

Introdução ao Método dos Elementos Finitos

Notas de aula PPGEM

Prof. Eduardo Lenz Cardoso



Conteúdo

I	Introdução ao Método dos Elementos Finitos	
1	Introdução	9
2	Equações diferenciais	11
2.1	Classificação e Exemplos	11
2.2	Equação de condução de calor 1D	12
2.2.1	Procedimento de solução para a equação de calor 1D	13
2.3	Modelo de barra	15
2.4	Uma breve revisão de gradiente e divergente	18
2.4.1	Gradiente	18
2.4.2	Divergente	19
2.5	Condução de calor 2D	19
2.6	Equação de equilíbrio de sólidos em 2D	21
2.7	Equações Diferenciais como Operadores	22
2.8	Equações Diferenciais Parciais Lineares	23
3	Método dos Resíduos Ponderados	25
3.1	Produto interno no espaço das funções	25
3.1.1	Convergência fraca	26
3.2	Método dos Resíduos Ponderados	26

4	Relaxação dos requisitos de continuidade	39
4.1	Integração por partes	39
4.1.1	Produto de funções de uma variável	39
4.1.2	Divergente do produto de um campo escalar por um campo vetorial	40
4.2	Utilizando a integração por partes com a forma fraca	41
4.2.1	Equação de quarta ordem da viga longa	43
4.3	Métodos de Galerkin	46

II

Elementos unidimensionais

5	Elemento finito de barra	51
5.1	Dedução do elemento finito de barra	52
5.1.1	Notação matricial	58
5.2	Nomenclatura e interpretação dos termos	59
5.3	Obtendo a matriz de rigidez por equilíbrio de forças	60
5.4	Simetria da matriz de rigidez	61
5.4.1	Sobreposição de barras	62
5.4.2	Automatização do procedimento de sobreposição de elementos	65
6	Elemento de barra: abordagem energética	67
6.1	Princípio da mínima energia potencial	67
6.2	Interpretando como um problema de perturbação da condição de equilíbrio - Princípio dos trabalhos virtuais (PTV)	69
6.3	Utilizando o PMEP para obter o elemento de barra	69
6.4	Interpretando a solução do MEF como um problema de otimização	71
7	Rotação de sistemas de referência	73
8	Elemento de barra em 2D - treliças	77
8.1	Treliças bidimensionais	78
8.1.1	Mapeamento local-global	80
8.1.2	Numeração dos graus de liberdade	81
9	Implementação computacional - treliças	83
9.1	Etapas de uma análise por elementos finitos	83
9.2	Entrada de dados	84
9.3	Pré-processamento	85
9.3.1	Montagem da matriz de rigidez global	86
9.3.2	Montagem do vetor global de forças concentradas	87
9.3.3	Aplicação das condições de contorno de Dirichlet (apoios)	87
9.3.4	Pós-processamento: cálculo dos esforços internos	90
10	Elemento finito de viga longa	93
10.1	Elemento Finito de Viga Longa	93

10.2	Sobreposição de elementos de viga	99
11	Elemento Finito de Pórtico Plano	101
11.1	Algoritmos para implementação computacional	103
11.2	Exemplo de pórtico plano	106
12	C.C. de Dirichlet não homogêneas	109
12.1	Técnica 1 - Rearranjo do sistema de equações	109
12.2	Técnica 2 - Lagrangiano	111

III

Elementos para elasticidade

13	Uma breve revisão de elasticidade linear	119
13.1	Relações entre deslocamentos e deformações	119
13.2	Relações entre deformações e tensões	120
13.3	Estados planos	121
13.3.1	Estado plano de tensão - EPT	122
14	Elementos Finitos aplicados à Elasticidade	125
14.1	Método dos resíduos ponderados	125
14.2	Método dos elementos finitos	127
14.3	Pós-processamento	128
15	Elemento bilinear isoparamétrico de 4 nós	129
15.1	Funções de interpolação	129
15.2	Cálculo das derivadas e mapeamento isoparamétrico	131
15.3	Cálculo das integrais de domínio	134
15.4	Cálculo das integrais de contorno	136
15.4.1	Aresta 1	136
15.4.2	Aresta 2	139
15.4.3	Aresta 3	140
15.4.4	Aresta 4	141
15.5	Cálculo de tensões	143
15.6	Exemplo	144
16	Cisalhamento parasita	151
16.1	Modos de deformação de um elemento finito	153
16.2	Problemas dominados por flexão	156
17	Implementação computacional	159
17.1	Entrada de dados	159
17.2	Montagem da matriz de rigidez local de um elemento	160
17.3	Montagem da matriz de rigidez global	160

17.4	Montagem do vetor de forças distribuídas	163
17.5	Cálculo de tensões em um elemento	167

IV

Exemplos de implementação computacional na linguagem Julia

18	Treliças 2D	171
19	Pórtico plano	177
20	Bilinear isoparamétrico de 4 nós	183



Introdução ao Método dos Elementos Finitos

1	Introdução	9
2	Equações diferenciais	11
2.1	Classificação e Exemplos	
2.2	Equação de condução de calor 1D	
2.3	Modelo de barra	
2.4	Uma breve revisão de gradiente e divergente	
2.5	Condução de calor 2D	
2.6	Equação de equilíbrio de sólidos em 2D	
2.7	Equações Diferenciais como Operadores	
2.8	Equações Diferenciais Parciais Lineares	
3	Método dos Resíduos Ponderados	25
3.1	Produto interno no espaço das funções	
3.2	Método dos Resíduos Ponderados	
4	Relaxação dos requisitos de continuidade	39
4.1	Integração por partes	
4.2	Utilizando a integração por partes com a forma fraca	
4.3	Métodos de Galerkin	

1. Introdução

Este texto compila minhas notas de aula associadas ao curso de Introdução ao Método dos Elementos Finitos, principalmente no contexto da disciplina do PPGEM/UDESC. Este texto tem como objetivo auxiliar o processo de ensino e não tem como objetivo substituir os livros sobre o assunto. Jamais utilize somente uma bibliografia para estudar qualquer assunto!

O método dos elementos finitos é um método para a solução de equações diferenciais. Assim, iniciamos o texto com uma breve revisão de Equações diferenciais e também deduzimos algumas equações diferenciais de interesse. Em seguida, apresentamos o método dos resíduos ponderados, que é a base matemática utilizada neste texto para a dedução do método dos elementos finitos. Maiores detalhes sobre as equações de elasticidade e relações constitutivas são apresentadas em um apêndice sobre o assunto. Aconselha-se ao leitor não familiarizado com o assunto a consultar o apêndice antes de ler os capítulos sobre a aplicação do método para problemas de elasticidade.

Por fim, sugere-se a leitura detalhada das minhas notas de aula de Fundamentos de matemática, tanto a parte de análise quanto a de cálculo numérico, bem como a literatura sugerida no plano de aula da disciplina.

2. Equações diferenciais

2.1 Classificação e Exemplos

Equação diferencial é uma equação que apresenta derivadas ou diferenciais de uma função desconhecida (chamada de variável dependente). Se a função a determinar depender de apenas uma variável independente, dizemos que a equação diferencial é **ordinária** (EDO), podendo ser escrita na forma

$$F(x, y(x), y'(x), \dots, y^n(x)) = 0$$

com x sendo a variável independente e $y(x)$ a variável dependente. Se a função a determinar depender de mais de uma variável independente a equação diferencial é dita **parcial** (EDP), podendo ser escrita na forma

$$F(x, y, z, w(x, y, z), w^x(x, y, z), \dots) = 0$$

onde x , y e z são variáveis independentes e $w(x, y, z)$ é a variável dependente.

Quanto a classificação, dizemos que a **ordem** de uma equação diferencial está associada a mais alta ordem de diferenciação que existe na equação diferencial. Assim, em ordem crescente

$$\frac{dy(x)}{dx} = -a$$

temos uma EDO de primeira ordem e

$$\frac{d^2y(x)}{dx^2} + \frac{dy(x)}{dx} = y(x)$$

uma EDO de segunda ordem.

O **grau** de uma equação diferencial é o expoente a qual está elevado a derivada de maior ordem da equação diferencial. Assim, na equação

$$\frac{d^2y(x)}{dx^2} + \left(\frac{dy(x)}{dx}\right)^2 = y(x)$$

é de primeiro grau, pois a maior derivada (ordem) está sendo elevada a 1. Desta forma, a equação

$$\left(\frac{d^2y(x)}{dx^2}\right)^3 + \left(\frac{dy(x)}{dx}\right)^2 = y(x)$$

é de terceiro grau.

Uma equação diferencial é dita **linear** se tem a forma

$$C_n(x)\frac{d^n f(x)}{dx^n} + C_{n-1}(x)\frac{d^{n-1} f(x)}{dx^{n-1}} + \dots + C_0(x)f(x) = g(x)$$

onde os coeficientes são funções lineares de x e a equação diferencial é de primeiro grau. Se $g(x)$ for nulo, a equação é dita **homogênea linear**.

Existem diversas equações diferenciais que tem sentido físico e que precisam ser bem compreendidas. No que segue, vamos apresentar algumas deduções de equações diferenciais.

2.2 Equação de condução de calor 1D

Um exemplo muito simples, mas importante, de equação diferencial é a condução de calor unidimensional. Seja uma barra perfeitamente isolada nas direções y e z ao longo do seu comprimento L , como ilustrado na Figura 2.1.

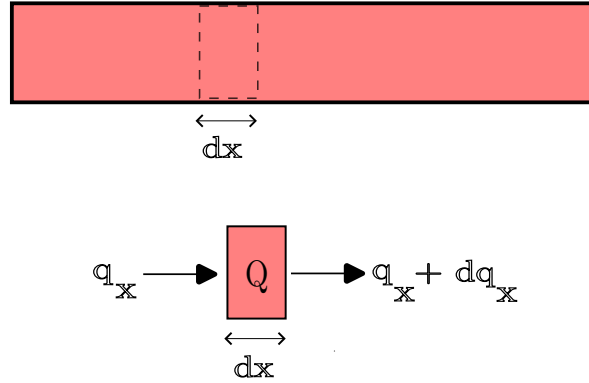


Figura 2.1: Condução de calor unidimensional.

Se extrairmos um elemento diferencial $dx [m]$ ao longo do comprimento, podemos realizar um balanço de fluxo de calor. Para isto, iremos considerar o fluxo de calor que entra no volume diferencial, $q_x [W/m^2]$ e o fluxo de calor que sai, $q_x + dq_x$. O termo dq_x é utilizado para levarmos em consideração os acréscimos associados ao termo de geração volumétrica de calor, $Q [W/m^3]$ que pode estar presente no interior do corpo. Assim, no volume diferencial em uma posição x , teremos

$$-q_x(x) dA + (q_x(x) + dq_x(x)) dA + Q(x) dV = 0 \quad [W] \quad (2.1)$$

como condição para equilíbrio de fluxos (aqui foi adotada a notação de sinais usual no teorema da divergência, onde o que "entra" no volume é negativo). Nesta equação temos que $dA = dydz$ e $dV = dx dy dz$. Simplificando a equação e dividindo todos os termos restantes por dV obtemos

$$\boxed{\frac{dq_x(x)}{dx} + Q(x) = 0} \quad [W/m^3] \quad (2.2)$$

que é uma Equação Diferencial ordinária, de primeira ordem e primeiro grau, onde a variável dependente é o fluxo $q_x(x)$ e a variável independente é a posição x .

Se quisermos obter o campo de temperaturas, $T(x)$, devemos modificar essa equação, inserindo a relação entre o fluxo de calor e a temperatura. Utilizando a lei de Fourier,

$$q_x(x) = -k(x) \frac{dT(x)}{dx} \quad (2.3)$$

tal que

$$\boxed{\frac{d}{dx} \left(-k(x) \frac{dT(x)}{dx} \right) + Q(x) = 0} \quad (2.4)$$

que é uma EDO de segunda ordem e de primeiro grau, sendo que agora a temperatura $T(x)$ é a variável dependente. A lei de Fourier, que relaciona um fluxo à outra grandeza, por meio de uma ou mais propriedades do material, é comumente chamada de **relação constitutiva**.

2.2.1 Procedimento de solução para a equação de calor 1D

A solução geral da Equação (2.4) pode ser obtida por uma dupla integração. Por simplicidade, vamos considerar a condutividade térmica $k(x)$ constante. A equação de equilíbrio se torna

$$\frac{d}{dx} \left(-k \frac{dT(x)}{dx} \right) = -Q(x) \quad (2.5)$$

que integrada uma vez resulta em

$$-k \frac{dT(x)}{dx} = - \int_0^L Q(x) dx + C_1 \quad (2.6)$$

onde C_1 é uma constante de integração. É interessante notar que o l.h.s da equação é o próprio fluxo (Lei de Fourier), tal que

$$q_x(x) = - \int_0^L Q(x) dx + C_1. \quad (2.7)$$

Por fim, integramos mais uma vez, obtendo

$$-kT(x) = - \int_0^L \int_0^L Q(x) dx dx + C_1 x + C_2 \quad (2.8)$$

ou

$$\boxed{T(x) = \frac{1}{k} \left(\int_0^L \int_0^L Q(x) dx dx - C_1 x - C_2 \right)} \quad (2.9)$$

em que C_2 é outra constante de integração. A solução obtida na Eq. (2.9) mostra uma característica interessante na solução de equações diferenciais: a solução é na verdade uma família de soluções, que serão particularizadas devido as **condições de contorno** do problema em análise (não confundir equação com problema: a equação é a geral de condução de calor e o problema depende das propriedades do material e também das condições de temperatura e fluxo que estão sendo impostas).

Em se tratando de condições de contorno, observamos que existem basicamente três possibilidades:

- De **Dirichlet** (também conhecidas como **essenciais** ou de **primeiro tipo**), quando trazem informações sobre o valor da função em pontos do contorno;
- De **Neuman** (também conhecidas como **naturais** ou de **segundo tipo**), quando trazem informações sobre a derivada da função em pontos do contorno;
- De **Robin** (também conhecidas como de **terceiro tipo**) quando são expressas como uma combinação linear de valores da função em pontos de contorno com valores da derivada da função nestes mesmos pontos.

■ **Exemplo 2.1** Aplicação das condições de contorno na equação 1D de condução de calor

Por simplicidade, vamos assumir que a geração de calor é nula, isto é, $Q(x) = 0$. Neste caso, a solução da equação diferencial é

$$T(x) = -\frac{1}{k}(C_1x + C_2). \quad (2.10)$$

Vamos assumir que conhecemos as temperaturas nas extremidades do domínio, tal que

- $T(0) = T_0$
- $T(L) = T_L$

Como estamos impondo diretamente valores da variável dependente no contorno, estamos aplicando duas condições de contorno de **Dirichlet**, ou essenciais. Assim, utilizando a Eq. (2.10) duas vezes

$$T(0) = -\frac{C_2}{k} = T_0 \implies C_2 = -T_0k \quad (2.11)$$

e

$$T(L) = -\frac{C_1L + C_2}{k} = T_L \implies C_1 = k\frac{T_0 - T_L}{L} \quad (2.12)$$

e, com isso, a solução particularizada para o problema em questão é

$$T(x) = \frac{T_L - T_0}{L}x + T_0 \quad (2.13)$$

que é uma variação linear de temperatura entre as duas extremidades do domínio (a solução correta).

■

■ **Exemplo 2.2** Aplicação de condições de contorno na equação diferencial de condução de calor 1D

Caso as informações sobre o contorno fossem diferentes, não iríamos modificar a resposta da equação geral de nossa EDO (exemplo anterior). Supondo, por exemplo, que $q_x(0) = -q_0$ (entrando) e $T(L) = T_L$, então teremos uma condição de contorno de **Neumann**, ou natural, em $x = 0$ e uma condição de **Dirichlet**, ou essencial, em $x = L$.

Podemos considerar a condição de contorno de Neuman utilizando a lei de Fourier, pois

$$q_x(0) = -k \frac{dT(x)}{dx} \Big|_{x=0} = -q_0 \quad (2.14)$$

tal que utilizando a Eq. (2.7) obtemos

$$C_1 = -q_0. \quad (2.15)$$

A condição de contorno de Dirichlet pode ser utilizada como no exemplo anterior, pois estamos impondo o valor da variável dependente

$$T(L) = -\frac{1}{k}(C_1L + C_2) = T_L \quad (2.16)$$

e, substituindo o valor de C_1

$$-\frac{1}{k}(-q_0L + C_2) = T_L \quad (2.17)$$

resultando em

$$C_2 = q_0L - T_Lk \quad (2.18)$$

tal que a solução completa para essas condições de contorno é

$$T(x) = -\frac{1}{k}(-q_0x + q_0L - T_Lk) = \frac{q_0}{k}x - \frac{q_0}{k}L + T_L. \quad (2.19)$$

■

Como já comentado, é interessante notar que uma mesma equação diferencial pode ser utilizada para modelar problemas físicos completamente diferentes. Vamos estudar um problema de mecânica dos sólidos.

2.3 Modelo de barra

Vamos considerar a mesma geometria do problema de condução de calor 1D, mas agora para um problema de mecânica dos sólidos. Seja o domínio unidimensional ilustrado na Fig. 2.2, submetido a uma força de corpo $b_x(x)$ [N/m^3] na direção x .

