^1 \mathbf{N}^T(-1, \eta) \bar{\mathbf{t}}(-1, \eta) \sqrt{\left(\frac{\partial x}{\partial \eta}(-1, \eta)\right)^2 + \left(\frac{\partial y}{\partial \eta}(-1, \eta)\right)^2} d\eta \\
 &\quad + \int_{-1}^1 \mathbf{N}^T(\xi, -1) \bar{\mathbf{t}}(\xi, -1) \sqrt{\left(\frac{\partial x}{\partial \xi}(\xi, -1)\right)^2 + \left(\frac{\partial y}{\partial \xi}(\xi, -1)\right)^2} d\xi.
 \end{aligned} \quad (4.80)$$

A integral de linha é dividida em cada uma das quatro arestas do elemento, $\ell = 1, 2, 3, 4$. O comprimento de arco nas novas variáveis é dado por $\mathcal{J}_\ell d\Gamma_\ell^e$. Esse valor é obtido utilizando

a regra da cadeia,

$$dx \stackrel{\text{def}}{=} \frac{\partial x}{\partial \xi} d\xi + \frac{\partial x}{\partial \eta} d\eta, \quad (4.81)$$

sendo que para cada lado do elemento o valor de ξ ou η é fixo igual a 1 ou -1. Nesse caso, para o lado onde $\xi = 1$ os diferenciais cartesianos ficam

$$\begin{aligned} dx &= \frac{\partial x}{\partial \eta} d\eta, \\ dy &= \frac{\partial y}{\partial \eta} d\eta, \end{aligned} \quad (4.82)$$

o comprimento de arco dado por esses diferenciais, para cálculo da integral de linha, é obtido pelo teorema de pitágoras,

$$\mathcal{J}_\ell d\Gamma_\ell^e = \sqrt{\left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2}. \quad (4.83)$$

Cada uma das integrais é então calculada utilizando Quadratura Gaussiana de dois pontos,

$$\begin{aligned} \mathbf{P}_t &= \sum_r^2 \mathbf{N}^T(1, \eta) \bar{\mathbf{t}}(1, \eta) \sqrt{\left(\frac{\partial x}{\partial \eta}(1, \eta)\right)^2 + \left(\frac{\partial y}{\partial \eta}(1, \eta)\right)^2} \\ &+ \sum_r^2 \mathbf{N}^T(\xi, 1) \bar{\mathbf{t}}(\xi, 1) \sqrt{\left(\frac{\partial x}{\partial \xi}(\xi, 1)\right)^2 + \left(\frac{\partial y}{\partial \xi}(\xi, 1)\right)^2} \\ &+ \sum_r^2 \mathbf{N}^T(-1, \eta) \bar{\mathbf{t}}(-1, \eta) \sqrt{\left(\frac{\partial x}{\partial \eta}(-1, \eta)\right)^2 + \left(\frac{\partial y}{\partial \eta}(-1, \eta)\right)^2} \\ &+ \sum_r^2 \mathbf{N}^T(\xi, -1) \bar{\mathbf{t}}(\xi, -1) \sqrt{\left(\frac{\partial x}{\partial \xi}(\xi, -1)\right)^2 + \left(\frac{\partial y}{\partial \xi}(\xi, -1)\right)^2} \end{aligned} \quad (4.84)$$

A implementação para construção desse vetor está no código a seguir,

```

1  def load_traction_vector(self, traction_bc, t=1):
2      """Build element load vector due traction_bction boundary condition
3
4      """
5      gp = np.array([
6          [-1.0/np.sqrt(3), -1.0],
7          [1.0/np.sqrt(3), -1.0]],
8          [[1.0, -1.0/np.sqrt(3)],
9           [1.0, 1.0/np.sqrt(3)]],
10         [[-1.0/np.sqrt(3), 1.0],
11          [1.0/np.sqrt(3), 1.0]],
12         [[-1.0, -1.0/np.sqrt(3)],
13          [-1.0, 1/np.sqrt(3)]]])
14
15     pt = np.zeros(8)
16
17     # loop for specified boundary conditions
18     for key in traction_bc(1, 1).keys():
19         line = key[1]
20
21         for ele_boundary_line, ele_side in zip(self.at_boundary_line,
22                                               self.side_at_boundary):
23             # Check if this element is at the line with traction
24             if line == ele_boundary_line:
```

```

25
26     # perform the integral with GQ
27     for w in range(2):
28         N, dN_ei = self.shape_function(xez=gp[ele_side, w])
29         _, _, arch_length = self.jacobian(self.xyz, dN_ei)
30
31         dL = arch_length[ele_side]
32         x1, x2 = self.mapping(self.xyz)
33
34         pt[0] += N[0] * traction_bc(x1, x2, t)[key][0] * dL
35         pt[1] += N[0] * traction_bc(x1, x2, t)[key][1] * dL
36         pt[2] += N[1] * traction_bc(x1, x2, t)[key][0] * dL
37         pt[3] += N[1] * traction_bc(x1, x2, t)[key][1] * dL
38         pt[4] += N[2] * traction_bc(x1, x2, t)[key][0] * dL
39         pt[5] += N[2] * traction_bc(x1, x2, t)[key][1] * dL
40         pt[6] += N[3] * traction_bc(x1, x2, t)[key][0] * dL
41         pt[7] += N[3] * traction_bc(x1, x2, t)[key][1] * dL
42
43     else:
44         # Catch element that is not at boundary
45         continue
46
47     return pt

```

4.11.6 Vetor de carregamento \mathbf{P}_e

O modelo constitutivo pode incluir também um termo devido a deformações iniciais de origem não mecânica,

$$\boldsymbol{\sigma} \stackrel{\text{mat}}{=} \mathbf{C}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0). \quad (4.85)$$

Nesse caso, quando na forma fraca, o vetor \mathbf{P}_e pode ser formado,

$$\begin{aligned} \mathbf{P}_e &\stackrel{\text{def}}{=} \int_{\Omega^e} \mathbf{B}^T \mathbf{C} \boldsymbol{\varepsilon}_0 d\Omega^e \\ &= \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T(\xi, \eta) \mathbf{C} \boldsymbol{\varepsilon}_0 \mathcal{J} d\xi \wedge d\eta. \end{aligned} \quad (4.86)$$

Utilizando Quadratura Gaussiana de dois pontos nas duas direções chega-se a,

$$\mathbf{P}_e = \sum_r^2 \sum_s^2 \mathbf{B}^T(\xi_r, \eta_s) \mathbf{C} \boldsymbol{\varepsilon}_0 \mathcal{J}(\xi_r, \eta_s). \quad (4.87)$$

A rotina abaixo mostra o procedimento para cálculo desse vetor

```

1     def load_strain_vector(self, t=1):
2         """Build the element vector due initial strain
3
4         """
5         C = self.c_matrix(t)
6
7         gauss_points = self.XEZ / np.sqrt(3.0)
8
9         pe = np.zeros(8)
10        for gp in gauss_points:
11            _, dN_ei = self.shape_function(xez=gp)
12            dJ, dN_xi, _ = self.jacobian(self.xyz, dN_ei)
13
14            B = np.array([
15                [dN_xi[0, 0], 0, dN_xi[0, 1], 0, dN_xi[0, 2], 0,
16                 dN_xi[0, 3], 0],
17                [0, dN_xi[1, 0], 0, dN_xi[1, 1], 0, dN_xi[1, 2], 0,

```

```

18         dN_xi[1, 3]],
19         [dN_xi[1, 0], dN_xi[0, 0], dN_xi[1, 1], dN_xi[0, 1],
20         dN_xi[1, 2], dN_xi[0, 2], dN_xi[1, 3], dN_xi[0, 3]])
21
22     pe += (B.T @ C @ self.eps0)*dJ
23
24     return pe

```

4.11.7 Classe do elemento

Para organização da arquitetura do programa optou-se por organizar os elementos em uma classe pai que contém atributos básicos de todos os elementos e classes especializadas que definem um tipo em particular. As classes especializadas herdam da class pai os atributos básicos. Essa arquitetura é popular entre os programas que utilizam o paradigma orientado objeto. A vantagem dessa abordagem é que a biblioteca de elementos pode ser facilmente estendida, bastando adicionar um novo módulo contendo a definição de uma classe que contenha funções chaves: `shape_func()`, `mapping()`, `jacobian()`, `stiffness_matrix()`, `load_body_vector()`, `load_strain_vector()` e `load_traction_vector()`. Essas funções devem retornar as matrizes e vetores elementares. A montagem das matrizes globais é efetuada utilizando os graus de liberdade (dof) em arrays de incidência. A Figura 4.12 mostra o esquema da classe juntamente com os atributos e métodos que são definidos.

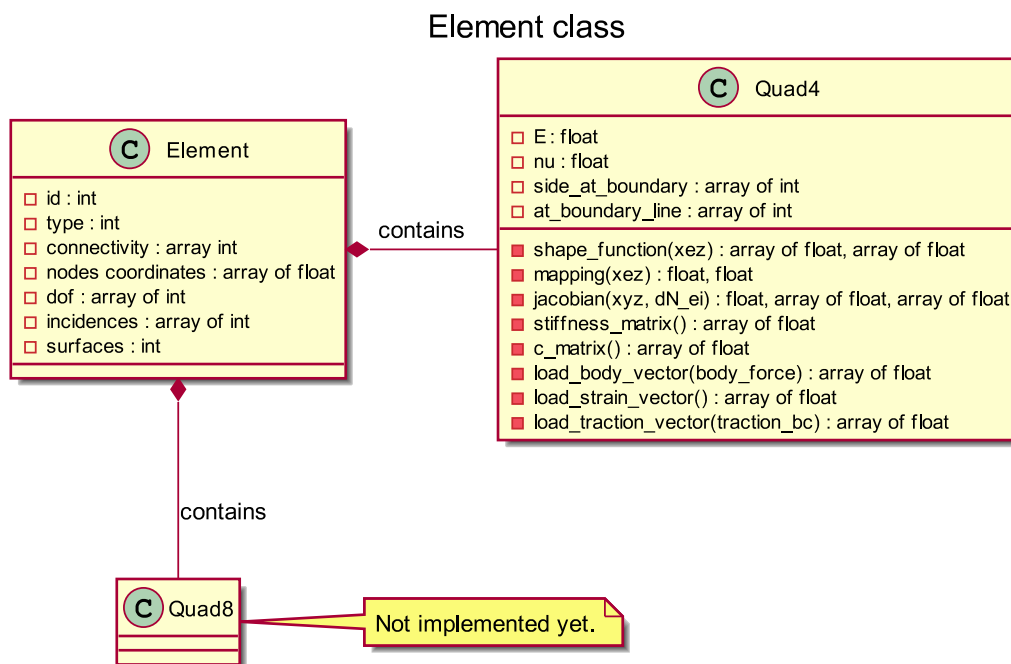


Figura 4.12 – Classe pai dos elementos e classe específica que herda os atributos.

4.12 Solução do problema estático

O problema estático da elasticidade plana linear é resolvido seguindo o diagrama mostrado na Fig. 4.13. Em um *loop*, as matrizes elementares são montadas e já alocadas na incidência correspondente a cada grau de liberdade afetado pelo elemento. Em seguida, as condições de contorno de deslocamento são aplicadas alterando as matrizes globais e o sistema linear é resolvido utilizando o método `np.linalg.solve()`.

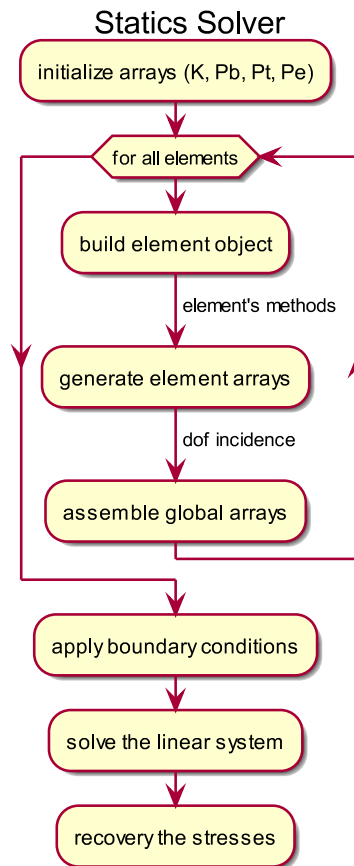


Figura 4.13 – Rotina de cálculo para solução do problema estático.

As condições de contorno de deslocamento são aplicadas substituindo o valor especificado no vetor \mathbf{P} e alterando a linha correspondente na matriz de rigidez \mathbf{K} . Na matriz de rigidez a linha passa a ter apenas entradas nulas exceto na coluna correspondente ao grau de liberdade onde a condição de contorno foi especificada. Por exemplo, uma matriz 4×4 qualquer e um vetor \mathbf{P} qualquer,

$$\begin{bmatrix} 10 & 2 & 0 & 0 \\ 2 & 13 & 3 & 0 \\ 0 & 4 & 8 & 2 \\ 0 & 0 & 11 & 3 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 9 \\ 7 \end{bmatrix}. \quad (4.88)$$

Caso o valor de μ_1 seja especificado por uma condição do problema, a linha que corresponde à μ_1 é substituída por entradas com exceção da entrada que afeta diretamente μ_1 e o valor

de P_1 é substituído pelo valor da condição de contorno especificada

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 13 & 3 & 0 \\ 0 & 4 & 8 & 2 \\ 0 & 0 & 11 & 3 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} = \begin{bmatrix} u^* \\ 5 \\ 9 \\ 7 \end{bmatrix}, \quad (4.89)$$

dessa maneira, a primeira equação do sistema torna-se

$$\mu_1 = u^*. \quad (4.90)$$

4.13 Malha

A malha para discretização do domínio em elementos é feita utilizando o programa gmsh ou especificada manualmente.

O programa gmsh, (GEUZAINÉ; REMACLE, 2009), é um *software* livre desenvolvido para ser simples, rápido e leve em tarefas como geração de geometria com seu módulo de CAD (*computer-aided design*), geração de malha e pós-processamento.

Os programas desenvolvidos nesse trabalho requerem dois arquivos, um contendo a geometria (.geo) e outro contendo a malha, (.msh). Os arquivos podem ser criados manualmente e devem ter o seguinte formato, primeiro o .geo:

```

1 Point(1) = {0, 0, 0, 1.0};
2 Point(2) = {1, 0, 0, 1.0};
3 Point(3) = {1, 1, 0, 1.0};
4 Point(4) = {0, 1, 0, 1.0};
5 Line(1) = {1, 2};
6 Line(2) = {2, 3};
7 Line(3) = {3, 4};
8 Line(4) = {4, 1};
9 Physical Line(5) = {4};
10 Physical Line(6) = {1};
11 Physical Line(7) = {2};
12 Physical Line(8) = {3};
13 Line Loop(9) = {4, 1, 2, 3};
14 Plane Surface(10) = {9};
15 Physical Surface(11) = {10};

```

Os parâmetros `Physical Line` e `Physical Surface` são utilizados para se identificar os elementos onde condições de contorno podem ser especificadas. O número entre parênteses identifica o elemento. Os primeiros três números na definição do ponto são as coordenadas x, y, z ; o último número especifica o tamanho da elemento da malha nesse ponto.

O arquivo da malha .msh:

```

1 $MeshFormat
2 2.2 0 8
3 $EndMeshFormat
4 $Nodes
5 9
6 1 0 0 0
7 2 1 0 0
8 3 1 1 0
9 4 0 1 0
10 5 0.5 0 0
11 6 1 0.5 0
12 7 0.5 1 0
13 8 0 0.5 0
14 9 0.4 0.6 0
15 $EndNodes
16 $Elements
17 12
18 1 1 2 6 1 1 5
19 2 1 2 6 1 5 2
20 3 1 2 7 2 2 6
21 4 1 2 7 2 6 3
22 5 1 2 8 3 3 7
23 6 1 2 8 3 7 4
24 7 1 2 5 4 4 8
25 8 1 2 5 4 8 1
26 9 3 2 11 10 1 5 9 8
27 10 3 2 11 10 5 2 6 9
28 11 3 2 11 10 9 6 3 7
29 12 3 2 11 10 8 9 7 4
30 $EndElements

```

nele é necessário especificar as coordenadas x, y, z de todos os nós que compõem a malha e os elementos. Existem vários tipos de elementos, o número na segunda coluna indica o tipo do elemento, no caso, o número 1 indica o elemento: 2-node line e o número 3 indica o elemento 4-node quadrangle. A terceira coluna na especificação do elemento indica o número de *tags* que existem para identificar o elemento, no caso, todos os elementos possuem duas etiquetas. Por último, os números que restam são os identificadores dos nós que foram o elemento, ou seja, conectividade.

Para ler esses arquivos foi criado um módulo `gmsch.py` responsável por interpretar essas linhas e converte-las em *arrays* ou *dicionários*. O módulo criado é composto por uma classe, `Parse()` cujos atributos são os parâmetros que definem a malha.

O código a seguir mostra a implementação dessa classe,

```

1 import numpy as np
2 import os
3 import re
4
5
6 def find_num(string):
7     """Find all numbers in a string
8
9     """
10    num = re.findall(r'[+-]?(?:\d+(?:\.\d*)?|\.\d+)', string)
11    return num
12
13
14 class Parse(object):
15     """Parse the .geo and .msh file into dictionaries or lists or arrays

```


Parse class

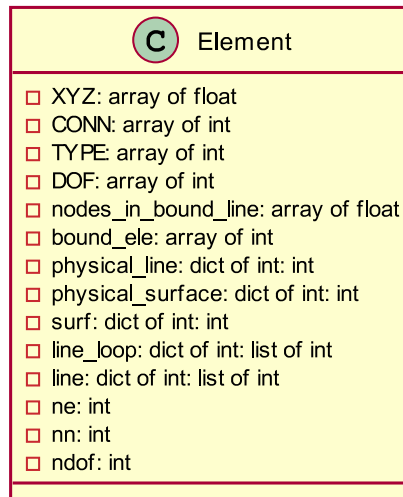


Figura 4.14 – Parse class inside the gmsh module

```

16
17
18
19 """
20 def __init__(self, filename):
21     geo_path = os.path.join(filename+'.geo')
22     geo_file = open(geo_path, 'r')
23
24     # physical_line_tag: line_tag
25     self.physical_line = {}
26     # physical_surf_tag: surface_tag
27     self.physical_surf = {}
28     # surf_tag: line_loop_tag
29     self.surf = {}
30     # line_loop_tag: [line1_tag line2_tag line3_tag ...]
31     self.line_loop = {}
32     # line_tag: [node1_tag node2_tag]
33     self.line = {}
34
35 for txt_line in geo_file:
36     num_list = find_num(txt_line)
37
38     if txt_line.startswith('Physical Line'):
39         nl = [int(f) - 1 for f in num_list]
40         self.physical_line[nl[0]] = nl[1]
41
42     if txt_line.startswith('Plane Surface'):
43         nl = [int(f) - 1 for f in num_list]
44         self.surf[nl[0]] = nl[1]
45
46     if txt_line.startswith('Physical Surface'):
47         nl = [int(f) - 1 for f in num_list]
48         self.physical_surf[nl[0]] = nl[1]
49
50     if txt_line.startswith('Line('):
51         nl = [int(f) - 1 for f in num_list]
52         self.line[nl[0]] = nl[1:]
53
54     if txt_line.startswith('Line Loop'):
55         nl = [abs(int(f)) - 1 for f in num_list]
56         self.line_loop[nl[0]] = nl[1:]
57
58 msh_path = os.path.join(filename+'.msh')
59 msh_file = open(msh_path, 'r')
60
61 # node_tag: [node1 node2]

```

```

61     XYZ = {}
62     # element_tag: [node1_tag node2_tag node3_tag node4_tag]
63     CONN = {}
64     # element_tag: physical_surf_tag
65     self.surf_of_ele = {}
66     # [line_tag node1_tag node2_tag]
67     self.nodes_in_bound_line = []
68     # element TYPE: [e1_type, e2_type ...]
69     TYPE = []
70
71     e_i = 0
72     for txt_line in msh_file:
73         num_list = find_num(txt_line)
74
75         # nodes coordinates xyz
76         if len(num_list) == 4:
77             n_tag = int(num_list[0]) - 1
78             XYZ[n_tag] = [float(f) for f in num_list[1:3]]
79
80         if len(num_list) == 9:
81             conn = [int(f) - 1 for f in num_list[5:]]
82             CONN[e_i] = conn
83             self.surf_of_ele[e_i] = int(num_list[4]) - 1
84             TYPE.append(int(num_list[1]))
85             e_i += 1
86
87         if len(num_list) == 7:
88             nl = [int(f) - 1 for f in num_list[4:]]
89             self.nodes_in_bound_line.append([nl[0], nl[1], nl[2]])
90
91     self.nodes_in_bound_line = np.array(self.nodes_in_bound_line)
92     self.XYZ = np.array(list(XYZ.values()))
93     self.CONN = np.array(list(CONN.values()))
94     self.TYPE = np.array(TYPE)
95     self.ne = len(CONN)
96     self.nn = len(XYZ)
97
98     # DEPENDS ON ELEMENT TYPE!
99     # DOF = [[dof1_e1, dof2_e1, ... dof8_e1]
100            # [dof1_e2, dof2_e2, ... dof8_e2]]
101     DOF = []
102     for e, conn in enumerate(self.CONN):
103         DOF.append([2 * conn[0], 2 * conn[0] + 1,
104                   2 * conn[1], 2 * conn[1] + 1,
105                   2 * conn[2], 2 * conn[2] + 1,
106                   2 * conn[3], 2 * conn[3] + 1])
107     self.DOF = np.array(DOF)
108
109     # Number of total degree of freedom
110     self.ndof = 2*self.nn
111
112     # [ele side_of_ele_at_bound bound_line]
113     bound_ele = []
114     for e, conn in enumerate(self.CONN):
115         for l, n1, n2 in self.nodes_in_bound_line:
116
117             if np.all([n1, n2] == self.CONN[e, 0:2]):
118                 bound_ele.append([e, 0, 1])
119
120             if np.all([n1, n2] == self.CONN[e, 1:3]):
121                 bound_ele.append([e, 1, 1])
122
123             if np.all([n1, n2] == self.CONN[e, 2:4]):
124                 bound_ele.append([e, 2, 1])
125
126             if np.all([n1, n2] == self.CONN[e, :-3]):
127                 bound_ele.append([e, 3, 1])
128
129     self.bound_ele = np.array(bound_ele)
130     self.gmsh = 1.0

```

4.14 Verificação

Nessa seção as funções serão testadas separadamente. Primeiro a função que aplica as condições de contorno de deslocamento, em seguida o método que cria o vetor devido às forças de corpo e por fim o método que cria o vetor devido aos vetores tensão no contorno.

4.14.1 Condições de Contorno de Deslocamento

Para verificar as condições de contorno de deslocamento o método proposto em (MOSALAM, 2011) é utilizado. Nele, o deslocamento é especificado igual a 0 nos nós 0, 3 e 7; deslocamento igual a 0.5 nos nós 4, 6; e deslocamento igual a 1 nos nós 1, 5 e 2. Esse campo de deslocamentos produz uma deformação unitária na direção x . Nesse caso, os campos de deformação e tensão são dados por

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{E}{1-\nu^2} \\ \frac{\nu E}{1-\nu^2} \\ 0 \end{bmatrix}, \quad (4.91)$$

para $E = 1000[Pa]$ e $\nu = 0.3$ o valor do campo de tensão é

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} 1098.9 \\ 329.67 \\ 0 \end{bmatrix} \quad (4.92)$$

O *script* utilizado para especificar esse problema está mostrado no seguinte código,

```
1 import numpy as np
2 from elastopy.model import Build
3 from elastopy.mesh import gmsh
4 from elastopy.material import Material
5 from elastopy.solvers import statics
6 from elastopy.postprocess import plotter
7
8 mesh_file = 'patch'
9 mesh = gmsh.Parse(mesh_file)
10 model = Build(mesh)
11 material = Material(E={9: 1000}, nu={9: 0.3})
12
13
14 def body_forces(x1, x2, t=1):
15     return np.array([0.0, 0.0])
16
17
18 def traction_imposed(x1, x2, t=1):
19     return {}
20
21
22 def displacement_imposed(x1, x2):
23     return {}
```

```

24     ('nodes', 0, 3, 7): [0.0, 0.0],
25     ('nodes', 4, 6): [0.5, 0.0],
26     ('nodes', 1, 5, 2): [1.0, 0.0]}
27
28 U, SIG = statics.solver(model, material, body_forces,
29                       traction_imposed, displacement_imposed)
30
31 plotter.model(model, ele=True, nodes_label=True, edges_label=True,
32              ele_label=True)
33 plotter.model_deformed(model, U, magf=0.1, ele=True)
34
35 plotter.show()

```

O resultado para as tensões em cada nó estão expressas abaixo. A primeira coluna corresponde as tensões normais ao plano x , a segunda coluna as tensões normais ao plano y e a terceira a tensão cisalhante.

```

1 print(SIG)
2 [[ 1.09890110e+03  3.29670330e+02  8.57766930e-13]
3  [ 1.09890110e+03  3.29670330e+02 -1.70803542e-13]
4  [ 1.09890110e+03  3.29670330e+02 -6.72576966e-15]
5  [ 1.09890110e+03  3.29670330e+02 -3.04422246e-13]
6  [ 1.09890110e+03  3.29670330e+02 -2.13504428e-13]
7  [ 1.09890110e+03  3.29670330e+02 -6.98062777e-14]
8  [ 1.09890110e+03  3.29670330e+02  1.31910040e-13]
9  [ 1.09890110e+03  3.29670330e+02  4.54879834e-13]
10 [ 1.09890110e+03  3.29670330e+02  3.98031801e-14]]

```

A Figura 4.15 mostra a geometria juntamente com a numeração de cada elemento do problema e o campo de deslocamento especificado.

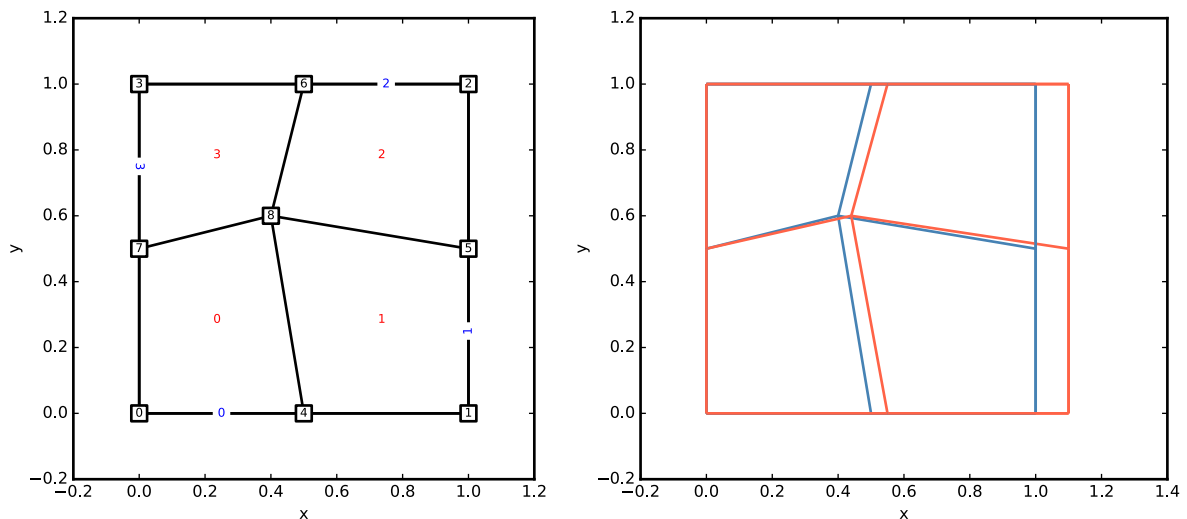


Figura 4.15 – Verificação do método aplica as condições de contorno de deslocamento, o fator de ampliação do campo de deslocamento é 0.1.

4.14.2 Forças de Corpo

Para o problema onde existe apenas forças de corpo distribuídas, no caso, a força gravitacional,

$$\mathbf{b} = \begin{bmatrix} 0 \\ -\rho g \end{bmatrix}, \quad (4.93)$$

onde, ρ é a massa específica do material e g é a aceleração gravitacional. A geometria do problema é uma chapa de 1[m] de largura e 5[m] de comprimento engastada na face superior. Pela equação do equilíbrio, Eq. (4.9), e constatando que todas as componentes do tensor tensão serão nulas exceto σ_{yy} ,

$$\frac{\partial \sigma_{yy}}{\partial y} - \rho g = 0, \quad (4.94)$$

que por integração direta tem solução

$$\sigma_{yy} = \rho g y + cte. \quad (4.95)$$

com condição de contorno $\sigma_{yy} = 0$ em $y = 0$, implica $cte = 0$. Nesse caso, para $\rho g = 1000[N/m^2]$, $\nu = 0$, $E = 10^6[Pa]$, no engaste o valor da tensão normal ao plano y é

$$\sigma_{yy} = 1000(5) = 5000[Pa]. \quad (4.96)$$

O valor encontrado pelo programa, Fig. 4.16, é de 4930.1347 [Pa] para uma malha com 126 elementos e 4994.18 [Pa] para uma malha com 2568.