Neste caso, podemos realizar o balanço diferencial dos fluxos (tensões normais) no contorno do elemento diferencial (aqui iremos utilizar a notação cartesiana, com positivo na direção de x e negativo se oposto), tal que o somatório de forças em x resulta em

$$-\sigma_{xx}(x)dydz + (\sigma_{xx}(x) + d\sigma_{xx}(x))dydz + b_x(x)dV = 0 \quad [N] \quad (2.20)$$

e, simplificando os termos e dividindo pelo diferencial de volume, obtemos

$$\boxed{\frac{d\sigma_{xx}(x)}{dx} + b_x(x) = 0} \quad [N/m^3] \quad (2.21)$$

que é uma equação diferencial ordinária de primeira ordem e primeiro grau. A variável dependente é a tensão normal $\sigma_{xx}(x)$ e a variável independente é a coordenada x . Compare as Eqs. (2.21) e (2.2) e verifique que a equação diferencial é idêntica, só mudando o significado do fluxo e do termo fonte.

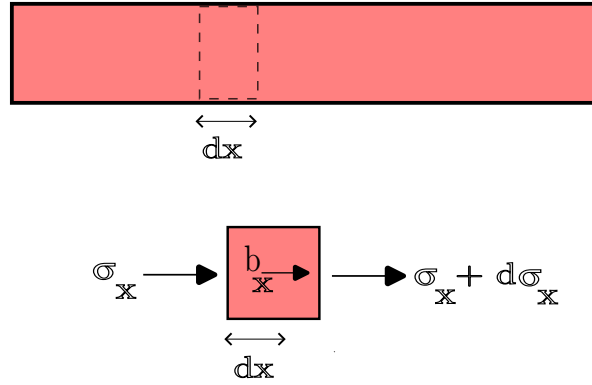


Figura 2.2: Equilíbrio diferencial de um problema sólido 1D.

Da mesma forma que fizemos no problema de condução de calor, podemos considerar uma **relação constitutiva**. No caso da barra, podemos utilizar a famosa lei de Hooke

$$\sigma_{xx}(x) = E(x)\epsilon_{xx}(x) = E(x)\frac{du_x(x)}{dx} \quad (2.22)$$

em que ϵ_{xx} é a deformação normal na direção x , $E(x)$ é o módulo de elasticidade longitudinal do material na posição x e $u_x(x)$ o deslocamento axial. Inserindo a relação da Eq. (2.22) na Eq. (2.21) obtemos

$$\frac{d}{dx} \left(E(x) \frac{du(x)}{dx} \right) + b_x(x) = 0 \quad (2.23)$$

que é uma equação diferencial ordinária de segunda ordem e primeiro grau, em que a variável dependente agora é o deslocamento axial $u_x(x)$.

Como identificamos que a equação diferencial é praticamente idêntica ao problema unidimensional de condução de calor (temos uma troca de sinal, devido a relação constitutiva), já temos a solução analítica para a variável dependente quando a propriedade do material é constante (no caso $E(x)$), Eq. (2.9), que pode ser reescrita como

$$u_x(x) = \frac{1}{E} \left(- \int_0^L \int_0^L b_x(x) dx dx + C_1 x + C_2 \right). \quad (2.24)$$

A nomenclatura das condições de contorno é a mesma, pois essa classificação vem da teoria de equações diferenciais e não dos problemas físicos. Assim, ainda teremos as condições de contorno de Dirichlet, Neumann e Robin só mudando o significado físico para a aplicação de sólidos em relação ao problema de condução de calor.

■ **Exemplo 2.3** Barra engastada e submetida a uma força na extremidade livre

Seja o problema ilustrado na Figura 2.3. A barra está presa na extremidade da esquerda, tal que $u_x(0) = 0[m]$, e submetida a uma força distribuída $p[N/m^2]$ na extremidade da direita. Assumimos que o módulo de elasticidade longitudinal E é constante e que o comprimento total é $L[m]$. Também assumimos que não existe força de corpo.

Com isso, podemos utilizar a solução da Eq. (2.24)

$$u_x(x) = \frac{1}{E} (C_1 x + C_2) \quad (2.25)$$

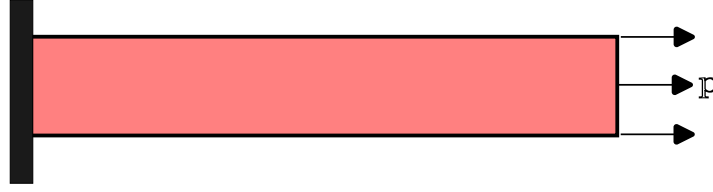


Figura 2.3: Barra presa na extremidade esquerda e submetida a um carregamento distribuído homogêneo $p [N/m^2]$ na extremidade livre.

e as constantes devem ser obtidas pelas condições de contorno.

Começando pela informação sobre o carregamento, podemos notar que em $x = L$ temos um carregamento distribuído p que tem unidade $[N/m^2]$, ou seja, é um fluxo de força. Utilizando a relação constitutiva

$$p(L) = E \frac{du_x(L)}{dx} \quad (2.26)$$

e, derivando a Eq. (2.25) em relação a x e usando a Equação anterior, obtemos

$$C_1 = p. \quad (2.27)$$

OU seja, a força distribuída é uma condição de contorno de Neumann, pois é relacionada à derivada da variável dependente da EDO. A condição de que a barra está presa em $x = 0$ é uma condição imposta diretamente sobre a variável dependente, tal que é uma condição de contorno de Dirichlet. Utilizando a Eq. (2.25)

$$u_x(0) = \frac{1}{E} C_2 = 0 \implies C_2 = 0. \quad (2.28)$$

Com isso, obtemos a solução

$$u_x(x) = \frac{p}{E} x. \quad (2.29)$$

Uma solução mais conhecida do problema de barra é assumir que a força não é aplicada de forma distribuída na extremidade, mas como uma força concentrada $F [N]$. Neste caso, podemos utilizar o teorema de Saint-Venant e descrever p como sendo o fluxo causado por essa força na seção transversal $A [m^2]$

$$p = \frac{F}{A} \quad (2.30)$$

tal que a solução pode ser escrita de forma alternativa

$$u_x(x) = \frac{F}{EA} x. \quad (2.31)$$

■

Iremos utilizar essas equações unidimensionais ao longo do curso, de tal forma que a sua dedução e propriedades devem ser bem fixadas pelo leitor. Caso alguma dúvida sobre a parte física esteja precisando de um reforço, esta seria uma boa hora!

Vamos discutir os mesmos problemas, mas para o caso bidimensional.

2.4 Uma breve revisão de gradiente e divergente

Duas ferramentas importantes para entendermos a dedução (e o significado) de ED em 2 ou 3 dimensões são os operadores gradiente e divergente. Iremos discutir tais operadores para sistemas de referência cartesianos tradicionais.

2.4.1 Gradiente

Relembrando, se temos um campo escalar $T(x,y,z)$ (ou seja, a cada posição do espaço temos um valor escalar), então o **gradiente** de tal campo é denotado por

$$\nabla T(x,y,z) = \left\{ \begin{array}{c} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{array} \right\}_{(x,y,z)} \quad (2.32)$$

ou seja, a cada posição do espaço, o gradiente de um campo escalar retorna um vetor. O significado é a direção e o sentido de aumento do valor do campo T e a magnitude do vetor indica a taxa de aumento do campo escalar nessa direção. Podemos pensar no gradiente como um **operador linear**

$$\nabla() = \left\{ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{array} \right\} \quad (2.33)$$

o que torna muito mais fácil a generalização da operação para campos não escalares. Por exemplo, podemos aplicar o gradiente em um campo vetorial \mathbf{v} . Para isso, escrevemos o vetor em sua base $\mathbf{v} = v_x \mathbf{e}_x + v_y \mathbf{e}_y + v_z \mathbf{e}_z$, em que \mathbf{e}_i são os vetores de base. Assim,

$$\nabla \mathbf{v} = \nabla v_x \mathbf{e}_x + \nabla v_y \mathbf{e}_y + \nabla v_z \mathbf{e}_z \quad (2.34)$$

ou

$$\nabla \mathbf{v} = \left\{ \begin{array}{c} \frac{\partial v_x}{\partial x} \\ \frac{\partial v_x}{\partial y} \\ \frac{\partial v_x}{\partial z} \end{array} \right\} \mathbf{e}_x + \left\{ \begin{array}{c} \frac{\partial v_y}{\partial x} \\ \frac{\partial v_y}{\partial y} \\ \frac{\partial v_y}{\partial z} \end{array} \right\} \mathbf{e}_y + \left\{ \begin{array}{c} \frac{\partial v_z}{\partial x} \\ \frac{\partial v_z}{\partial y} \\ \frac{\partial v_z}{\partial z} \end{array} \right\} \mathbf{e}_z \quad (2.35)$$

que pode ser descrito como um **tensor de segunda ordem**

$$\nabla \mathbf{v} = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_y}{\partial x} & \frac{\partial v_z}{\partial x} \\ \frac{\partial v_x}{\partial y} & \frac{\partial v_y}{\partial y} & \frac{\partial v_z}{\partial y} \\ \frac{\partial v_x}{\partial z} & \frac{\partial v_y}{\partial z} & \frac{\partial v_z}{\partial z} \end{bmatrix} \quad (2.36)$$

ou seja, cada coluna é o gradiente de uma componente do vetor \mathbf{v} .

2.4.2 Divergente

O divergente nada mais é do que o produto escalar (ou interno) do operador gradiente por um campo. Seja o vetor \mathbf{v} , o divergente de tal vetor é definido como

$$\nabla \cdot \mathbf{v} = \left\{ \begin{array}{c} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{array} \right\} \cdot \left\{ \begin{array}{c} v_x \\ v_y \\ v_z \end{array} \right\} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \quad (2.37)$$

ou seja, a aplicação do divergente em um campo vetorial 'abaixa' a dimensão para um campo escalar. Também podemos aplicar o divergente em um tensor de segunda ordem

$$\nabla \cdot \mathbf{S} = \nabla \cdot \left(\left\{ \begin{array}{c} S_{xx} \\ S_{yx} \\ S_{zx} \end{array} \right\} \mathbf{e}_x + \left\{ \begin{array}{c} S_{xy} \\ S_{yy} \\ S_{zy} \end{array} \right\} \mathbf{e}_y + \left\{ \begin{array}{c} S_{xz} \\ S_{yz} \\ S_{zz} \end{array} \right\} \mathbf{e}_z \right) \quad (2.38)$$

tal que

$$\nabla \cdot \mathbf{S} = \left(\frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{yx}}{\partial y} + \frac{\partial S_{zx}}{\partial z} \right) \mathbf{e}_x + \left(\frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y} + \frac{\partial S_{zy}}{\partial z} \right) \mathbf{e}_y + \left(\frac{\partial S_{xz}}{\partial x} + \frac{\partial S_{yz}}{\partial y} + \frac{\partial S_{zz}}{\partial z} \right) \mathbf{e}_z$$

que é um vetor

$$\nabla \cdot \mathbf{S} = \left\{ \begin{array}{c} \frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{yx}}{\partial y} + \frac{\partial S_{zx}}{\partial z} \\ \frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y} + \frac{\partial S_{zy}}{\partial z} \\ \frac{\partial S_{xz}}{\partial x} + \frac{\partial S_{yz}}{\partial y} + \frac{\partial S_{zz}}{\partial z} \end{array} \right\}. \quad (2.39)$$

O conceito de divergente é muito importante em mecânica do contínuo, como será visto nas deduções a seguir. Resumindo, todas as equações de balanço envolvendo uma porção (domínio) de material terão o gradiente da grandeza que está sendo utilizada para o cálculo do balanço. Isso ocorre pois o divergente de um campo mede o quanto este campo está contraindo ou expandindo no entorno do ponto, ou fluxo pontual. Assim, um divergente positivo de um campo vetorial significa que os vetores estão 'saindo' do ponto e um divergente negativo indica que os vetores estão 'entrando' no ponto. Um divergente nulo indica que temos a mesma quantidade entrando do que saindo (ou que o campo é nulo).

Com isso, podemos entender o **teorema da divergência**

$$\int_{\Omega} \nabla \cdot \mathbf{v} d\Omega = \int_{\Gamma} \mathbf{v} \cdot \mathbf{n} d\Gamma \quad (2.40)$$

que diz que a soma de tudo que está sendo 'gerado' ou 'consumido' pelo domínio (lado esquerdo) deve passar pelo contorno (lado direito). O Produto escalar $\mathbf{v} \cdot \mathbf{n}$ é a projeção do vetor \mathbf{v} na direção normal \mathbf{n} do contorno Γ .

2.5 Condução de calor 2D

Vamos estender o problema 1D de condução de calor para o caso 2D, pela inclusão de fluxos de calor na direção Y , de acordo com a Figura 2.4.

Realizando o balanço de fluxos

$$\begin{aligned} & -q_x(x, y) dy dz + (q_x(x, y) + dq_x(x, y)) dy dz \\ & -q_y(x, y) dx dz + (q_y(x, y) + dq_y(x, y)) dx dz + Q(x, y) dV = 0 \end{aligned} \quad (2.41)$$

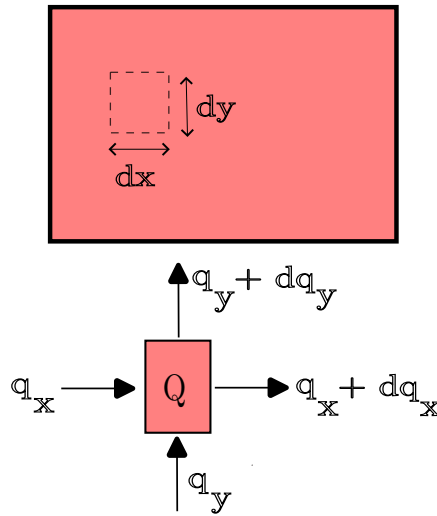


Figura 2.4: Elemento infinitesimal 2D para condução de calor.

e com o cancelamento de termos iguais mas com sinais opostos e também pela normalização pelo diferencial de volume, obtemos

$$\frac{\partial q_x(x,y)}{\partial x} + \frac{\partial q_y(x,y)}{\partial y} + Q(x,y) = 0 \quad [W] \quad (2.42)$$

onde o termo de derivada parcial, ∂ , foi utilizado para compatibilidade com a literatura de equações diferenciais parciais.

A equação obtida é uma equação diferencial parcial (EDP) de primeira ordem e de primeiro grau. A variável dependente é o vetor de fluxo de calor $\mathbf{q}(x,y) [W/m^2]$ em cada ponto do domínio e as variáveis independentes são as coordenadas x e y . Obviamente, se assumirmos que não temos variação na direção y , retornamos ao caso 1D, obtendo uma EDO. Uma maneira mais compacta de escrever a Eq. (2.42) é utilizando o conceito de divergente

$$\nabla \cdot \mathbf{q}(x,y) + Q(x,y) = 0. \quad (2.43)$$

Da mesma forma que fizemos no caso 1D, também podemos utilizar uma relação constitutiva. Assumindo um material em regime linear, podemos escrever

$$\mathbf{q}(x,y) = -\mathbf{k}(x,y) \nabla T(x,y) \quad (2.44)$$

em que $\nabla T(x,y)$ é o vetor gradiente de temperatura

$$\nabla T(x,y) = \left\{ \begin{array}{c} \frac{\partial T(x,y)}{\partial x} \\ \frac{\partial T(x,y)}{\partial y} \end{array} \right\} \quad (2.45)$$

e a matriz $\mathbf{k}(x,y)$, para um material isotrópico, é

$$\mathbf{k}(x,y) = \begin{bmatrix} k(x,y) & 0 \\ 0 & k(x,y) \end{bmatrix} \quad (2.46)$$

Substituindo a relação constitutiva na EDP, obtemos

$$\nabla \cdot (-\mathbf{k}(x,y)\nabla \mathbf{q}(x,y)) + Q(x,y) = 0 \quad (2.47)$$

ou

$$-\frac{\partial}{\partial x} \left(k(x,y) \frac{\partial T(x,y)}{\partial x} \right) - \frac{\partial}{\partial y} \left(k(x,y) \frac{\partial T(x,y)}{\partial y} \right) + Q(x,y) = 0. \quad (2.48)$$

Também é muito comum expressar essa equação como

$$\boxed{\nabla^2 T(x,y) = -Q(x,y)} \quad (2.49)$$

onde ∇^2 é conhecido como operador de Laplace (Laplaciano) e a equação recebe o nome de equação de Poisson. Caso o termo fonte seja nulo, então chamamos a equação de EDP de Laplace. Como pode ser verificado, essa equação é uma EDP de segunda ordem e de primeiro grau.

2.6 Equação de equilíbrio de sólidos em 2D

Vamos estender o problema 1D de sólidos para o caso 2D, pela inclusão de tensões normais na direção Y e também das tensões cisalhantes σ_{xy} , de acordo com a Figura 2.5.

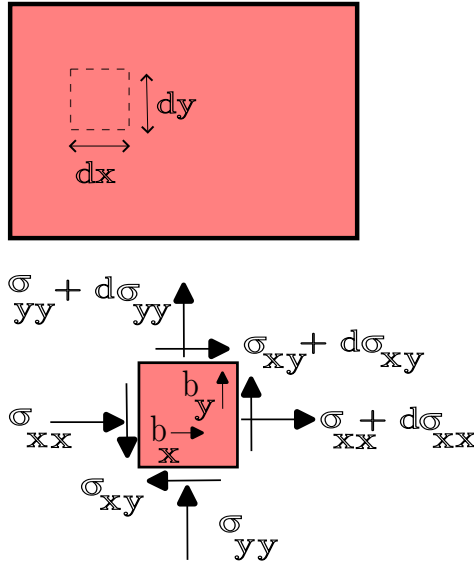


Figura 2.5: Elemento infinitesimal 2D para o problema de sólidos.