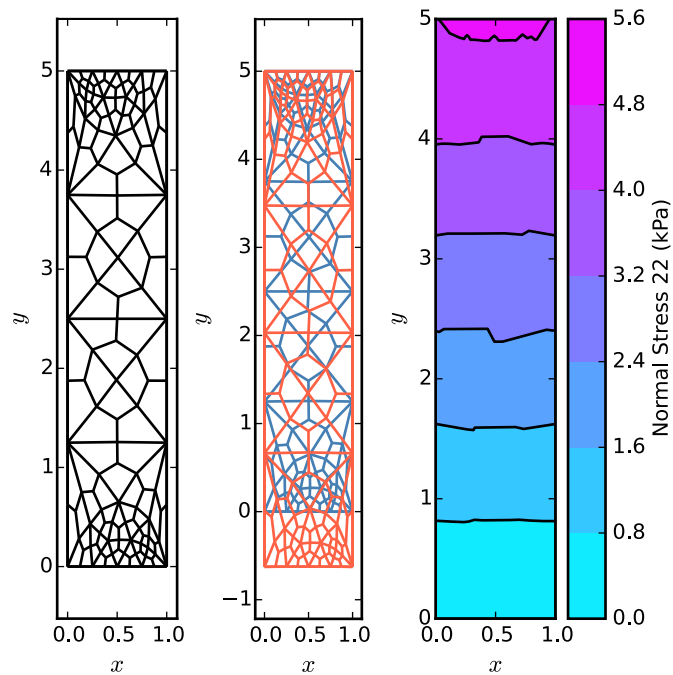


Figura 4.16 – Verificação do método que cria o vector de carregamento devido às forças de corpo.

4.14.3 Condição de Contorno com Vetor Tensão Aplicado

Para validade o método que cria o vetor de carregamento devido ao campo de vetor tensão aplicado no contorno, utiliza-se o procedimento conhecido como *patch test*, (TAYLOR et al., 1986).

Esse procedimento, como descrito em Mosalam (2011), consiste em, a partir de um campo constante de tensões no plano x , na direção x , igual a $1 [Pa]$. A geometria utilizada é a mesma utilizada para verificar a função que aplica as condições de contorno de deslocamento, um quadrado com quatro elementos. A partir desse campo de tensão, o campo de deslocamento pode ser determinado,

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} \Rightarrow \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} 1/E \\ -\nu/E \\ 0 \end{bmatrix}. \quad (4.97)$$

Para $E = 1000 [Pa]$ e $\nu = 0.3$, o campo de deformações é igual a

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} 0.001 \\ -0.0003 \\ 0 \end{bmatrix} \quad (4.98)$$

O campo de tensões constante igual a $\sigma_{xx} = 1 [Pa]$ pode ser definido como um vetor tensão distribuído no contorno à esquerda e à direita, ver Fig. 4.17. Uma vez que qualquer seção de corte feita na geometria implica numa distribuição de tensões internas unitária e positiva (tração).

O *script* a seguir mostra como esse problema pode ser especificado. Nele as forças de corpo são nulas e o vetor tensão é especificado nas linhas de contorno 1 e 3. As condições de deslocamento foram aplicadas para garantir que o corpo não desloque como corpo rígido, sem deformação.

```
1 import numpy as np
2 from elastopy.model import Build
3 from elastopy.mesh import gmsh
4 from elastopy.material import Material
5 from elastopy.solvers import statics
6 from elastopy.postprocess import plotter
7
8 mesh_file = 'patch'
9 mesh = gmsh.Parse(mesh_file)
10 model = Build(mesh)
11 material = Material(E={9: 1000}, nu={9: 0.3})
12
13
14 def b_force(x1, x2, t=1):
15     return np.array([0.0,
16                     0.0])
17
18
19 def trac_bc(x1, x2, t=1):
20     return {
```

```

21     ('line', 3): [-1.0, 0.0],
22     ('line', 1): [1.0, 0.0]}
23
24
25 def displ_bc(x1, x2):
26     return {'node', 0): [0.0, 0.0],
27            ('node', 1): ['free', 0.0]}
28
29 U, SIG = statics.solver(model, material, b_force,
30                        trac_bc, displ_bc)
31
32 plotter.model(model, ele=True, nodes_label=True,
33              ele_label=True, edges_label=True)
34 plotter.model_deformed(model, U, magf=100, ele=True)
35
36 plotter.show()

```

O resultado para os deslocamentos nodais são

```

1 print(U.round(4))
2 [ 0.      0.      0.001 -0.      0.001 -0.0003 -0.      -0.0003 0.0005
3  0.      0.001 -0.0001 0.0005 -0.0003 -0.      -0.0002 0.0004 -0.0002]

```

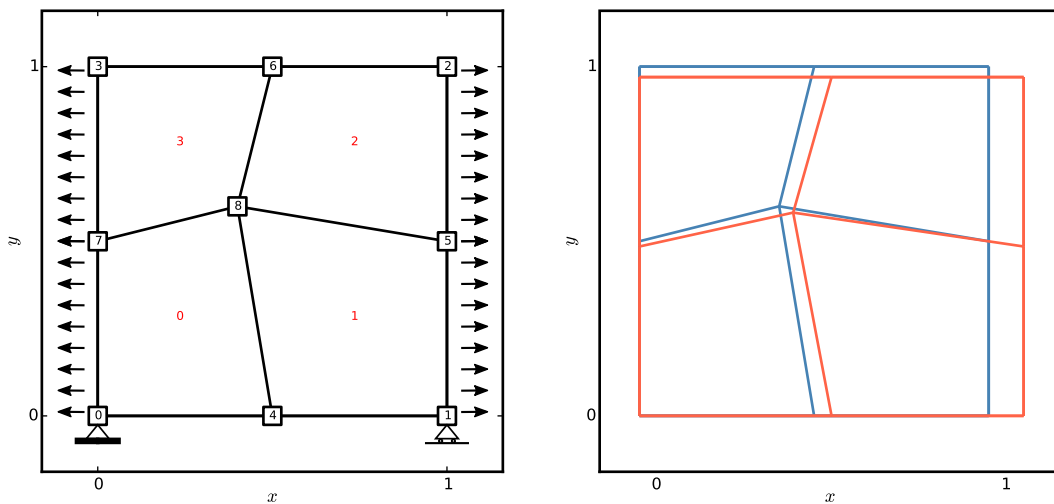


Figura 4.17 – Verificação do método que cria o vector de carregamento devido aos vetores tensão no contorno, o deslocamento possui um fator de ampliação de 100.

4.15 Exemplos

Nessa seção serão apresentados alguns exemplos a fim de demonstrar as capacidades da biblioteca desenvolvida. O arquivo de entrada, já exposto nos exemplos de verificação mostrados na seção anterior, apresenta os seguintes componentes:

1. Importação dos módulos e classes do pacote;
2. Definição do arquivo da malha;
3. Definição das propriedades do material;
4. Definição da força de corpo e das condições de contorno;
5. Chamar o *solver*;
6. Plotar os resultados;

O pacote `elastopy`, *elasticity in python*, está disponibilizado no repositório oficial PyPi, Python Package Index. Para fazer *download* do código fonte e disponibilizar o pacote para ser importado pelo Python local basta na linha de comando digitar `pip install elastopy`. Outra opção é o *download* direto através do repositório no github. Nesse caso, o código fica disponível mesmo enquanto em desenvolvimento.

4.15.1 Viga bi Apoiada

Nesse exemplo uma viga é carregada com um vetor tensão vertical igual a $-1 [N/m]$. O módulo de elasticidade é igual a $1000 [Pa]$. As condições de contorno de deslocamento foram aplicadas na linha neutra (*LN*) e na linha inferior (*Bottom*), as diferenças entre esses casos no que diz respeito as tensões normais na fibra extrema na seção de momento máximo está apresentada no gráfico abaixo. Percebe-se que a diferença é significativa para uma discretização com menos de 50 elementos.

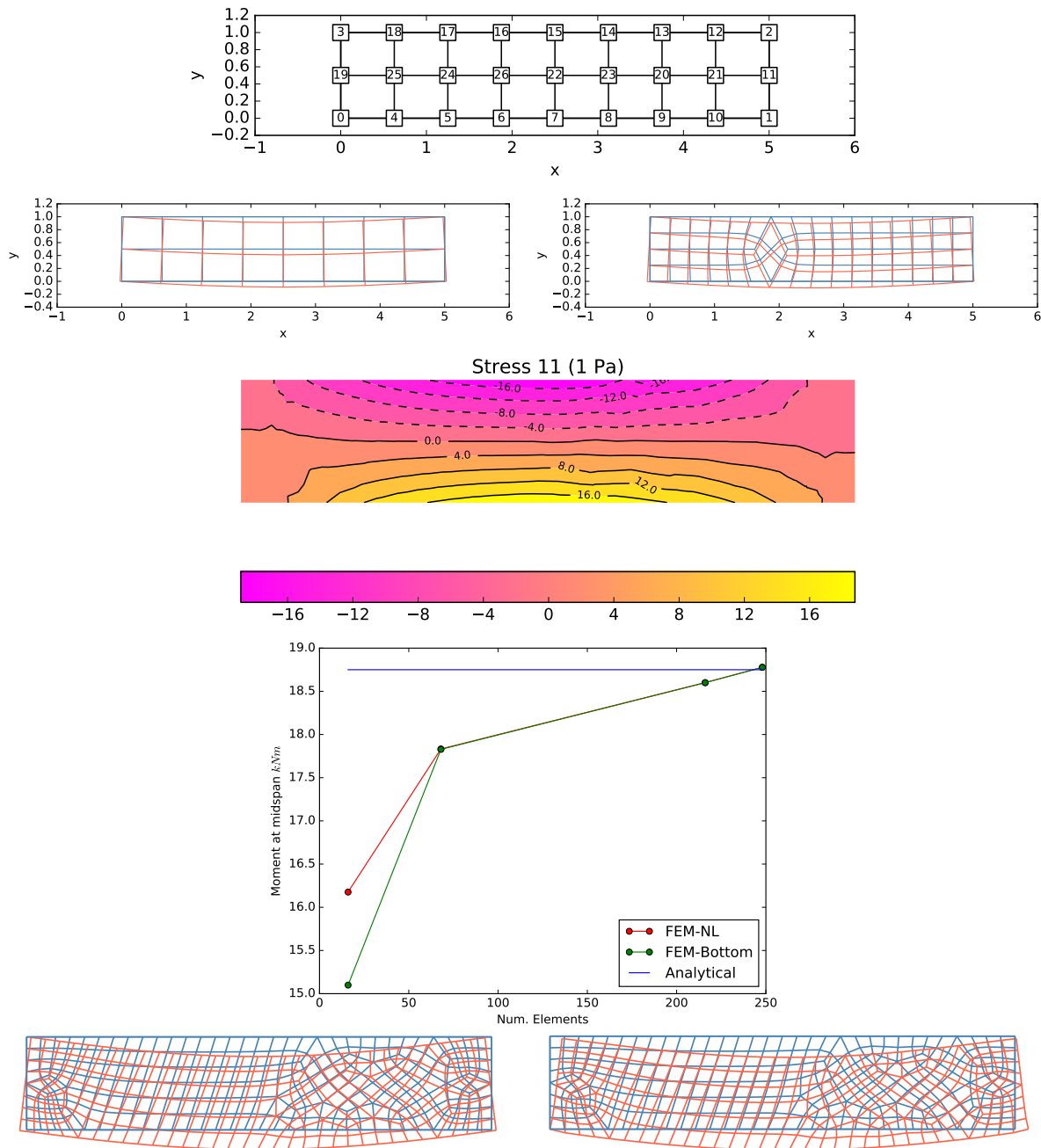


Figura 4.18 – Resultado para deslocamento e tensão normal na direção x no plano x para uma viga com condições de contorno de deslocament na linha neutra e na linha inferior; vetor tensão distribuído uniformemente na linha superior. O deslocamento foi ampliado em 2 vezes.

4.15.2 Geometria com Furo

Nesse exemplo será demonstrado a capacidade do programa em tratar geometria com furo. O domínio escolhido para o exemplo está na Fig. 4.19 e apresenta 208 elementos. As condições de contorno de deslocamento são as mínimas suficientes para não haver deslocamento de corpo rígido. As condições de vetor tensão no contorno são tensão unitária do lado esquerdo e direito apontando em sentidos opostos de forma a criar uma tensão constante unitária de tração no corpo.

O resultado para o campo de deslocamentos e para a tensão normal no plano x estão na Fig. 4.20, nele pode-se notar que a tensão $\sigma_x = 0$ no extremo do furo, em $y = 1$. O valor da tensão pode ser verificado utilizando a relação empírica mostrada nos livros de mecânica dos sólidos. Nesse caso um coeficiente é obtido por meio de um gráfico, esse coeficiente multiplica o valor da tensão nominal na seção desconsiderando o furo. O valor da tensão na seção está plotado na Fig. 4.21, o resultado obtido pelo programa se aproximou do resultado empírico, sendo que o numérico foi maior.

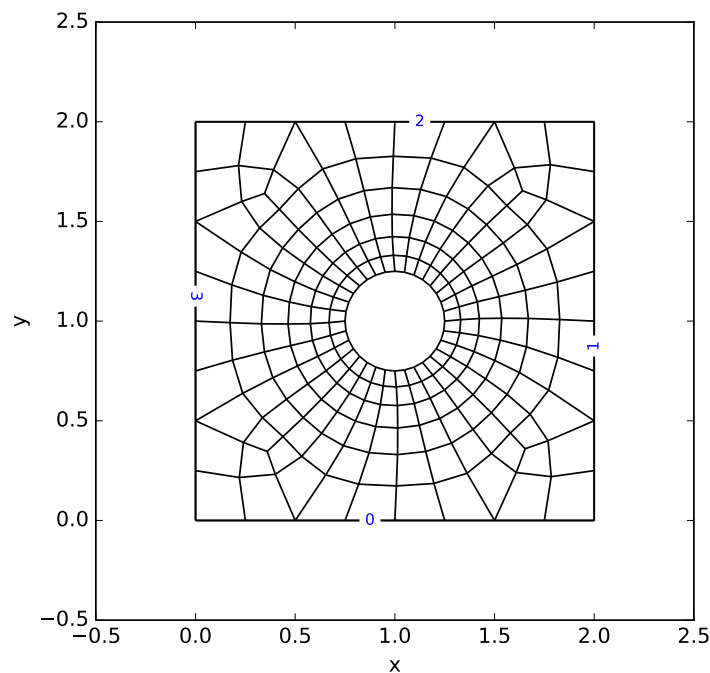


Figura 4.19 – Geometria com furo circular.

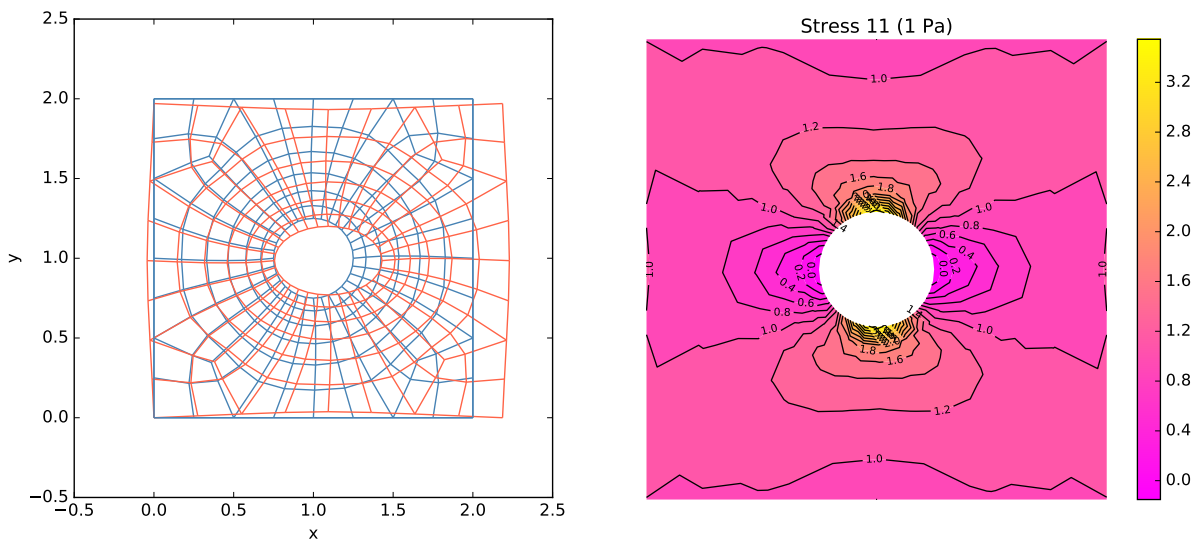


Figura 4.20 – Resultado para deslocamento e tensão normal na direção x no plano x numa malha com 208 elementos.

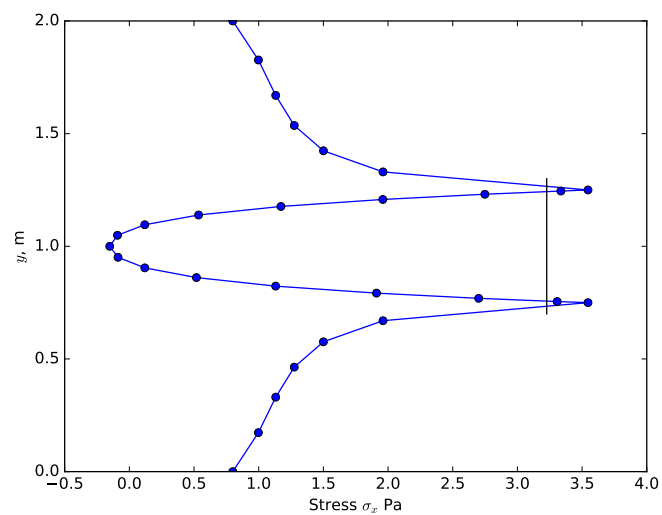


Figura 4.21 – Resultado para tensão normal na direção x no plano x na seção central que passa pelo furo e acompanhando a borda direita do furo, a linha reta na região do furo delimita o valor da tensão máxima obtida de forma empírica para um $K_t = 2,42 = \sigma_{nom}/\sigma_{max}$, sendo o $\sigma_{nom} = 1,334 Pa$.

4.15.3 Barragem Gravidade

Nesse exemplo será mostrado a capacidade do programa em lidar com geometria mais complexas e com descontinuidade nas propriedades do material. O problema consistem em uma barragem gravidade com carregamento hidrostático e peso próprio. A barragem será modelada com e sem a fundação, para se verificar os efeitos da mesma na análise. As tensões obtidas serão comparadas com tensões obtidas pelo método gravidade, que é um método analítico para se obter tensões em barragens. O resultado analítico é obtido por meio do software CADAM. A Figura 4.24 apresenta essa comparação. Na elevação à 40 m, os resultados para o modelo com fundação apresenta uma grande variação pois os resultados não fora extraídos em nós na mesma elevação mas utilizando o resultado em nós aproximadamente a 40 m.

A barragem em questão é do tipo gravidade e possui perfil convencional mostrado na Fig. 4.22. O modelo com fundação possui 1172 elementos enquanto o modelo sem fundação apresenta 972 elementos. A pressão hidrostática é aplicada na linha 8 e 9 no modelo com fundação; e na linha 6 no modelo sem fundação. As condições de contorno de deslocamento são especificadas de forma que a linha 0 tenha deslocamento x e y restringido, e nas linha 1 e 10 apenas na direção x .

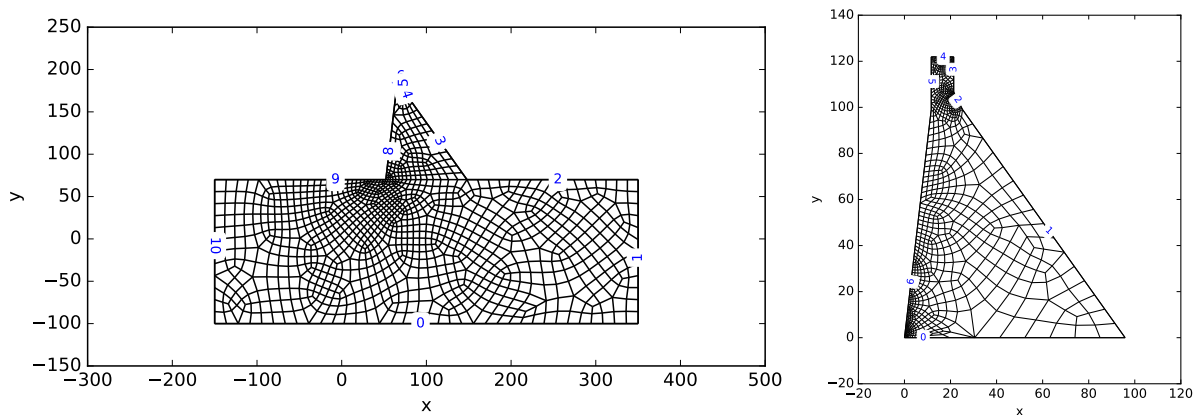


Figura 4.22 – Geometria da barragem analisada, à esquerda o modelo com a fundação e à direita o modelo sem.

As propriedades dos materiais podem ser estabelecidas por meio do seguinte código

```
1 material = Material(E={s[0]: 20e9, s[1]: 30e9}, # Pa
2                 nu={s[0]: 0.2, s[1]: 0.33},
3                 dnsty={s[0]: 2400, s[1]: 3000}) # kg/m3
```

Os resultados para o campo de deslocamento está mostrado na Fig. 4.23, devido à restrição ao deslocamento na base da barragem sem a fundação, as diferenças entre o campo de deslocamento são significativas. O resultado para o campo de tensões é

apresentado na Fig. 4.25, nele observa-se que as tensões apresentam qualitativamente a mesma distribuição, no entanto os valores do modelo sem fundação

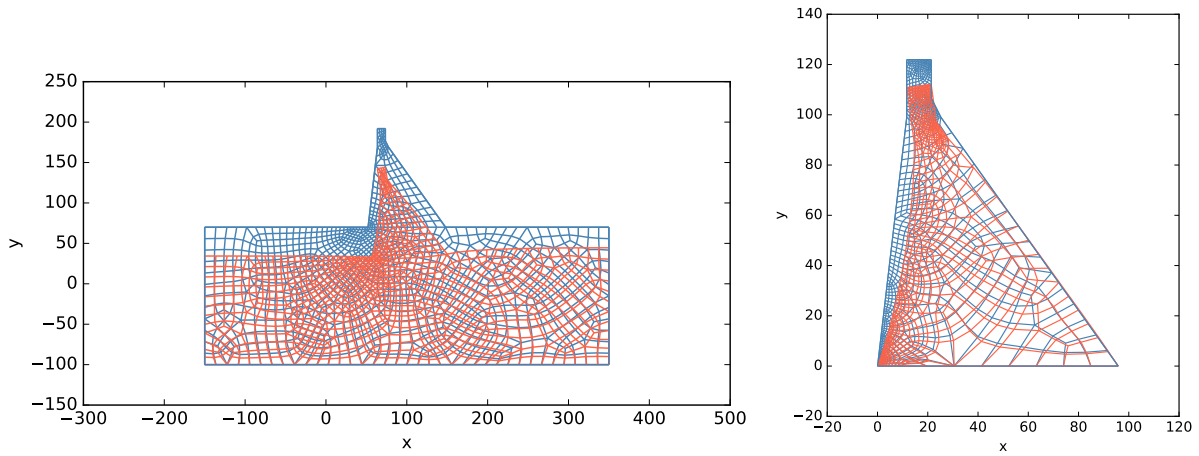


Figura 4.23 – Resultado do campo de deslocamentos ampliado 2000 vezes.

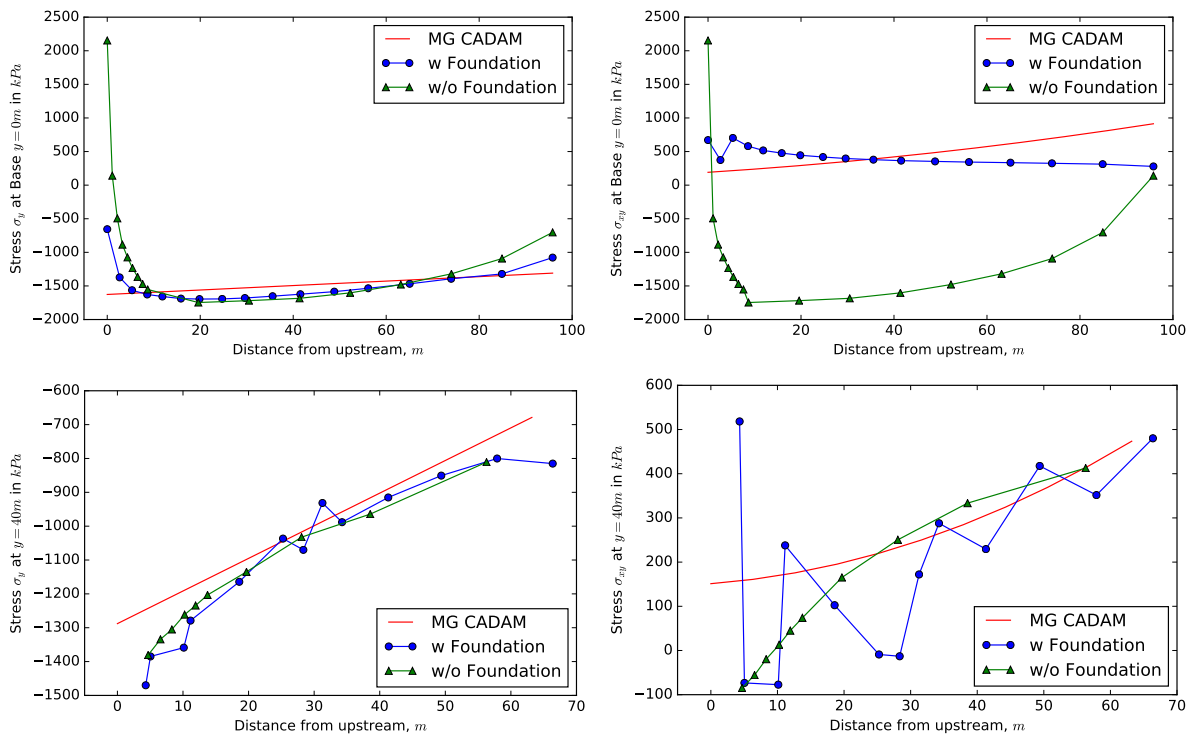


Figura 4.24 – Resultado da comparação entre as tensões, σ_x e σ_{xy} , obtidas do modelo com fundação, modelo sem fundação e pelo método analítico em duas alturas: $y = 0m$, na base e $y = 40m$.

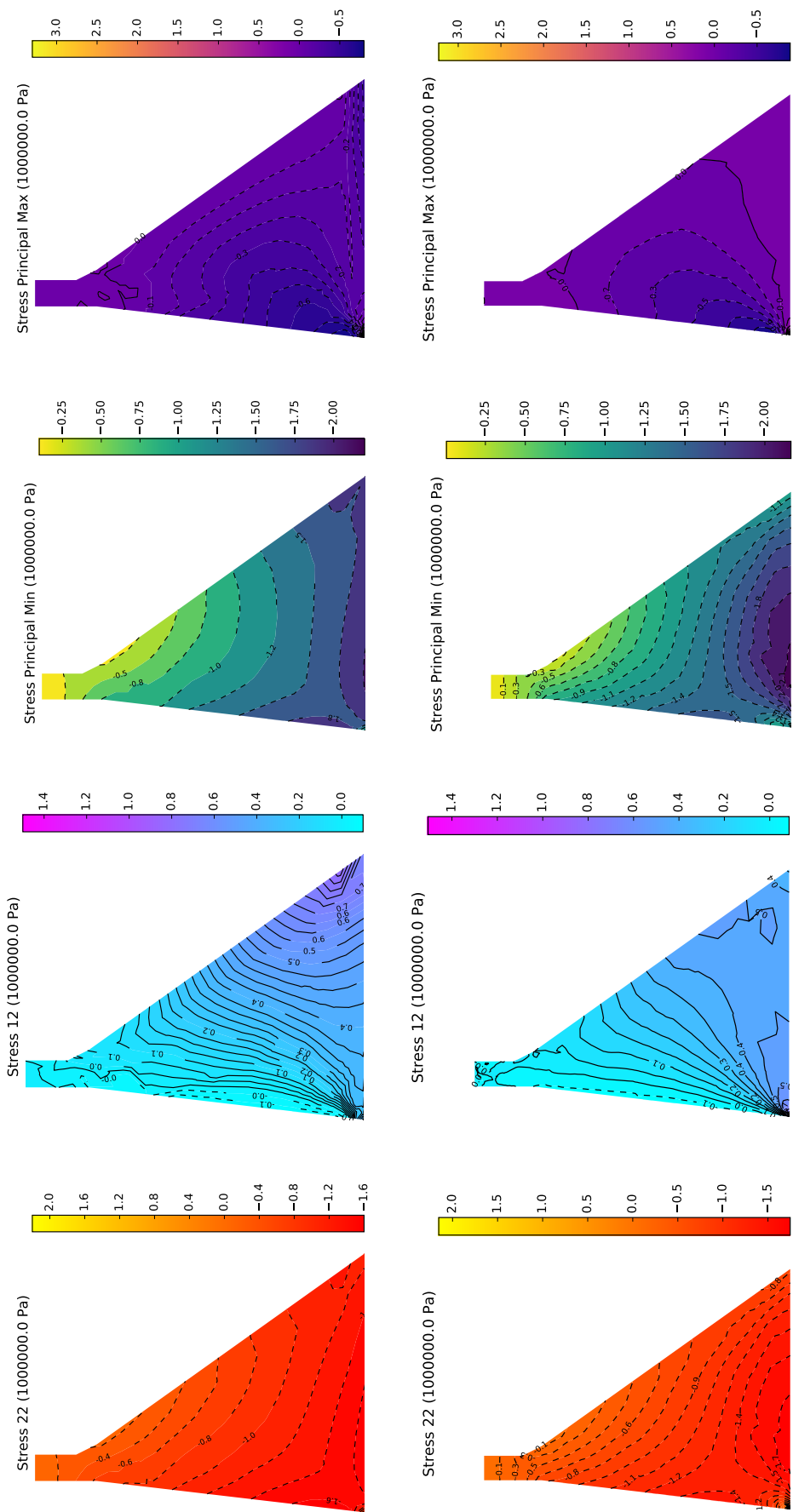


Figura 4.25 – Resultados para o campo de tensões, os campos na parte superior são do modelo com fundação e os campos na parte inferior são referentes ao modelo sem fundação.