O somatório de forças em X pode ser escrito como

$$\begin{aligned} & -\sigma_{xx}(x,y) dydz + (\sigma_{xx}(x,y) + d\sigma_{xx}(x,y)) dydz \\ & -\sigma_{xy}(x,y) dx dz + (\sigma_{xy}(x,y) + d\sigma_{xy}(x,y)) dx dz + b_x(x,y) dV = 0 \quad [N] \end{aligned} \quad (2.50)$$

e, como fizemos no caso 1D e também no problema de condução de calor 2D, podemos cancelar termos iguais mas com sinais contrários e normalizar pelo diferencial de volume, obtendo

$$\frac{\partial \sigma_{xx}(x,y)}{\partial x} + \frac{\partial \sigma_{xy}(x,y)}{\partial y} + b_x(x,y) = 0 \quad [N/m^3]. \quad (2.51)$$

Realizando agora o somatório de forças em Y

$$-\sigma_{yy}(x, y) dx dz + (\sigma_{yy}(x, y) + d\sigma_{yy}(x, y)) dx dz - \sigma_{xy}(x, y) dy dz + (\sigma_{xy}(x, y) + d\sigma_{xy}(x, y)) dy dz + b_y(x, y) dV = 0 \quad [N] \quad (2.52)$$

que resulta em

$$\frac{\partial \sigma_{xy}(x, y)}{\partial x} + \frac{\partial \sigma_{yy}(x, y)}{\partial y} + b_y(x, y) = 0 \quad [N/m^3]. \quad (2.53)$$

Com isso, temos um **sistema de equações diferenciais parciais** de primeira ordem

$$\begin{cases} \frac{\partial \sigma_{xy}(x, y)}{\partial x} + \frac{\partial \sigma_{yy}(x, y)}{\partial y} + b_x(x, y) = 0 \\ \frac{\partial \sigma_{xy}(x, y)}{\partial x} + \frac{\partial \sigma_{yy}(x, y)}{\partial y} + b_y(x, y) = 0 \end{cases} \quad (2.54)$$

e a variável dependente é o tensor de tensão em cada ponto do domínio $\boldsymbol{\sigma}(x, y)$, que é um tensor de segunda ordem. A equação de balanço em termos de tensões também pode ser escrita na forma compacta

$$\boxed{\nabla \cdot \boldsymbol{\sigma}(x, y) + \mathbf{b}(x, y) = \mathbf{0}} \quad [N/m^3] \quad (2.55)$$

Considerando um problema elástico linear, podemos escrever a relação constitutiva

$$\boldsymbol{\sigma}(x, y) = \mathbf{C}(x, y) \boldsymbol{\epsilon}(x, y) \quad (2.56)$$

em que

$$\boldsymbol{\epsilon}(x, y) = \mathbf{L} \mathbf{u}(x, y) \quad (2.57)$$

é a relação entre deslocamentos \mathbf{u} e a deformação. O operador linear \mathbf{L} é

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}. \quad (2.58)$$

Juntando a relação constitutiva e o operador \mathbf{L} , Podemos obter um sistema de equações diferenciais de segunda ordem em termos do campo de deslocamentos \mathbf{u}

$$\boxed{\nabla \cdot (\mathbf{C}(x, y) \mathbf{L} \mathbf{u}(x, y)) + \mathbf{b}(x, y) = \mathbf{0}} \quad [N/m^3]. \quad (2.59)$$

2.7 Equações Diferenciais como Operadores

Conforme visto nas seções anteriores, podemos interpretar uma equação diferencial como sendo a aplicação de um operador (Linear ou não), sobre uma função, mapeando esta para um outro espaço vetorial. Assim, a equação diferencial

$$\nabla^2 T(x, y) = -Q(x, y)$$

pode ser interpretada como a aplicação do operador ∇^2 sobre o campo escalar $T(x, y)$, mapeando para o campo escalar $-Q(x, y)$. Assim, podemos utilizar toda a teoria de operadores lineares para estudar as propriedades de uma equação diferencial, como se essa fosse a simples aplicação de um operador. Em especial, caso os espaços vetoriais sejam finito dimensionais, então o operador poderá ser escrito na forma matricial!

2.8 Equações Diferenciais Parciais Lineares

Uma EDP linear tem a forma geral

$$Af_{xx} + 2Bf_{xy} + Cf_{yy} + Df_x + Ef_y + Ff + G = 0$$

onde f_{xx} denota derivada segunda em relação a x , por exemplo. Os coeficientes A, B, C, D, E, F e G dependem apenas das variáveis independentes, tendo como requisito a condição

$$A^2 + B^2 + C^2 \neq 0.$$

Quanto a classificação, as EDP lineares podem ser classificadas como:

- Hiperbólicas se $\Delta > 0$ (pois a expressão é a mesma de uma hipérbole),
- Elípticas se $\Delta < 0$ (pois a expressão é a mesma de uma elipse),
- Parabólicas se $\Delta = 0$ (pois a expressão é a mesma de uma parábola),

onde $\Delta = B^2 - 4AC$ é o discriminante da EDP. Alguns exemplos:

- $f_{xx} - f_{yy} = 0$ é hiperbólica;
- $f_{xx} + f_{yy} = 0$ é Elíptica;
- $f_{xx} - f_y = 0$ é parabólica;
- Equações de Laplace e de Poisson são elípticas;
- Equação da onda é hiperbólica;
- Equação de Fourier é parabólica;

é importante salientar que a classificação da EDP permite a avaliação de quais métodos de solução poderão ser empregados para a sua correta solução numérica. Ainda mais interessante, é a interpretação física associada a esta classificação, pois:

- Hiperbólica: Está associada a leis de conservação, como no caso do teorema da divergência. Neste tipo de equação diferencial, a solução se comporta como a propagação de uma onda, ou seja, se aplicarmos uma perturbação nas condições de contorno ou nas condições iniciais, cada ponto do domínio demora um certo tempo para sentir o efeito.
- Elíptica: Suas soluções não podem ter descontinuidades nas derivadas. Diferentemente das equações hiperbólicas, não existe uma propagação de informação ao longo do domínio, tal que este tipo de equação é adequado para descrever problemas que não dependam do tempo.
- Parabólica: Está associada a problemas cuja solução se comporta de forma semelhante a difusão de calor em um sólido, ao longo do tempo.

Exercício 2.1 Classifique as equações diferenciais vistas neste capítulo e discuta o significado físico das condições de contorno. ■

Exercício 2.2 Classifique a equação diferencial

$$\frac{d^2\theta(t)}{dt^2} + \frac{g}{l}\sin(\theta(t))$$

Exercício 2.3 Estenda a equação diferencial de condução de calor em um meio isotrópico e linear para o caso 3D. ■

3. Método dos Resíduos Ponderados

Considere uma equação diferencial

$$F(x, y(x), y_x(x), \dots, y_{(n)}(x)) = 0 \quad (3.1)$$

com $x \in [0, L]$. Sabemos que a solução desta equação diferencial é uma função (variável dependente) $y(x)$, que deve ser n -diferenciável, ou seja, pertencente ao conjunto de funções C^n . Ainda, $y(x)$ deve satisfazer a equação diferencial em **todos** os pontos do domínio ($\forall x \in [0, L]$), o que é um requisito bastante **forte**.

O objetivo deste capítulo é introduzir um método para a solução dessas ED, por meio de algumas hipóteses que 'enfraqueçam' os requisitos da solução. Antes de prosseguir, vamos relembrar alguns conceitos básicos (Esses conceitos são discutidos com mais detalhes nas notas de aulas de Fundamentos de Matemática).

3.1 Produto interno no espaço das funções

O conceito de produto interno, ou escalar, entre dois vetores no \mathbb{R}^3 é bem conhecido desde a graduação. Sejam dois vetores \mathbf{u} e \mathbf{v} . O produto interno é definido como

$$\mathbf{u} \cdot \mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle = u_x v_x + u_y v_y + u_z v_z \quad (3.2)$$

ou seja, a soma dos produtos de cada componente dos dois vetores. Essa lógica se mantém para vetores definidos no \mathbb{R}^n , tal que

$$\mathbf{u} \cdot \mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^n u_i v_i. \quad (3.3)$$

Também podemos lidar com **espaço de funções** onde os 'vetores' são funções. Como visto no material de Fundamentos de Matemática, podemos estender todos os conceitos vistos em Álgebra para o \mathbb{R}^n para esses espaços, em especial, o conceito de produto interno. Sejam duas funções $u(x)$ e $v(x)$ definidas em $x \in [a, b]$. O produto interno é a 'soma' do

produto das funções em todas as suas 'componentes', ou seja, em todos os infinitos pontos do intervalo

$$\langle u(x), v(x) \rangle = \int_a^b u(x)v(x) dx. \quad (3.4)$$

Essa forma de produto interno é chamada de produto interno canônico no espaço das funções e permite as mesmas interpretações do produto interno no \mathbb{R}^3 . Em especial, se $u(x)$ e $v(x)$ forem **ortogonais**, o seu produto interno será nulo.

3.1.1 Convergência fraca

Uma sequência de funções $u_k(x)$ é tal que o comportamento da sequência depende de uma 'enumeração' $k \in \mathbb{N}$. Um exemplo é a sequência

$$u_k(x) = \sin(k\pi x) \quad x \in [0, 1] \quad k \in \mathbb{N}. \quad (3.5)$$

Dizemos que a sequência $u_k(x)$ **converge fortemente** para uma função $u(x)$ se

$$\lim_{k \rightarrow \infty} u_k(x) = u(x) \quad \forall x \quad (3.6)$$

ou seja, temos que avaliar o limite em k para **cada ponto** x . Esse é um requisito muito forte e nem sempre conseguimos garantir que uma sequência vá convergir em todos os pontos exatamente.

Um outro conceito de convergência é muito mais flexível. Suponha que queremos utilizar a sequência $u_k(x)$ para realizar uma operação posterior \mathcal{O} e estamos mais preocupados com o resultado dessa operação do que com a sequência em si. Assim, estamos satisfeitos se

$$\mathcal{O}(u_k(x)) \implies \mathcal{O}(u(x)). \quad (3.7)$$

Uma operação de interesse é justamente o produto interno da sequência por uma função $w(x)$, tal que

$$\langle w(x), u_k(x) \rangle \implies \langle w(x), u(x) \rangle \quad (3.8)$$

ao que chamamos de **convergência fraca**.

Exercício 3.1 Ler a seção 9.2 do texto de Fundamentos de Matemática, acompanhando os exemplos. ■

3.2 Método dos Resíduos Ponderados

Supondo, por um momento, que tenhamos uma função $\tilde{y}(x)$ que satisfaça os requisitos de diferenciabilidade e as condições de contorno, mas que **não** solucione a EDO em todos os pontos. Neste caso, a introdução de $\tilde{y}(x)$ na equação diferencial irá resultar em

$$F(x, \tilde{y}(x), \tilde{y}_x(x), \dots, \tilde{y}_{(n)}(x)) = r(x) \quad (3.9)$$

onde $r(x)$ é conhecido como **resíduo**. Obviamente, se $\tilde{y}(x) = y(x) \implies r(x) = 0(x)$. Assim, uma 'boa solução' será aquela que minimiza o resíduo, sendo justamente a estratégia utilizada nos métodos que serão abordados aqui.

A satisfação da equação diferencial em todos os pontos, conforme visto acima, torna o procedimento de solução bastante complicado. Assim, ao invés do requisito **forte**, podemos buscar a solução **fraca** (no sentido de convergência) por meio da operação

$$\langle w(x), r(x) \rangle = \langle w(x), 0(x) \rangle = 0 \quad (3.10)$$

onde $w(x)$ é conhecida como função teste (ou peso). Essa expressão nada mais é, no espaço das funções, do que

$$\int w(x)r(x) dx = 0 \quad (3.11)$$

que pode ser interpretada como uma média ponderada (por $w(x)$) da função resíduo ao longo do domínio de solução, dando origem ao nome **resíduos ponderados**. Certamente, podemos realizar uma interpretação mais geométrica, uma vez que sabemos que a condição da Eq. (3.10) implica em **ortogonalidade** entre a função resíduo e a função teste. Assim, podemos dizer que estamos ortogonalizando o resíduo em relação a um espaço, definido pelas funções teste.

■ **Exemplo 3.1** Vamos considerar a equação diferencial

$$\frac{d^2 T(x)}{dx^2} = 1, \quad x \in (0, 1) \quad (3.12)$$

com condições de contorno de Dirichlet $T(0) = T(1) = 0$. Essa equação tem solução exata

$$T(x) = -\frac{x}{2} + \frac{x^2}{2}. \quad (3.13)$$

Vamos propor a solução aproximada

$$\tilde{T}(x) = a_0 \sin(\pi x) \quad (3.14)$$

em que $a_0 \in \mathbb{R}$ é uma amplitude. A solução $\tilde{T}(x)$ satisfaz as condições de contorno essenciais e também as condições de diferenciabilidade.

O resíduo será

$$r(x) = \frac{d^2 \tilde{T}(x)}{dx^2} - 1 \quad (3.15)$$

tal que sua ortogonalização em relação a uma função peso $w(x)$ resulta em

$$\int_0^1 \left(\frac{d^2 \tilde{T}(x)}{dx^2} - 1 \right) w(x) dx = 0 \quad (3.16)$$

ou, desenvolvendo a derivada e agrupando os termos,

$$\int_0^1 -\pi^2 w(x) a_0 \sin(\pi x) dx = \int_0^1 w(x) dx \quad (3.17)$$

tal que a 'melhor' amplitude a_0 deve satisfazer

$$a_0 = - \frac{\int_0^1 w(x) dx}{\pi^2 \int_0^1 w(x) \sin(\pi x) dx}. \quad (3.18)$$

■

Neste ponto fica claro que a escolha de $w(x)$ irá definir a aproximação (e sua 'qualidade'). De fato, diferentes escolhas para $w(x)$ levam a diferentes soluções. Embora a escolha de $w(x)$ não esteja diretamente associada a variável dependente da ED, vamos utilizar a hipótese de que $w(x)$ **também satisfaz as condições de contorno essenciais do problema**. Embora seja possível relaxar esse requisito, não iremos nos aprofundar nesta questão pois ela não é necessária para o desenvolvimento do método dos elementos finitos em um curso introdutório.

Vamos estudar algumas opções para a função peso $w(x)$ por meio de alguns exemplos.

■ **Exemplo 3.2** Mesma função da solução aproximada

Vamos escolher a mesma função que foi utilizada para a aproximação de $T(x)$

$$w(x) = b_0 \sin(\pi x) \quad (3.19)$$

onde $b_0 \in \mathbb{R}$ é uma amplitude. Neste caso, a solução obtida no exemplo anterior é

$$a_0 = -\frac{\int_0^1 b_0 \sin(\pi x) dx}{\pi^2 \int_0^1 b_0 \sin(\pi x) \sin(\pi x) dx} \quad (3.20)$$

resultando em

$$a_0 = -\frac{4}{\pi^3} \quad (3.21)$$

tal que a melhor aproximação (considerando a escolha de que $w(x)$ utiliza a mesma função do que $\tilde{T}(x)$) será

$$\tilde{T}(x) = -\frac{4}{\pi^3} \sin(\pi x) \quad (3.22)$$

A Figura 3.1 mostra a comparação entre a solução aproximada e a analítica.

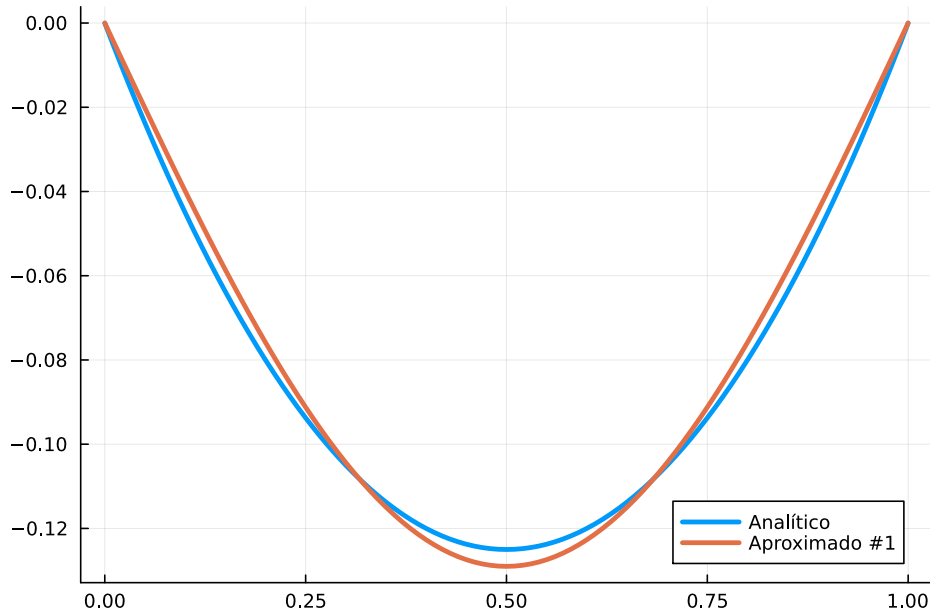


Figura 3.1: Aproximação obtida para $w(x) = b_0 \sin(\pi x)$ comparada com a solução analítica.