4.16 Problema Dinâmico

A utilização do modelo matemático da elasticidade de linear é útil para simulação de problemas estáticos onde os deslocamentos são pequenos. Caso o carregamento aplicado no sistema mecânico seja aplicado de forma lenta, as forças inerciais que surgem podem ser desprezadas e o problema ser tratado de forma estática, (WILSON, 1998, p. 12-1). No entanto, se o carregamento aplicado ocorrer de forma rápida ou descontínua uma análise dinâmica deve ser realizada.

O modelo matemático para o problema mecânico elástico é obtido considerando no equilíbrio as forças de natureza inercial, ou pseudo forças. As forças inerciais decorrem do Princípio de d’Alambert, que diz “Se um sistema de referência está em movimento acelerado, o movimento produz o mesmo efeito que uma força externa, fictícia ou inercial, imposta ao sistema”, (LANCZOS, 1952, p. 100).

Nesse caso as forças inerciais, \mathbf{F}_I , são dadas por,

$$\mathbf{F}_I = \partial_t \mathbf{p} \quad (4.99)$$

onde, \mathbf{p} é o momento linear. A Figura 4.26 abaixo mostra o digrama de classificação para o problema dinâmico. A dimensão do tempo no diagrama de classificação implica numa extensão do diagrama para inclusão de um novo plano. A diferença entra o problema estático e dinâmico é a inclusão da força inercial no equilíbrio. Portanto o somatório de forças não é mais nulo e sim igual a força inercial resultante. A relação entre o momento linear \mathbf{p} e a velocidade \mathbf{v} representa uma relação constitutiva. A parte dinâmica é representada no diagrama por meio de um plano horizontal enquanto a parte estática permanece vertical, (TONTI, 2013, p. 227).

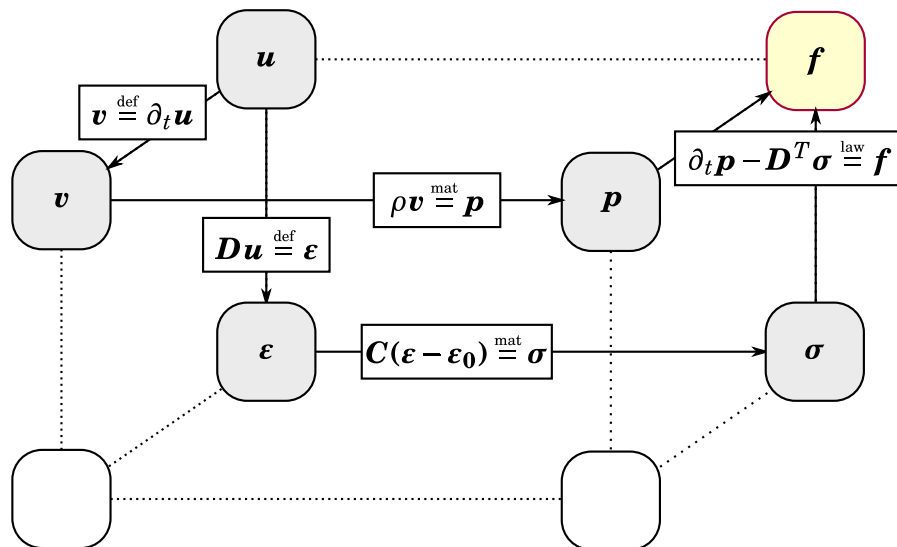


Figura 4.26 – Diagrama de classificação para o problema elástico dinâmico. (TONTI, 2013)

5 Elasticidade Linear: Estruturas

Nesse capítulo será tratado o procedimento para solução numérica de sólidos que compõem a classe de membros estruturais. Membros estruturais como vigas e pilares possuem a peculiaridade de possuir uma dimensão muito menor do que as outras e isso possibilita uma simplificação teórica. O procedimento consiste na definição da energia potencial para elementos de barras e vigas e sua posterior minimização.

5.1 Introdução

A análise estrutural, ou mecânica das estruturas, consiste num *membro teórico da mecânica dos sólidos*. A análise em si consiste na determinação da resposta da estrutura à solicitação externa (mecânica ou não-mecânica como no caso da temperatura). Dado um conjunto de forças agindo nos nós e os deslocamentos prescritos, ou seja, condições de contorno, objetiva-se determinar os deslocamentos nodais e as reações de apoio.

As estruturas são compostas por membros ligados entre si. Os *membros estruturais*, ou elementos, podem ser lineares, planos ou sólidos tridimensionais. Vigas e pilares por apresentarem uma dimensão predominante sobre a outra podem ser encaixados dentro do primeiro grupo, elementos lineares. As lajes são modeladas como elementos planos ou pelo modelo de grelha onde uma malha de barras simula o comportamento de uma placa contínua. A terceira categoria, de sólidos em três dimensões, engloba todos os elementos estruturais que não podem ser modelados como elemento linear ou plano.

Na realidade, todos os elementos estruturas são sólidos tridimensionais, no entanto, a simplificação da idealização de certos elementos permite o desenvolvimento de uma teoria prática e conveniente ao uso.

As equações da elasticidade linear para os sólidos, discutidas no Capítulo 4, são gerais, portanto, estruturas incluem-se nessa modelagem de sólidos. No entanto, a abordagem em estruturas discretizadas por *elementos estruturais* conta com a solução destas equações em cada elemento individualmente como função dos valores do contorno, ou extremidades. Esses valores são então compatibilizados nos pontos onde os elementos se conectam, nos nós. Essa compatibilização gera um conjunto de equações algébricas que

podem ser resolvidas, (PRZEMIENIECKI, 1985).

5.1.1 Modelo estrutural

O modelo estrutural, ou idealização estrutural, usado para representar a realidade consiste num conjunto de elementos ligados entre si por meio de nós, trata-se de uma discretização da estrutura real em elementos discretos. O processo de discretização consiste em reduzir o número de graus de liberdade para um valor finito.

Os graus de liberdade, *degrees of freedom*, ou *dof*, no contexto estrutural são os possíveis movimentos que um nó pode ter. Em duas dimensões, um nó pode transladar na direção x e y e rotacionar em torno do eixo z , portanto, 3 *dof*. Em 3 dimensões o nó pode transladar nos três eixos e rotacionar em torno dos três eixos, portanto 6 *dof*.

A convenção de sinal adotada é:

- Translação e forças na direção do eixo são positivas;
- Rotação e momentos são positivas quando, utilizando a regra da mão direita, o vetor aponta no sentido positivo do eixo. Em 2d, uma rotação é positiva quando gira no sentido anti-horário.

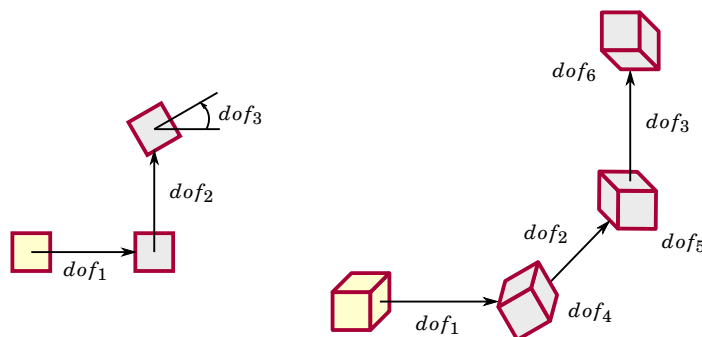


Figura 5.1 – Graus de liberdade de um nó no plano e no espaço.

5.2 Discretização

Discretização é o processo responsável pela subdivisão do domínio de forma a reduzir o número de *dof*. No contexto da mecânica das estruturas a discretização ocorre na divisão da estruturas em *elementos estruturais*. Os elementos podem ser: barra, viga, pórtico e suas variações espaciais. A diferença de cada elemento é devido ao tipo de aproximação feita para a variável primária, deslocamento u . Num elemento de barra, por exemplo, a função deve ser contínua no elemento e satisfazer as condições de contorno

no suporte, portanto, uma função linear já é adequada. Já num elemento de viga, o requerimento de continuidade aumenta, a derivada da função também deve ser contínua.

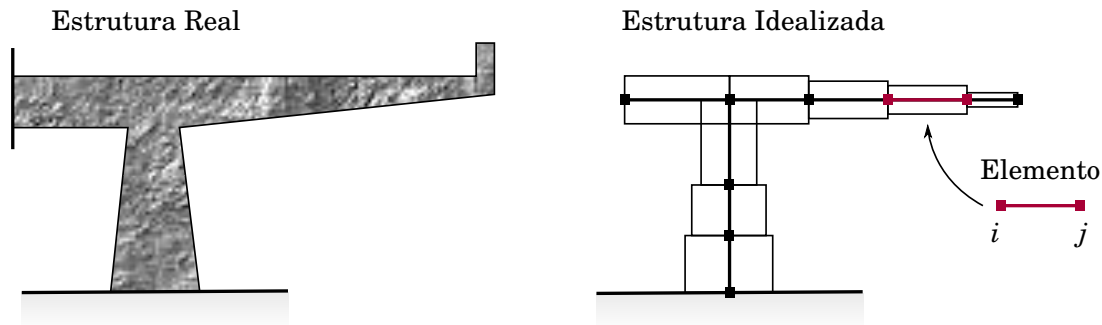


Figura 5.2 – Discretização de um elemento estrutural.

A formulação do elemento pode ser feita através de uma abordagem puramente mecânica, ou seja, mecânica dos sólidos, como em Felippou (2013), ou utilizando a abordagem variacional (princípios de minimização), Felippa (2004) e Rao (2013). A primeira é interessante para apreciação do problema e introdução à metodologia de análise geral.

5.3 Formulação por Métodos Variacionais

A metodologia por métodos variacionais consiste em obtenção de uma solução aproximada que minimize um funcional quadrático, no caso, a energia potencial total. A metodologia possui a seguinte sequência de análise:

1. Definir o funcional quadrática associado ao problema físico. No caso mecânico, esse potencial é dado pela energia potencial total.
2. Definir o conjunto admissível para a solução da variável primária. Um conjunto admissível é aquele que satisfaz as condições cinemáticas, ou sejam, para o campo de deslocamentos $u(x)$:

- Deve ser contínuo ao longo do comprimento, ou seja, continuamente diferenciável,

$$u(x) \in C^0 \quad \text{em } x \in [0, L], \quad (5.1)$$

- Deve satisfazer as condições de contorno naturais, de deslocamento.

3. Invocar o princípio de minimização da energia potencial total que diz que a solução do problema, u^* é aquela que faz o potencial estacionário,

$$\delta\Pi = 0, \quad \text{se } u(x) = u^*. \quad (5.2)$$

4. Discretizar o problema. Nesse ponto o método dos elementos finitos tem início. Utiliza-se do fato de que a integral no domínio pode ser dividida pela soma da integral em subdomínios, no caso, os *elementos finitos*,

$$\Pi \equiv \sum_{e=1}^{ne} \Pi^e, \quad (5.3)$$

5. Assume-se uma solução aproximada, $u \approx \bar{u}$. Nesse ponto defini-se *qual o tipo de elemento* é necessário para melhor descrever o problema. Esse trabalho limita-se, no que tange à mecânica das estruturas, a elementos de barra, viga e pórtico que é a superposição dos dois primeiros.

5.3.1 Energia Potencial Total

O funcional da energia potencial total é dado por,

$$\Pi \stackrel{\text{def}}{=} U - W, \quad (5.4)$$

onde, U é a escalar que denota a energia interna das deformações elásticas e W denota a energia desprendida do trabalho de forças externas. A energia de deformação elástica interna pode ser obtida através da soma da energia em cada ponto material pela relação,

$$U = \int_V \mathcal{U} dV, \quad (5.5)$$

onde \mathcal{U} é a *densidade de energia de deformação*, dada por $\mathcal{U} = \frac{1}{2}\sigma\varepsilon$, o que representa a área abaixo da curva tensão-deformação de materiais elásticos, Figura 5.3.

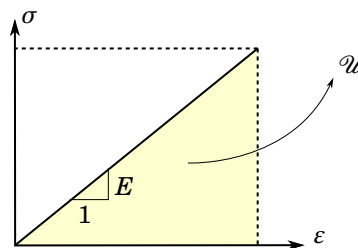


Figura 5.3 – Curva tensão-deformação e área abaixo do gráfico que indica a densidade de energia de deformação.

Integrando o valor da densidade de energia de deformação ao longo da seção transversal obtém-se

$$U = \int_V \frac{1}{2}\sigma\varepsilon dV. \quad (5.6)$$

O trabalho da força externa é igual ao produto da força externa pelo deslocamento na direção da força. Ou seja, para uma força transversal w e deflexão v , num elemento de viga

$$W = \int_0^L wv dx. \quad (5.7)$$

5.3.2 Princípio de Minimização da Energia Potencial Total

A solução de deslocamento que satisfaz as equações governantes fazem com que o potencial seja estacionário.

Após o processo de discretização prossegue-se a análise para resolver o problema mecânico por meio da minimização da energia potencial total.

A energia potencial total para cada elemento pode ser escrita como,

$$\Pi^e = U^e - W^e, \quad (5.8)$$

a energia proveniente das deformações elásticas e o trabalho da força externa são dados de forma geral matricial por,

$$U^e = \frac{1}{2}(\mathbf{u}^e)^T \mathbf{K}^e \mathbf{u}^e \quad (5.9)$$

$$W^e = (\mathbf{u}^e)^T \mathbf{p}^e, \quad (5.10)$$

onde, \mathbf{K}^e é a matriz de rigidez elementar mostra para o elemento de barra e de viga nas seções seguintes. O vetor \mathbf{p}^e contém os carregamentos nodais. A energia de deformação elástica e o trabalho da força externa são funções do deslocamento. Isso ocorre pois no processo de construção dessas quantidades optou-se por substituir as tensões pelos deslocamentos. Nesse caso, esse procedimento é chamado de método dos deslocamentos, (RAO, 2011, p. 296).

A variação da energia potencial total com respeito ao deslocamento,

$$\begin{aligned} 0 &= \delta \Pi^e \\ &= (\delta \mathbf{u}^e)^T \frac{\partial \Pi^e}{\partial \mathbf{u}^e} \\ &= (\delta \mathbf{u}^e)^T [\mathbf{K}^e \mathbf{u}^e - \mathbf{f}^e] = 0. \end{aligned} \quad (5.11)$$

como a variação é arbitrária, para a equação ser nula, a expressão entre chaves deve ser nula.

$$\mathbf{K}^e \mathbf{u}^e = \mathbf{f}^e. \quad (5.12)$$

Essa expressão pode ser entendida considerando $\delta \mathbf{u}^e$ como um deslocamento virtual e a igualdade como sendo decorrente do teorema dos trabalhos virtuais.

5.4 Elemento de Barra

O elemento de barra é um elemento unidimensional, ou seja, a dimensão longitudinal é predominante com relação às dimensões na seção. Esse elemento resiste apenas à esforços axiais. Ele é utilizado na modelagem de treliças, cabos, correntes e cordas, (FELIPPA, 2004).

5.4.1 Esforço Interno Axial

O elemento de barra apresenta apenas *esforço interno axial*. A distribuição de tensões axiais, ou normais, σ_α na seção pode ser substituída por uma resultante equivalente, através da integração na área, denominada força normal \mathcal{N} , ou axial. Por meio de um corte no elemento, relaciona-se a força normal na seção com a força de extremidade independente. Cada uma das extremidades da barra apresenta uma força axial, no entanto, como o elemento deve estar em equilíbrio estático (caso contrário estaria se deslocando como um corpo rígido) essas forças são iguais e em sentidos opostos.

5.4.2 Deformação Axial

No elemento de barra existe apenas deformação axial. A deformação axial é, por definição teórica,

$$v_1 = \int_0^L \varepsilon_\alpha(x) dx. \quad (5.13)$$

Caso a deformação específica ε_α for constante ao longo do elemento a deformação $v_1 = \varepsilon_\alpha L$. Como a deformação específica axial é calculada a partir da uma variação no comprimento, $\varepsilon_\alpha = \Delta L/L$, tem-se que $v_1 = \Delta L$, ou seja, o alongamento ou encurtamento do elemento.

5.4.3 Diagrama de Tonti: Modelo de Barra

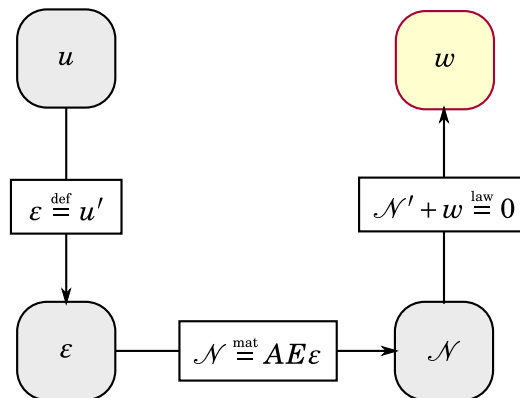


Figura 5.4 – Diagrama de Tonti para o modelo matemático de barra. (FELIPPA, 2005)

5.4.4 EPT Elemento de Barra

Energia potencial total (EPT) para o elemento de barra.

$$\begin{aligned}
 U^e &= \frac{1}{2} \int_V \sigma \varepsilon dV \\
 &= \frac{1}{2} \int_V (E\varepsilon)\varepsilon dV \\
 &= \frac{1}{2} \int_x (EA\varepsilon)\varepsilon dx \\
 &= \frac{1}{2} \int_x u' EA u' dx
 \end{aligned} \tag{5.14}$$

Como a EPT apresenta a primeira derivada da variável primária, deslocamento u , implica que os deslocamentos devem ser contínuos, ou seja, apresentarem primeira derivada definida. O número de derivadas é conhecido com *índice variacional*. Além de serem contínuas os deslocamentos devem também satisfazer as condições de contorno de deslocamento, naturais. A partir dessas condições, define-se o conjunto de funções admissíveis para a solução. As funções de aproximação, Rayleigh-Ritz, também devem satisfazer essas condições, no entanto, formam um conjunto *finito*. Para esse elemento, uma solução na forma de combinação linear de *polinômios lineares*, N_1 e N_2 , é suficiente para satisfazer as condições impostas,

$$u^e(x) = N_{u1}^e u_1^e + N_{u2}^e u_2^e = \begin{bmatrix} N_{u1}^e & N_{u2}^e \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} = \mathbf{N}^e \mathbf{u}^e, \tag{5.15}$$

Os subscritos nas variáveis primárias de deslocamentos, u , indicam o índice correspondentes aos graus de liberdade, nesse caso, há apenas 1 grau de liberdade por nó. A Figura 5.5 mostra a interpretação da aproximação da solução por funções de forma N . As funções interpoladoras são dadas por

$$\begin{aligned}
 N_{u1} &= 1 - \xi, \\
 N_{u2} &= \xi,
 \end{aligned} \tag{5.16}$$

com ξ sendo uma coordenada adimensional, *coordenada natural*, pelo qual é possível padronizar o elemento como sendo um de 1 unidade de comprimento, ou seja, $\xi \in [0, 1]$. Nesse caso,

$$\xi(x) = \frac{x - x_1}{L} \tag{5.17}$$

onde x_1 é a coordenada do nó 1, ou nó na extremidade esquerda.

Para um elemento de barra, as relações cinemáticas, ou deformação-deslocamento,

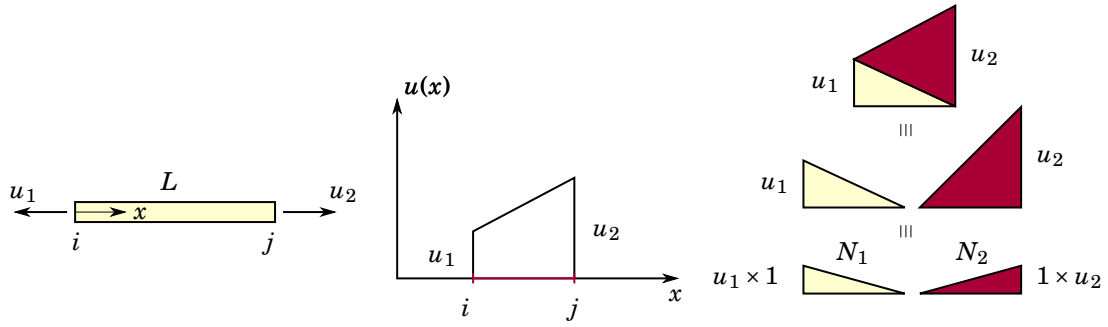


Figura 5.5 – Graus de liberdade em um elemento de barra e funções de forma linear para aproximação. (FELIPPA, 2004)

podem ser escrita utilizando a solução na forma aproximada, $u(x) = \mathbf{N}^e \mathbf{u}^e$,

$$\begin{aligned}
 \varepsilon &= \frac{du^e}{dx} \\
 &= \left[\frac{dN_1^e}{dx} \quad \frac{dN_2^e}{dx} \right] \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} \\
 &= \frac{1}{L} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} \\
 &= \mathbf{B} \mathbf{u}^e,
 \end{aligned} \tag{5.18}$$

onde \mathbf{A} é a matriz cinemática, ou matriz de deformação-deslocamentos. Como as funções de forma N são lineares num elemento, a matriz cinemática possui termos constantes.

Utilizando a função de forma para aproximar a solução, a energia de deformação elástica num elemento de barra pode ser obtida por

$$\begin{aligned}
 U^e &= \frac{1}{2} \int_x \varepsilon(EA)\varepsilon dx \\
 &= \frac{1}{2} \int_x (\mathbf{B} \mathbf{u}^e)^T (EA) \mathbf{B} \mathbf{u}^e dx \\
 &= \frac{1}{2} (\mathbf{u}^e)^T \mathbf{K}^e \mathbf{u}^e.
 \end{aligned} \tag{5.19}$$

o termo transposto surgiu pois é a forma explícita de um produto interno entre vetores. Nesse ponto utilizou-se a hipótese de que o elemento é *prismático*, por tanto, a rigidez axial, EA , é constante. De forma geral a matriz de rigidez elementar, \mathbf{K}^e , pode ser obtida por

$$\mathbf{K}^e = \int_x (\mathbf{B})^T (EA) \mathbf{B} dx. \tag{5.20}$$

5.5 Elemento de Viga

Vigas são elementos estruturas unidimensionais responsáveis por sustentar esforços transversais. O mecanismo responsável por proporcionar essa sustentação é o da

flexão, a barra quando flexiona apresenta uma distribuição de tensões normais linearmente distribuída em sua seção transversal. A distribuição de tensões ocorre de forma a produzir uma região tracionada e outra comprimida. Essas tensões geram resultantes na seção, através da integral. A resultante mais importante é o momento fletor,

$$\mathcal{M} = \int_A y\sigma \, dA \quad (5.21)$$

O momento na seção é o principal mecanismo pelo qual o carregamento é transportado para os apoios, (FELIPPA, 2004).

5.5.1 Hipóteses da Teoria de Vigas

O modelo matemático que descreve o comportamento de elementos estruturais sujeitos à carregamento transversal mais utilizado é a teoria de *Euler-Bernoulli*. Essa teoria clássica apresenta uma série de hipóteses,

- Simetria planar: O eixo longitudinal da peça é reto e as seções possuem um eixo longitudinal de simetria.
- Variação da seção transversal: A peça pode ser prismática, sem variação da geometria da seção, ou variar continuamente.
- Seções transversais planas, ortogonais ao eixo longitudinal, permanecem planas e normais após flexão. Isso implica em

$$u = -y \frac{\partial v}{\partial x} = -yv' = -y\theta, \quad (5.22)$$

onde, u e v são os deslocamentos axial e transversal, v' é a tangente à deflexão, representada também pela rotação θ . A Figura 5.6 mostra visualmente essa relação. Essa hipótese também implica que a cinemática da seção pode ser descrita por apenas dois parâmetros independentes: deformação axial, ε_a , e a curvatura, κ .

- Energia de deformação: A energia interna de deformação considera apenas deformações devido à flexão, todas as outras contribuições são desprezadas.
- Linearidade: As deflexões transversais, rotações e deformações são consideradas pequenas de forma que o erro de truncamento na série de Taylor seja desprezível.
- Modelo material é assumido como *elástico e isotrópico*.

5.5.2 Diagrama de Tonti: Modelo de Viga

O diagrama de classificação para o modelo matemático da viga de Euler-Bernoulli é mostrado na Fig. 5.7.

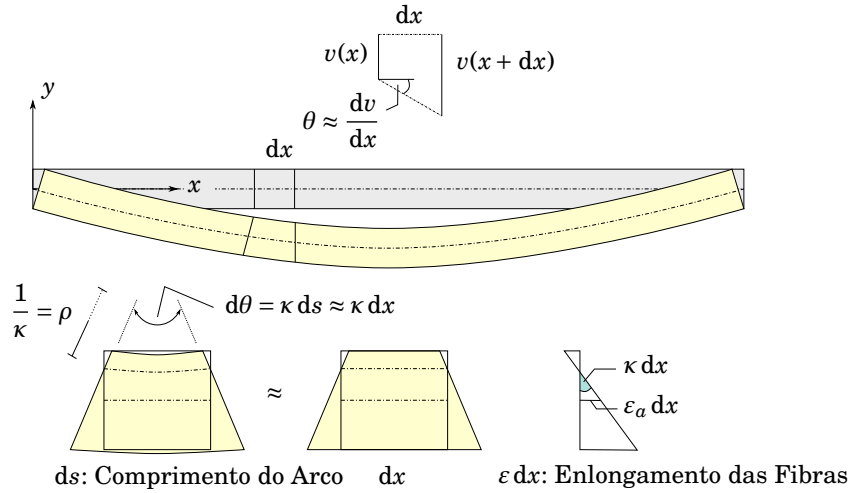


Figura 5.6 – Interpretação da relação aproximada da deflexão, v , com curvatura, $\kappa = 1/\rho$. Dentre as aproximações estão: $\kappa \approx d^2v/dx^2$, $\theta \approx \tan(\theta)$ e $ds \approx dx$. (FELIPPOU, 2013)

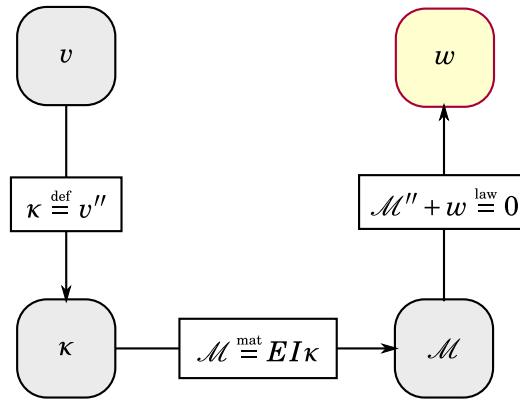


Figura 5.7 – Diagrama de Tonti para o modelo matemático de viga. (FELIPPA, 2005)

5.5.3 EPT Elemento de Viga

O modelo para vigas de Euler-Bernoulli, discutido na seção 5.5.1, gera funções de aproximação do tipo *polinômios de Hermite*. Esse modelo negligência deformações devido ao cisalhamento no cálculo da energia interna, (FELIPPA, 2004). Portanto, a energia de deformação interna é devido a apenas deformações de flexão e tensões normais.

$$\begin{aligned}
 U^e &= \frac{1}{2} \int_V \sigma \epsilon dV \\
 &= \frac{1}{2} \int_V E(\kappa y)^2 dV \\
 &= \frac{1}{2} \int_x EI \kappa^2 dx \\
 &= \frac{1}{2} \int_x v'' EI v'' dx.
 \end{aligned}
 \tag{5.23}$$

utilizou-se $\epsilon = \kappa y$, proveniente das relações do arco de circunferência com o ângulo do arco e da curvatura. Para satisfazer os requerimentos do funcional, de índice 2, a função v de

deslocamento deve ser contínua e ter sua derivada também contínua, diz-se C^1 contínua. Então, os deslocamentos v e rotações $v' = \theta$ devem ser contínuos ao longo do elemento.