É interessante notar que, embora estejamos comparando $\tilde{T}(x)$ com $T(x)$ no gráfico da Figura 3.1, o método tem como foco a média ponderada do resíduo. Assim, utilizando a Eq. (3.15) e a solução aproximada obtida aqui, obtemos

$$r(x) = \frac{4}{\pi} \sin(\pi x) - 1, \quad (3.23)$$

com gráfico ilustrado na Figura 3.2.

■

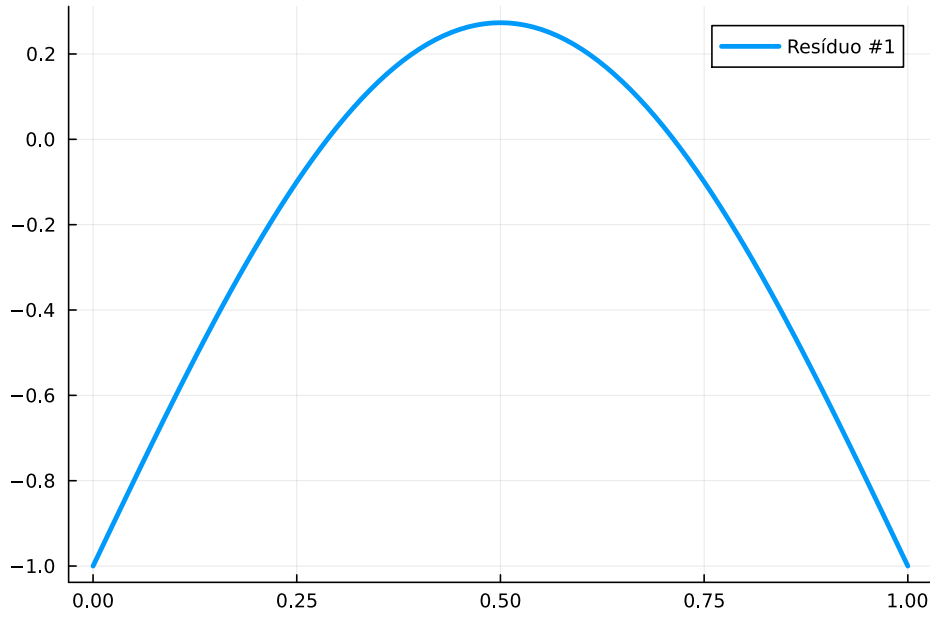


Figura 3.2: Resíduo obtido com $w(x) = b_0 \sin(\pi x)$.

■ **Exemplo 3.3** Uso de outra função de ponderação

Vamos considerar a função de ponderação

$$w(x) = b_0 \delta(x - 0.5) \quad (3.24)$$

que é um delta de Dirac centrado em $x = 0.5$. Utilizando a propriedade de que

$$\int_{-\infty}^{\infty} f(x) \delta(x - x_0) dx = f(x_0), \quad (3.25)$$

para avaliarmos a Eq. (3.18), obtemos

$$a_0^\delta = -\frac{1}{\pi^2 \sin(0.5\pi)} \quad (3.26)$$

tal que

$$\tilde{T}^\delta(x) = -\frac{1}{\pi^2 \sin(0.5\pi)} \sin(\pi x), \quad (3.27)$$

com o gráfico ilustrado na figura 3.5.

O resíduo, neste caso, será

$$r^\delta(x) = \frac{\sin(\pi x)}{\sin(0.5\pi)} - 1 \quad (3.28)$$

com o comportamento do gráfico da Figura 3.4. É muito interessante notar que o resíduo é nulo exatamente no ponto em que aplicamos o delta de Dirac.

■

■ **Exemplo 3.4** Uso de diferentes funções para $w(x)$

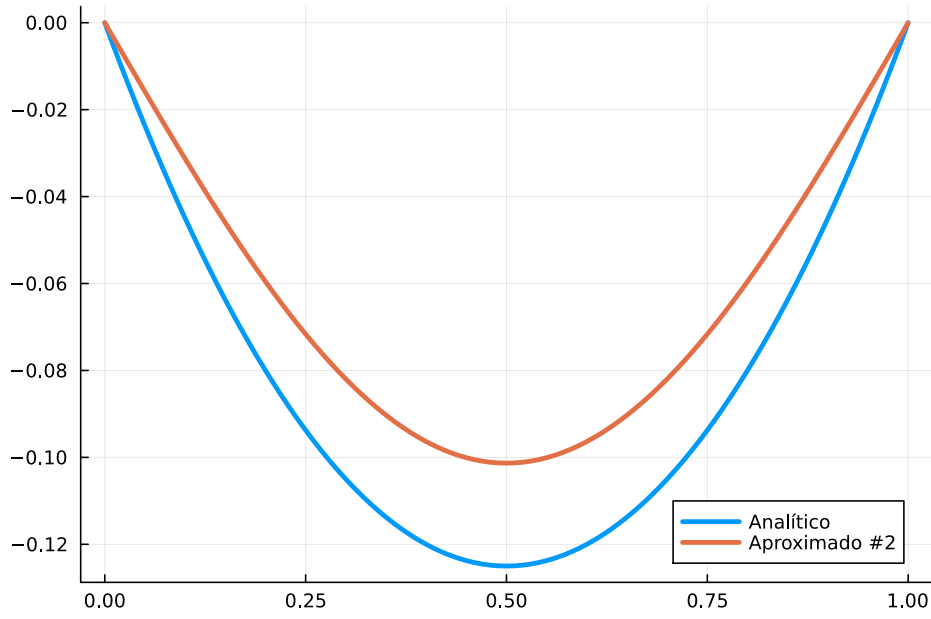


Figura 3.3: Aproximação obtida para $w(x) = b_0\delta(x - 0.5)$ comparada com a solução analítica.

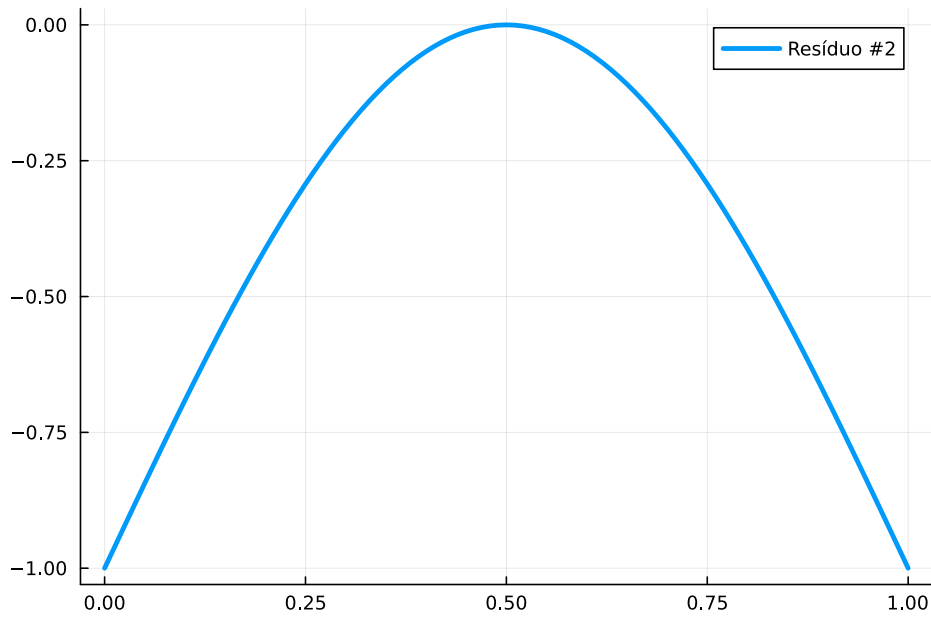


Figura 3.4: Resíduo obtido para $w(x) = b_0\delta(x - 0.5)$.

Por fim, podemos considerar $w(x) = b_0(H(x - 0) - H(x - 1))$, correspondendo a função degrau (Heaviside) ativa somente no domínio, resultando em

$$a_0 = -\frac{1}{2\pi} \quad (3.29)$$

tal que

$$\tilde{T}^H(x) = -\frac{1}{2\pi} \sin(\pi x) \quad (3.30)$$

com resíduo

$$r^H(x) = \frac{\pi \sin(\pi x)}{2} - 1 \quad (3.31)$$

conforme ilustrado nas Figuras 3.5 e 3.6, respectivamente.

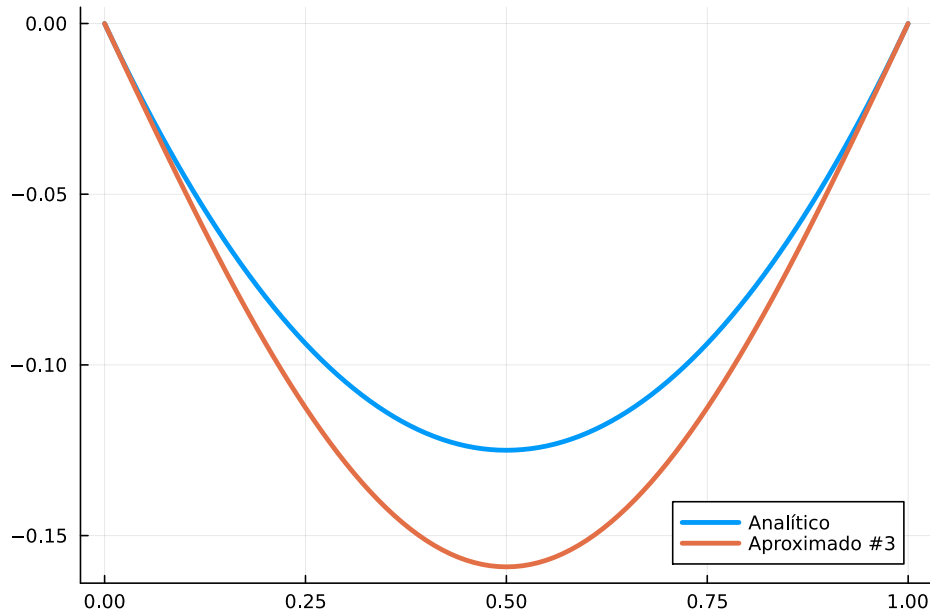


Figura 3.5: Aproximação obtida para $w(x) = b_0(H(x-0) - H(x-1))$ comparada com a solução analítica.

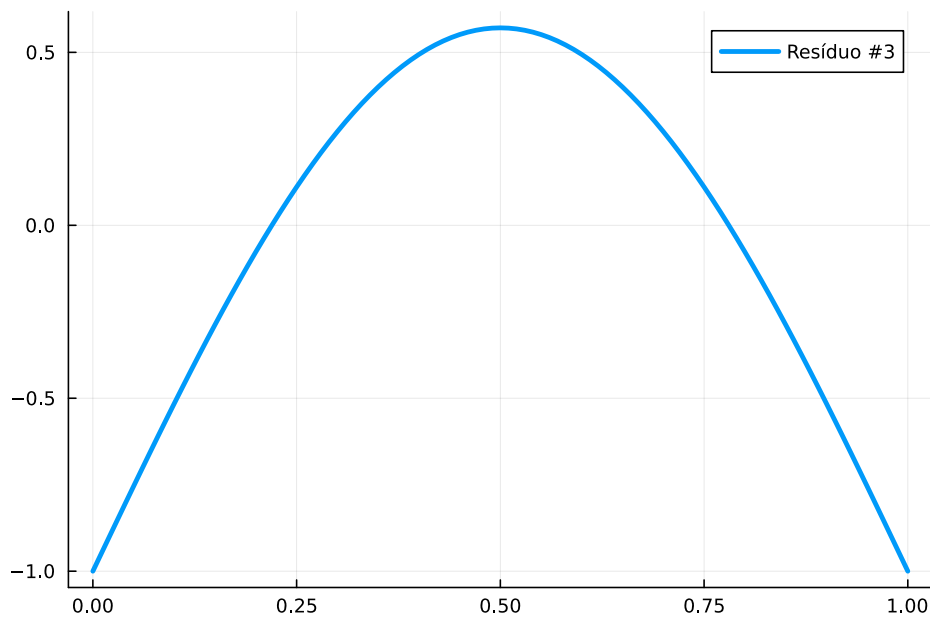


Figura 3.6: Resíduo obtido para $w(x) = b_0(H(x-0) - H(x-1))$.

■

Avaliando os três exemplos anteriores, fica difícil comparar qual é a melhor opção para $w(x)$. Comparando as soluções para $\tilde{T}(x)$ em um mesmo gráfico, Figura 3.7, observamos que cada uma das opções tem características diferentes. A solução obtida com a mesma função fica mais próxima da solução analítica ao longo de todo o domínio, mas a solução obtida com o Heaviside aproxima bem a função perto das extremidades do domínio.

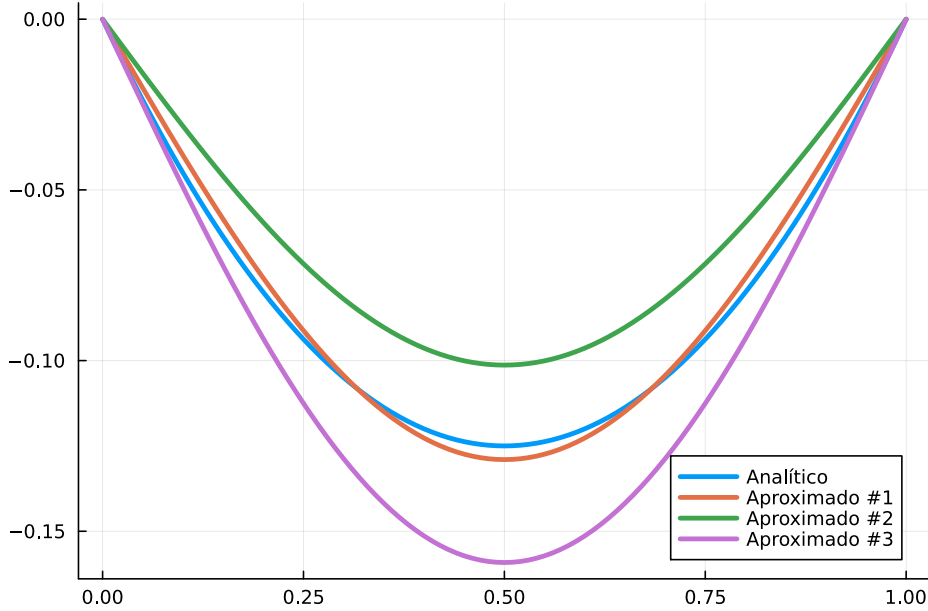


Figura 3.7: Comparação das soluções $\tilde{T}(x)$ para os três primeiros exemplos.

Uma comparação mais justa é via os resíduos, já que o método usa o resíduo como parâmetro a ser aproximado, Figura 3.8.

Uma outra questão muito importante é que a aproximação dos fluxos é de grande interesse para aplicações práticas, pois fluxos tem significado físico relevante, como fluxo de calor e a tensão mecânica. O fluxo da solução analítica é

$$q(x) = \frac{dT(x)}{dx} = x - 0.5 \quad (3.32)$$

e os fluxos obtidos com as soluções aproximadas são

$$\tilde{q}(x) = -\frac{4\cos(\pi x)}{\pi^2} \quad (3.33)$$

para a mesma base,

$$q^\delta(x) = -\frac{\cos(\pi x)}{(\pi \sin(0.5\pi))} \quad (3.34)$$

para o delta de Dirac e

$$q^H(x) = -\frac{\cos(\pi x)}{2} \quad (3.35)$$

para o Heaviside. A Figura 3.9 ilustra os comportamentos no domínio. Neste caso, observamos que a escolha do Heaviside dá o melhor fluxo no contorno.

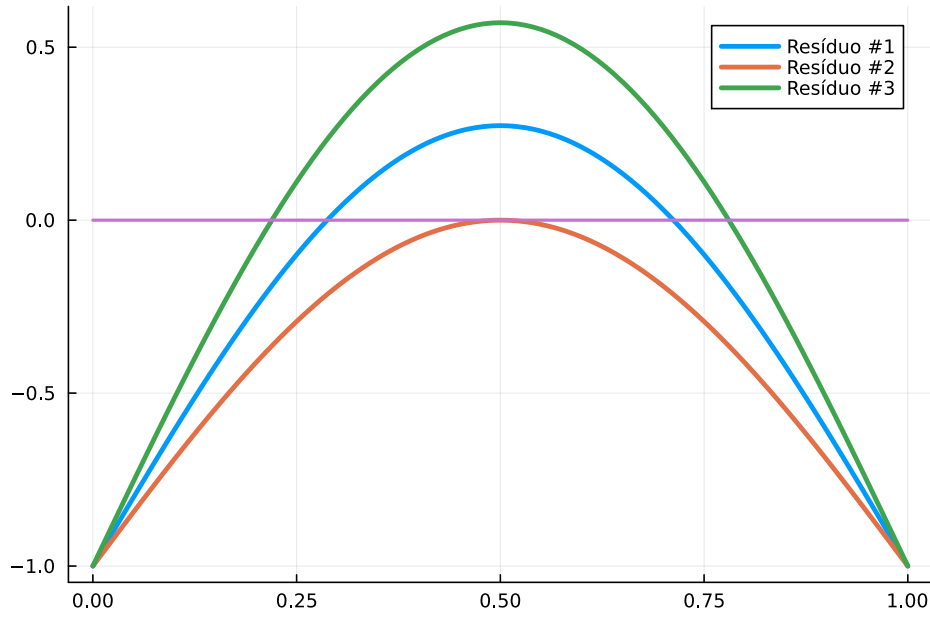


Figura 3.8: Comparação dos resíduos das diferentes soluções $\tilde{T}(x)$ para os três primeiros exemplos.

Neste ponto é interessante recordar o que fizemos até agora: Dada uma equação diferencial, propomos uma solução aproximada $\tilde{T}(x)$ que deve satisfazer as condições de contorno essenciais. Também escolhemos uma função de ponderação $w(x)$, com o mesmo requisito de satisfazer as condições de contorno essenciais. O que devemos olhar com mais calma neste momento é o que significa 'propor' uma solução. Quando descrevemos uma função, podemos utilizar uma **combinação linear** de termos de **base**, na forma

$$\tilde{T}(x) = \sum_{i=0}^n a_i \phi_i(x) \quad (3.36)$$

com o requisito de que as funções de base $\phi_i(x)$ sejam **linearmente independentes**. Quando fizemos a proposta $\tilde{T}(x) = a_0 \sin(\pi x)$, utilizamos somente um termo no somatório, ou seja, o conjunto de base é limitado a só uma função. No entanto, podemos propor algo mais 'refinado', tal como

$$\tilde{T}(x) = a_0 \sin(\pi x) + a_1 \sin(2\pi x) + a_2 \sin(3\pi x) \quad (3.37)$$

por exemplo. Todas as funções da base propostas acima satisfazem a condição de contorno e, também, são linearmente independentes. O mesmo raciocínio pode ser utilizado para propormos uma função de ponderação $w(x)$. Vamos a um exemplo

■ **Exemplo 3.5** Utilização de uma combinação linear para descrever $\tilde{T}(x)$

Vamos utilizar a função aproximada

$$\tilde{T}(x) = a_0 \sin(\pi x) + a_1 \sin(2\pi x) + a_2 \sin(3\pi x) \quad (3.38)$$

e vamos descrever a função de ponderação **na mesma base**

$$w(x) = b_0 \sin(\pi x) + b_1 \sin(2\pi x) + b_2 \sin(3\pi x) \quad (3.39)$$

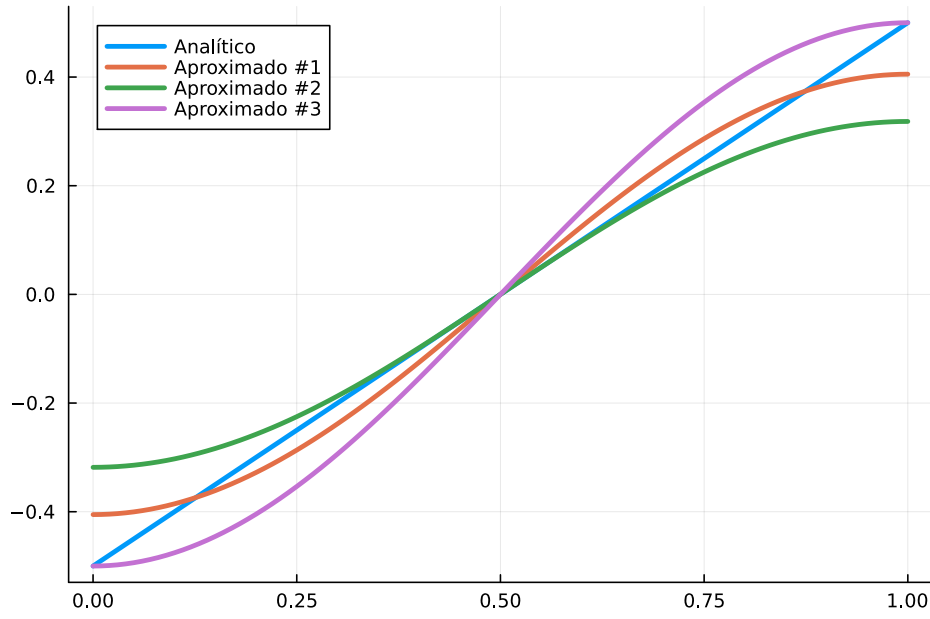


Figura 3.9: Fluxos obtidos com as diferentes estratégias de solução.

Como a equação diferencial é a mesma dos exemplos anteriores, observamos que o resíduo continua sendo

$$r(x) = \frac{d^2 \tilde{T}(x)}{dx^2} - 1 \quad (3.40)$$

e que o método dos resíduos ponderados preconiza que

$$\int_0^1 w(x) \left(\frac{d^2 \tilde{T}(x)}{dx^2} - 1 \right) dx = 0. \quad (3.41)$$

Assim, considerando as propostas para $\tilde{T}(x)$ e $w(x)$, obtemos

$$-((3\pi^3 a_0 + 6)b_0 + (12\pi^3 a_1 - 3)b_1 + (27\pi^3 a_2 + 2)b_2) = 2b_2 + 3b_1 + 6b_0 \quad (3.42)$$

e devemos observar que o nosso interesse é obter as amplitudes a_i e não as b_i . Assim, podemos separar a expressão por termos b_i em comum, obtendo

$$b_0) \quad -3\pi^3 a_0 - 6 = 6 \quad (3.43)$$

$$b_1) \quad -12\pi^3 a_1 + 3 = 3 \quad (3.44)$$

$$b_2) \quad -27\pi^3 a_2 - 2 = 2 \quad (3.45)$$

$$(3.46)$$

que correspondem a três equações lineares em a_i , com solução

$$a_0 = -\frac{4}{\pi^3} \quad (3.47)$$

$$a_1 = 0 \quad (3.48)$$

$$a_2 = -\frac{4}{27\pi^3} \quad (3.49)$$

tal que

$$\tilde{T}(x) = -\frac{4}{\pi^3} \sin(\pi x) - \frac{4}{27\pi^3} \sin(3\pi x) \quad (3.50)$$

que pode ser visualizado na Figura 3.10.

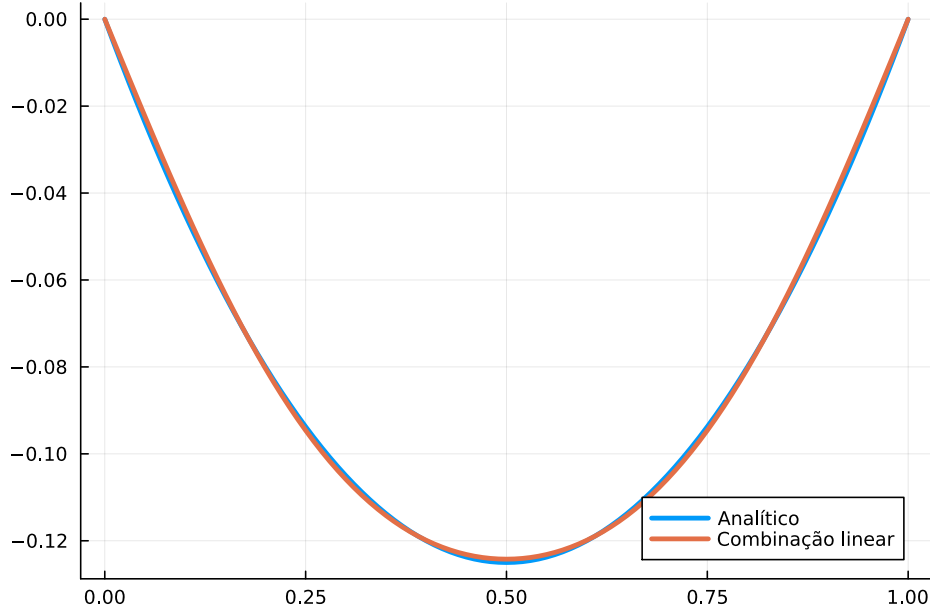


Figura 3.10: Solução $\tilde{T}(x)$ obtida com a base $\phi = \{\sin(i\pi x)\}$ para $i = 0, 1, 2$.

Aqui podemos notar algumas características interessantes. Em primeiro lugar, o coeficiente a_0 foi o mesmo obtido no primeiro exemplo, em que utilizamos a mesma base para $\tilde{T}(x)$ e $w(x)$, como no caso deste exemplo. Isso quer dizer que se mantivermos a mesma estrutura do problema e do espaço de solução, podemos 'refinar' a solução pela consideração de mais termos de base, sem que os coeficientes dos termos já computados seja alterado. Outra questão interessante, e que é específica para o problema em questão, é o fato de que obtivemos $a_1 = 0$. Isso pode ser explicado pelo fato de a solução ser necessariamente simétrica em relação aos extremos do domínio e um seno com frequência par não é simétrico. Assim, o método naturalmente anulou esse termo de base.

É interessante notar que não só melhoramos a descrição de $\tilde{T}(x)$, mas também do resíduo, que passa a ser

$$r(x) = \frac{4 \sin(3\pi x)}{3\pi} + \frac{4 \sin(\pi x)}{\pi} - 1 \quad (3.51)$$

conforme ilustrado na Figura 3.11 onde verificamos que temos resíduo nulo em 4 pontos agora. Observamos, também, uma melhora no fluxo

$$q(x) = -\frac{(4 \cos(3\pi x))}{9\pi^2} - \frac{4 \cos(\pi x)}{\pi^2} \quad (3.52)$$

conforme ilustrado no gráfico da Figura 3.12.

■

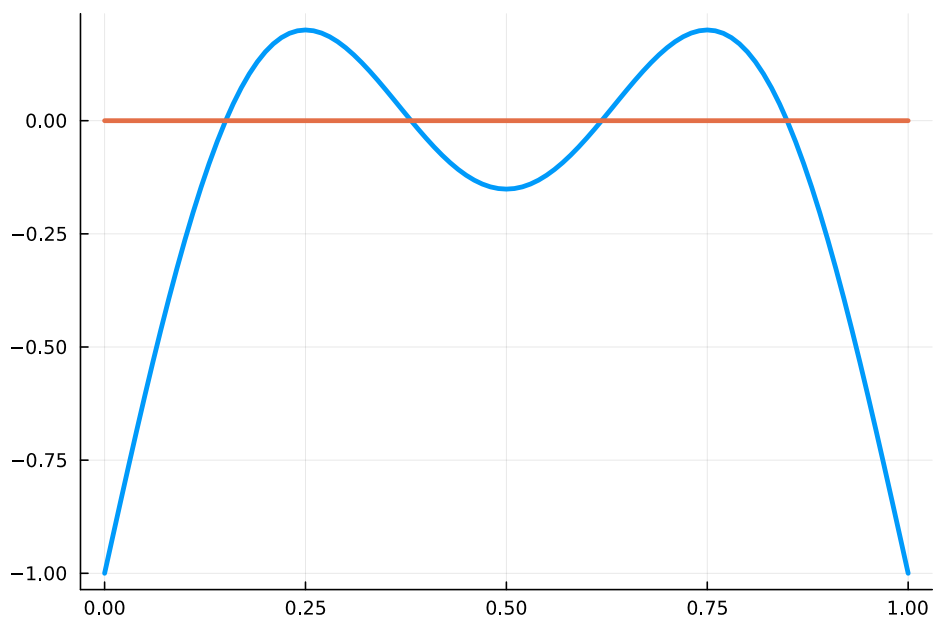


Figura 3.11: Resíduo da solução com três termos.

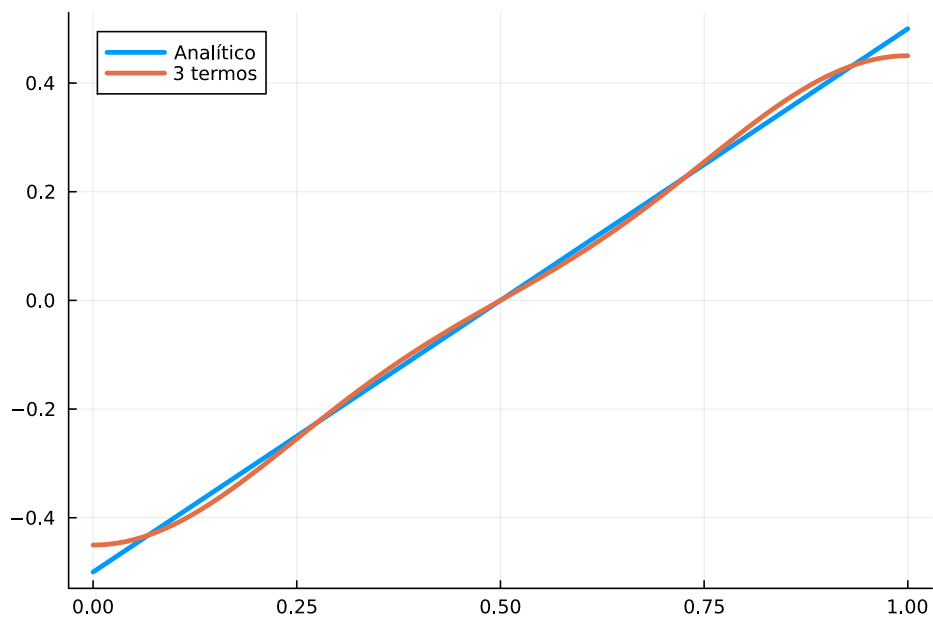


Figura 3.12: Fluxo da solução com três termos.

Exercício 3.2 Solucione o problema anterior utilizando a base polinomial $\phi = \{1, x, x^2\}$ para descrever $\tilde{T}(x)$. Realize o mesmo estudo de utilizar diferentes tipos de funções de ponderação e compare sempre com a solução analítica.

Dica 1: antes de sair utilizando o polinômio, lembre-se que devemos satisfazer as condições de contorno $T(0) = T(1) = 0$

Dica 2: utilize o mesmo número de termos de base para $\tilde{T}(x)$ e para a função de ponderação $w(x)$.



4. Relaxação dos requisitos de continuidade

Continuando, podemos verificar, após o extenso exemplo discutido no capítulo anterior, que o método dos Resíduos Ponderados depende de duas escolhas importantes:

- Da classe de funções utilizadas para descrever a variável dependente da equação diferencial (aproximação);
- Da classe de funções utilizadas para a função teste.

Em especial, tivemos o cuidado de selecionar uma família de funções $\tilde{T}(x)$ que:

- Satisfizesse as condições de contorno;
- Pertencesse ao espaço das funções $2\times$ diferenciáveis (C^2).

A primeira condição é fundamental, mas podemos trabalhar um pouco mais a abordagem utilizada no primeiro exemplo, de modo a relaxar ainda mais os requisitos de continuidade da solução.

Antes de apresentarmos o procedimento, vamos fazer uma pequena revisão do procedimento de integração por partes.

4.1 Integração por partes

A ferramenta mais importante que iremos utilizar neste capítulo é a integração por partes. Por isso, vamos revisar o procedimento.

4.1.1 Produto de funções de uma variável

Vamos começar com o caso mais simples, de funções de uma variável. Vamos definir duas funções $u(x)$ e $v(x)$. Seja o produto dessas funções

$$y(x) = u(x)v(x) \tag{4.1}$$

com derivada (vou deixar de escrever as dependências de x para simplificar a notação)

$$\frac{dy}{dx} = \frac{d(uv)}{dx} = \frac{du}{dx}v + u\frac{dv}{dx} \tag{4.2}$$

tal que podemos rearranjar os termos, obtendo

$$u \frac{dv}{dx} = \frac{d(uv)}{dx} - \frac{du(x)}{dx} v. \quad (4.3)$$

Integrando ambos os lados da equação acima em relação à x

$$\int u \frac{dv}{dx} dx = \int \frac{d(uv)}{dx} dx - \int \frac{du(x)}{dx} v dx \quad (4.4)$$

e podemos verificar facilmente que o primeiro termo do lado direito é a integral da derivada, que são operações 'inversas'. Com isso

$$\boxed{\int u \frac{dv}{dx} dx = uv - \int \frac{du(x)}{dx} v dx.} \quad (4.5)$$

É interessante notar que no caso de integrais definidas, o procedimento de simplificação realizado acima implica em considerar o valor do produto nos extremos do intervalo, tal que

$$\int_a^b u \frac{dv}{dx} dx = \int_a^b \frac{d(uv)}{dx} dx - \int_a^b \frac{du(x)}{dx} v dx \quad (4.6)$$

resulta em

$$\boxed{\int_a^b u \frac{dv}{dx} dx = uv|_a^b - \int_a^b \frac{du(x)}{dx} v dx = u(b)v(b) - u(a)v(a) - \int_a^b \frac{du(x)}{dx} v dx.} \quad (4.7)$$

4.1.2 Divergente do produto de um campo escalar por um campo vetorial

O procedimento visto anteriormente pode ser estendido para casos bem mais gerais, que serão de interesse para nós ao longo deste texto. Quando lidamos com funções de mais de uma variável ou com funções não escalares, podemos deduzir a fórmula da integração por partes a partir da regra do produto, como fizemos na dedução acima. Um exemplo é o caso do divergente do produto de uma função escalar u e de um vetor \mathbf{v}

$$\nabla \cdot (u\mathbf{v}) = u\nabla \cdot \mathbf{v} + \nabla u \cdot \mathbf{v}. \quad (4.8)$$

Integrando no domínio Ω

$$\int_{\Omega} \nabla \cdot (u\mathbf{v}) d\Omega = \int_{\Omega} u\nabla \cdot \mathbf{v} d\Omega + \int_{\Omega} \nabla u \cdot \mathbf{v} d\Omega \quad (4.9)$$

e podemos utilizar o teorema da divergência na integral do lado esquerdo, pois

$$\int_{\Omega} \nabla \cdot (u\mathbf{v}) d\Omega = \int_{\Gamma} u\mathbf{v} \cdot \mathbf{n} d\Gamma \quad (4.10)$$

em que Γ é o contorno de Ω e \mathbf{n} é a normal em Γ (apontando 'para fora' do domínio). Assim, rearranjando os termos, obtemos

$$\boxed{\int_{\Omega} u\nabla \cdot \mathbf{v} d\Omega = \int_{\Gamma} u\mathbf{v} \cdot \mathbf{n} d\Gamma - \int_{\Omega} \nabla u \cdot \mathbf{v} d\Omega.} \quad (4.11)$$

Exercício 4.1 Prove a distributividade (lado direito) da Eq. (4.8). ■

Exercício 4.2 Suponha que o vetor \mathbf{v} seja obtido pela aplicação do gradiente em um campo escalar v

$$\mathbf{v} = \nabla v. \quad (4.12)$$

Deduz novamente a Eq. (4.11) para esse caso. ■

4.2 Utilizando a integração por partes com a forma fraca

Vamos continuar considerando a equação diferencial

$$\frac{d^2 T(x)}{dx^2} = 1, \quad x \in (0, 1) \quad (4.13)$$

e a ortogonalização de seu resíduo

$$\int_0^1 w(x) \left(\frac{d^2 \tilde{T}(x)}{dx^2} - 1 \right) dx = 0. \quad (4.14)$$

Reagrupando a equação, obtemos

$$\int_0^1 w(x) \frac{d^2 \tilde{T}(x)}{dx^2} dx = \int_0^1 w(x) dx \quad (4.15)$$

e, aplicando **integração por partes** no lado esquerdo, obtemos

$$w(x) \frac{d\tilde{T}}{dx} \cdot \mathbf{n} \Big|_0^1 - \int_0^1 \frac{dw(x)}{dx} \frac{d\tilde{T}(x)}{dx} dx = \int_0^1 w(x) dx, \quad (4.16)$$

e enfatizamos que temos um produto escalar no contorno, pois a derivada de \tilde{T} em relação a x é um vetor (mesmo que em 1D). Podemos identificar os seguintes resultados:

- A integral de domínio agora contém derivadas de primeira ordem, tanto de $\tilde{T}(x)$ quanto de $w(x)$;
- Surgiram termos de contorno (como nosso exemplo é 1D, o contorno consiste em $x = 0$ e $x = 1$), dependentes dos valores do fluxo $\frac{d\tilde{T}(x)}{dx}$ e de $w(x)$ nestas posições.