Com o objetivo de se aproximar a solução, por Rayleigh-Ritz, procura-se no conjunto de soluções admissíveis um subconjunto finito contido nele. Um conjunto de soluções possíveis pode ser obtido por polinômios cúbicos, uma família em particular é conveniente ao problema: *polinômios cúbicos de Hermite*. Esses polinômios interpoladores são convenientes pois podem ser construídos por 4 parâmetros, deslocamentos nas extremidades e tangente nas extremidades, ou seja, parâmetros que definem o movimento da viga.

Como é necessário quatro parâmetros para definir o polinômio interpolador, o elemento possui 4 graus de liberdade a serem determinados, ou seja, dois graus de liberdade por nó, deslocamento e rotação em cada nó. A solução aproximada, então, pode ser definida como sendo uma combinação linear dos polinômios cúbicos de Hermite,

$$v^e = \begin{bmatrix} N_{v1}^e & N_{\theta1}^e & N_{v2}^e & N_{\theta2}^e \end{bmatrix} \begin{bmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{bmatrix} = \mathbf{N}^e \mathbf{u}^e \quad (5.24)$$

As funções interpoladores são dadas por

$$\begin{aligned} N_{v1} &= \frac{1}{4}(1 - \xi)^2(2 + \xi), \\ N_{\theta1} &= \frac{1}{8}L(1 - \xi)^2(1 + \xi), \\ N_{v2} &= \frac{1}{4}(1 + \xi)^2(2 - \xi), \\ N_{\theta2} &= -\frac{1}{8}L(1 + \xi)^2(1 - \xi), \end{aligned} \quad (5.25)$$

onde, ξ é a coordenada natural adimensional de um elemento padrão normalizado. Esse elemento é definido como de comprimento de uma unidade seu centro está situado na origem, $\xi = 0$. Dessa forma, a posição da extremidade a esquerda de um elemento, na coordenada local 0, vale $\xi = -1$ na coordenada natural, e a extremidade à direita, coordenada L, vale $\xi = 1$. A relação entre as coordenadas natural e cartesianas é dada por,

$$\xi(x) = \frac{2x}{L} - 1 \quad (5.26)$$

As relações cinemáticas para o elemento de viga são dadas por meio relação entre

a curvatura do elemento e os deslocamentos nodais,

$$\begin{aligned}
\kappa &= \frac{d^2 v^e(x)}{dx^2} \\
&= \frac{d^2 v^e(\xi)}{d\xi^2} \left(\frac{d\xi}{dx} \right)^2 + \frac{dv^e(\xi)}{d\xi} \frac{d^2 \xi}{dx^2} \\
&= \frac{4}{L^2} \frac{d^2 v^e(\xi)}{d\xi^2} \\
&= \frac{4}{L^2} \frac{d^2 \mathbf{N}^e}{d\xi^2} \mathbf{u}^e \\
&= \mathbf{B} \mathbf{u}^e,
\end{aligned} \tag{5.27}$$

onde foi utilizado a regra da cadeia na mudança de variável e regra da derivada de produtos. A matriz cinemática é obtida pela segunda derivada das funções de forma em (5.25) e é dada por,

$$\mathbf{B} = \frac{1}{L} \begin{bmatrix} \frac{6\xi}{L} & 3\xi - 1 & \frac{-6\xi}{L} & 3\xi + 1 \end{bmatrix}. \tag{5.28}$$

Utilizando a forma aproximada da solução, a energia de deformação elástica no elemento de viga assume a forma

$$\begin{aligned}
U^e &= \frac{1}{2} \int_x \kappa (EI) \kappa \, dx \\
&= \frac{1}{2} \int_x (\mathbf{B} \mathbf{u}^e)^T (EI) \mathbf{B} \mathbf{u}^e \, dx \\
&= \frac{1}{2} \int_x (\mathbf{u}^e)^T \mathbf{B}^T (EI) \mathbf{B} \mathbf{u}^e \, dx \\
&= \frac{1}{2} (\mathbf{u}^e)^T \mathbf{K}^e \mathbf{u}^e,
\end{aligned} \tag{5.29}$$

onde, a matriz de rigidez elementar é dada por

$$\mathbf{K}^e = \int_x \mathbf{B}^T (EI) \mathbf{B} \, dx. \tag{5.30}$$

Como a matriz cinemática, \mathbf{B} , possui componentes em função das coordenadas naturais, ξ , faz-se uma mudança de variáveis na relação integral em (5.30). A matriz de rigidez pode

ser obtida por,

$$\begin{aligned}
\mathbf{K}^e &= EI \int_x \mathbf{B}^T \mathbf{B} dx \\
&= EI \int_{-1}^1 \mathbf{B}^T \mathbf{B} \frac{dx}{d\xi} d\xi \\
&= EI \int_{-1}^1 \mathbf{B}^T \mathbf{B} \frac{L}{2} d\xi \\
&= EI \frac{L}{2} \int_{-1}^1 \begin{bmatrix} \frac{6\xi}{L} \\ 3\xi - 1 \\ -\frac{6\xi}{L} \\ 3\xi + 1 \end{bmatrix} \begin{bmatrix} \frac{6\xi}{L} & 3\xi - 1 & -\frac{6\xi}{L} & 3\xi + 1 \end{bmatrix} d\xi \\
&= \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ & 4L^2 & -6L & 2L^2 \\ & & 12 & -6L \\ \text{sim.} & & & 4L^2 \end{bmatrix}.
\end{aligned} \tag{5.31}$$

Essa matriz pode também ser obtida por meio da formulação por Mecânica do Sólidos de forma mais intuitiva, utilizando apenas equilíbrio de forças nodais. Notar que a matriz de rigidez elementar depende do comprimento do elemento, L , de propriedades da seção transversal, I , e de propriedades do material E .

O trabalho de uma carga externa distribuída (considerada uniforme) é dado por,

$$\begin{aligned}
W^e &= \int_0^L wv^e dx \\
&= \int_0^L w\mathbf{N}^e \mathbf{u}^e dx \\
&= \int_{-1}^1 w\mathbf{N}^e \mathbf{u}^e \frac{dx}{d\xi} d\xi \\
&= \frac{1}{2}wL \int_{-1}^1 \mathbf{N}^e \mathbf{u}^e d\xi \\
&= (\mathbf{u}^e)^T \mathbf{p}^e.
\end{aligned} \tag{5.32}$$

onde foi utilizado a igualdade $\mathbf{N}\mathbf{u} = \mathbf{u}^T \mathbf{N}^T$. O vetor \mathbf{p}^e é dado por

$$\begin{aligned}
\mathbf{p}^e &= \frac{1}{2}wL \int_{-1}^1 (\mathbf{N}^e)^T d\xi \\
&= \frac{1}{2}wL \int_{-1}^1 \begin{bmatrix} \frac{1}{4}(1-\xi)^2(2+\xi) \\ \frac{1}{8}L(1-\xi)^2(1+\xi) \\ \frac{1}{4}(1+\xi)^2(2-\xi) \\ -\frac{1}{8}L(1+\xi)^2(1-\xi) \end{bmatrix} d\xi = \frac{1}{2}wL \begin{bmatrix} 1 \\ \frac{1}{6}L \\ 1 \\ -\frac{1}{6}L \end{bmatrix}.
\end{aligned} \tag{5.33}$$

Esse mesmo resultado é obtido na formulação por mecânica dos sólidos onde foi calculado a curvatura gerada pelo esforço distribuído, as respectivas deformações elementares e através das relações constitutivas os esforços internos gerados.

5.6 Procedimento de Solução

O procedimento de solução do problema estrutural consiste em montar as matrizes elementares, montar a matriz global por meio da incidência da cada elemento nos respectivos graus de liberdade, montar os vetores de carregamento nodal e carregamento equivalente para cargas distribuídas e finalmente resolver o sistema linear descrito em 5.12

5.7 Verificação

A verificação será feita através de um exemplo de uma viga biapoiada com balanço. O carregamento da ocorre nos dois vãos à direita e valer $w = 5kN/m$. No início do balanço o momento pode ser calculado passando a resultante do balanço para o apoio,

$$\mathcal{M}(20) = 5 \cdot 5 \cdot \frac{5}{2} = 62.5kNm \quad (5.34)$$

tracionando em cima.

```

1  from sapy import displmethod
2  from sapy import element
3  from sapy import gmsh
4  from sapy import structure
5  from sapy import plotter
6  import matplotlib.pyplot as plt
7
8
9  mesh_file = 'patch4'
10
11 mesh = gmsh.Parse(mesh_file)
12
13 ele = element.Data()
14 for i in range(len(mesh.con)):
15     ele.E[i] = 1000          # MPa
16     ele.I[i] = 833.3        # mm4
17     ele.A[i] = 100          # mm2
18     ele.TYPE[i] = '2dFrame'
19
20 bound = {0: [1, 1, 0],
21           2: [0, 1, 0]}
22
23 nodal_load = {}
24
25 ele.w[1] = [0, -5]          # kN/m
26 ele.w[2] = [0, -5]
27
28 model = structure.Builder(mesh, ele, bound, nodal_load)
29
30 U, Q, V = displmethod.solver(mesh, model, ele, nodal_load)
31

```

```

32 plotter.undeformed(model, axis='off')
33 plotter.deformed(model, U, V, ele, magf=800, axis='off')
34 plotter.moments(model, Q, ele, magf=0.05, axis='off')
35 plt.show()

```

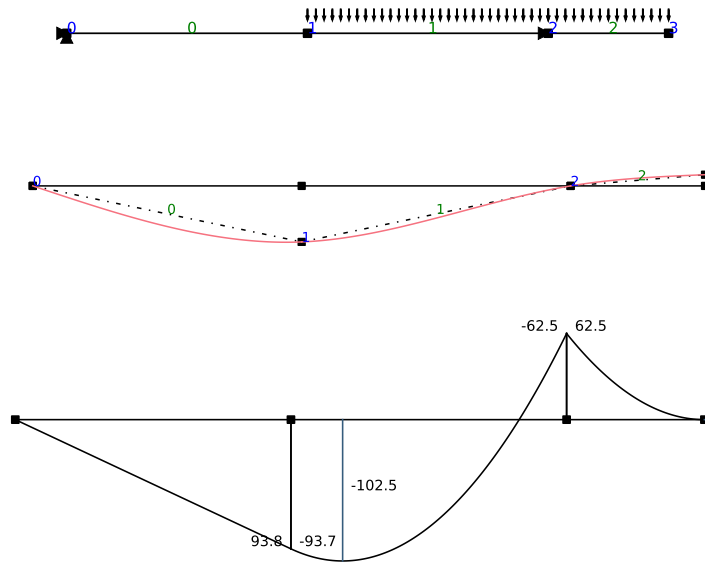


Figura 5.8 – Viga com carregamento uniformemente distribuído $w = -5$

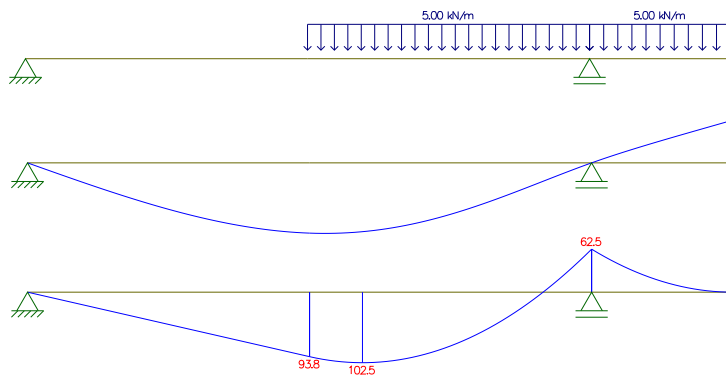


Figura 5.9 – Resultado obtido utilizando o programa ftool para viga com carregamento uniformemente distribuído $w = -5$

6 Condução Térmica

Nesse capítulo o problema da condução é apresentado juntamente com o modelo matemático utilizado para descrever o caso estacionário e transiente. O procedimento para solução numérica e implementação computacional também são discutidos. Por fim, o programa é verificado e alguns exemplos são mostrados.

6.1 Introdução

O problema de condução térmica faz parte da teoria da *termodinâmica não equilibrada*, ou termodinâmica irreversível. A termodinâmica não equilibrada possibilita uma descrição macroscópica de processos irreversíveis, (GROOT; MAZUR, 2011, p. 3). Os processos irreversíveis são caracterizados por meio da segunda lei da termodinâmica através da entropia. Considerando o equilíbrio local de regiões no sistema, podem-se definir variáveis termodinâmicas. O equilíbrio local é definido de forma que cada região atinge esse estado independentemente e que a interação entre as regiões, ou subsistemas, é fraca. Quantidades físicas extensivas associadas a cada região em equilíbrio local sujeitam-se às leis fundamentais de conservação, por exemplo, conservação de energia, (BELLAC; MORTESSAGNE; BATROUNI, 2004, p. 336).

A termodinâmica não equilibrada é baseada numa descrição espacial, Euleriana, enquanto a termodinâmica do equilíbrio é baseada numa descrição material, Lagrangiana. No contexto não equilibrado, a variável de configuração temperatura é uma função do ponto o que permite a avaliação do seu gradiente, conhecido como *força termodinâmica*, (TONTI, 2013; GROOT; MAZUR, 2011).

A termodinâmica pode ser definida como a ciência da *energia*. Energia pode ser entendida, de maneira não rigorosa, como sendo a quantidade utilizada em sistemas que passam por mudanças, (CENGEL; others, 2005, p. 2). A energia de um sistema pode ser microscópica ou macroscópica. A energia macroscópica está associada ao movimento e à influência de efeitos externos como a gravidade. A energia microscópica está associada com a estrutura molecular e as atividades das moléculas, (CENGEL; others, 2005, p. 53-55).

Outra definição de energia é *a quantidade que se mantém conservada quando*

associada a uma simetria temporal na função Lagrangiana de um sistema. Essa definição decorre do teorema de Noether (1882-1935), matemática alemã, que diz: *toda simetria da ação (Lagrangiana) tem uma lei de conservação associada*, (HANC; TULEJA; HANCOVA, 2004).

O problema fundamental da condução térmica consiste em:

- Dado:
 - Um corpo,
 - Um intervalo de tempo,
 - A natureza do material,
 - As condições de contorno,
 - A temperatura inicial em todos os pontos do corpo,
 - A distribuição e intensidade das fontes internas de calor no tempo e no espaço,
- Encontrar:
 - A temperatura em todos os pontos do corpo em todos os instantes que se seguem.

6.2 Variáveis de Interesse

As variáveis de interesse são as *quantidades físicas* que nos permitem descrever o fenômeno físico. Estas podem ser classificadas quanto o seu papel na teoria física. Classificam-se as variáveis em *de fonte, de configuração e de energia*.

6.2.1 Variáveis de Fonte

As variáveis de fonte são aquelas que denotam *produção*, ver Apêndice B.5.1. No problema do calor as seguintes variáveis de fonte podem ser destacadas: o *calor*, denotado por Q , a *corrente de calor*, denotado por Φ , e a *densidade de corrente de calor*, denotada por q .

Calor é uma forma particular de transferência de energia entre dois sistemas devido à variação de temperatura, (CENGEL; BOLES, 1994), (TONTI, 2013).

Corrente de calor, ou fluxo térmico, é a *taxa* de calor através de uma superfície num intervalo de tempo. O termo *corrente* denota *taxa de escoamento*, ou seja, quanto escoar por

unidade de tempo. O termo corrente de calor é preferido ao invés de fluxo térmico, pois a palavra fluxo pode ser ambígua.¹

Densidade de corrente de calor denota a taxa de escoamento por unidade de área. Outro termo utilizado para essa variável é fluxo de calor, mas, como discutido anteriormente, esse termo possui ambiguidades que são eliminadas com a introdução da nomenclatura: *corrente* e *densidade*. O termo *densidade* indica uma relação de razão com a área da superfície pela qual o escoamento atravessa.

6.2.2 Variáveis de Configuração

No problema do calor as seguintes variáveis de configuração são utilizadas: a *temperatura*, denotada por T , e o *gradiente da temperatura*, denotado por $\mathbf{g} = \nabla T$.

6.3 Modelo Matemático

O modelo matemático para o problema do calor, ou condução térmica, pode ser obtido, em sua forma diferencial, através da consideração de balanço energético em um volume finito. Em seguida, por um processo de limite, chega-se ao modelo na forma forte. Outros ingredientes utilizado no modelo são: a *definição* da variável gradiente de temperatura, que representa um passo intermediário, e a *lei constitutiva do material*, lei de Fourier.

A lei de Fourier é uma lei material que relaciona a densidade de corrente de calor, medida em $[W/m^2]$ com o gradiente de temperatura no ponto, medido em $[K/m]$. A constante que relaciona essas duas medidas é a condutividade λ , medida em $[W/Km]$. O gradiente de temperatura é o vetor que aponta na direção de crescimento da temperatura. O sinal negativo se deve ao fato de que a energia passa de uma região de maior temperatura para uma região de menor, desse modo o gradiente de temperatura aponta para a região com a maior temperatura mas a energia é transferida em sentido contrário. Portanto, o sinal faz com que o sentido da transferência de energia coincida com o gradiente de temperatura.

$$\mathbf{q}^{\text{law}} = -\lambda \mathbf{g} \quad (6.1)$$

Nesse caso, o material foi considerado isotrópico.

¹ Fluxo pode significar escoamento. Escoamento é a passagem de algo através de uma superfície. Fluxo pode significar taxa de escoamento, ou seja, uma quantidade por unidade de tempo, por exemplo, taxa de escoamento de energia *watt* $[J/s]$. Fluxo pode significar densidade da taxa de escoamento, $[watt/m^2]$. No caso do magnetismo, a variável que denota fluxo magnético está associada à um instante e uma superfície e não denotam escoamento de algo através de uma superfície, (TONTI, 2013, p. 31).

6.3.1 Diagrama de Tonti: Condução Térmica

O diagrama de classificação para o problema da transferência de calor está expresso na Fig. 6.1.

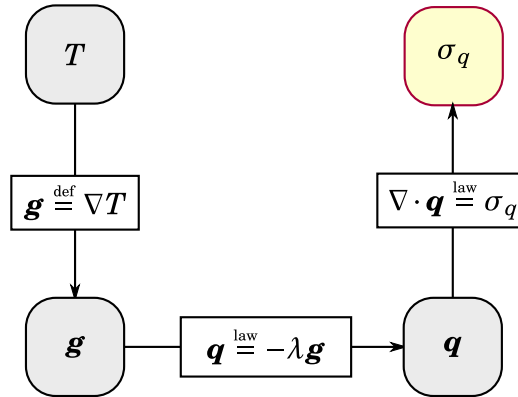


Figura 6.1 – Diagrama de Tonti para o problema da condução térmica em regime permanente. (TONTI, 2013)

6.3.2 Modelo na Forma Forte

Combinando a equação de balanço de energia com a Lei de Fourier e a definição de gradiente de temperatura, ou seja, percorrendo o caminho primário (horário) no diagrama em 6.1, chega-se a

$$\nabla \cdot \lambda \nabla T + \sigma_q = 0, \quad \text{em } \Omega. \quad (6.2)$$

O problema de segunda ordem é completo com duas condições de contorno que podem ser:

$$\begin{aligned} T(x_i) &= \bar{T}, & \text{em } \Gamma_T \\ \lambda \nabla T \cdot \mathbf{n} &= \bar{q}, & \text{em } \Gamma_f \end{aligned} \quad (6.3)$$

onde, \mathbf{n} é o vetor normal à Γ . A condição de contorno de Neumann, ou de densidade de corrente, ou de fluxo, no contorno pode ocorrer devido a uma densidade de corrente de calor no contorno ou convecção com o material que envolve Ω . Nesse caso, um fluxo indica que energia entra no sistema. Em um contexto unidimensional, verifica-se esse fato da seguinte forma: se o fluxo no contorno é positivo, ou seja, um gradiente que aponta para esquerda (sentido de entrar no domínio); caso o fluxo seja especificado negativo, o gradiente aponta para direita, logo, indica energia saindo do sistema definido de 0 a um comprimento L . O valor \bar{T} é a temperatura especificado no contorno em K ou $^{\circ}C$ e \bar{q} é a densidade de corrente de calor dada em $[W/m^2]$ no contorno especificado.

6.3.3 Modelo na Forma Fraca

Para se obter a forma fraca do modelo matemático utiliza-se o procedimento de resíduos ponderados. O resíduo é a equação homogênea na forma forte, Eq. (6.2), e a ponderação é feita por meio de uma função teste escalar $w(x_i)$. A minimização do resíduo então fica,

$$\int_{\Omega} (\nabla \cdot \lambda \nabla T + \sigma_q) w \, d\Omega = 0 \quad (6.4)$$

Utilizando a relação na Eq. (6.5) que decorre da regra do produto na diferenciação,

$$\nabla \cdot (w \nabla T) \stackrel{\text{def}}{=} w(\nabla \cdot \nabla T) + \nabla w \cdot \nabla T, \quad (6.5)$$

a Eq. (6.4) fica,

$$\int_{\Omega} \nabla w \cdot \nabla \lambda T \, d\Omega - \int_{\Omega} \nabla \cdot (w \lambda \nabla T) \, d\Omega - \int_{\Omega} w \sigma_q \, d\Omega = 0. \quad (6.6)$$

Usando o teorema de divergência, Apêndice B.7, a equação fica,

$$\int_{\Omega} \nabla w \cdot \lambda \nabla T \, d\Omega - \int_{\Gamma} w \lambda \nabla T \cdot \mathbf{n} \, d\Gamma - \int_{\Omega} w \sigma_q \, d\Omega = 0. \quad (6.7)$$

6.4 Discretização

O processo de discretização, como discutido anteriormente, consiste em substituir a integral em todo domínio pela soma da integral nos domínios definidos pelos elementos. Assim, a equação na forma fraca fica,

$$\sum_{e=1}^{ne} \int_{\Omega^e} \nabla w \cdot \lambda \nabla T \, d\Omega^e - \int_{\Gamma^e} w \lambda \nabla T \cdot \mathbf{n} \, d\Gamma^e - \int_{\Omega^e} w \sigma_q \, d\Omega^e = 0. \quad (6.8)$$

6.5 Aproximação

Aproximando a solução em cada elemento por meio da combinação linear de funções conhecidas,

$$T \approx \mathbf{N} \boldsymbol{\tau} \quad (6.9)$$

onde \mathbf{N} é a matriz com a coleção de funções de forma e $\boldsymbol{\tau}$ é o vetor com as coordenadas generalizadas, ou coeficientes da combinação linear não conhecidos. Diferente do problema da elasticidade, a temperatura é um campo escalar, portanto há apenas um grau de liberdade por nó. A matriz com as funções de forma explicitamente para um elemento plano de quatro nós é dada por,

$$\mathbf{N} = \begin{bmatrix} N_1(x_i) & N_2(x_i) & N_3(x_i) & N_4(x_i) \end{bmatrix} \quad (6.10)$$

Utilizando o método de Galerkin, a função teste é aproximada por,

$$w \approx \mathbf{N}\boldsymbol{\omega} \quad (6.11)$$

onde o espaço de aproximação é definido de forma que nos pontos onde a temperatura for conhecido o valor de w é nulo, ou seja,

$$w = 0, \quad \text{em } \Gamma_T \quad (6.12)$$

O gradiente da temperatura utilizando na aproximação proposta é dado por,

$$\nabla T = \nabla \mathbf{N}\boldsymbol{\tau} = \mathbf{B}\boldsymbol{\tau} = \begin{bmatrix} \nabla N_1(x_i) & \nabla N_2(x_i) & \nabla N_3(x_i) & \nabla N_4(x_i) \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}. \quad (6.13)$$

6.6 Construção do Elemento

Substituindo a solução aproxima na Eq. 6.8, considerando apenas um elemento,

$$\int_{\Omega^e} \mathbf{B}\boldsymbol{\omega} \cdot \lambda \mathbf{B}\boldsymbol{\tau} \, d\Omega^e - \int_{\Gamma^e} \mathbf{N}\boldsymbol{\omega} \bar{q} \, d\Gamma^e - \int_{\Omega^e} \mathbf{N}\boldsymbol{\omega} \sigma_q \, d\Omega^e = 0. \quad (6.14)$$

Fatorando os coeficientes em $\boldsymbol{\omega}$ e alterando a notação do produto interno, a equação fica,

$$\int_{\Omega^e} \lambda \mathbf{B}^T \mathbf{B}\boldsymbol{\tau} \, d\Omega^e - \int_{\Gamma^e} \mathbf{N} \bar{q} \, d\Gamma^e - \int_{\Omega^e} \mathbf{N} \sigma_q \, d\Omega^e = 0. \quad (6.15)$$

6.6.1 Matrizes Elementares

As matrizes elementares podem ser formadas retirando os coeficientes constantes da integração,

$$\mathbf{K}\boldsymbol{\tau} = \mathbf{P} \quad (6.16)$$

onde,

$$\begin{aligned} \mathbf{K} &\stackrel{\text{def}}{=} \int_{\Omega^e} \lambda \mathbf{B}^T \mathbf{B} \, d\Omega^e, \\ \mathbf{P} &\stackrel{\text{def}}{=} \int_{\Gamma^e} \mathbf{N} \bar{q} \, d\Gamma^e + \int_{\Omega^e} \mathbf{N} \sigma_q \, d\Omega^e \equiv \mathbf{P}_q + \mathbf{P}_\sigma. \end{aligned} \quad (6.17)$$

As integrais são resolvidas utilizando a mudança de coordenadas para variáveis do espaço isoparamétrico conforme discutido na Seção 4.10.1 e utilizando quadratura Gaussiana, Apêndice B.13.

6.7 Solução do Problema Estacionário

A solução do problema estacionário consiste em montar as matrizes e vetores elementares, montar as matrizes globais, aplicar as condições de contorno e resolver o sistema de equações linear. Processo equivalente ao especificado no diagrama na Fig. 4.13.

6.8 Verificação

O procedimento para verificação da solução numérica será efetuado com a comparação direta dos resultados com a solução analítica para o problema unidimensional. Considerando simetria com relação ao eixo y , pode-se tratar o problema como em unidimensional. Nesse caso, a equação da transferência de calor é dada por,

$$\begin{aligned}\frac{d}{dx} \left(\lambda \frac{dT}{dx} \right) + \sigma_q &= 0 \\ \lambda \frac{dT}{dx}(L) &= \bar{q} \\ T(0) &= T_0\end{aligned}\tag{6.18}$$

Integrando duas vezes e aplicando as condições de contorno, obtém-se:

$$T(x) = T_0 + \frac{1}{\lambda} (\bar{q} + \sigma_q L)x - \frac{\sigma_q}{2\lambda} x^2\tag{6.19}$$

Para o caso em que não há fonte interna de energia,

$$T(x) = T_0 + \frac{\bar{q}}{\lambda} x\tag{6.20}$$

O resultado na Fig. 6.3 mostra um exemplo onde $\bar{q} = -20W/^\circ C m$ na linha 1 e $T = 50^\circ C$ na linha 3, portanto, existe um fluxo de calor saindo através do contorno 1. Usando a Eq. 6.21, a temperatura em $x = L$ é $T(L) = 30^\circ C$.

Considerando o mesmo problema anterior com uma geração interna de calor constante de $\sigma_q = 40W/m^2$. Utilizando a Eq. 6.21, a temperatura em $x = 0$ continua igual a $T_0 = 50^\circ C$ e em $T(L/2) = 55^\circ C$ e $T(L) = 50$

O código utilizado para processar esse exemplo de verificação está mostrado a seguir,

```
1 from diffuspy import steadystate
2 from diffuspy import gmsh
3 from diffuspy import plotter
4 from diffuspy.material import Material
5
6 model_name = 'patch-refined'
7
8 model = gmsh.Parse(model_name)
9
10 s = list(model.surf.keys())
11 material = Material(cndtvt={s[0]: 1},
```

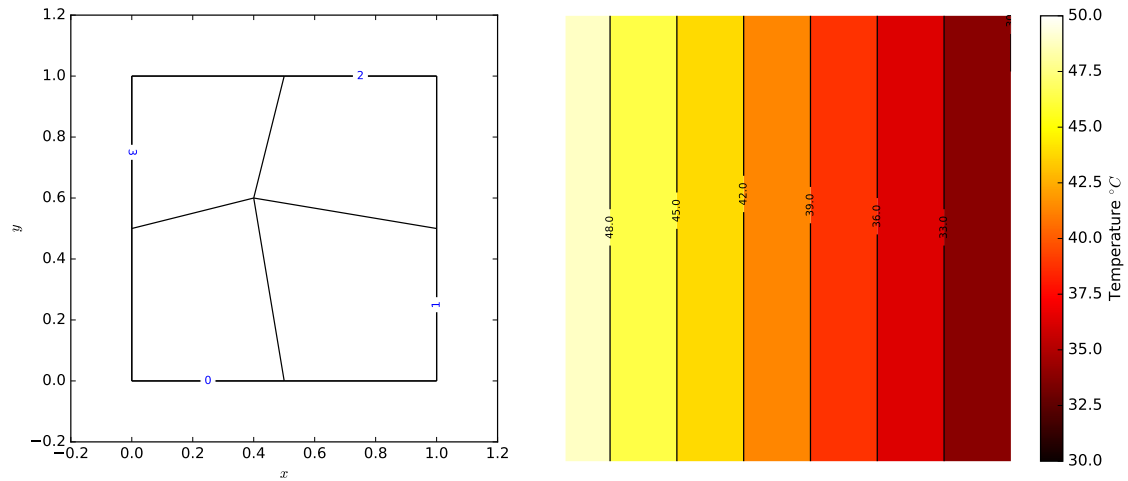


Figura 6.2 – Campo de temperatura com $T = 50^{\circ}\text{C}$ na linha 3 e $\bar{q} = -20\text{W/m}$ na linha 1, $\lambda = 1\text{W/}^{\circ}\text{Cm}$

```

12         spcfht={s[0]: 1},
13         dnsty={s[0]: 0.1})
14
15 def internal_heat(x1, x2, t=1):
16     return 0
17
18 def flux_bc(x1, x2, t=1):
19     return {1: -20}
20
21 def temperature_bc(x1, x2, t=1):
22     return {3: 50}
23
24 T = steadystate.solver(model, material, internal_heat, flux_bc,
25                         temperature_bc)
26
27 plotter.contour(model, T)
28 plotter.model(model, ele=True, edges_label=True)
29 plotter.show()

```

6.9 Exemplos

Nessa seção serão mostrados alguns exemplos para demonstrar as capacidades da ferramenta criada. A primeira série de exemplos são utilizados para demonstrar como as propriedades dos materiais pode ser definida. Primeiro as propriedades são constantes, em seguida são definidas por meio de uma função que delimita uma região específica e por último funções do espaço.

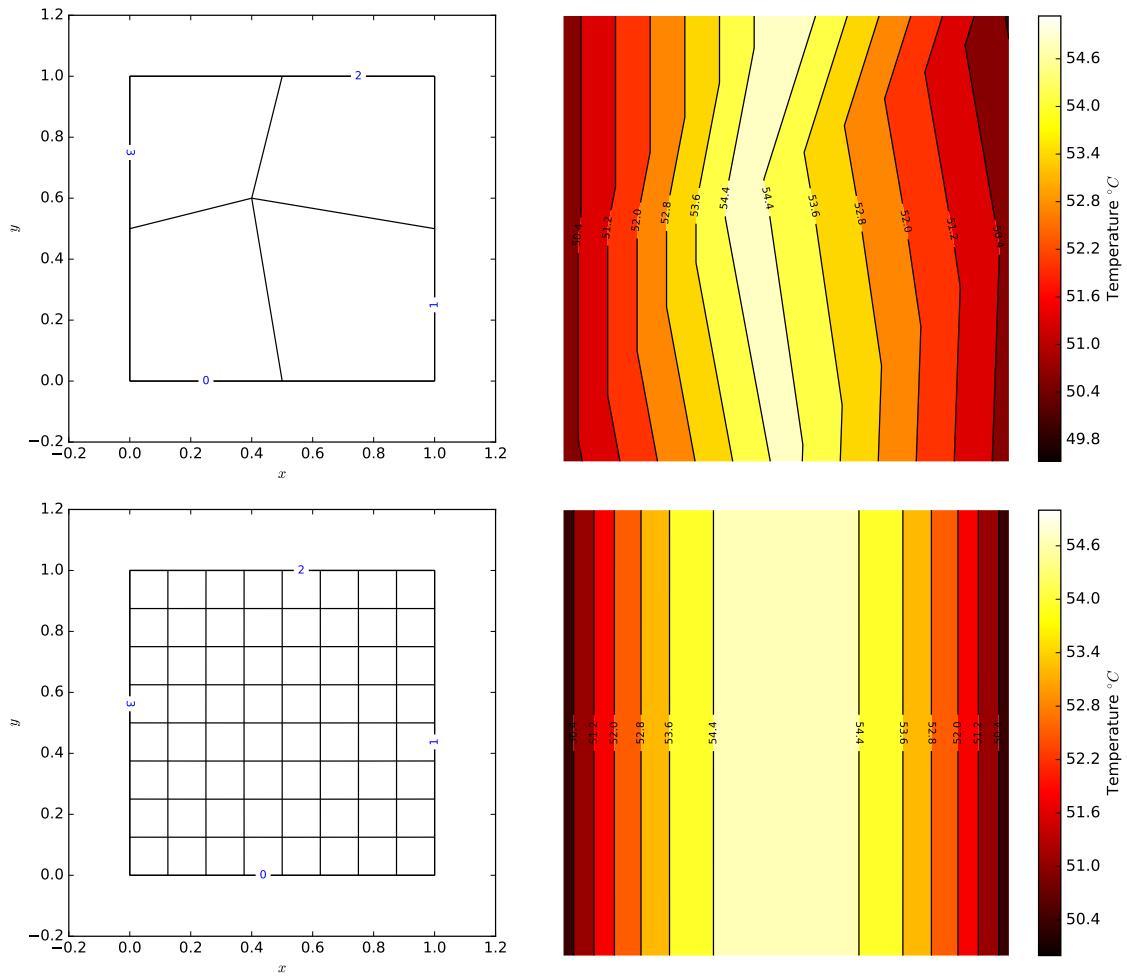


Figura 6.3 – Campo de temperatura com com $T = 50^{\circ}\text{C}$ na linha 3, $\bar{q} = -20\text{W/m}$ na linha 1, $\lambda = 1\text{W}^{\circ}\text{Cm}$ e com geração interna de calor $\sigma_q = 40\text{W/m}^2$

6.9.1 Condutividade Constante

A condutividade constante é especificada por meio de argumentos da classe `Material`. As condições de contorno são especificadas por meio de funções. Nesse caso, definiu-se uma temperatura nas bordas 1 e 3 e corrente de calor nula nas restantes.

```
1 s = list(model.surf.keys())
2 material = Material(cndtvt={s[0]: 1},
3                   spcfht={s[0]: 1},
4                   dnsty={s[0]: 0.1})
5
6 def temperature_bc(x1, x2, t=1):
7     return {1: 32, 3: 50}
```

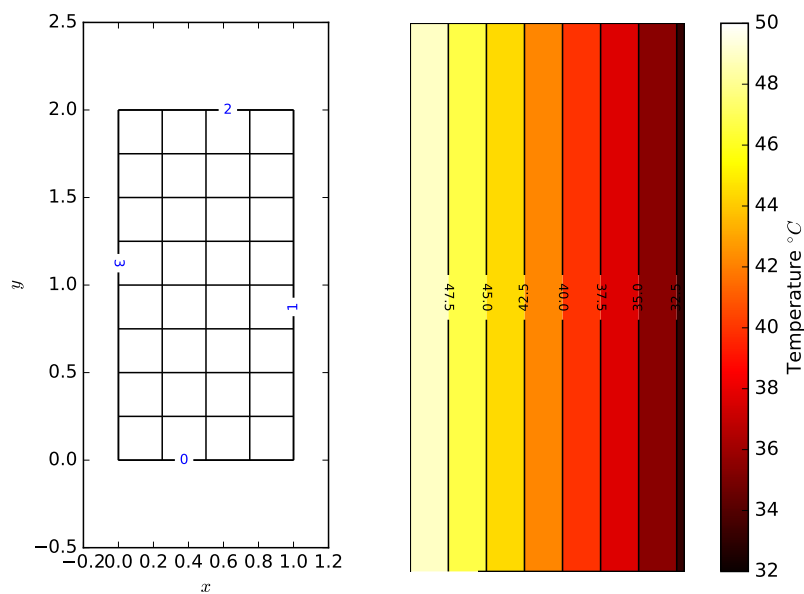


Figura 6.4 – Campo de temperatura com condutividade constante

6.9.2 Condutividade Descontínua Constante

Utilizando operações booleanas em Python, ou seja, operações que retornem *True* ou *False*, é possível definir de forma conveniente domínios com propriedades materiais distintas. Além disso, pode-se definir propriedades que variem com as coordenadas espaciais e temporais por meio da definição da função. Outro método seria criar outra `physical_surface` no `gsmh` e incluir a outra propriedade para essa superfície no dicionário: `Material(cndtvt={s[0]: 10, s[1]:100})`, onde `s[0]` indica uma superfície e `s[1]` outra.

```
1 h1, h2 = .3, .5
2 d1, d2 = .4, 1
3 k1, k2 = 100, 5
4
5 def conductivity(x1, x2, t=1):
6     return ((h1 <= x1 <= h1+d1 and h2 <= x2 <= h2+d2)*k1 +
7             (x1 < h1 or x2 < .5)*k2 +
8             (x1 > h1+d1 or x2 > h2+d2)*k2)
9
10 s = list(model.surf.keys())
11 material = Material(cndtvt={s[0]: conductivity},
12                    spcfht={s[0]: 1},
13                    dnsty={s[0]: 0.1})
14
15 def temperature_bc(x1, x2, t=1):
16     return {1: 32, 3: 50}
```

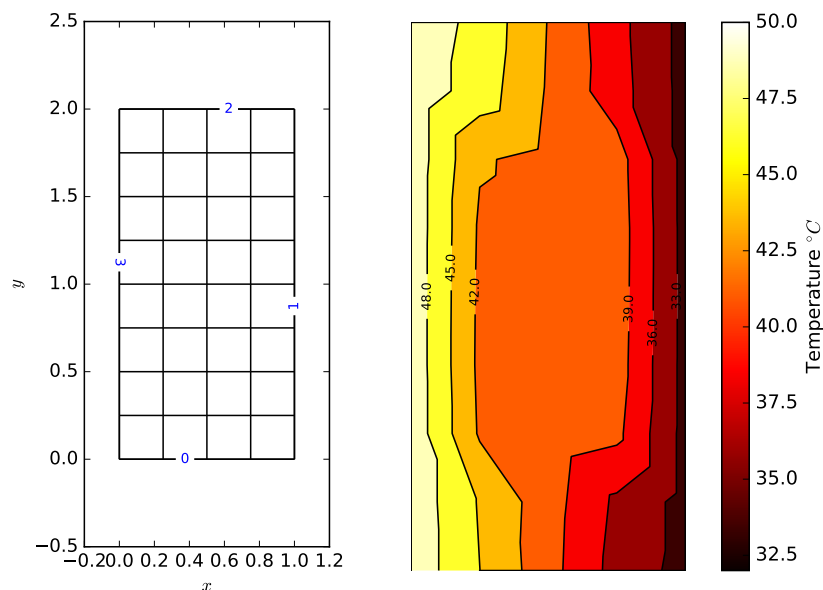


Figura 6.5 – Campo de temperatura com condutividade descontínua não variável

6.9.3 Condutividade Variável

Nesse exemplo a condutividade do material varia com a posição.

```
1 def conductivity(x1, x2, t=1):  
2     return (5*x1)**2 + (2*x2)**2  
3  
4 s = list(model.surf.keys())  
5 material = Material(cndtvt={s[0]: conductivity},  
6                     spcfht={s[0]: 1},  
7                     dnsty={s[0]: 0.1})  
8  
9 def temperature_bc(x1, x2, t=1):  
10    return {1: 32, 3: 50}
```

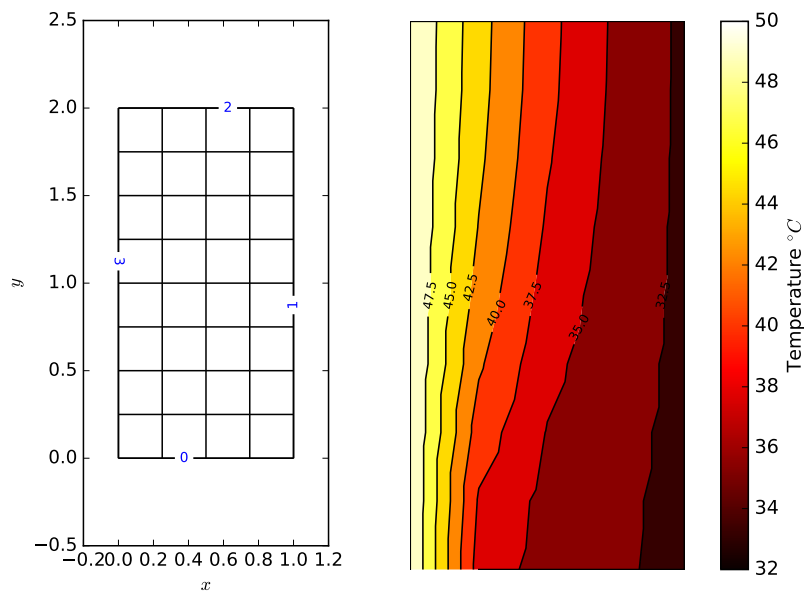


Figura 6.6 – Campo de temperatura com condutividade variável no espaço

6.9.4 Geometria Variável

Nesse exemplo será demonstrado a capacidade do programa em resolver e pós processar o caso em que a geometria apresenta descontinuidade como um furo. Será especificado por meio do dicionário a temperatura no contorno do furo e no contorno externo.

```
1 def temperature_bc(x1, x2, t=1):  
2     return {5: 10, 6: 10, 7: 10, 4: 10, 1: 20, 2: 20, 3: 20, 0: 20}
```

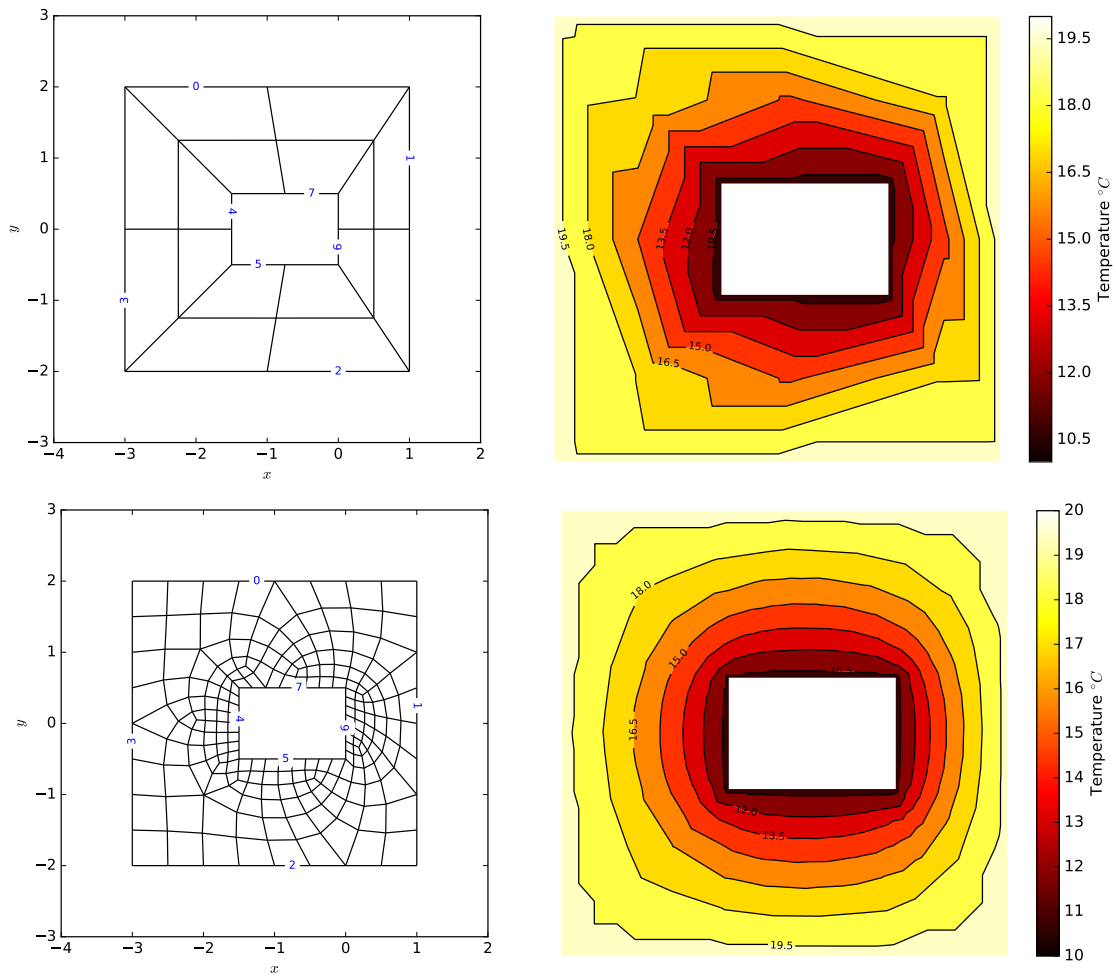


Figura 6.7 – Campo de temperatura em geometria com furo

6.10 Problema Transiente

O problema transiente de transferência de calor é obtido quando se considera a variação da energia interna do corpo, u . A variação da energia interna do corpo está associada com mudança de temperatura. Essa relação constitutiva do material é dada por,

$$\partial_t u \stackrel{\text{mat}}{=} \rho c \partial_t T, \quad (6.21)$$

onde, ρ é a densidade em $[kg/m^3]$, c é o calor específico dado em $[J/kgK]$. A Figura 6.8, mostra o diagrama de classificação para o problema transiente.

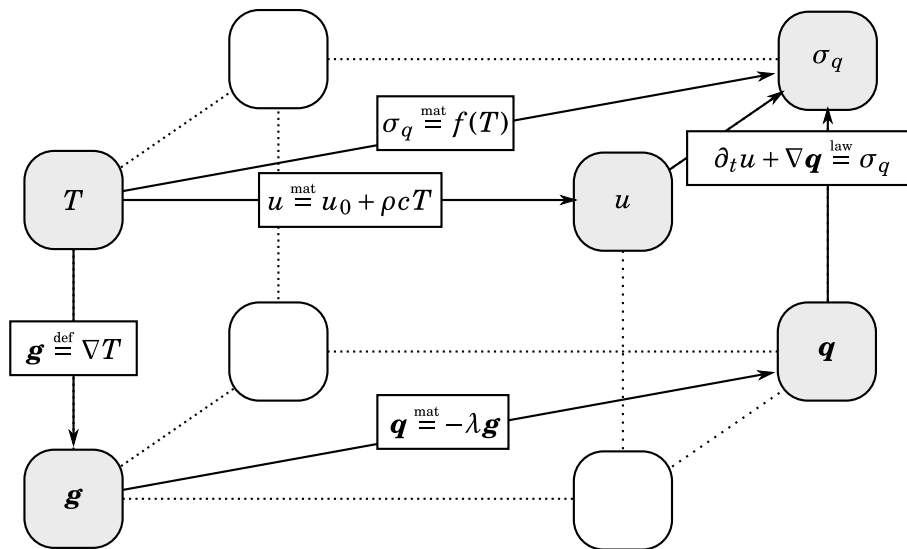


Figura 6.8 – Diagrama de Tonti para o problema da condução térmica transiente com uma fonte de calor dependente da temperatura. (TONTI, 2013)

7 Conclusão

Nesse capítulo serão apresentadas as conclusões e considerações finais sobre o trabalho. Essas são divididas em três aspectos. O primeiro diz respeito à metodologia de solução numérica, Método dos Elementos Finitos. O segundo aspecto está ligado ao desenvolvimento do programa em Python. O terceiro está relacionado às considerações gerais como a utilização do software Gmsh e tópicos para futuros trabalhos.

7.1 Método dos Elementos Finitos

O método dos elementos finitos representa uma técnica de solução numérica importante em vários ramos da engenharia. Com o advento de computadores pessoais a utilização de programas que produzem soluções numéricas tornou-se comum. Contudo, é importante ter o conhecimento do procedimento utilizado para se obter a solução numérica.

O ingrediente básico para se buscar a previsão de algum fenômeno físico é o modelo matemático que representa o fenômeno. O conhecimento sobre o modelo matemático permite a avaliação dos resultados de forma que estes não violem hipóteses utilizadas na criação do modelo. Como exemplo pode-se citar a hipótese de pequenos deslocamentos no modelo da elasticidade linear. A solução numérica do modelo consiste em dois passos básicos: discretização e aproximação. O processo de discretização pode ocorrer por diversos métodos: diferenças finitas, elementos finitos, volume finitos, elementos de contorno, dentre outros. A aproximação no caso dos elementos finitos é definido pela escolha dos elementos. O resultado desse procedimento é o valor do campo em pontos discretos que devem ser então interpretados.

Soluções numéricas são facilmente obtidas por meio de *softwares* comerciais. A avaliação dos resultados constitui a etapa mais importante do processo. O conhecimento das equações que modelam o fenômeno físico permite ao usuário, por meio de simplificações, calcule manualmente a resposta do problema de forma a ter noção qualitativa e quantitativa do resultado esperado na análise numérica.

7.2 Python

Os programas foram desenvolvidos em Python pelas seguintes razões:

- Python é uma linguagem de fácil leitura, isso faz com que a colaboração em Python ocorra de forma mais fácil. Além disso permite uma maior compreensão do programa apenas com a leitura do mesmo, dessa forma é possível revisar o código com maior facilidade.
- Python apresenta diversas bibliotecas feitas por usuários, bibliotecas como: *numpy*, *matplotlib*, *networkx*, *scipy*, *sympy* que podem ser utilizadas por qualquer um. Dessa forma, os programas podem ser criados dependendo de outros programas;

Os softwares foram licenciados utilizando GPL, *GNU General Public License v3* que de forma simplificada diz: o usuário pode copiar, distribuir e modificar desde que as mudanças sejam registradas no código fonte. Qualquer modificação de *software* sob a GPL deve estar também sob essa mesma licença.

O código dos programas está hospedado na plataforma github onde é feito o controle de versão, <<https://github.com/nasseralkmim>>.

7.3 Gmsh

O programa Gmsh é um software livre extremamente poderoso. O seu desenvolvimento teve início em 1997, ou seja, são quase 2 décadas de constante melhorias e aprimoramento. Nesse trabalho, uma grande parte do tempo foi alocada no pós processamento dos resultados utilizando as bibliotecas *matplotlib* e *networkx*. A utilização dessas bibliotecas permitiu uma maior independência do programa desenvolvido, no entanto, como a malha é criada utilizando o Gmsh, a utilização desse mesmo programa para o pós-processamento não acarreta em grandes problemas, pelo contrário, apenas vantagens. Uma vez que o programa Gmsh apresenta um poderoso e versátil módulo dedicado ao pós processamento de resultados. Com isso, o tempo alocado para detalhes técnicos de visualização dos resultados podem ser utilizado no estudo e desenvolvimento do *solver*.

Para utilizar o pós-processamento do programa gmsh, basta acrescentar no arquivo *.msh* uma seção *NodeData*. Por exemplo, para o campo de temperatura:

```
1 $NodeData
2 1
3 "Temperature"
4 1
5 0.0
6 3
```

```

7 0
8 1
9 9
10 1 10.0
11 2 11.0
12 3 12.0
13 4 13.0
14 5 14.0
15 6 15.0
16 7 16.0
17 8 17.0
18 9 18.0
19 $EndNodeData

```

e para o campo de deslocamento:

```

1 $NodeData
2 1
3 "Displacement"
4 1
5 0.0
6 3
7 0
8 3
9 9
10 1 0 0 0
11 2 1 0 0
12 3 1 0 0
13 4 0 0 0
14 5 .5 0 0
15 6 1 0 0
16 7 .5 0 0
17 8 0 0 0
18 9 .4 0 0
19 $EndNodeData

```

onde, o primeiro número representa o número de etiquetas do tipo *string*, no caso, o nome do resultado a ser mostrado. O próximo inteiro mostra o número de etiquetas do tipo *real*, que indica o tempo associado ao dado. O último inteiro indica o número de etiquetas inteiras, no caso, 3. Sendo que o primeiro inteiro indica o índice referente ao tempo (0 é o padrão), o segundo indica o número de componentes no campo a ser mostrado (no caso 1), e o último é o número de entidades (no caso, 9 nós). Carregando esse o arquivo `.msh` com essa nova seção, o programa Gmsh irá apresentar um novo item na árvore, *post-processing*. O resultado mostrado na Fig. 7.1 é obtido pelo Gmsh.

7.4 Trabalhos Futuros

Os programas desenvolvidos, `elastopy`, `diffuspy` e `sapy`, continuaram em desenvolvimento.

O `sapy` precisa de mais atenção quanto ao módulo de pós processamento. Atualmente, na versão 0.1, ele plota os diagramas de momento fletor rotacionando os resultados de acordo com a configuração do elemento. Tendo em vista que esse procedimento é com-

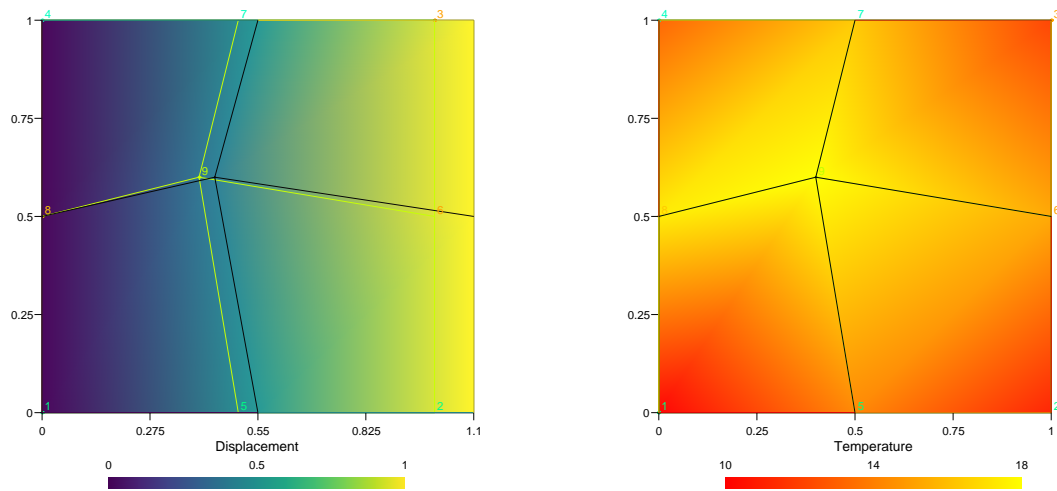


Figura 7.1 – Resultado utilizando Gmsh como ferramenta de pós-processamento

plexo, uma nova abordagem deve ser utilizada. O pacote matplotlib permite rotacionar o gráfico inteiro, portanto esse pode ser um caminho a ser seguido.

O programa elastopy pode ser estendido com relação ao elemento utilizado. Atualmente, versão 0.1, o programa possui implementado o elemento plano de quatro nós. A extensão do programa para três dimensões pode ser feita de forma simples e o pós processamento pode ser passado para o gmsh. O problema dinâmico já foi implementado, no entanto requer testes. A discretização temporal é feita por meio do método de Newmark. O programa diffuspy segue a mesma linha do programa elastopy.

Tópicos de estudo como: otimização topológica, otimização estrutural e mecânica não linear podem requerer implementações semelhantes as feitas. Além disso a experiência na construção dos programas pode ser útil em trabalhos futuros.

Referências

- ALNAES, M. et al. The fenics project version 1.5. *Archive of Numerical Software*, v. 3, n. 100, 2015. ISSN 2197-8263. Disponível em: <<http://journals.ub.uni-heidelberg.de/index.php/ans/article/view/20553>>. Citado na página 144.
- ANALYTICS, C. *Numba — Numba*. <<http://numba.pydata.org/>>. Citado na página 146.
- ARCHER, G. C. *Object-Oriented Finite Element Analysis*. Tese (Doutorado) — University of California at Berkeley, 1996. Citado 2 vezes nas páginas 8 e 146.
- ARGYRIS, J. H.; KELSEY, S. *Energy Theorems and Structural Analysis*. Boston, MA: Springer US, 1960. ISBN 978-1-4899-5852-5 978-1-4899-5850-1. Citado na página 7.
- BARBER, J. R. *Elasticity*. Dordrecht; Boston: Kluwer Academic Publishers, 2002. OCLC: 54909163. ISBN 978-0-306-48395-0 978-1-4020-0966-2 978-1-4020-0964-8. Citado 3 vezes nas páginas 30, 31 e 118.
- BATHE. *Klaus-Jürgen Bathe-Finite Element Procedures-Prentice Hall (1996).pdf*. 1996. Citado 8 vezes nas páginas vi, 8, 12, 25, 42, 46, 52 e 53.
- BATHE, K.-J. *Finite element procedures*. [S.l.]: Klaus-Jurgen Bathe, 2006. Citado na página 12.
- BELLAC, M. L.; MORTESSAGNE, F.; BATROUNI, G. G. *Equilibrium and Non-Equilibrium Statistical Thermodynamics*. Cambridge, UK; New York: Cambridge University Press, 2004. OCLC: 144618541. ISBN 978-0-511-19228-9 978-0-511-64817-5 978-0-511-60657-1. Citado na página 94.
- BRENNER, S.; SCOTT, R. *The mathematical theory of finite element methods*. [S.l.]: Springer Science & Business Media, 2007. v. 15. Citado na página 5.
- BUCALEM, M. L.; BATHE, K.-J. *The Mechanics of Solids and Structures - Hierarchical Modeling and the Finite Element Solution*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. (Computational Fluid and Solid Mechanics). ISBN 978-3-540-26331-9 978-3-540-26400-2. Citado na página 45.
- CENGEL, Y. A.; BOLES, M. A. Thermodynamics: an engineering approach. *Sea*, v. 1000, p. 8862, 1994. Citado na página 95.
- CENGEL, Y. A.; others. *Thermodynamics An Engineering Approach 5th Edition By Yunus A Cengel: Thermodynamics An Engineering Approach*. [S.l.]: Digital Designs, 2005. Citado na página 94.