O primeiro item indica que graças a este simples procedimento de integração por partes podemos procurar a solução aproximada, $\tilde{T}(x)$ em um espaço menos restrito do que C^2 (neste caso, funções da classe C^1 satisfazem a relação).

O segundo item também é interessante, pois pelo procedimento de integração por partes obtivemos naturalmente o fluxo no contorno. Por este motivo, estas condições de contorno são chamadas de "naturais", pois informações sobre a derivada da variável dependente no contorno estão associadas a condições de contorno de Neumann, ou naturais.

Ainda, se assumirmos que $w(x)$ tem que satisfazer as condições de contorno essenciais de $\tilde{T}(x)$, então verificamos que sempre saberemos ao menos um dos valores em $x = 0$ ou em $x = 1$ (contorno), tal que o termo 'novo' que apareceu após a integração por partes sempre é conhecido.

De fato, problemas físicos descritos por equações diferenciais, tais como a condução de calor e o problema de mecânica dos sólidos, são muitas vezes chamados de BVP (*boundary value problems* ou problemas de valores no contorno, pois para termos a solução precisamos especificar as condições de contorno. Outra questão interessante é que um problema só é

bem definido (em termos de condições de contorno), quando sabemos **ou** a condição de Dirichlet **ou** a de Neumann, isto é, temos que saber ao menos uma em cada ponto do contorno.

■ **Exemplo 4.1** Utilizando os termos de contorno

Como exemplo, vamos considerar a nossa equação diferencial com as condições de contorno de Dirichlet $T(0) = 0$ e de Neumann, $\left. \frac{dT(x)}{dx} \right|_1 = 2$. Vamos, também, procurar a nossa solução em um espaço com base $\phi = \{1, x, x^2\}$. De modo a satisfazer a condição de contorno essencial em $x = 0$ podemos trabalhar com equações do tipo

$$\tilde{T}(x) = a_1x + a_2x^2 \quad (4.17)$$

e, como função teste, podemos utilizar $w(x)$ descrita no mesmo espaço, ou seja

$$w(x) = b_1x + b_2x^2. \quad (4.18)$$

Assim, inserindo estas informações em

$$w(x) \frac{d\tilde{T}}{dx} \Big|_0^1 - \int_0^1 \frac{dw(x)}{dx} \frac{d\tilde{T}(x)}{dx} = \int_0^1 w(x) dx \quad (4.19)$$

obtemos, para a integral de domínio no lado esquerdo,

$$\int_0^1 \frac{d\tilde{T}(x)}{dx} \frac{dw(x)}{dx} = \frac{(4a_2 + 3a_1)b_2 + (3a_2 + 3a_1)b_1}{3} \quad (4.20)$$

para a integral de domínio no lado direito

$$\int_0^1 w(x) dx = \frac{2b_2 + 3b_1}{6}, \quad (4.21)$$

para o contorno em $x = 0$

$$0(2a_2x + a_1) \quad (4.22)$$

e, para o contorno em $x = 1$,

$$(b_1 + b_2)2. \quad (4.23)$$

Assim, agrupando todos os termos, obtemos

$$[2(b_1 + b_2) - 0] - \frac{(4a_2 + 3a_1)b_2 + (3a_2 + 3a_1)b_1}{3} = \frac{2b_2 + 3b_1}{6} \quad (4.24)$$

de tal forma que, agrupando por termos de b em comum, obtemos

$$b_1) \quad 2 - \frac{3a_2 + 3a_1}{3} = \frac{1}{2} \quad (4.25)$$

$$b_2) \quad 2 - \frac{4a_2 + 3a_1}{3} = \frac{1}{3} \quad (4.26)$$

e, organizando na forma $\mathbf{Ax} = \mathbf{b}$, obtemos

$$\begin{bmatrix} 3 & 3 \\ 3 & 4 \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 4.5 \\ 5 \end{Bmatrix} \quad (4.27)$$

tal que $a_1 = 1$ e $a_2 = 0.5$. Com isto, obtemos uma solução aproximada

$$\tilde{T}(x) = x + \frac{1}{2}x^2 \quad (4.28)$$

com

$$\frac{d\tilde{T}(x)}{dx} = 1 + x. \quad (4.29)$$

Que é a solução exata para o problema, pois usamos a base da solução analítica. ■

Exercício 4.3 Resolva o problema anterior utilizando

$$\tilde{T}(x) = a_1 x. \quad (4.30)$$

Exercício 4.4 Resolva o problema descrito por

$$\frac{d^2 T(x)}{dx^2} = x, \quad x \in (0, 1) \quad (4.31)$$

com $T(0) = 1$ e $\left. \frac{dT(x)}{dx} \right|_{x=1} = 0$.

Obtenha a solução analítica e também a solução pelo método dos resíduos ponderados, com e sem integração por partes, explorando diferentes espaços de solução. ■

Exercício 4.5 Resolva o problema descrito por

$$\frac{d^2 T(x)}{dx^2} = -2, \quad x \in (0, 1) \quad (4.32)$$

com $T(1) = -1$ e $\left. \frac{dT(x)}{dx} \right|_{x=0} = 1$.

Obtenha a solução analítica e também a solução pelo método dos resíduos ponderados, com e sem integração por partes, explorando diferentes espaços de solução. **Dica:** cuidado com a orientação do vetor normal na face com $x = 0$ (o vetor aponta para a esquerda). ■

4.2.1 Equação de quarta ordem da viga longa

O poder da integração por partes fica mais evidente no caso de uma equação de quarta ordem, como no caso da famosa equação da linha elástica de uma viga longa. Esta equação é dada por

$$\frac{d^2}{dx^2} \left(E(x) I(x) \frac{d^2 v(x)}{dx^2} \right) = q(x) \quad (4.33)$$

em que $E(x)$ é o módulo de elasticidade longitudinal, $I(x)$ é o momento de inércia da seção transversal, $q(x)$ é um carregamento distribuído e $v(x)$ é o deslocamento transversal ao

longo do comprimento. Outra medida que é associada diretamente ao deslocamento da linha elástica é a rotação da linha elástica

$$\theta(x) = \frac{dv(x)}{dx}. \quad (4.34)$$

Propondo uma solução aproximada $\tilde{v}(x)$, podemos definir o resíduo como

$$r(x) = \frac{d^2}{dx^2} \left(E(x)I(x) \frac{d^2 \tilde{v}(x)}{dx^2} \right) - q(x) \quad (4.35)$$

tal que o procedimento de ortogonalização do resíduo resulta em

$$\int_0^L w(x) \left(\frac{d^2}{dx^2} \left(E(x)I(x) \frac{d^2 \tilde{v}(x)}{dx^2} \right) - q(x) \right) dx = 0 \quad (4.36)$$

ou

$$\int_0^L w(x) \left(\frac{d^2}{dx^2} \left(E(x)I(x) \frac{d^2 \tilde{v}(x)}{dx^2} \right) \right) dx = \int_0^L w(x)q(x) dx. \quad (4.37)$$

Por estas equações fica claro que o espaço de funções que satisfaz esta equação diferencial é, no mínimo, da classe C^4 , o que é bastante restritivo. Com o objetivo de amenizar este requisito, podemos realizar uma integração por partes no termo da esquerda, resultando em

$$w(x) \frac{d}{dx} \left(E(x)I(x) \frac{d^2 \tilde{v}(x)}{dx^2} \right) \Big|_0^L - \int_0^L \frac{dw(x)}{dx} \frac{d}{dx} \left(E(x)I(x) \frac{d^2 \tilde{v}(x)}{dx^2} \right) dx \quad (4.38)$$

de onde podemos verificar que mais uma integração por partes ainda é possível. Efetuando mais uma integração por partes na integral de domínio da equação anterior, obtemos

$$\frac{dw(x)}{dx} EI \frac{d^2 \tilde{v}(x)}{dx^2} \Big|_0^L - \int_0^L \frac{d^2 \tilde{v}(x)}{dx^2} E(x)I(x) \frac{d^2 w(x)}{dx^2} dx \quad (4.39)$$

resultando em

$$\int_0^L \frac{d^2 \tilde{v}(x)}{dx^2} E(x)I(x) \frac{d^2 w(x)}{dx^2} dx = \int_0^L w(x)q(x) dx \quad (4.40)$$

$$- w(x) \frac{d}{dx} \left(E(x)I(x) \frac{d^2 \tilde{v}(x)}{dx^2} \right) \Big|_0^L + \frac{dw(x)}{dx} EI \frac{d^2 \tilde{v}(x)}{dx^2} \Big|_0^L \quad (4.41)$$

e podemos identificar algumas grandezas nos termos de contorno. Iniciamos com a equivalência do esforço cortante

$$- \frac{d}{dx} \left(E(x)I(x) \frac{d^2 \tilde{v}(x)}{dx^2} \right) \Big|_0^L = V(x) \Big|_0^L \quad (4.42)$$

e do momento fletor

$$EI \frac{d^2 \tilde{v}(x)}{dx^2} \Big|_0^L = M(x) \Big|_0^L, \quad (4.43)$$

que são condições de contorno de Neumann (ou naturais). Assim,

$$\boxed{\int_0^L \frac{d^2 \tilde{v}(x)}{dx^2} E(x)I(x) \frac{d^2 w(x)}{dx^2} dx = \int_0^L w(x)q(x) dx + w(x)V(x) \Big|_0^L + \frac{dw(x)}{dx} M(x) \Big|_0^L} \quad (4.44)$$

Neste ponto, podemos verificar que após **duas** integrações por partes obtivemos uma expressão que impõe um requisito de continuidade de, no mínimo, C^2 para $\tilde{v}(x)$ e $w(x)$.

■ **Exemplo 4.2** Viga engastada com carregamento distribuído

Considerando o caso de uma viga engastada submetida a um carregamento $q(x) = q_0$, de acordo com a figura 4.1. Assuma que $E(x)$ e $I(x)$ são constantes.

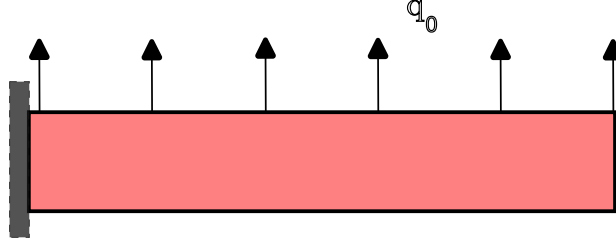


Figura 4.1: Viga engastada submetida a um carregamento distribuído constante.

Vamos considerar um polinômio cúbico para descrever o deslocamento $\tilde{v}(x)$. Aqui devemos ter um cuidado muito especial, pois o polinômio tem que descrever corretamente as condições de contorno essenciais, que são $v(0) = 0$ e $\theta(0) = 0$. Assim, partindo de um polinômio geral

$$\tilde{v}(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \quad (4.45)$$

devemos satisfazer

$$\tilde{v}(0) = a_0 + a_1 \cdot 0 + a_2 \cdot 0^2 + a_3 \cdot 0^3 \implies a_0 = 0 \quad (4.46)$$

e

$$\tilde{\theta}(0) = a_1 + 2a_2 \cdot 0 + 3a_3 \cdot 0 \implies a_1 = 0 \quad (4.47)$$

o que restringe o polinômio para

$$\tilde{v}(x) = a_2x^2 + a_3x^3. \quad (4.48)$$

Vamos utilizar a mesma base para a função peso, tal que

$$w(x) = b_2x^2 + b_3x^3. \quad (4.49)$$

Com isso, podemos avaliar os termos da Eq. (4.44)

$$\int_0^L \frac{d^2w(x)}{dx^2} EI \frac{d^2\tilde{v}(x)}{dx^2} dx = EI (12a_3b_3L^3 + (6a_2b_3 + 6a_3b_2)L^2 + 4a_2b_2L) \quad (4.50)$$

e

$$\int_0^L w(x)q_0 dx = \frac{q_0 (3b_3L^4 + 4b_2L^3)}{12}. \quad (4.51)$$

Os termos de contorno de Dirichlet, ou essenciais, são $w(0) = 0$ e $w'(0) = 0$, tal que ambos os termos de contorno se anulam em $x = 0$. Como não temos forças ou momentos concentrados

em $x = L$, também observamos que $V(L) = M(L) = 0$. Assim, todos os termos de contorno são nulos neste exemplo.

Agrupando os termos, obtemos

$$EI(12a_3b_3L^3 + (6a_2b_3 + 6a_3b_2)L^2 + 4a_2b_2L) = \frac{q_0(3b_3L^4 + 4b_2L^3)}{12} \quad (4.52)$$

tal que, separando por termos comuns em b , obtemos

$$b_2) \quad 6EIa_3L^2 + 4EIa_2L = \frac{q_0L^3}{3} \quad (4.53)$$

$$b_3) \quad 12EIa_3L^3 + 6EIa_2L^2 = \frac{q_0L^4}{4} \quad (4.54)$$

dando origem ao sistema linear

$$EI \begin{bmatrix} 4L & 6L^2 \\ 6L^2 & 12L^3 \end{bmatrix} \begin{Bmatrix} a_2 \\ a_3 \end{Bmatrix} = \begin{Bmatrix} \frac{q_0L^3}{3} \\ \frac{q_0L^4}{4} \end{Bmatrix} \quad (4.55)$$

com solução $a_2 = \frac{5q_0L^2}{24EI}$ e $a_3 = -\frac{q_0L}{12EI}$, tal que

$$\tilde{v}(x) = \frac{5q_0L^2}{24EI}x^2 - \frac{q_0L}{12EI}x^3 \quad (4.56)$$

que pode ser comparada a solução analítica

$$v(x) = \frac{q_0x^2}{24EI}(6L^2 - 4Lx + x^2) \quad (4.57)$$

que é um polinômio de grau 4. A Figura 4.2 ilustra a comparação das soluções para $q_0 = 1$, $EI = 1$ e $L = 1$. ■

Exercício 4.6 Solucione o problema da viga engastada com carregamento distribuído constante, utilizando um polinômio de quarto grau. Compare com a solução analítica. ■

Exercício 4.7 Considere o problema de uma viga engastada com um momento concentrado na ponta livre. Proponha um conjunto de base e solucione o problema utilizando o procedimento descrito neste capítulo. ■

Exercício 4.8 Considere o problema de uma viga bi-apoiada, submetida a um carregamento distribuído com variação linear. Proponha uma solução em um espaço com bases senoidais. Investigue o refino da solução. ■

4.3 Métodos de Galerkin

Os métodos de Galerkin são métodos que convertem uma equação diferencial descrita por um operador contínuo em um operador discreto, por meio do uso de bases finito dimensionais, da forma fraca e de projeções. Isso foi basicamente o que estudamos nos capítulos anteriores.

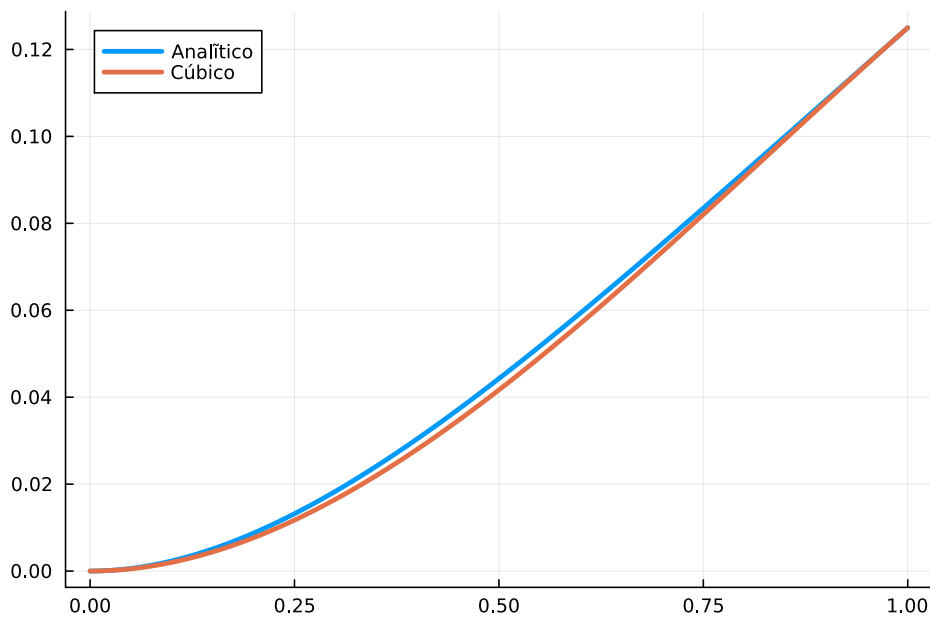


Figura 4.2: Comparação entre a solução aproximada com um polinômio de terceiro grau e a solução analítica de quarto grau. para o problema da viga longa.

Embora o nome Galerkin seja advindo do matemático Russo Boris Galerkin, geralmente o conceito é bem geral e existem denominações um pouco mais específicas.

Quando utilizamos conjuntos de bases diferentes para a função aproximada e para a função teste, geralmente chamamos o procedimento de Petrov-Galerkin e, no caso particular em que utilizamos as mesmas bases, de Bubnov-Galerkin. Também é comum citarmos o método de Ritz-Galerkin, que é ainda mais restritivo, pois assume que a forma bilinear do operador deve ser positivo-definida. Isso geralmente está associado a alguma medida de energia de um sistema físico, de tal forma que para os problemas que estudamos aqui, pode haver uma confusão entre o método de Ritz-Galerkin e o de Bubnov-Galerkin.

No contexto da literatura de Elementos Finitos, é muito comum denotar o procedimento de utilizar as mesmas bases como sendo o método de Galerkin tão somente.



Elementos unidimensionais

5	Elemento finito de barra	51
5.1	Dedução do elemento finito de barra	
5.2	Nomenclatura e interpretação dos termos	
5.3	Obtendo a matriz de rigidez por equilíbrio de forças	
5.4	Simetria da matriz de rigidez	
6	Elemento de barra: abordagem energética	67
6.1	Princípio da mínima energia potencial	
6.2	Interpretando como um problema de perturbação da condição de equilíbrio - Princípio dos trabalhos virtuais (PTV)	
6.3	Utilizando o PMEP para obter o elemento de barra	
6.4	Interpretando a solução do MEF como um problema de otimização	
7	Rotação de sistemas de referência	73
8	Elemento de barra em 2D - treliças	77
8.1	Treliças bidimensionais	
9	Implementação computacional - treliças	83
9.1	Etapas de uma análise por elementos finitos	
9.2	Entrada de dados	
9.3	Pré-processamento	
10	Elemento finito de viga longa	93
10.1	Elemento Finito de Viga Longa	
10.2	Sobreposição de elementos de viga	
11	Elemento Finito de Pórtico Plano	101
11.1	Algoritmos para implementação computacional	
11.2	Exemplo de pórtico plano	
12	C.C. de Dirichlet não homogêneas	109
12.1	Técnica 1 - Rearranjo do sistema de equações	
12.2	Técnica 2 - Lagrangiano	

5. Elemento finito de barra

A barra é o modelo estrutural mais simples, decorrente de uma série de hipóteses simplificativas. Sua equação diferencial de equilíbrio foi deduzida no começo deste texto

$$\frac{d}{dx} \left(E(x) \frac{du_x(x)}{dx} \right) + b_x(x) = 0 \quad x \in (0, L) \quad (5.1)$$

correspondendo a uma EDO linear de segunda ordem. A variável dependente é o deslocamento axial $u_x(x)$ e a variável independente é a coordenada x (seção transversal). Essa equação diferencial foi utilizada em boa parte do texto anterior que apresentou o método dos resíduos ponderados, de tal forma que já temos uma boa noção das propriedades matemáticas. Por simplicidade, vamos assumir que o módulo de elasticidade $E(x)$ é constante em x . A solução da Eq. (5.1) é a função $u_x(x)$ que satisfaz a EDO exatamente em todos os pontos, além de satisfazer as condições de contorno (solução forte). A barra é definida na Figura 5.1, com um sistema de referência local x (centroidal), em que também indicamos o deslocamento axial $u_x(x)$ de uma seção transversal qualquer.

Como vimos nos capítulos anteriores, podemos lançar mão do método dos Resíduos Ponderados e também da integração por partes para obtermos uma aproximação $\tilde{u}_x(x)$. O

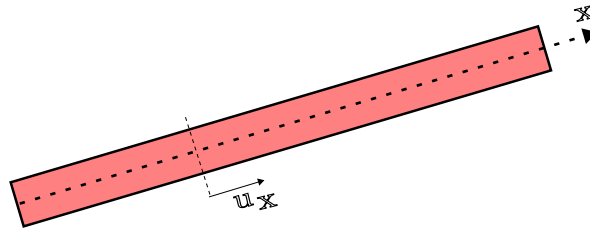


Figura 5.1: Barra descrita por um sistema de coordenadas local x .

resíduo é definido como

$$r(x) = E \frac{d^2 \tilde{u}_x(x)}{dx^2} + b_x(x) \quad (5.2)$$

e definimos a ortogonalização do resíduo em relação a uma função peso $w(x)$ como

$$\int_0^L w(x) \left(E \frac{d^2 \tilde{u}_x(x)}{dx^2} + b_x(x) \right) dx = 0 \quad (5.3)$$

tal que

$$\int_0^L w(x) E \frac{d^2 \tilde{u}_x(x)}{dx^2} dx = - \int_0^L w(x) b_x(x) dx. \quad (5.4)$$

Integrando por partes o termo do lado esquerdo resulta em

$$w(x) E \frac{d\tilde{u}_x(x)}{dx} \cdot \mathbf{n} \Big|_0^L - \int_0^L \frac{dw(x)}{dx} E \frac{d\tilde{u}_x(x)}{dx} dx = - \int_0^L w(x) b_x(x) dx \quad (5.5)$$

ou

$$\boxed{\int_0^L \frac{dw(x)}{dx} E \frac{d\tilde{u}_x(x)}{dx} dx = \int_0^L w(x) b_x(x) dx + w(x) E \frac{d\tilde{u}_x(x)}{dx} \cdot \mathbf{n} \Big|_0^L}. \quad (5.6)$$

Agora precisamos propor uma expressão para $\tilde{u}_x(x)$ que satisfaça as condições de contorno, mas as condições de contorno vão mudar para cada problema.

Por exemplo, podemos supor que a aproximação utiliza uma base polinomial linear

$$\tilde{u}_x(x) = a_0 + a_1 x \quad (5.7)$$

mas essa expressão precisará ser modificada, a depender das informações das condições de contorno (como fizemos nos capítulos anteriores).

5.1 Dedução do elemento finito de barra

O método dos Elementos Finitos nada mais é do que uma maneira estruturada de propormos a função $\tilde{u}_x(x)$ de forma 'genérica', de tal forma que podemos obter uma solução que irá acomodar diferentes valores de condições de contorno.

Para isso, vamos iniciar escolhendo dois **pontos discretos** ao longo da barra. Por simplicidade, vamos selecionar os extremos da barra, como ilustrado na Fig. 5.2

- Ponto $x = 0$ será chamado de nó 1;
- Ponto $x = L$ será chamado de nó 2

Com isso, vamos assumir que os deslocamentos axiais nestes pontos são 'conhecidos'

- $u_x(0) = U_1$;
- $u_x(L) = U_2$.

Com isso, podemos avaliar a nossa proposta de solução

$$\tilde{u}_x(x) = a_0 + a_1 x \quad (5.8)$$

nos pontos 'conhecidos', tal que

$$\tilde{u}_x(0) = U_1 = a_0 + a_1 0 \implies a_0 = U_1 \quad (5.9)$$

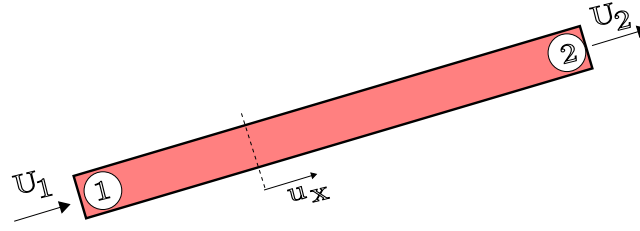


Figura 5.2: Nós 1 e 2 e seus deslocamentos axiais.

e

$$\tilde{u}_x(L) = U_2 = a_0 + a_1 L \implies a_1 = \frac{U_2 - U_1}{L} \quad (5.10)$$

de tal forma que

$$\tilde{u}_x(x) = U_1 + \frac{U_2 - U_1}{L}x. \quad (5.11)$$

Neste ponto, podemos reescrever a expressão anterior como

$$\tilde{u}_x(x) = \left(1 - \frac{x}{L}\right)U_1 + \left(\frac{x}{L}\right)U_2 \quad (5.12)$$

ou

$$\tilde{u}_x(x) = N_1(x)U_1 + N_2(x)U_2. \quad (5.13)$$

As funções de base

$$N_1(x) = \left(1 - \frac{x}{L}\right) \quad (5.14)$$

e

$$N_2(x) = \left(\frac{x}{L}\right) \quad (5.15)$$

são chamadas de funções de interpolação ou funções de forma.

Neste ponto, devemos observar algumas informações importantes:

- Os valores de deslocamentos nodais são 'discretos' (pontuais) e não dependem da variável x ;
- A aproximação $\tilde{u}_x(x)$ continua sendo linear, mas agora estamos utilizando as bases $N_i(x)$ ao invés da base polinomial canônica $\phi = \{1, x\}$;
- Não estamos mais utilizando coeficientes genéricos a_i mas os valores discretos de deslocamentos nodais como coeficientes da interpolação;

Em relação as funções de interpolação, notamos que essas possuem as seguintes propriedades:

- Tem a mesma ordem da interpolação original;
- A função associada ao nó 1, $N_1(x)$ assume valor unitário em $x = 0$ e é nula em $x = L$;
- A função associada ao nó 2, $N_2(x)$ assume valor unitário em $x = L$ e é nula em $x = 0$.

Continuando, podemos considerar então a interpolação da Eq. (5.13) na forma fraca da Eq. (5.6). Seguindo o procedimento de Bubnov-Galerkin, utilizamos a mesma base para interpolar a função de ponderação

$$w(x) = N_1(x)W_1 + N_2(x)W_2. \quad (5.16)$$

Vamos avaliar os termos necessários para calcular a Eq. (5.6).

Começando com as derivadas do termo do lado esquerdo

$$\frac{dw(x)}{dx} = \frac{d}{dx} (N_1(x)W_1 + N_2(x)W_2) = \frac{dN_1}{dx}W_1 + \frac{dN_2}{dx}W_2 \quad (5.17)$$

e

$$\frac{d\tilde{u}_x(x)}{dx} = \frac{d}{dx} (N_1(x)U_1 + N_2(x)U_2) = \frac{dN_1}{dx}U_1 + \frac{dN_2}{dx}U_2, \quad (5.18)$$

de tal forma que o primeiro termo da Eq. (5.6) pode ser escrito como

$$\begin{aligned} \int_0^L \frac{dw(x)}{dx} E \frac{d\tilde{u}_x(x)}{dx} dx = \int_0^L \left(\frac{dN_1(x)}{dx} E \frac{dN_1(x)}{dx} U_1 W_1 + \frac{dN_2(x)}{dx} E \frac{dN_1(x)}{dx} U_1 W_2 \right. \\ \left. + \frac{dN_1(x)}{dx} E \frac{dN_2(x)}{dx} U_2 W_1 + \frac{dN_2(x)}{dx} E \frac{dN_2(x)}{dx} U_2 W_2 \right) dx \end{aligned} \quad (5.19)$$

ou

$$\begin{aligned} \int_0^L \frac{dw(x)}{dx} E \frac{d\tilde{u}_x(x)}{dx} dx = W_1 \int_0^L \frac{dN_1(x)}{dx} E \frac{dN_1(x)}{dx} dx U_1 + W_2 \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_1(x)}{dx} dx U_1 \\ + W_1 \int_0^L \frac{dN_1(x)}{dx} E \frac{dN_2(x)}{dx} dx U_2 + W_2 \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_2(x)}{dx} dx U_2. \end{aligned} \quad (5.20)$$

O primeiro termo do lado direito depende de como vamos representar as forças de corpo $b_x(x)$. A ideia do método dos Elementos Finitos é representar todas as grandezas por meio dos valores discretos nas posições dos nós. Desta forma, podemos utilizar

$$b_x(x) = N_1(x)b_1 + N_2(x)b_2 \quad (5.21)$$

em que b_1 e b_2 são os valores nodais. Observe que esse procedimento permite apenas a descrição de forças de corpo com variação linear no domínio. Com isso,

$$\int_0^L w(x)b_x(x) dx = \int_0^L (N_1(x)W_1 + N_2(x)W_2) (N_1(x)b_1 + N_2(x)b_2) dx \quad (5.22)$$

tal que

$$\begin{aligned} \int_0^L w(x)b_x(x) dx = W_1 \left(b_1 \int_0^L N_1(x)N_1(x) dx + b_2 \int_0^L N_1(x)N_2(x) dx \right) \\ + W_2 \left(b_1 \int_0^L N_2(x)N_1(x) dx + b_2 \int_0^L N_2(x)N_2(x) dx \right). \end{aligned} \quad (5.23)$$

Por fim, devemos avaliar os termos de contorno

$$w(x)E \frac{d\tilde{u}_x(x)}{dx} \cdot \mathbf{n} \Big|_0^L. \quad (5.24)$$

Para isso, devemos lembrar que o termo

$$E \frac{d\tilde{u}_x(x)}{dx} \cdot \mathbf{n} = t_n(x) \quad [N/m^2] \quad (5.25)$$

é o carregamento distribuído aplicado em uma das extremidades. Como discutido no exemplo de barra no capítulo inicial deste texto, podemos descrever esse carregamento como

$$t_n = \frac{F}{A} \quad (5.26)$$

em que F é a força normal aplicada na extremidade e A a área da seção transversal. O sinal de t_n (e de F_n) é positivo se a força estiver na mesma direção da normal da face e negativo se for contrário.

Assim,

$$w(x)E \frac{d\tilde{u}_x(x)}{dx} \cdot \mathbf{n} \Big|_0^L = (N_1(L)W_1 + N_2(L)W_2) \frac{F_2}{A} - (N_1(0)W_1 + N_2(0)W_2) \frac{F_1}{A} \quad (5.27)$$

em que F_1 é a força normal aplicada no nó 1 e F_2 a força normal aplicada no nó 2.