CHANDRASEKHARAIHAH, D. S.; DEBNATH, L. *Continuum Mechanics*. Boston: Academic Press, 1994. ISBN 978-0-12-167880-7. Citado 5 vezes nas páginas 29, 30, 34, 39 e 40.

CIMRMAN, R. SfePy - write your own FE application. In: BUYL, P. de; VAROQUAUX, N. (Ed.). *Proceedings of the 6th European Conference on Python in Science (EuroSciPy 2013)*. [S.l.: s.n.], 2014. p. 65–70. [Http://arxiv.org/abs/1404.6391](http://arxiv.org/abs/1404.6391). Citado na página 144.

CLOUGH, R. W. Early history of the finite element method from the view point of a pioneer. *International journal for numerical methods in engineering*, v. 60, n. 1, p. 283–287, 2004. Citado na página 7.

CLOUGH, R. W.; WILSON, E. L. *Stress Analysis of a Gravity Dam by the Finite Element Method*. 1962. Citado 2 vezes nas páginas 2 e 8.

CODINA, R. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, v. 156, n. 1, p. 185–210, abr. 1998. ISSN 0045-7825. Citado na página 9.

COOK, R. *Concepts and Applications of Finite Element Analysis*. 2002. Citado 6 vezes nas páginas 8, 12, 13, 17, 19 e 45.

CRUZ, F. *Programming the ENIAC*. 2013. <<http://www.columbia.edu/cu/computinghistory/eniac.html>>. Citado na página 1.

DADVAND, P.; ROSSI, R.; OÑATE, E. An Object-oriented Environment for Developing Finite Element Codes for Multi-disciplinary Applications. *Archives of Computational Methods in Engineering*, v. 17, n. 3, p. 253–297, set. 2010. ISSN 1134-3060, 1886-1784. Citado na página 9.

DICTIONARY, T. F. *ENIAC | Article about ENIAC by The Free Dictionary*. 2003. <<http://encyclopedia2.thefreedictionary.com/ENIAC>>. Citado na página 1.

FELIPPA, C. Advanced finite element methods. Department of Aerospace Engineering Sciences, University of Colorado, 2005. Citado 15 vezes nas páginas vi, vii, viii, 11, 12, 16, 26, 37, 41, 42, 43, 46, 84, 88 e 134.

FELIPPA, C. A. Introduction to finite element methods. *Course Notes, Department of Aerospace Engineering Sciences, University of Colorado at Boulder, available at <http://www.colorado.edu/engineering/Aerospace/CAS/courses.d/IFEM.d>*, 2004. Citado 7 vezes nas páginas viii, 5, 81, 83, 86, 87 e 88.

FELIPPOU, F. J. *First course on matrix structural analysis*. [S.l.: s.n.], 2013. Citado 10 vezes nas páginas viii, ix, 81, 88, 120, 123, 125, 127, 130 e 131.

FORDE, B. W. R.; FOSCHI, R. O.; STIEMER, F. S. *Object-Oriented Finite Element Analysis*. 1989. Citado na página 8.

GANDER, M. J.; WANNER, G. From euler, ritz, and galerkin to modern computing. *Siam Review*, SIAM, v. 54, n. 4, p. 627–666, 2012. Citado na página 6.

GANDER, M. J.; WANNER, G. From Euler, Ritz, and Galerkin to Modern Computing. *SIAM Review*, v. 54, n. 4, p. 627–666, jan. 2012. ISSN 0036-1445, 1095-7200. Citado 2 vezes nas páginas 6 e 7.

- GARTLING, D. K. Convective heat transfer analysis by the finite element method. *Computer Methods in Applied Mechanics and Engineering*, v. 12, n. 3, p. 365–382, dez. 1977. ISSN 0045-7825. Citado na página 9.
- GEUZAINÉ, C.; REMACLE, J.-F. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, v. 79, n. 11, p. 1309–1331, 2009. Citado na página 62.
- GONZALEZ, O.; STUART, A. M. *A First Course in Continuum Mechanics*. Leiden: Cambridge University Press, 2008. OCLC: 437240799. ISBN 978-0-511-45513-1 978-1-282-38994-6. Citado na página 34.
- GOVINDJEE, S. *A first course on variational methods in structural mechanics and engineering*. 2014. Citado 2 vezes nas páginas vi e 14.
- GROOT, S. R. D.; MAZUR, P. *Non-Equilibrium Thermodynamics*. [S.l.]: Dover Publications, 2011. Citado na página 94.
- GURTIN, M. *Gurtin1981.pdf*. 1981. Citado na página 32.
- HALL, J. F.; CHOPRA, A. K. Two-dimensional dynamic analysis of concrete gravity and embankment dams including hydrodynamic effects. *Earthquake Engineering & Structural Dynamics*, v. 10, n. 2, p. 305–332, 1982. Citado na página 9.
- HANC, J.; TULEJA, S.; HANCOVA, M. Symmetries and conservation laws: Consequences of Noether's theorem. *American Journal of Physics*, v. 72, n. 4, p. 428, 2004. ISSN 00029505. Citado na página 95.
- HUGHES, T. J. *The finite element method: linear static and dynamic finite element analysis*. [S.l.]: Courier Corporation, 2012. Citado na página 12.
- KAVAMURA, E. E. *Estudo Do Comportamento Termo-Estrutural Bidimensional de Barragens de Concreto Utilizando O Método Dos Elementos Finitos*. 2005. Citado na página 10.
- LANCZOS, C. *The Variational Principles of Mechanics*. [S.l.: s.n.], 1952. Citado na página 78.
- LEGER, P.; KATSOULI, M. Seismic stability of concrete gravity dams. *Earthquake engineering & structural dynamics*, v. 18, n. 6, p. 889–902, 1989. Citado na página 9.
- LEGER, P.; VENTURELLI, J.; BHATTACHARJEE, S. S. Seasonal temperature and stress distributions in concrete gravity dams. Part 1: Modelling. *Canadian journal of civil engineering*, v. 20, n. 6, p. 999–1017, 1993. Citado na página 9.
- MARSDEN, J. E.; HUGHES, T. J. *Mathematical foundations of elasticity*. [S.l.]: Courier Corporation, 1994. Citado na página 34.
- MEEK, J. A brief history of the beginning of the finite element method. *International journal for numerical methods in engineering*, v. 39, p. 3761–3774, 1996. Citado 2 vezes nas páginas 5 e 6.
- MILLMAN, K. J.; AIVAZIS, M. Python for Scientists and Engineers. *Computing in Science & Engineering*, v. 13, n. 2, p. 9–12, mar. 2011. ISSN 1521-9615. Citado na página 1.

- MOSALAM. *Mosalam Course Textbook.pdf*. 2011. Citado 4 vezes nas páginas 35, 49, 66 e 69.
- NASCIMENTO, C. A. d. N. *Um Estudo Comparativo Analítico-Numérico de Tensões Locais E Globais Em Barragens Gravidade de Concreto*. 2016. Citado na página 10.
- NASTRAN. 2016. Page Version ID: 726339404. Citado na página 2.
- OLVER, P. J. *Numerical analysis lectures notes*. [S.l.: s.n.], 2008. Citado na página 20.
- PATZÁK, B.; BITTNAR, Z. Design of object oriented finite element code. *Advances in Engineering Software*, v. 32, n. 10, p. 759–767, 2001. Citado na página 8.
- PHUNPENG, V.; BAIZ, P. M. Mixed finite element formulations for strain-gradient elasticity problems using the FEniCS environment. *Finite Elements in Analysis and Design*, v. 96, p. 23–40, abr. 2015. ISSN 0168-874X. Citado na página 9.
- PRZEMIENIECKI, J. S. *Theory of matrix structural analysis*. [S.l.]: Courier Corporation, 1985. Citado na página 80.
- PURCELL, E. M. *Electricity and Magnetism*. Third edition. Cambridge: Cambridge University Press, 2013. ISBN 978-1-107-01402-2. Citado na página 139.
- RAO, S. S. *The Finite Element Method in Engineering*. 5. ed. ed. Amsterdam: Elsevier, Butterworth-Heinemann, 2011. OCLC: 846354670. ISBN 978-1-85617-661-3. Citado 4 vezes nas páginas 8, 42, 83 e 142.
- RAO, S. S. *The Finite Element Method in Engineering: Pergamon International Library of Science, Technology, Engineering and Social Studies*. [S.l.]: Elsevier, 2013. Citado 5 vezes nas páginas 2, 13, 19, 24 e 81.
- SCREMIN, J. J. *Análise Termo-Elástica 2d via Método Dos Elementos Finitos Aplicada a Estimativa de Falha Em Barragens de Ccr*. 2011. Citado na página 10.
- SLAUGHTER, W. S.; VERLAG, B. *The Linearized Theory of Elasticity*. 2003. Citado 8 vezes nas páginas vi, 30, 31, 34, 35, 37, 38 e 39.
- TAYLOR, R. L. *FEAP-Ein Finite Element Analysis Programm*. [S.l.]: Ing.-Gemeinschaft Klee & Wrigges, 1987. Citado na página 8.
- TAYLOR, R. L. et al. The patch test—a condition for assessing FEM convergence. *International Journal for Numerical Methods in Engineering*, v. 22, n. 1, p. 39–62, 1986. Citado na página 69.
- TEAM, C. *Cython: C-Extensions for Python*. <<http://cython.org/>>. Citado na página 146.
- TONTI, E. *The Mathematical Structure of Classical and Relativistic Physics*. New York, NY: Springer New York, 2013. (Modeling and Simulation in Science, Engineering and Technology). ISBN 978-1-4614-7421-0 978-1-4614-7422-7. Citado 18 vezes nas páginas vi, viii, 25, 26, 28, 29, 37, 41, 50, 78, 94, 95, 96, 97, 107, 137, 138 e 139.
- TONTI, E. Why starting from differential equations for computational physics? *Journal of Computational Physics*, v. 257, p. 1260–1290, jan. 2014. ISSN 00219991. Citado na página 27.

van der Walt, S.; COLBERT, S. C.; VAROQUAUX, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, v. 13, n. 2, p. 22–30, mar. 2011. ISSN 1521-9615. Citado na página 146.

VANDERPLAS, J. *Why Python Is Slow: Looking Under the Hood*. 2014. <<https://jakevdp.github.io/blog/2014/05/09/why-python-is-slow/>>. Citado 2 vezes nas páginas 145 e 146.

WANG, W.; KOLDITZ, O. Object-oriented finite element analysis of thermo-hydro-mechanical (THM) problems in porous media. *International Journal for Numerical Methods in Engineering*, v. 69, n. 1, p. 162–201, jan. 2007. ISSN 00295981, 10970207. Citado na página 9.

WILSON, E. *Three Dimensional Static and Dynamic Analysis Of Structures*. 1998. <<http://www.edwilson.org/Book/book.htm>>. Citado 2 vezes nas páginas 49 e 78.

ZIENKIEWICZ, O. C.; TAYLOR, R. L. *The Finite Element Method*. 5th ed. ed. Oxford ; Boston: Butterworth-Heinemann, 2000. ISBN 978-0-7506-5049-6 978-0-7506-5055-7 978-0-7506-5050-2. Citado na página 8.

ZOHDI, T. *Introduction to finite elements*. 2014. Citado 3 vezes nas páginas vi, 17 e 18.

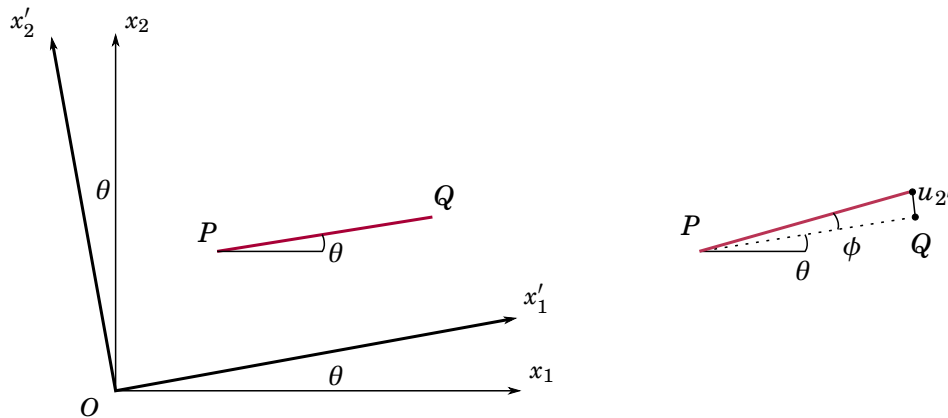
Apêndices

A Tópicos em Mecânica

A.1 Deformação Cisalhante

Essa demonstração é apresentada em (BARBER, 2002). Conhecidas as componentes do deslocamento u_1 e u_2 deseja-se encontrar a rotação ϕ de uma linha PQ inclinada em um ângulo θ com relação ao eixo x . Ver Figura A.1.

Figura A.1 – Rotação dos eixos.



Fonte: Elaborado pelo autor

Primeiramente, cria-se o eixo $Ox'_1x'_2$ com o segmento Ox'_1 paralelo à PQ . As rotações do segmento podem ser aproximadas *linearmente* por

$$\phi(PQ) = \frac{\partial u_2'}{\partial x_1'}. \quad (\text{A.1})$$

onde a derivada no eixo rotacionado é dada por

$$\frac{\partial}{\partial x_1'} = \nabla \cdot \mathbf{i}' \quad (\text{A.2})$$

$$= \mathbf{i} \cdot \mathbf{i}' \frac{\partial}{\partial x_1} + \mathbf{j} \cdot \mathbf{i}' \frac{\partial}{\partial x_2} \quad (\text{A.3})$$

$$= \cos(\theta) \frac{\partial}{\partial x_1} + \sin(\theta) \frac{\partial}{\partial x_2} \quad (\text{A.4})$$

com \mathbf{i} e \mathbf{j} como os vetores unitários nas direções x_1 e x_2 respectivamente. O valor dos deslocamentos no eixo rotacionado pode ser obtido por trigonometria,

$$u_{1'} = u_1 \cos(\theta) + u_2 \sin(\theta) \quad (\text{A.5})$$

$$u_{2'} = u_2 \cos(\theta) + u_1 \sin(\theta). \quad (\text{A.6})$$

substituindo esses valores na equação (A.23),

$$\phi(PQ) = \left(\cos(\theta) \frac{\partial}{\partial x_1} + \sin(\theta) \frac{\partial}{\partial x_2} \right) (u_2 \cos(\theta) + u_1 \sin(\theta)) \quad (\text{A.7})$$

$$= \frac{1}{2} \left(\frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2} \right) + \frac{1}{2} \left(\frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2} \right) \cos(2\theta) + \frac{1}{2} \left(\frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2} \right) \sin(\theta), \quad (\text{A.8})$$

utilizando as igualdades trigonométricas $\sin(a+b) = \sin(a)\cos(b) + \sin(b)\cos(a)$ e $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$. O primeiro termo da equação independe de θ , portanto, representa *rotação de corpo rígido* ω . O termo também independe do eixo coordenado. O raciocínio pode ser estendido para três dimensões gerando a rotação de corpo rígido em um plano devido aos deslocamentos. De forma geral, a rotação de corpo rígido pode ser obtida por

$$\boldsymbol{\omega} = \frac{1}{2} \nabla \times \mathbf{u} \quad (\text{A.9})$$

A deformação cisalhante ε_{ij} pode então ser definida como a diferença entre a rotação da linha na direção x_1 e a rotação de corpo rígido no plano em questão. Para o plano x_1x_2 tem-se,

$$\varepsilon_{12} = \frac{\partial u_2}{\partial x_1} - \frac{1}{2} \left(\frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2} \right) = \frac{1}{2} \left(\frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2} \right) \quad (\text{A.10})$$

ou de forma geral,

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (\text{A.11})$$

A.2 Formulação por Mecânica do Sólidos

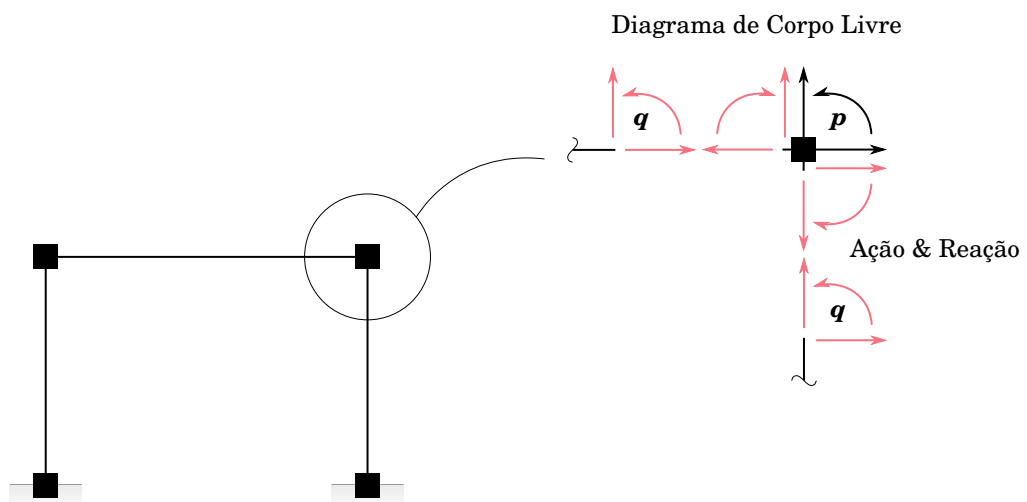
A formulação por mecânica do sólidos consiste na discretização por elementos e a escolha das variáveis em cada elemento por meio da extensão das equações no ponto material para a seção e em seguida para o elemento. Essa formulação é interessante pois não requer o formalismo matemático como no caso de métodos variacionais e portanto pode ser entendida com apenas o pré-requisito de mecânica dos sólidos.

As variáveis envolvidas no problema da mecânica das estruturas são os esforços e deformações internas, forças externas e deslocamentos nodais. Essas variáveis relacionam-se entre si por meio de *relações de equilíbrio, relações cinemáticas e relações constitutivas*.

A.2.1 Relação de Equilíbrio

As relações de equilíbrio são obtidas através do equilíbrio do *corpo livre* definido por cada *nó na estrutura*. Para estabelecer essas relações é necessário antes definir as variáveis da estática: *forças nodais* (F_N) e *forças de extremidade nos elementos* (F_{EE}). Essas variáveis são relacionadas através do equilíbrio estático no nó, ou seja, força resultante igual a zero.

Figura A.2 – Equilíbrio do corpo livre definido por um nó. As forças de extremidade no elemento, (F_{EE}), dadas pela letra q equilibram-se com as forças nodais (F_N) representadas pela letra p .



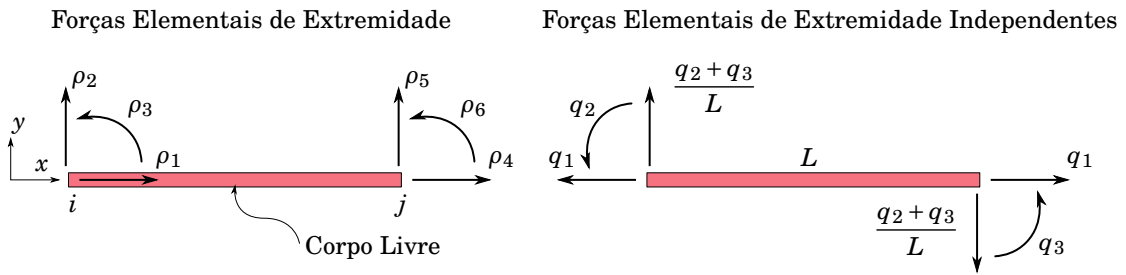
Fonte: Elaborada pelo autor

As forças nodais são condições de contorno especificadas. As forças de extremidade nos elementos *são as incógnitas do problema*. A quantidade de forças de extremidade depende do tipo de elemento em questão, no caso de treliças há apenas a força axial.

A.2.1.1 Forças Elementares de Extremidade Independentes

Num elemento de pórtico, que pode ser entendido como a superposição de um elemento de barra e um de viga, possui em cada extremidade 3 F_{EE} . Pelo equilíbrio do corpo livre definido pelo elemento, tem-se 3 equações da estática e 6 incógnitas. Da álgebra linear, esse sistema possui 3 variáveis dependentes e 3 independentes. É conveniente escolher os momentos em cada extremidade (ρ_3 e ρ_6) e o esforço axial (ρ_4) como variáveis independentes. Esses esforços são então *Forças elementares de extremidade independentes* (F_{EEI}), ou como em (FELIPPOU, 2013), *forças básicas elementares* (q). A Figura A.3, exemplifica esse processo.

Figura A.3 – Forças elementares de extremidade independentes.



Fonte: Elaborada pelo autor

As relações de equilíbrio geram o sistema

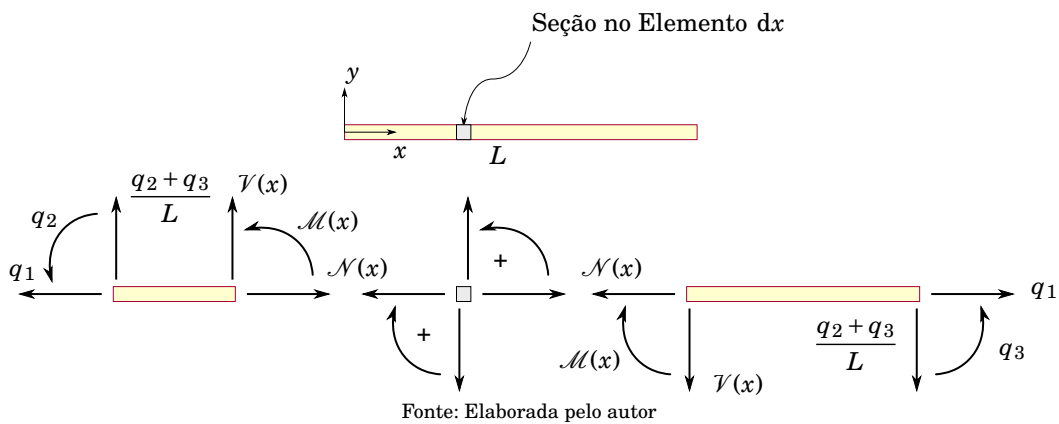
$$\begin{cases} \text{Em } x: & \rho_1 + \rho_4 = 0 \\ \text{Em } y: & \rho_2 + \rho_5 = 0 \\ \text{Em } i: & \rho_3 + \rho_6 + \rho_5 L = 0 \end{cases} \Rightarrow \begin{cases} \rho_1 = -\rho_4 \\ \rho_2 = \frac{\rho_3 + \rho_6}{L} \\ \rho_5 = -\frac{\rho_3 + \rho_6}{L} \end{cases} \quad (\text{A.12})$$

No contexto das *forças básicas elementares*, $\rho_4 = q_1$, $\rho_3 = q_2$ e $\rho_6 = q_3$. A escolha dessas três variáveis independentes é conveniente pois, a partir de q_1 obtém-se o diagrama de esforço normal e a partir de q_2 e q_3 obtém-se o diagrama de esforço cortante.

A.2.1.2 Forças de Extremidade e Esforços Internos

Uma vez resolvido o problema, e portanto conhecidas as forças de extremidade, os esforços internos seccionais: *momento fletor* \mathcal{M} , *esforço axial* \mathcal{N} , *esforço cortante* \mathcal{V} podem ser determinados através do *equilíbrio do corpo livre definido pelo elemento*.

Figura A.4 – Equilíbrio do corpo livre definido por um elemento.



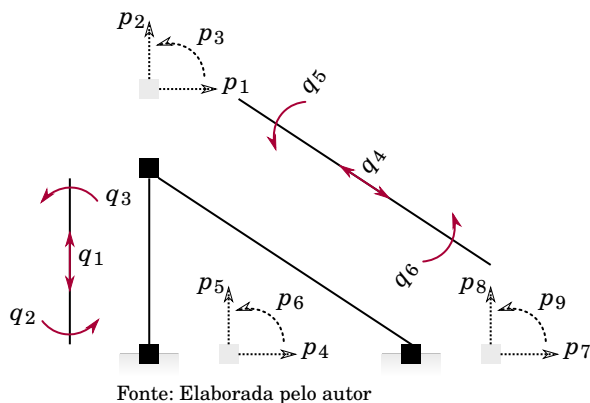
A.2.1.3 Equilíbrio Nodal

As relações de equilíbrio em cada nó produzem um sistema de equações na forma

$$\mathbf{p} = \mathbf{A}\mathbf{q} \quad (\text{A.13})$$

onde, \mathbf{p} é o vetor que contém as forças aplicadas nos nós, \mathbf{A} é a matriz do equilíbrio e \mathbf{q} é o vetor com as forças elementares de extremidade independentes, ou forças elementares básicas. Cada elemento possui um número de F_{EB} , em elementos de barra apenas 1 e em elementos de pórtico existem 3 F_{EB} . A soma do número de F_{EB} compõem o número de incógnitas no problema, dado por nq . A Figura A.5 mostra as forças elementares básicas para cada elemento.

Figura A.5 – Equilíbrio nodal do modelo estrutural. Na figura, q_1, q_2, \dots, q_6 , são as incógnitas $nq = 6$, p_1, p_2, p_3 indicam o valor nodal no grau de liberdade livre, $nf = 3$, e p_4, p_5, \dots, p_9 indicam os graus de liberdade restringidos por apoios.



Cada nó tem 3 graus ou liberdade, 3 *dof*. Uma relação estática pode ser gerada para cada um deles. Um grau de liberdade pode ser *livre* ou *restringido*. Na Figura A.5, os *dof* restringidos estão tracejados e os livres em linha cheia. Nos graus de liberdade livre, onde uma força nodal pode ser aplicada, p_1, p_2 e p_3 , pode-se formar uma equação válida, ou seja, útil para obtenção das incógnitas, F_{EB} . O número de graus de liberdade livre, *dof livres*, é dado por $nf = 3$. O número de graus de liberdade total é $nt = 3nn = nf + nr = 9$, ou seja, no plano, cada nó apresenta 3 *dof*, $nr = 6$ é o número de graus de liberdade restringidos.

Uma estrutura que apresenta mais incógnitas, nq , do que equações válidas, nf , é dita *hiperestático*. Caso $nq = nf$, a estrutura é *isostática* e caso $nq < nf$, *hipostática*. O grau de hiperestaticidade é dado pela diferença $nq - nf$. A estrutura mostrada na Figura A.5 tem grau de hiperestaticidade igual a $3 = 6 - 3$. O grau de hiperestaticidade também pode ser obtido por meio dos *dof* restringidos usando a relação $nr = nt - nf$.

A.2.2 Relação Cinemática

As relações cinemáticas, ou relações deslocamento-deformação, estabelecem a conexão entre os *deslocamentos nodais* do modelo estrutural com as *deformações internas* a cada elemento, (FELIPPOU, 2013). Para a descrição dessas relações as seguintes variáveis cinemáticas são utilizadas:

- *Modelo Estrutural*: deslocamentos nodais, \mathbf{u} . Por exemplo, um nó no plano pode transladar nas direções x e y e rotacionar em torno do eixo z ;
- *Elemento*: deslocamentos elementares de extremidade, $\boldsymbol{\mu}$, e deformações, \mathbf{v} , obtidas pela remoção do movimento de corpo rígido dos deslocamentos $\boldsymbol{\mu}$ (mais detalhes na Seção A.2.2.2);
- *Ponto Material*: deformação específica¹, ϵ ;
- *Seção*: deformação da seção, ϵ . As deformações na seção, assumindo a hipótese de que seções planas permanecem planas, pode ser descrita pela deformação específica axial (do eixo), ϵ_a , e a curvatura da seção κ . Essa hipótese facilita na definição de um conjunto de deformações para o elemento, \mathbf{v} .

De posse dessas variáveis deseja-se uma relação do tipo

$$\mathbf{v} = \mathbf{B}\mathbf{u}, \quad (\text{A.14})$$

análoga à Equação (4.44), das relações de equilíbrio estático, ambas *relacionam uma variável elementar e uma variável nodal*.

Para estabelecer essa relação é necessário primeiro definir o conjunto de *deformações elementares*, \mathbf{v} (todos os elementos), a partir das equações do ponto material, discutidas no Capítulo 4. As equações da mecânica dos meios contínuos podem ser extendidas para a seção a partir de hipóteses simplificadoras.

A.2.2.1 Deformações no Ponto Material e na Seção

A hipótese de que *seção transversais planas permanecem planas* após deformações implica numa *distribuição linear* de deformações na seção transversal. A distribuição linear das deformações permite que as deformações específicas numa seção sejam descritas por dois parâmetros (como uma reta). Os parâmetros utilizados são a deformação axial no eixo do elemento, ϵ_a , e a curvatura da seção, κ . A Figura 5.6 mostra, em um esquema

¹ No contexto da mecânica dos sólidos, no Capítulo 4, a deformação específica foi chamada apenas de deformação.

gráfico, a interpretação das aproximações feitas para a rotação num ponto, θ , e curvatura, κ .

A distribuição de deformações específicas na seção é dada por $\varepsilon(x, y) = \varepsilon_a + \kappa y$. Os parâmetros $\varepsilon_a(x)$ e $\kappa(x)$ são funções unicamente da posição no eixo x . Portanto, as deformações elementares serão obtidas pela integração desses valores ao longo do eixo x . Pelo fato da deformação ser dada pela soma entre ε_a e κy , esse dois valores são desacoplados, portanto divide-se a *deformação do elemento* em *axial*, dada por v_1 , e *de flexão*, v_2 e v_3 .

A.2.2.2 Deformações Elementares

As variáveis de interesse primário na análise cinemática de estruturas são as *deformações elementares* (dE), representadas por \mathbf{v} . Em um elemento de pórtico essas variáveis representam, *deformação axial* e *rotações em cada extremidade*, ou, mais precisamente, o ângulo entre a tangente da forma deformada do elemento e a corda elementar. A corda é a linha *reta* que liga as extremidades elementar. A variação do comprimento longitudinal, deformação axial, é dada por v_1 . Os ângulos da deformação em cada extremidade são dados por, v_2 e v_3 .

A deformação axial contínua, v_1 , pode ser calculada por

$$v_1 = \int_0^L \varepsilon_a \, dx = \varepsilon_a L, \quad (\text{A.15})$$

ou, no caso mais geral, superpõem-se um valor de deformação inicial ΔL_0 . As deformações de flexão contínuas², causadas por um campo de curvatura, podem ser calculadas através de relações geométricas aproximadas mostradas na Figura A.6. O valor dessas deformações é dado por,

$$v_2 = -\frac{1}{L} \int_0^L \kappa(L-x) \, dx \quad (\text{A.16})$$

$$v_3 = \frac{1}{L} \int_0^L \kappa x \, dx \quad (\text{A.17})$$

A convenção adotada para as deformações de flexão é: positivo quando o ângulo que vai da *corda* para a *tangente* é *anti-horário* e negativo caso contrário, *horário*. Nesse caso, o ângulo mostrado na Figura A.6, v_2 , é negativo, pois está no sentido horário.

A.2.2.3 Continuidade Nodal

A continuidade nodal é garantida através da relação entre *deformações elementares* e *deslocamentos nodais*. A continuidade nodal é analogamente ao equilíbrio nodal na

² Podem ainda existir deformações de natureza discreta como por exemplo rotações devido a presença de rótulas nas extremidades dos elementos.

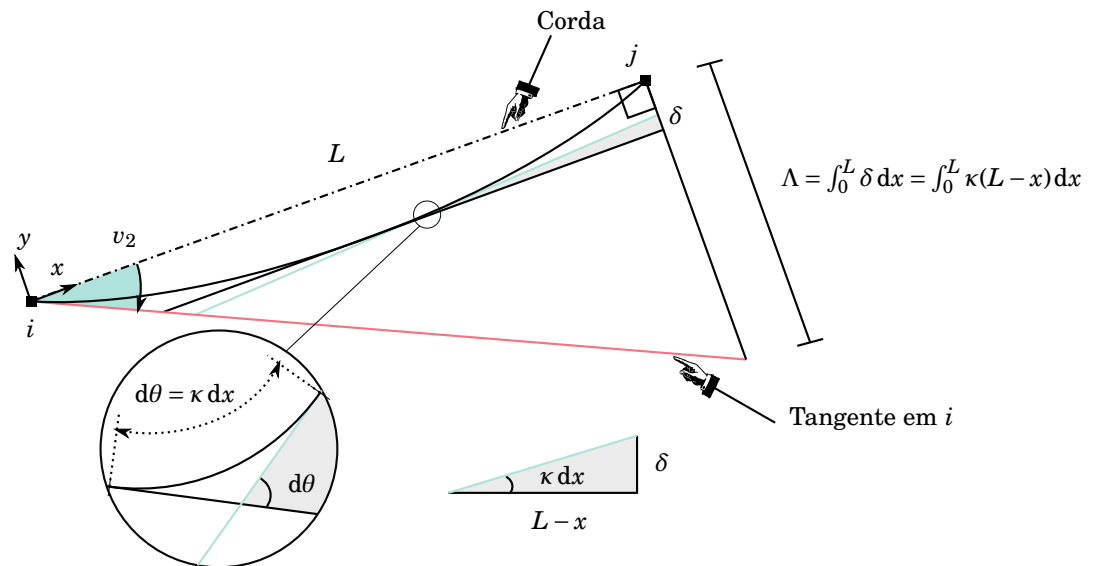


Figura A.6 – Deformação de flexão elemental contínua calculada considerando pequenas rotações, $\text{tg}(\theta) \approx \theta$. Em cada trecho dx da estrutura, um δ pode ser calculado. A soma dos δ produz um Λ que representa o cateto oposto no triângulo maior pelo qual é possível calcular a deformação de flexão, v_2 . (FELIPPOU, 2013)

estática, Seção A.2.1.3. A fim de estabelecer essa relação, é necessário definir algumas variáveis.

Além das variáveis primárias, (dE), existem variáveis acessórias utilizadas para relacionar o elemento, a extremidade do elemento, os nós e a região entre nós.

- **Elemento:** as variáveis cinemáticas do elemento são as *deformações elementares*, (dE), denotadas por \mathbf{v} .
- **Extremidade elemental:** as extremidades dos elementos tem a cinemática descrita por *deslocamentos de extremidade elemental*, (DEE), denotadas por $\boldsymbol{\mu}$.
- **Nós:** os nós possuem a cinemática descrita por *deslocamentos nodais*, (DN), \mathbf{u} . Os (DN) se relacionam com os (DEE) por meio de argumentos de continuidade.
- **Região entre nós:** possui variável cinemática denominada *deformação intra-nodal*, (dIN). Essa variável é definida com o intuito de fazer a conexão do elemento com o modelo estrutural, a região entre nós pode sofrer alongamento e rotações nas duas extremidades. A (dIN) se relaciona com a (dE) devido a argumentos de continuidade entre o modelo estrutural e o elemento.

O processo utilizado para estabelecer as relações entre \mathbf{v} e \mathbf{u} , deformações elementares e deslocamentos nodais, descritos em (FELIPPOU, 2013), consiste em:

1. Igualar os deslocamentos nodais, \mathbf{u} , com os deslocamentos de extremidade elementar, $\boldsymbol{\mu}$. As duas variáveis descrevem o mesmo movimento, no entanto, uma aplica-se diretamente aos nós, \mathbf{u} , e a outra à extremidade do elemento, $\boldsymbol{\mu}$.
2. Relacionar os deslocamentos do par de nós, que definem o elemento, com as deformações intra-nodais. Os deslocamentos do par de nós é equivalente aos deslocamentos das extremidades elementar, $\boldsymbol{\mu}$. As deformações intra-nodais são definidas antes de se encaixar os elementos no modelo estrutural.
3. Igualar as deformações intra-nodais com as deformações elementares, \mathbf{v} . Nesse ponto o elemento é incluído no modelo estrutural.

A.2.2.3.1 Passo 1: Relação entre u e μ

Por continuidade, as extremidades elementares seguem os deslocamentos nodais, portanto, em um elemento geral

$$\mathbf{u}^e = \boldsymbol{\mu}^e \quad (\text{A.18})$$

onde, \mathbf{u}^e indica os deslocamentos nodais num par de nós que definem um elemento.

A.2.2.3.2 Passo 2: Relação entre μ e as dIN

As deformações intra-nodais (dIN), como definidas anteriormente, são convenientemente expressas em função dos deslocamentos da extremidade elementar, $\boldsymbol{\mu}$, nas coordenadas locais, $\bar{\boldsymbol{\mu}}$,

$$\text{Enlongamento: } \bar{\mu}_4 - \bar{\mu}_1, \quad (\text{A.19})$$

$$\text{Rotação em } i : \bar{\mu}_3 - \frac{\bar{\mu}_5 - \bar{\mu}_2}{L}, \quad (\text{A.20})$$

$$\text{Rotação em } j : \bar{\mu}_6 - \frac{\bar{\mu}_5 - \bar{\mu}_2}{L}, \quad (\text{A.21})$$

onde, as variáveis $\bar{\boldsymbol{\mu}}$ estão definidas conforme a Figura . Nesse ponto utilizou-se argumentos de *linearização* para calcular o enlongamento e as rotações, aproximou-se o arco da rotação por uma reta sem mudança de comprimento. Portanto, o enlongamento pode ser calculado por meio dos deslocamentos axiais apenas e as rotações por meio de deslocamentos transversais ao elementos ($\tan(\theta) \approx \theta$).

As coordenadas podem ser transformadas de global para local por meio das relações,

$$\begin{aligned}
 \bar{\mu}_1 &= \mu_1 \cos \alpha + \mu_2 \sin \alpha \\
 \bar{\mu}_2 &= -\mu_1 \sin \alpha + \mu_2 \cos \alpha \\
 \bar{\mu}_3 &= \mu_3 \\
 \bar{\mu}_4 &= \mu_4 \cos \alpha + \mu_5 \sin \alpha \\
 \bar{\mu}_5 &= -\mu_4 \sin \alpha + \mu_5 \cos \alpha \\
 \bar{\mu}_6 &= \mu_6,
 \end{aligned}
 \tag{A.22}$$

ou, em forma matricial $\bar{\boldsymbol{\mu}} = \mathbf{G}\boldsymbol{\mu}$. O ângulo α é definido pela inclinação do elemento no espaço. As relações em (A.22) são obtidas por trigonometria.

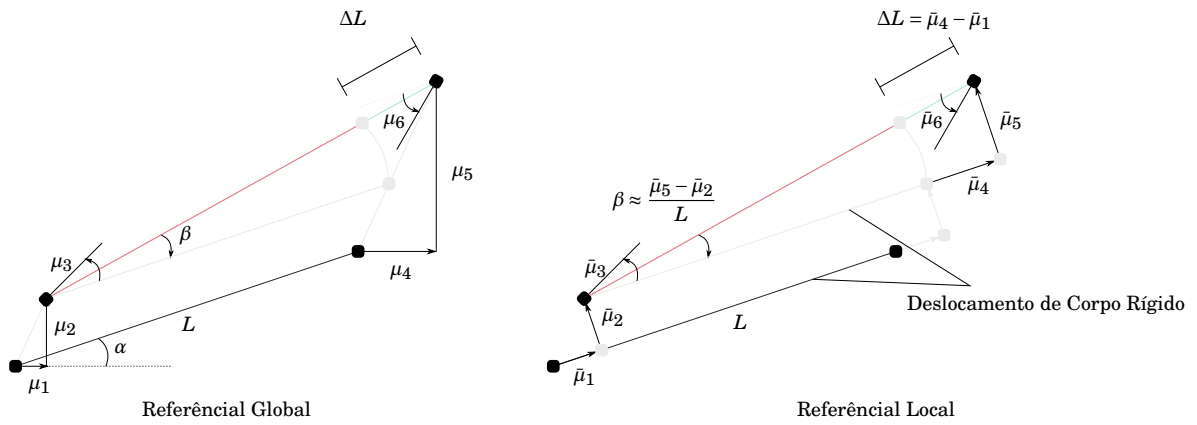


Figura A.7 – Relação entre deformação intra-nodal e deslocamentos de extremidade, $\boldsymbol{\mu}$. A figura explicita o movimento de corpo rígido devido aos deslocamentos de extremidade e também as componentes nas referências dos eixos global e local. O ângulo α é a inclinação do elemento no espaço com relação ao eixo x global, o ângulo β é a inclinação da corda do elemento. (FELIPPOU, 2013)

A.2.2.3.3 Passo 3: Relação entre dIN e v

Também por continuidade, ao encaixar o elemento no modelo estrutural, as dIN devem ser iguais as deformações elementares,

$$\begin{aligned}
 v_1 &= \Delta L = \bar{\mu}_4 - \bar{\mu}_1 \\
 v_2 &= \bar{\mu}_3 - \frac{\bar{\mu}_5 - \bar{\mu}_2}{L} \\
 v_3 &= \bar{\mu}_6 - \frac{\bar{\mu}_5 - \bar{\mu}_2}{L}.
 \end{aligned}
 \tag{A.23}$$

Então, como os deslocamentos de extremidade elementar, $\boldsymbol{\mu}$, consideram translação e rotação das extremidades, existe, movimento de corpo rígido e deformações envolvidas nessas variáveis. As dIN e conseqüentemente, \boldsymbol{v} , consideram apenas a parcela de deformação de $\boldsymbol{\mu}$, portanto, é dito que as mesmas são obtidas pela remoção dos deslocamentos de corpo rígido de $\boldsymbol{\mu}$.

Utilizando a matriz de transformação \mathbf{G} , que leve os deslocamentos de extremidade das coordenadas locais para as globais, pode-se escrever, para cada elemento,

$$\mathbf{v}^e = \mathbf{B}^e \boldsymbol{\mu}^e. \quad (\text{A.24})$$

A matriz cinemática elementar, \mathbf{B}^e , possui dimensão 3×6 para um elemento de pórtico plano, e 1×6 para elemento de barra. A matriz para o elemento de pórtico plano, obtida pelas relações em (A.23), utilizando a transformação de coordenadas por meio de \mathbf{G} , é dada por,

$$\mathbf{A}^e = \begin{bmatrix} -\cos \alpha & -\sin \alpha & 0 & \cos \alpha & \sin \alpha & 0 \\ \frac{-\sin \alpha}{L} & \frac{\cos \alpha}{L} & 1 & \frac{\sin \alpha}{L} & \frac{-\cos \alpha}{L} & 0 \\ \frac{-\sin \alpha}{L} & \frac{\cos \alpha}{L} & 0 & \frac{\sin \alpha}{L} & \frac{-\cos \alpha}{L} & 1 \end{bmatrix} = \begin{bmatrix} -\frac{\Delta X}{L} & -\frac{\Delta Y}{L} & 0 & \frac{\Delta X}{L} & \frac{\Delta Y}{L} & 0 \\ \frac{\Delta Y}{L^2} & \frac{\Delta X}{L^2} & 1 & \frac{\Delta Y}{L^2} & -\frac{\Delta X}{L^2} & 0 \\ -\frac{\Delta Y}{L^2} & \frac{\Delta X}{L^2} & 0 & \frac{\Delta Y}{L^2} & -\frac{\Delta X}{L^2} & 1 \end{bmatrix}, \quad (\text{A.25})$$

onde, ΔX e ΔY são as componentes que definem o comprimento L do elemento. Considerando a assemblagem de todos os elementos,

$$\mathbf{v} = \mathbf{B} \mathbf{u}. \quad (\text{A.26})$$

com,

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}^{(1)} \\ \mathbf{v}^{(2)} \\ \vdots \\ \mathbf{v}^{(ne)} \end{bmatrix}_{3ne \times 1}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}^{(1)} \\ \mathbf{B}^{(2)} \\ \vdots \\ \mathbf{B}^{(ne)} \end{bmatrix}. \quad (\text{A.27})$$

A matriz \mathbf{B} envolve todos os graus de liberdade da estrutura, no entanto, como existem graus de liberdade restringidos por condições de contorno, pode-se separá-la em uma parte referente aos *graus de liberdade livre*, \mathbf{B}_f , e uma parte referente aos *graus de liberdade restringidos*, \mathbf{B}_d .

A matriz \mathbf{B} pode ser formada por meio do processo de assemblagem descrito acima, conveniente para o computador. Nesse caso, é produzido uma matriz para cada elemento individualmente para em seguida construir uma matriz global por meio de *índices de incidência* dos graus do liberdade do elemento na estrutura como um todo.

Na solução de problemas manualmente, é possível construir a matriz cinemática manualmente por meio da utilização de vetores unitários que compõem a base do espaço de dimensão igual ao número de graus de liberdade. Esse processo é utilizado em álgebra linear para encontrar a matriz da transformação linear. Fisicamente isso significa aplicar um deslocamento unitário em cada grau de liberdade \mathbf{u} e relacioná-lo com deformações elementares, \mathbf{v} . Esse processo é convenientemente expresso graficamente. O exemplo a seguir mostra o procedimento para construção da matriz cinemática,

A.2.3 Relação Constitutiva

A relação constitutiva é aquela que estabelece a ligação entre as variáveis de deformação e variáveis de tensão num ponto material. Num material elástico e com hipóteses de linearidade, a tensão e deformação são relacionadas pelo módulo de elasticidade, E .

$$\varepsilon = \frac{\sigma}{E} + \varepsilon_0 \quad (\text{A.28})$$

onde, ε_0 representa deformações iniciais, causada por efeitos não mecânicos, variações térmicas, por exemplo.

A.2.3.1 Elemento de Barra

Num elemento de barra as deformações são dadas pela integral da deformação específica axial, ε_a , ver Figura 5.6. Integrando essa deformação ao longo do elemento obtém-se a deformação elementar, v_1 .

$$v_1 = \int_0^L \varepsilon_a dx + \Delta L_0 \quad (\text{A.29})$$

utilizando a hipótese de material elástico linear, (B.2),

$$v_1 = \int_0^L \left(\frac{\mathcal{N}}{EA} + \varepsilon_0 \right) dx + \Delta L \quad (\text{A.30})$$

onde \mathcal{N} é a força normal na seção, que corresponde, em um elemento de barra, à própria força básica axial do elemento, q_1 . Como q_1 é constante ao longo do elemento e supõem-se o material homogêneo e prismático, E e A constantes, e ε_0 também constante. A integral fica,

$$v_1 = \frac{L}{EA} q_1 + v_{0_1} \quad (\text{A.31})$$

onde, $v_{0_1} = \varepsilon_0 L + \Delta L$ é a combinação das deformações não mecânicas. Portanto, diz-se que a deformação elementar e o esforço interno elementar estão ligados por meio da matriz de *flexibilidade*, $\mathbf{F} = L/EA$. O nome flexibilidade é dado pois quando maior for o seu valor maior será a deformação, portanto, mais flexível.

A.2.3.2 Elemento de Viga

Num elemento de viga as deformações, rotações nas extremidades, são calculadas por meio das integrais em (4.48) e (4.49). O campo de curvatura utilizado é obtido considerando a distribuição de momentos no elemento. Utiliza-se a superposição das deformações devido à aplicação de momentos em cada extremidade individualmente. No elemento, onde não há forças transversais, a distribuição de momentos longitudinalmente é uniforme

linear. A curvatura está ligada ao momento fletor por meio da relação $\kappa EI = M$, pois, na seção $\varepsilon = \kappa y$ (cinemática), $\sigma = My/I$ (equilíbrio) e $\sigma = E\varepsilon$ (constituição elástica).

Caso $q_3 = 0$, tem-se o seguinte resultado

$$v_2 = \left(\frac{L}{3EI}\right)q_2, \quad v_3 = \left(-\frac{L}{6EI}\right)q_2, \quad (\text{A.32})$$

e caso $q_2 = 0$,

$$v_2 = \left(-\frac{L}{6EI}\right)q_3, \quad v_3 = \left(\frac{L}{3EI}\right)q_3. \quad (\text{A.33})$$

Ambos estão mostrados na Figura A.9.

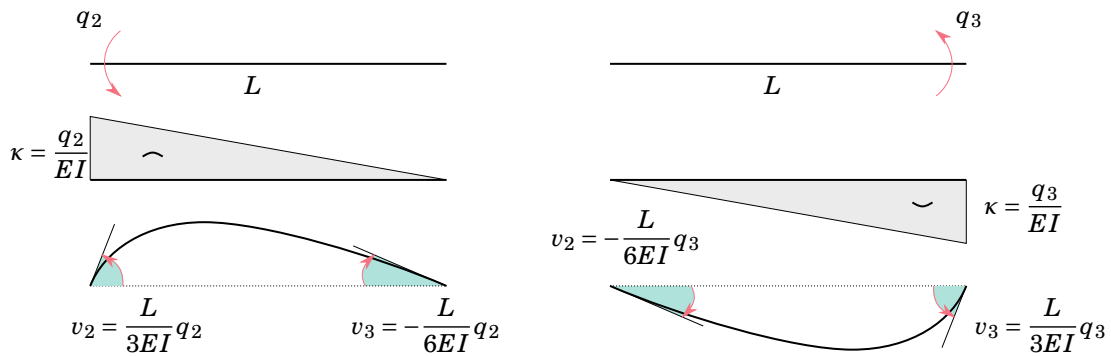


Figura A.8 – Deformação elementar de flexão devido ao esforço interno de momento fletor. O diagrama de distribuição da curvatura, na parte central, é obtido por meio do diagrama de momento fletor. Na parte inferior está o elemento em sua forma deformada com a indicação das deformações elementares nas extremidades que foram calculadas por meio da integral em (4.48) e (4.49). (FELIPPOU, 2013)

Caso haja carregamento no elemento, forças distribuídas, procede-se da mesma maneira: primeiro define-se a distribuição de momentos, em seguida encontra-se a curvatura para finalmente calcular as deformações por meio das integrais em (4.48) e (4.49).

No caso de um carregamento constante distribuído, w .

$$v_2 = -\frac{wL^3}{24EI}, \quad v_3 = \frac{wL^3}{24EI} \quad (\text{A.34})$$

O efeito de uma variação de temperatura, produz um campo de temperatura constante no elemento, gerando

$$v_2 = -\frac{\kappa_t L}{2}, \quad v_3 = \frac{\kappa_t L}{2}. \quad (\text{A.35})$$

Todas essas relações podem ser resumidas pela expressão

$$\begin{bmatrix} v_2 \\ v_3 \end{bmatrix} = \frac{L}{6EI} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} q_2 \\ q_3 \end{bmatrix} + \begin{bmatrix} -\frac{\kappa_t L}{2} \\ \frac{\kappa_t L}{2} \end{bmatrix} + \begin{bmatrix} -\frac{wL^3}{24EI} \\ \frac{wL^3}{24EI} \end{bmatrix}, \quad (\text{A.36})$$

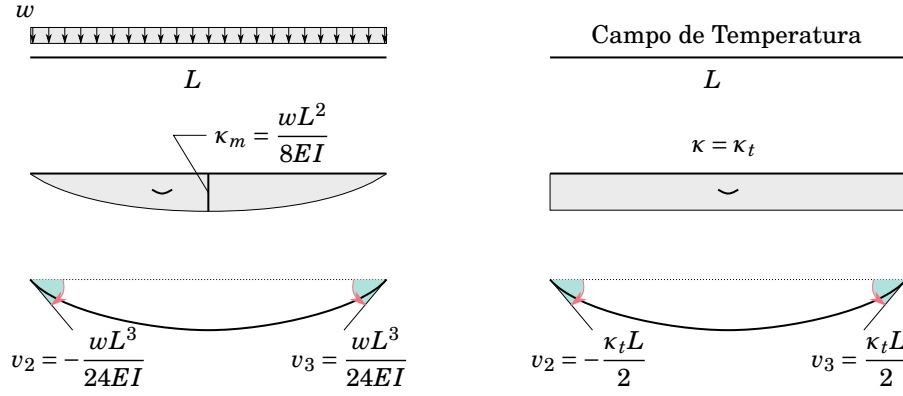


Figura A.9 – Deformação elementar de flexão devido ao carregamento distribuído (esquerda) e devido à campo de temperatura que produz curvatura constante (direita). O elemento em sua forma deformada com a indicação das deformações elementares nas extremidades que foram calculadas por meio da integral em (4.48) e (4.49). (FELIPPOU, 2013)

ou de forma resumida,

$$\mathbf{v}^e = \mathbf{F}^e \mathbf{q}^e + \mathbf{v}_0^e, \quad (\text{A.37})$$

a matriz \mathbf{F} é chamada de *flexibilidade*. A forma inversa da relação é dada por,

$$\begin{aligned} \mathbf{q}^e &= (\mathbf{F}^e)^{-1}(\mathbf{v}^e - \mathbf{v}_0^e) \\ &= \mathbf{C}^e \mathbf{v}^e + \mathbf{q}_0^e \end{aligned} \quad (\text{A.38})$$

onde, $(\mathbf{F}^e)^{-1} = \mathbf{C}^e$ e $\mathbf{q}_0^e = -\mathbf{C}^e \mathbf{v}_0^e$. A matriz \mathbf{C}^e é a matriz de rigidez elementar. Explicitamente,

$$\begin{bmatrix} q_2 \\ q_3 \end{bmatrix} = \frac{2EI}{L} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} v_2 \\ v_3 \end{bmatrix} + \begin{bmatrix} EI\kappa_0 \\ -EI\kappa_0 \end{bmatrix} + \begin{bmatrix} \frac{wL^2}{12} \\ -\frac{wL^2}{12} \end{bmatrix}. \quad (\text{A.39})$$

A matriz de constituição global é formada por meio das matrizes elementares posicionadas em blocos na diagonal,

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1^e & 0 & 0 & \dots \\ 0 & \mathbf{C}_2^e & 0 & \dots \\ 0 & 0 & \mathbf{C}_3^e & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{A.40})$$

A.2.4 Resumo das Equações

Em resumo, as equações do modelo linear elástico para estruturas podem ser expressas por

$$\begin{aligned} (\text{Equilíbrio}) \quad \mathbf{p} &= \mathbf{A}^T \mathbf{q}, \\ (\text{Cinemática}) \quad \mathbf{v} &= \mathbf{A} \mathbf{u}, \\ (\text{Constitutiva}) \quad \mathbf{q} &= \mathbf{C} \mathbf{v} + \mathbf{q}_0. \end{aligned} \quad (\text{A.41})$$

Essas equações estão em sua versão global, ou seja, referem-se ao graus de liberdade globais da estrutura. A matriz cinemática e a matriz de equilíbrio se relacionam por $\mathbf{A} = \mathbf{B}^T$. Essa dualidade decorre do princípio dos trabalhos virtuais e é chave na construção do método dos deslocamentos.

No contexto geral da técnica dos elementos finitos o princípio dos trabalhos virtuais enquadra a formulação como uma derivada dos residuais ponderados. Como na mecânica dos sólidos o princípio dos trabalhos virtuais é embutido na dualidade entre a matriz cinemática e de equilíbrio, e conseqüentemente no método dos deslocamentos, diz-se apenas que a formulação se dá por meio da mecânica dos sólidos.

A.2.5 Método dos Deslocamentos

Pelo fato do problema mecânico ser formado pelo acoplamento de três equações e possuir diferentes incógnitas (deslocamentos e esforços), diferentes caminhos podem ser seguidos para a solução do problema. O método dos deslocamentos é um deles. Nesse método a variável primária são os deslocamentos.

Sua formulação consiste em, substituir os esforços na equação de equilíbrio pelas deformações,

$$\mathbf{p} = \mathbf{B}^T \mathbf{C} \mathbf{v} + \mathbf{B}^T \mathbf{q}_0 \quad (\text{A.42})$$

e em seguida utilizar as relações cinemáticas,

$$\mathbf{p} = \mathbf{B}^T \mathbf{C} \mathbf{B} \mathbf{u} + \mathbf{B}^T \mathbf{q}_0. \quad (\text{A.43})$$

Chamando $\mathbf{K} = \mathbf{B}^T \mathbf{C} \mathbf{B}$ de matriz de rigidez global e $\mathbf{p}_0 = \mathbf{B}^T \mathbf{q}_0$ é o vetor das forças que equilibram as *solicitações não mecânicas* geradas por um campo de temperatura, por exemplo. Nesse procedimento a solução é obtida através da rigidez.

$$\mathbf{p} = \mathbf{K} \mathbf{u} + \mathbf{p}_0 \quad (\text{A.44})$$

A matriz de rigidez pode então ser obtida pela produto da matriz cinemática transposta com a matriz constitutiva e pela matriz cinemática novamente. Como a matriz cinemática pode ser formada elemento por elemento e em seguida montada em sua forma global, pode ser feito o mesmo com a matriz de rigidez. Esse procedimento é conhecido como *Método da Rigidez Direta*. O fato de o método dos deslocamentos gerar um conjunto de equações lineares que podem ser resolvidas diretamente por meio de álgebra linear representa uma vantagem uma vez que sistemas lineares são facilmente resolvidos no computador. O fato de se gerar a matriz elemento por elemento também é vantagem pois a implementação computacional é facilitada, uma vez que a matriz cinemática depende apenas de constantes geométricas.

Na formulação variacional, 5.3, a integral da energia de deformação elástica foi resolvida num elemento padrão, em coordenadas naturais, e o resultado foi uma matriz com termos constantes para cada elemento. No caso da formulação por mecânica dos sólidos a mudança de coordenadas para o elemento padrão não é feita uma vez que a matriz cinemática é a responsável por levar as deformações na referência do elemento para a referência global e nela já é embutida os parâmetros geométricos. Como a matriz cinemática pode ser facilmente formada apenas com dados geométricos, não é necessário fazer mudança de coordenada para um elemento padrão.

B Preliminares Matemáticas

B.1 Notação Geral

A notação utilizada nesse trabalho segue as recomendações de (FELIPPA, 2005).

B.1.1 Escalar

Letras Gregas e outras letras minúsculas e maiúsculas.

$$\alpha \tag{B.1}$$

B.1.2 Vetor

Letras minúsculas em negrito e letras gregas em negrito.

$$\mathbf{u}, \boldsymbol{\theta} \tag{B.2}$$

B.1.3 Matrix

Letras maiúsculas em negrito

$$\mathbf{K}, \mathbf{M} \tag{B.3}$$

B.1.4 Tensor de Segunda Ordem

$$\underline{\sigma}, \underline{\varepsilon} \tag{B.4}$$

B.1.5 Tensor de Quarta Ordem

$$\underline{\underline{C}} \quad (\text{B.5})$$

B.2 Diferentes Igualdades

Identidade, vale para todos os valores das variáveis.

$$a^2 - b^2 \equiv (a + b)(a - b) \quad (\text{B.6})$$

Equação, vale apenas para a variável incógnita x .

$$3x^2 - 2x = 5 \quad (\text{B.7})$$

Definição.

$$\frac{df}{dx} \stackrel{\text{def}}{=} \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (\text{B.8})$$

Leis universais, não dependem do material, não contem parâmetros de materiais.

$$\partial_t \rho + \text{div} \mathbf{J}^{\text{law}} \stackrel{\text{def}}{=} 0 \quad (\text{B.9})$$

Leis materiais, dependem de parâmetros de materiais. São leis particulares.

$$\sigma \stackrel{\text{mat}}{=} \mathbf{E} \epsilon \quad (\text{B.10})$$

B.3 Notação Indicial

$$u_i = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (\text{B.11})$$

$$u_{i,i} = u_1 + u_2 + u_3 \quad (\text{B.12})$$

B.4 Calculo Multivariacional

B.4.1 Gradiente de um Escalar

$$\nabla\phi \equiv \phi_{,i} \equiv \frac{\partial\phi}{\partial x_i} \equiv \begin{bmatrix} \frac{\partial\phi}{\partial x_1} \\ \frac{\partial\phi}{\partial x_2} \\ \frac{\partial\phi}{\partial x_3} \end{bmatrix} \quad (\text{B.13})$$

B.4.2 Gradiente de um Vetor

$$\nabla\mathbf{u} \equiv \nabla u_i \equiv u_{i,j} \equiv \frac{\partial u_i}{\partial x_j} \equiv \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3} \end{bmatrix} \quad (\text{B.14})$$

B.4.3 Divergente de um Vetor

$$\text{div}\mathbf{u} \equiv \nabla \cdot \mathbf{u} \equiv u_{i,i} \equiv \sum_{i=1}^3 \frac{\partial u_i}{\partial x_i} \equiv \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3} \quad (\text{B.15})$$

B.4.4 Divergente de um Tensor de Segunda Ordem

$$\text{div}\underline{\sigma} \equiv \nabla \cdot \underline{\sigma} \equiv \sigma_{ij,j} \equiv \sum_{j=1}^3 \frac{\partial \sigma_{ij}}{\partial x_j} \equiv \begin{bmatrix} \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} + \frac{\partial \sigma_{13}}{\partial x_3} \\ \frac{\partial \sigma_{21}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} + \frac{\partial \sigma_{23}}{\partial x_3} \\ \frac{\partial \sigma_{31}}{\partial x_1} + \frac{\partial \sigma_{32}}{\partial x_2} + \frac{\partial \sigma_{33}}{\partial x_3} \end{bmatrix} \quad (\text{B.16})$$

B.5 Classificação das Variáveis

A necessidade de definir atributos quantitativos às observações físicas liga a ciência física à matemática.

Quantidades físicas são valores numéricos associados à uma dimensão física, ou unidade física. Essas quantidades podem ser classificadas de acordo com a teoria física a qual pertencem, a natureza matemática ou se são constantes, parâmetros ou variáveis. A natureza matemática engloba escalares, vetores, tensores e outros. Um parâmetro é uma quantidade física característica de um material, sistema ou processo que é constante no contexto em que aparece, mas varia em outras circunstâncias. As variáveis são todas as quantidades físicas que não são nem constantes nem parâmetros, (TONTI, 2013). As variáveis físicas podem ser classificadas de acordo com o *papel que desempenham* na teoria física ou sua natureza *global ou local*. Os papéis desempenhados pelas variáveis físicas nos permite classificá-las como variáveis de *configuração, fonte e energia*.

B.5.1 Variáveis de Fonte

De acordo com (TONTI, 2013), variáveis de estado são *todas as variáveis que descrevem a fonte de um campo; a causa de um fenômeno e todas as variáveis ligadas à elas por meio de operações de soma, diferença, divisão por um comprimento, uma área, um volume ou um intervalo; por uma operação de limite, e conseqüentemente derivada espacial e temporal; por integrais de linha, superfície, volume e integrais temporais*. Todas essas operação não podem conter constantes físicas.

Em mecânica dos meios contínuos, são variáveis de fonte: todas as variáveis da estática e dinâmica, por exemplo, tensão, momento, torque, tensão de superfície e força de corpo.

Em termodinâmica , são variáveis de fonte: entropia.

B.5.2 Variáveis de Configuração

De acordo com (TONTI, 2013), variáveis de configuração são *todas as variáveis que descrevem a configuração de um sistema físico, em particular, o potencial de campos. Pertencem à essa classe todas as variáveis ligadas a um potencial por operações de soma, diferença, divisão por um comprimento, uma área, um volume ou um intervalo; por uma operação de limite, e conseqüentemente derivada espacial e temporal; por integrais de linha, superfície, volume e integrais temporais*. Todas essas operação não podem conter constantes físicas.

A mesma variável de configuração pode apresentar papéis diferentes em teorias distintas. Por exemplo, a massa, em mecânica das partículas é um parâmetro física e em mecânica dos fluido é uma variável de fonte.

Em mecânica dos meios contínuos, as variáveis de configuração são todas as variáveis cinemáticas e geométricas.

B.5.3 Variáveis de Energia

As variáveis de energia são *todas as variáveis obtidas pela multiplicação de uma variável de configuração por uma variável de fonte e as variáveis ligadas à essas por meio de operações de soma, diferença, divisão por um comprimento, uma área, um volume ou um intervalo; por uma operação de limite, e conseqüentemente derivada espacial e temporal; por integrais de linha, superfície, volume e integrais temporais*

B.6 Equações Físicas

São exemplos de equações usadas em físicas:

- Equação constitutiva
- Equação material
- Equação de balanço
- Equação de compatibilidade
- Equação de campo
- Equação de conservação
- Equação de continuidade
- Equação de movimento
- Equação de restrição

As equações podem ser divididas, de acordo com (TONTI, 2013), em cinco classes:

1. **Equações Definidoras** definem novas variáveis em termos de variáveis conhecidas.
2. **Equações Topológicas** ligam o valor de um variável associada com um elemento espacial, linha, superfície ou volume, com outra variável associada à fronteira. Não dependem de parâmetros do meio material. Inclui equações de balanço, continuidade, conservação, circulação.
3. **Equações Comportamentais** especificam um comportamento particular de um fenômeno, ou uma classe particular de transformação. Não contém constantes físicas. Especificam transformação, movimento ou materiais por meio de condições nas variáveis físicas. Um exemplo é a condição de fluido incompressível, $\nabla \cdot \mathbf{v} = 0$. Outro exemplo é a condição estacionária, $\partial_t \mathbf{v} = 0$.
4. **Equações Constitutivas** ligam variáveis de configuração com variáveis de fonte da mesma teoria. Elas são caracterizadas por conter parâmetros físicos que dependem do meio material. Possuem caráter empírico, portanto são *fenomenológicas*. Exemplos são a lei de Hooke e a lei de Ohm.

5. **Equações de Interação** ligam variáveis de configuração com variáveis de fonte de outra teoria. Podem ser *irreversíveis* como no caso termoelástico ou *reversíveis* como no caso piezoelétrico.

B.7 Teorema da Divergência

O divergente de uma função vetorial $\mathbf{v}(\mathbf{P})$ é uma escalar obtida por:

1. Em uma região definida por uma superfície S , considerar o *fluxo* através do contorno dessa região,

$$\phi = \int_S \mathbf{v} \cdot d\mathbf{a}. \quad (\text{B.17})$$

onde, $d\mathbf{a}$ é o vetor normal a um elemento infinitesimal de área.

2. Dividindo o volume definido pela região S em partes menores, o fluxo total da função $\mathbf{v}(\mathbf{P})$ continua o mesmo, pois, em superfícies internas, resultado da divisão do volume, o fluxo da função em um dos lados é o mesmo, porém com sinal contrário, pois o vetor $d\mathbf{a}$ aponta em sentido oposto. Portanto, escreve-se

$$\phi = \sum_{i=1}^n \phi_i \quad (\text{B.18})$$

ou seja, o fluxo total é o fluxo referente as superfícies dos volumes resultantes da divisão. No caso, o volume original foi dividido em n volumes menores.

3. Como o fluxo diminui caso a região diminui de tamanho, toma-se a razão entre esses dois valores a fim de encontrar um valor finito.
4. Calcular a *razão* entre o *fluxo* e o *volume* da região.
5. Calcular o *limite* da razão quando a região é reduzida ao ponto.

$$\text{div} \mathbf{v} \stackrel{\text{def}}{=} \lim_{V_i \rightarrow 0} \frac{1}{V_i} \int_{S_i} \mathbf{v} \cdot d\mathbf{a}_i \quad (\text{B.19})$$

Nesse ponto encontra-se a escalar $\psi(\mathbf{P}) = \text{div} \mathbf{v}(\mathbf{P})$. Essa demonstração pode ser encontrada em (PURCELL, 2013). Pode-se dizer que o *divergente* surge de um valor escalar associado a uma *superfície*, (TONTI, 2013).

B.8 Regra do Produto do Cálculo Multivariável

A regra de produto para a derivada unidimensional é dada por

$$(vu)' \equiv v'u + vu' \quad (\text{B.20})$$

Estendendo essa propriedade para o caso multivariável tem-se

$$(\sigma_{ij}v_i)_{,j} \equiv \sigma_{ij,j}v_i + \sigma_{ij}v_{i,j}, \quad \text{ou} \quad \nabla \cdot (\underline{\sigma}\mathbf{v}) \equiv (\nabla \cdot \underline{\sigma})\mathbf{v} + (\nabla\mathbf{v}) : \underline{\sigma} \quad (\text{B.21})$$

onde,

$$\nabla \cdot \underline{\sigma} \equiv \sigma_{ij,j} \quad \text{e} \quad \nabla\mathbf{v} \equiv v_{i,j}. \quad (\text{B.22})$$

Essa relação pode ser verificada para o primeiro termo, $i = j = 1$, por exemplo,

$$\partial_x(\sigma_{11}v_1) = (\partial_x\sigma_{11})v_1 + \sigma_{11}(\partial_xv_1) \quad (\text{B.23})$$

onde $\partial_x \equiv \partial/\partial x$

B.9 Regra da Cadeia

A regra da cadeia para funções multivariáveis é dada por,

$$\frac{\partial}{\partial x} f(\xi, \eta) \stackrel{\text{def}}{=} \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial f}{\partial \eta} \frac{\partial \eta}{\partial x} \quad (\text{B.24})$$

B.10 Produto Duplo de Tensores

O produto duplo entre tensores é definido como

$$\underline{\mathbf{E}} : \underline{\boldsymbol{\varepsilon}} \equiv E_{ijkl}\varepsilon_{kl} \equiv \sum_k \sum_l E_{ijkl}\varepsilon_{kl} \quad (\text{B.25})$$

onde o subscrito kl repetido indica contração dupla, nesse caso o resultado é um tensor de segunda ordem.

B.11 Kronecker Delta

Por definição $\delta_{ij} = 1$ para $i = j$ e $\delta_{ij} = 0$ para $i \neq j$.

B.12 Introdução ao Cálculo Variacional

O cálculo variacional preocupa-se com a determinação de máximos e mínimos. Um funcional é definido como uma função de outras funções. O procedimento para encontrar uma solução que faz o estacionário por meio de tentativas é chamado de *cálculo de variações*.

Dado o funcional

$$\Pi = \int_{\Omega} I \left(x, \phi(x), \frac{d\phi(x)}{dx} \right) d\Omega \quad (\text{B.26})$$

objetiva-se encontrar $\phi(x)$ que faz o funcional Π ser estacionário (ou mínimo). Por meio de uma tentativa de função $\bar{\phi}$, que pode ser expressa por

$$\bar{\phi}(x) = \phi(x) + \delta\phi(x) \quad (\text{B.27})$$

onde, $\delta\phi(x)$ é uma variação em ϕ , ou seja, uma mudança arbitrária, infinitesimal, para um valor fixo em x . O operador δ é chamado de operador variacional. Definindo a variação de um funcional como

$$\delta I = \frac{\partial I}{\partial \phi} \delta\phi + \frac{\partial I}{\partial \phi_x} \delta\phi_x + \frac{\partial I}{\partial \phi_{xx}} \delta\phi_{xx} + \frac{\partial I}{\partial x} \delta x \quad (\text{B.28})$$

onde $\phi_x = \frac{\partial \phi}{\partial x}$. Essa expressão é semelhante à expressão do diferencial de uma função, $df = \frac{df}{dx} dx$, no entanto, multivariável. Como deseja-se encontrar a variação de I para um valor fixo em x , implica em $\delta x = 0$.

A variação em $\delta\Pi$ correspondente a variações na solução ϕ é dada por

$$\delta\Pi = \int_{\Omega} \delta I dx \quad (\text{B.29})$$

para ser estacionária, a variação $\delta\Pi = 0$.

$$\delta\Pi = \int_{\Omega} \left(\frac{\partial I}{\partial \phi} \delta\phi + \frac{\partial I}{\partial \phi_x} \delta\phi_x + \frac{\partial I}{\partial \phi_{xx}} \delta\phi_{xx} \right) dx = 0 \quad (\text{B.30})$$

Integrando por parte os termos $\frac{\partial I}{\partial \phi_x} \delta\phi_x$ e $\frac{\partial I}{\partial \phi_{xx}} \delta\phi_{xx}$ e usando a propriedade comutativa do operador variacional, $\delta\left(\frac{d\phi}{dx}\right) = \frac{d}{dx}(\delta\phi)$, chega-se a

$$\begin{aligned} \delta\Pi = \int_{\Omega} \left[\frac{\partial I}{\partial \phi} - \frac{d}{dx} \left(\frac{\partial I}{\partial \phi_x} \right) + \frac{d}{dx^2} \left(\frac{\partial I}{\partial \phi_{xx}} \right) \right] \delta\phi d\Omega \\ + \left[\frac{\partial I}{\partial \phi} - \frac{d}{dx} \left(\frac{\partial I}{\partial \phi_{xx}} \right) \right] \delta\phi \Big|_{\Omega} + \left[\left(\frac{\partial I}{\partial \phi_{xx}} \right) \delta\phi_x \right] \Big|_{\Omega} = 0. \end{aligned} \quad (\text{B.31})$$

Como a variação $\delta\phi$ é arbitrária, cada termo fatorado por ela deve ser consequentemente nulo.

$$\frac{\partial I}{\partial \phi} - \frac{d}{dx} \left(\frac{\partial I}{\partial \phi_x} \right) + \frac{d}{dx^2} \left(\frac{\partial I}{\partial \phi_{xx}} \right) = 0 \quad (\text{B.32})$$

$$\left. \frac{\partial I}{\partial \phi} - \frac{d}{dx} \left(\frac{\partial I}{\partial \phi_{xx}} \right) \right|_{\Omega} = 0 \quad (\text{B.33})$$

$$\left. \frac{\partial I}{\partial \phi_{xx}} \right|_{\Omega} = 0. \quad (\text{B.34})$$

A primeira equação é a forma forte do modelo, em sua forma de equação diferencial, chamada de *Equação de Euler* ou *Euler-Lagrange*. As demais duas equações são as condições de contorno.

B.13 Quadratura Gaussiana

Existem diversos métodos para integração. O método da Quadratura Gaussiana é considerado o mais conveniente na implementação do Método dos elementos finitos, (RAO, 2011, p. 148).

Dado a seguinte integral,

$$I = \int_{-1}^1 f(x) dx, \quad (\text{B.35})$$

que representa a área abaixo da função $f(x)$ no intervalo $[-1, 1]$. A forma mais simples de se aproximar essa área é escolher um ponto médio, f_m entre $f(-1)$ e $f(1)$, e multiplicar essa altura por uma base de tamanho $1 - (-1) = 2$. Esse resultado,

$$I \approx 2f_m, \quad (\text{B.36})$$

representa a área de um retângulo de base 2 e altura f_m .

Outro método para aproximar a área abaixo da curva é o método trapezoidal,

$$I \approx \frac{(f(-1) + f(1))2}{2} \equiv 1f(-1) + 1f(1) \quad (\text{B.37})$$

que representa a área de um trapézio.

Generalizando essa idéia, em vez de escolher os pontos extremos, como no caso do trapézio, ou o ponto médio; escolhe-se pontos de tal forma que o erro seja mínimo. Sendo o erro a diferença entre o valor da integral e o valor da aproximação. Nesse caso,

$$I \approx \sum_{i=1}^{ngp} w_i f(x_i) \quad (\text{B.38})$$

onde, w_i são pesos, ngp é o número de pontos de integração e x_i são pontos específicos onde a função é calculada.

Os pontos onde as funções são calculadas são obtidos como as raízes do polinômio de Legendre.

Em duas dimensões procede-se da mesma maneira calculando a integral em cada variável separadamente.

C Aspectos Computacionais

Nesse apêndice serão tratados tópicos relacionados à linguagem de programação Python.

C.1 Python

No presente trabalho, os desenvolvimentos dos programas foram efetuados utilizando a linguagem de programação Python. A linguagem foi criada por Guido Van Rossum, em fevereiro de 1991. Dentre suas características, pode-se destacar:

1. **Python é interpretada.** Em Python, não há a etapa de compilação o que facilita o processo de verificação e correção do código.
2. **Python é orientada-objeto.** O paradigma de programação orientado ao objeto consiste na utilização de *classes* e *objetos*. Esses elementos são utilizados para, por meio de *abstração*, reunir um conjunto de dados e procedimentos em um conjunto lógico e apropriado ao problema específico.
3. **Python é de alto-nível com semântica dinâmica:** O alto nível se deve ao fato de que sua sintaxe é distante do código de máquina e mais próxima da linguagem humana. A característica de ser dinâmica implica que o interprete é capaz de discernir o tipo da variável no momento da execução. No caso estático, as variáveis devem ter o tipo declarado no momento de sua criação. Pelo fato de ser dinâmica é necessário cuidado nas manipulações exercidas em uma variável para evitar complicações no entendimento da rotina.

Em Python, todos os elementos da linguagem são *objetos*. Objeto é o elemento primário na construção das rotinas. Eles são criados por meio de *classes*. As *classes* contém instruções para a criação de um objeto. Uma classe pode gerar quantos objetos forem necessários, diz-se que cada objeto é uma *instância* da classe. As *classes* são uma ferramenta útil na abstração de um conjunto de dados e métodos de forma que estes sejam contidos em uma cápsula lógica, ou seja, o conjunto apresenta um sentido quando

agrupado. A cápsula semântica consiste na nomenclatura dada ao objeto *instanciado pela classe*. Como exemplo pode-se citar a classe `Model`, apresentada a seguir, que é responsável por conter os *atributos*, ou características, do modelo estrutural estudado.

A linguagem apresenta alguns pontos positivos:

1. Possui sintaxe simples e de fácil aprendizado.
2. É livre e grátis. Livre no sentido de se ter o acesso ao código fonte e grátis quanto ao custo.
3. Possui comunidade crescente e ativa. A comunidade é responsável por criar um gigantesco banco de dados com soluções de problemas e responsável pela criação de bibliotecas que podem ser utilizadas como dependência do seu programa. Alguns exemplos de pacotes de terceiros: *numpy*, *matplotlib*, *simpy*, *scipy*.
4. É multiplataforma. Um código em Python pode ser interpretado sem problemas em todos os sistemas operacionais.
5. Possui um console interativo. A interatividade com a *shell* permite a rápida prototipagem e realização de testes. Além disso também apresenta *debugger* interativo o que facilita o encontro de erros.

C.2 Softwares em Python

Existem atualmente uma série de programas livres e comerciais desenvolvidos para solução de equações diferenciais utilizando o método dos elementos finitos. Dentre os *softwares* livres mais conhecidos estão: Code-Aster, FeniCS e CalculiX. O projeto FEniCS, (ALNAES et al., 2015), possui uma coleção de componentes que operam em conjunto. Dentre esses componentes, está o DOLFIN, que é escrito em C++ e Python. Softwares escritos exclusivamente em Python para soluções de equações diferenciais não são comuns, em parte devido a problemas de performance. O programa SfePy, (CIMRMAN, 2014), foi escrito quase na totalidade em Python. Apenas rotinas que demandavam processamento elevado foram escritas em C.

O programa SfePy, *simple finite elements in python*, funciona da seguinte forma:

1. O problema é definido por meio de um arquivo de *input* que contém:
 - a) O nome do arquivo com a malha em uma *string*;
 - b) As propriedades dos materiais em um dicionário;
 - c) As regiões características do domínio em um dicionário no qual é necessário especificar por meio de *strings* a geometria das regiões;

- d) O campo que será aproximada em um dicionário que especifica o domínio em questão, a regiões onde será aproximada e outros parâmetros;
- e) As variáveis de interesse e a função teste também em um dicionário;
- f) As condições de contorno em um dicionário que utiliza as chaves definidas no dicionário das regiões;
- g) A própria equação em um dicionário cuja chave é o nome do problema e o valor uma *string* com a equação na forma fraca, as equações são construídas por meio de termos matemático pré-estabelecidos e documentados em forma de tabela;
- h) O *solver* utilizado um dicionário com valores especificando qual método para resolver o problema linear e não linear;

2. Finalmente, invoca-se o programa através da linha de comando.

O programa permite uma grande flexibilidade na definição do modelo matemático a ser resolvido, no entanto, introduz muita complexidade na criação do arquivo de entrada. O programa possui uma série de definições como por exemplo os termos para formar equações. Isso é feito para que o programa seja o mais geral possível, contudo, recai sobre o usuário o conhecimento das extensa lista de funções e parâmetros a serem declarados. A malha de elementos é produzida por outro programa e inserida no arquivo de entrada. O programa reconhece diferentes formatos para descrição da malha.

A idéia básica por trás do desenvolvimento das bibliotecas em Python nesse trabalho é a de modularidade. Em vez de criar um programa complexo que recebe uma extensa quantidade de parâmetros cria-se um conjunto de pacotes que resolvem um problema em particular.

C.3 Performance em Python

A linguagem Python é comumente atribuída a característica de ser lenta. Essa característica se deve ao fato de Python ser uma linguagem dinâmica e interpretada. O fato de ser dinâmica indica que o *tipo* das variáveis não é declarado no momento de sua criação. Em C, por exemplo, as variáveis possuem seu tipo definido no momento de sua criação. Isso faz com que o compilador saiba de antemão como a variável se comporta baseada em seu tipo, (VANDERPLAS, 2014).

Como tudo em Python é tratado com um objeto, no momento de execução sabe-se apenas que a variável é um objeto. Nesse ponto, o interprete busca informações contidas no objeto para prosseguir a rotina. Enquanto um *array* em C acessa diretamente os dados, uma lista em Python possui referencias a ponteiros que referenciam, cada um, um objeto que finalmente possuem referência ao dado. Esse acesso à memória é uma das causas

de lentidão na execução dos programas, (VANDERPLAS, 2014). Uma simples solução a esse problema de acesso à memória é a utilização da biblioteca Numpy, (van der Walt; COLBERT; VAROQUAUX, 2011).

Atualmente existem diversas bibliotecas cujo objetivo é contornar o problema da performance em Python. As principais são: Cython, (TEAM,), e Numba, (ANALYTICS,). Cython é um compilador estático de otimização para o Python, com ele o código em Python é convertido para C. Numba é um pacote que permite acelerar a execução de funções escritas diretamente em Python. Funções que realizam operações matemáticas que demandam alta capacidade de processamento podem ser compiladas *just-in-time* com a inserção de um *decorator* @jit antes da função. A vantagem do Numba é a não necessidade de se alterar a linguagem nem mudar interpretes.

C.4 Programação Orientada Objeto

O paradigma presente nas primeiras implementações do Método dos Elementos Finitos baseava-se em um conjunto de procedimentos que guiavam os dados em uma sequência lógica. Essa abordagem clássica na produção de programas, *orientada a procedimento*, consiste em uma simples sequência de chamadas de funções que operam nos dados. A vantagem nessa abordagem é a simples e *direta* conexão dos procedimentos com os dados. Por um lado, essa conexão é interessante na construção de programas teste utilizados para verificar se a metodologia produz resultados corretos. No entanto, a ligação entre os dados e o procedimento impõe dificuldades na extensão do código. Dessa maneira, caso exista um interesse em expandi-lo, deve-se conhecer toda a rotina de execução para que o novo código se adeque aos procedimentos criados anteriormente. Tarefa não fácil, pois o código só tende a crescer e com isso sua manutenção fica inviável.

Como vantagem da lógica de procedimentos pode-se citar a modularidade do programa na divisão das tarefas em diversas funções que desempenham um papel específico em um dado de entrada e retornam um resultado. Essa filosofia de modularidade junto com a produção de módulos em Python facilita a compreensão da rotina. Com exemplo de desvantagem pode-se citar a criação de um novo para ser utilizado na análise. Caso as funções do programa não estejam preparadas para diferentes elementos, essa implementação requererá ajuste de toda a rotina.

Os primeiros programas produzidos para solução através dos elementos finitos foram programados seguindo esse paradigma, (ARCHER, 1996). A linguagem geralmente utilizada pela comunidade científica naquela época era o FORTRAN. Os programas continham milhares de linha de código e o conjunto de informações que define o problema estava intimamente ligado com a estrutura da dados do programa. Modificações exigiam

alto conhecimento do código em toda sua extensão. Os procedimentos para solução nesses casos são conhecidos como *hard coded*. O código gerado passa a ser inflexível, de difícil manutenção e aprimoramento.

Programação orientada a objeto consiste em uma maneira de se organizar a estrutura de um programa. Por meio da abstração dos objetos essenciais ao programa em *instâncias de classes* é possível criar uma modularidade ao programa, permitindo a sua expansão no futuro sem com que o novo código interfira no funcionamento do antigo. Os *objetos* então são containers que contém informações e métodos de forma que o conjunto faça sentido como uma entidade independente. Esse paradigma de *programação orientada objeto* possui quatro conceitos fundamentais: *classes*, *objetos*, *herança* e *polimorfismo*.

O par *classe* e *objeto* formam o cerne desse paradigma. As classes são responsáveis por definir objetos, que são ditos *instâncias da classe*. A criação de classes envolve o processo de *abstração*. Abstração, em programação, consiste em isolar características essenciais dos elementos, no caso, variáveis e funções. No momento em que um conjunto de informações passa a ter um sentido, defini-se a entidade objeto que encapsula essas informações num container único. Além de informações, um conjunto de métodos, que são funções, pode fazer parte de um objeto. O código abaixo mostra um exemplo de uma classe que cria um objeto para o modelo estrutural.

```

1  import numpy as np
2
3  class Model(object):
4      """Builds a Model object"""
5      def __init__(self):
6          self.ne = 4          # number of elements
7          self.nn = 9          # number of nodes
8          self.ndof = 18      # number of degrees of freedom
9          self.XYZ = np.array([[0, 0],
10                             [1, 0],
11                             [1, 1],
12                             [0, 1],
13                             [.5, 0],
14                             [1, .5],
15                             [.5, 1],
16                             [0, .5],
17                             [.5, .5]])
18          self.CONN = np.array([[0, 4, 8, 7],
19                               [4, 1, 5, 8],
20                               [8, 5, 2, 6],
21                               [7, 8, 6, 3]])
22          self.DOF = np.array([[0, 1, 8, 9, 16, 17, 14, 15],
23                               [8, 9, 2, 3, 10, 11, 16, 17],
24                               [16, 17, 10, 11, 4, 5, 12, 13],
25                               [14, 15, 16, 17, 12, 13, 6, 7]])
26          self.surf_of_ele = [0, 0, 0, 0] # surface 0 for all elements
27          self.physical_surf = [0]       # physical surfaces tag
28          self.TYPE = [3, 3, 3, 3]
29
30  model = Model()                    # Instantiate a Model object

```