Das propriedades das funções $N_i(x)$, identificamos que $N_1(L) = N_2(0) = 0$ e $N_1(0) = N_2(L) = 1$, tal que

$$w(x)E \frac{d\tilde{u}_x(x)}{dx} \cdot \mathbf{n} \Big|_0^L = W_2 \frac{F_2}{A} - W_1 \frac{F_1}{A}. \quad (5.28)$$

Por fim, se definirmos as forças F_1 e F_2 com os mesmos sentidos de U_1 e U_2 , verificamos que F_1 está na direção oposta a normal da face do nó 1, tal que devemos inverter o seu sinal, resultando em

$$w(x)E \frac{d\tilde{u}_x(x)}{dx} \cdot \mathbf{n} \Big|_0^L = W_2 \frac{F_2}{A} + W_1 \frac{F_1}{A}. \quad (5.29)$$

Por fim, considerando os resultados das Eqs. (5.20), (5.23) e (5.29), podemos agrupar por temos W em comum, tal que

$$\begin{aligned} W1) \quad & \int_0^L \frac{dN_1(x)}{dx} E \frac{dN_1(x)}{dx} dx U_1 + \int_0^L \frac{dN_1(x)}{dx} E \frac{dN_2(x)}{dx} dx U_2 = \\ & b_1 \int_0^L N_1(x)N_1(x) dx + b_2 \int_0^L N_1(x)N_2(x) dx + \frac{F_1}{A} \end{aligned} \quad (5.30)$$

e

$$\begin{aligned} W2) \quad & \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_1(x)}{dx} dx U_1 + \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_2(x)}{dx} dx U_2 = \\ & b_1 \int_0^L N_2(x)N_1(x) dx + b_2 \int_0^L N_2(x)N_2(x) dx + \frac{F_2}{A} \end{aligned} \quad (5.31)$$

resultando em um sistema de duas equações lineares, que pode ser descrito em uma forma matricial

$$\begin{bmatrix} \int_0^L \frac{dN_1(x)}{dx} E \frac{dN_1(x)}{dx} dx & \int_0^L \frac{dN_1(x)}{dx} E \frac{dN_2(x)}{dx} dx \\ \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_1(x)}{dx} dx & \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_2(x)}{dx} dx \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} \frac{F_1}{A} \\ \frac{F_2}{A} \end{Bmatrix} + \begin{Bmatrix} b_1 \int_0^L N_1(x)N_1(x) dx + b_2 \int_0^L N_1(x)N_2(x) dx \\ b_1 \int_0^L N_2(x)N_1(x) dx + b_2 \int_0^L N_2(x)N_2(x) dx \end{Bmatrix}$$

ou, multiplicando toda a equação pela área A ,

$$A \begin{bmatrix} \int_0^L \frac{dN_1(x)}{dx} E \frac{dN_1(x)}{dx} dx & \int_0^L \frac{dN_1(x)}{dx} E \frac{dN_2(x)}{dx} dx \\ \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_1(x)}{dx} dx & \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_2(x)}{dx} dx \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} + A \begin{Bmatrix} b_1 \int_0^L N_1(x)N_1(x) dx + b_2 \int_0^L N_1(x)N_2(x) dx \\ b_1 \int_0^L N_2(x)N_1(x) dx + b_2 \int_0^L N_2(x)N_2(x) dx \end{Bmatrix}.$$

Por fim, podemos avaliar as integrais da expressão anterior. Começando com as integrais da matriz de coeficientes do lado esquerdo, precisamos das derivadas

$$\frac{dN_1}{dx} = \frac{d}{dx} \left(1 - \frac{x}{L} \right) = -\frac{1}{L} \quad (5.32)$$

e

$$\frac{dN_2}{dx} = \frac{d}{dx} \left(\frac{x}{L} \right) = \frac{1}{L}. \quad (5.33)$$

Com isso, as integrais que definem a matriz são

$$\int_0^L \frac{dN_1(x)}{dx} E \frac{dN_1(x)}{dx} dx = \int_0^L \frac{-1}{L} E \frac{-1}{L} dx = \frac{E}{L}, \quad (5.34)$$

$$\int_0^L \frac{dN_1(x)}{dx} E \frac{dN_2(x)}{dx} dx = \int_0^L \frac{dN_2(x)}{dx} E \frac{dN_1(x)}{dx} dx = \int_0^L \frac{-1}{L} E \frac{1}{L} dx = -\frac{E}{L}, \quad (5.35)$$

e

$$\int_0^L \frac{dN_2(x)}{dx} E \frac{dN_2(x)}{dx} dx = \int_0^L \frac{1}{L} E \frac{1}{L} dx = \frac{E}{L}. \quad (5.36)$$

As integrais associadas ao termo de força de corpo são

$$\int_0^L N_1(x) N_1(x) dx = \frac{L}{3}, \quad (5.37)$$

$$\int_0^L N_1(x) N_2(x) dx = \int_0^L N_2(x) N_1(x) dx = \frac{L}{6} \quad (5.38)$$

e

$$\int_0^L N_2(x) N_2(x) dx = \frac{L}{3}. \quad (5.39)$$

Com isso, podemos escrever o sistema linear como

$$\boxed{\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} + AL \begin{Bmatrix} \frac{b_1}{3} + \frac{b_2}{6} \\ \frac{b_1}{6} + \frac{b_2}{3} \end{Bmatrix}}. \quad (5.40)$$

■ **Exemplo 5.1** Vamos utilizar o mesmo exemplo do capítulo inicial, onde temos uma barra de módulo de elasticidade constante E e área constante A . Vamos assumir que a barra está presa em $x = 0$ e está sendo submetida a uma força concentrada F_2 na extremidade da direita, $x = L$. Não existem forças de corpo.

Com isso, sabemos que no nó 1 ($x = 0$) o deslocamento U_1 é conhecido e nulo - condição de contorno de Dirichlet. No entanto, não sabemos a força que atua nesse nó. Da mesma forma, não sabemos o deslocamento no nó 2, mas sabemos que existe uma força concentrada F_2 - condição de contorno de Neumann. Como pode ser observado, em um problema de valor no contorno, ou sabemos o deslocamento ou sabemos a força, de tal modo que sempre poderemos operar com o sistema definido na Eq. (5.40). Assim, substituindo os valores

$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} 0 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} ? \\ F_2 \end{Bmatrix} \quad (5.41)$$

ou, desenvolvendo as equações para esse primeiro exemplo

$$\frac{EA}{L} 0 - \frac{EA}{L} U_2 = ? \quad (5.42)$$

e

$$-\frac{EA}{L}0 + \frac{EA}{L}U_2 = F_2 \quad (5.43)$$

e como nossa única incógnita da ED é U_2 (lembre-se que estamos solucionando uma ED e que o objetivo é obter a variável dependente, que é o deslocamento) precisamos de somente uma equação. Podemos

$$-\frac{EA}{L}0 + \frac{EA}{L}U_2 = F_2 \implies U_2 = \frac{F_2 L}{EA} \quad (5.44)$$

ver, então, que a segunda equação permite a obtenção da solução, pois

$$-\frac{EA}{L}0 + \frac{EA}{L}U_2 = F_2 \implies U_2 = \frac{F_2 L}{EA} \quad (5.45)$$

que é a solução analítica do problema. Mas como obtivemos a solução analítica se estamos utilizando um método de solução 'aproximada'? Se o leitor acompanhou com cuidado os primeiros capítulos deste texto, pode verificar que o método dos Resíduos Ponderados leva a solução analítica quando utilizamos a base correta (a que contém a solução correta). Nesta caso, já vimos no capítulo em que obtivemos soluções analíticas, que a solução para o deslocamento axial ao longo do comprimento é

$$u_x(x) = \frac{F}{EA}x \quad (5.46)$$

e, no nosso exemplo atual, a força aplicada na ponta é F_2 e estamos calculando o deslocamento axial no segundo nó, ou seja, em $x = L$.

Neste ponto, podemos verificar que o método permite:

- Informar as condições de contorno de Dirichlet diretamente no sistema de equações lineares;
- Informar as condições de contorno de Neumann diretamente no sistema de equações lineares;
- Obter as incógnitas (deslocamentos nodais) utilizando as equações do sistema linear.

Por fim, de posse dos deslocamentos nodais, podemos **interpolar** os deslocamentos no domínio, utilizando a proposta de interpolação

$$u_x(x) = N_1(x)U_1 + N_2(x)U_2 \quad (5.47)$$

que no nosso exemplo resulta em

$$u_x(x) = \left(1 - \frac{x}{L}\right)0 + \left(\frac{x}{L}\right)\frac{F_2 L}{EA} = \frac{F_2}{EA}x. \quad (5.48)$$

Portanto, para barras submetidas a carregamentos concentrados nos nós, teremos a solução analítica. ■

■ **Exemplo 5.2** Vamos considerar a mesma barra do exemplo anterior, mas vamos trocar a força concentrada por uma força de corpo constante b $[N/m^3]$. Neste caso, temos

$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} 0 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} ? \\ 0 \end{Bmatrix} + AL \begin{Bmatrix} \frac{b}{3} + \frac{b}{6} \\ \frac{b}{6} + \frac{b}{3} \end{Bmatrix} \quad (5.49)$$

e, novamente, temos que considerar somente a segunda equação

$$\frac{EA}{L}0 + \frac{EA}{L}U_2 = \frac{bLA}{2} \quad (5.50)$$

tal que

$$U_2 = \frac{bL^2}{2E}. \quad (5.51)$$

A interpolação ao longo do comprimento da barra resulta em

$$u_x(x) = \left(1 - \frac{x}{L}\right) 0 + \left(\frac{x}{L}\right) \frac{bL^2}{2E} = \frac{bL}{2E} x. \quad (5.52)$$

■

Exercício 5.1 Obtenha a solução analítica para o problema do exemplo anterior e compare com a solução obtida pelo método dos elementos finitos. Verifique, também, a qualidade da solução no nó 2. Qual é a explicação? ■

5.1.1 Notação matricial

A dedução que fizemos para a obtenção do sistema da Eq. (5.40) foi realizada da forma mais direta possível, para facilitar a compreensão de cada operação matemática. No entanto, o mesmo procedimento pode ser realizado de uma forma mais compacta se utilizamos algumas operações com matrizes e vetores.

Para começar, podemos reescrever a Eq. (5.13) na forma

$$u_x(x) = \mathbf{N}\mathbf{U} \quad (5.53)$$

em que

$$\mathbf{N} = [N_1(x) \quad N_2(x)] \quad (5.54)$$

é uma matriz de dimensão 1×2 contendo as funções de interpolação do elemento e

$$\mathbf{U} = \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} \quad (5.55)$$

é um vetor 2×1 contendo os deslocamentos nodais do elemento. Podemos observar que o produto de uma matriz 1×2 por um vetor 2×1 resulta em um escalar, como desejado. O mesmo vale para a função de ponderação, tal que

$$w_x(x) = \mathbf{N}\mathbf{W} \quad (5.56)$$

em que \mathbf{W} é um vetor contendo W_1 e W_2 .

Com esse resultado, podemos reescrever a Eq. (5.18) como

$$\frac{du_x(x)}{dx} = \frac{d}{dx}(\mathbf{N}\mathbf{U}) = \frac{d\mathbf{N}}{dx}\mathbf{U} = \begin{bmatrix} \frac{dN_1}{dx} & \frac{dN_2}{dx} \end{bmatrix} \mathbf{U} = \mathbf{B}\mathbf{U} \quad (5.57)$$

e chamamos a matriz \mathbf{B} de matriz de derivada das funções de interpolação. Essa matriz é muito importante na teoria de elementos finitos e será utilizada ao longo de todo o texto. Da mesma forma, podemos escrever a Eq. (5.17) como

$$\frac{dw(x)}{dx} = \frac{d}{dx}(\mathbf{N}\mathbf{W}) = \frac{d\mathbf{N}}{dx}\mathbf{W} = \begin{bmatrix} \frac{dN_1}{dx} & \frac{dN_2}{dx} \end{bmatrix} \mathbf{W} = \mathbf{B}\mathbf{W}. \quad (5.58)$$

Com essas definições, podemos verificar que as operações realizadas para obtermos a Eq. (5.20) podem ser condensadas como

$$\int_0^L \frac{dw}{dx} E \frac{du_x}{dx} dx = \int_0^L (\mathbf{B}\mathbf{W})^T E (\mathbf{B}\mathbf{U}) dx \quad (5.59)$$

e, como $(\mathbf{BW})^T = \mathbf{W}^T \mathbf{B}^T$

$$\int_0^L \frac{dw}{dx} E \frac{du_x}{dx} dx = \int_0^L \mathbf{W}^T \mathbf{B}^T E \mathbf{B} \mathbf{U} dx = \mathbf{W}^T \int_0^L \mathbf{B}^T E \mathbf{B} dx \mathbf{U}. \quad (5.60)$$

A Eq. (5.29) também pode ser escrita na forma compacta como

$$\frac{1}{A} \mathbf{W}^T \mathbf{F}. \quad (5.61)$$

Por fim, as operações que resultaram no vetor de forças de corpo, Eq. (5.22), podem ser escritas como

$$\int_0^L w(x) b_x(x) dx = \int_0^L (\mathbf{N} \mathbf{W})^T (\mathbf{N} \mathbf{b}) dx = \mathbf{W}^T \int_0^L \mathbf{N}^T \mathbf{N} \mathbf{b} dx \quad (5.62)$$

em que o vetor \mathbf{b} contém os valores nodais da força de corpo. Assim, agrupando os termos e multiplicando todos pela área A , obtemos

$$\mathbf{W}^T A \int_0^L \mathbf{B}^T E \mathbf{B} dx \mathbf{U} = \mathbf{W}^T \mathbf{F} + \mathbf{W}^T A \int_0^L \mathbf{N}^T \mathbf{N} \mathbf{b} dx \quad (5.63)$$

que é uma equação escalar. Se colocarmos o vetor \mathbf{W}^T em evidência

$$\mathbf{W}^T \left(A \int_0^L \mathbf{B}^T E \mathbf{B} dx \mathbf{U} - \mathbf{F} - A \int_0^L \mathbf{N}^T \mathbf{N} \mathbf{b} dx \right) = 0 \quad (5.64)$$

podemos verificar que a igualdade é satisfeita para quaisquer valores de \mathbf{W} , desde que o termo entre parêntesis seja igual a um vetor nulo, isto é

$$A \int_0^L \mathbf{B}^T E \mathbf{B} dx \mathbf{U} - \mathbf{F} - A \int_0^L \mathbf{N}^T \mathbf{N} \mathbf{b} dx = \mathbf{0} \quad (5.65)$$

ou

$$A \int_0^L \mathbf{B}^T E \mathbf{B} dx \mathbf{U} = \mathbf{F} + A \int_0^L \mathbf{N}^T \mathbf{N} \mathbf{b} dx \quad (5.66)$$

que é exatamente o sistema de equações que obtivemos na Eq. (5.40).

5.2 Nomenclatura e interpretação dos termos

O sistema linear de equações da Eq. (5.40) pode ser escrito como

$$\mathbf{K}_{2 \times 2} \mathbf{U}_{2 \times 1} = \mathbf{F}_{2 \times 1} + \mathbf{F}_{\mathbf{b} 2 \times 1} \quad (5.67)$$

e a matriz \mathbf{K}

$$\mathbf{K} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (5.68)$$

é chamada de matriz de rigidez (do elemento de barra). A rigidez é a relação entre a força aplicada em um ponto j e o deslocamento gerado em outro ponto i . Assim, definimos a rigidez K_{ij} como

$$F_i = K_{ij} U_j \quad (5.69)$$

e podemos verificar que a matriz de rigidez contém termos de rigidez relacionando forças e deslocamentos em pontos discretos (nós) do elemento. Como temos dois nós, temos uma matriz 2×2 .

O vetor \mathbf{F}_b é chamado de vetor de carregamento consistente de forças de corpo, pois ele utiliza as mesmas funções de interpolação que foram utilizadas para a dedução da matriz de rigidez. Sua interpretação é que a força de corpo, que é uma força distribuída por todo o domínio, é projetada para os nós, de forma discreta. Por exemplo, se tivermos um valor constante de força de corpo, b , obtemos

$$\mathbf{F}_b = \frac{bLA}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \quad (5.70)$$

ou seja, metade da força total bLA é aplicada em cada nó (de tal forma que a força total é conservada). Da teoria de mecânica dos sólidos, sabemos que não é correto concentrar forças distribuídas (tanto de superfície quanto de corpo), a não ser para calcular reações.

5.3 Obtendo a matriz de rigidez por equilíbrio de forças

Desconsiderando os termos de força de corpo para simplificar a análise, podemos verificar que a matriz de rigidez que foi obtida por resíduos ponderados também pode ser obtida por uma abordagem mais física. Seja a Figura 5.3, onde o sentido positivo do eixo X é para a direita.

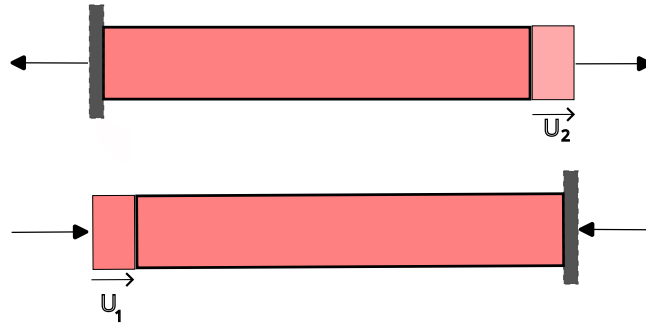


Figura 5.3: Dedução da matriz de rigidez por equilíbrio

Iniciamos aplicando um **deslocamento arbitrário** U_2 na posição do segundo nó e restringindo o deslocamento no primeiro nó (parte superior da Figura). Da teoria de barras, sabemos que a rigidez axial de uma barra é dada por

$$F = \frac{EA}{L} U \quad (5.71)$$

onde tanto a força F quanto o deslocamento U estão no mesmo ponto. Desta forma, podemos verificar que é necessário aplicar uma força positiva

$$F_2^2 = \frac{EA}{L} U_2 \quad (5.72)$$

na posição do nó 2. Vamos utilizar o super índice 2 para indicar que estamos aplicando o deslocamento unitário no segundo nó. No entanto, do equilíbrio, sabemos que deve surgir

uma **reação** de mesma magnitude, mas sentido contrário, na posição do nó 1, isto é

$$F_1^2 = -\frac{EA}{L}U_2. \quad (5.73)$$

Podemos realizar o mesmo procedimento, aplicando um deslocamento arbitrário no nó 1 e prendendo o nó 2. Neste caso, a força necessária para deslocar o nó 1 será

$$F_1^1 = \frac{EA}{L}U_1 \quad (5.74)$$

e termos uma reação no nó 2, com sentido negativo

$$F_2^1 = -\frac{EA}{L}U_1. \quad (5.75)$$

Como o sistema é **linear**, podemos considerar a aplicação das suas situações anteriores ao mesmo tempo pela **sobreposição de efeitos**. Em sistemas mecânicos, sabemos que devemos sempre respeitar o somatório de forças, tal que

$$F_1 = F_1^1 + F_1^2 = \frac{EA}{L}U_1 - \frac{EA}{L}U_2 \quad (5.76)$$

e

$$F_2 = F_2^1 + F_2^2 = -\frac{EA}{L}U_1 + \frac{EA}{L}U_2. \quad (5.77)$$

Esse sistema de equações lineares pode ser escrito na forma matricial, tal que

$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}. \quad (5.78)$$

A dedução por equilíbrio de forças é bem intuitiva, mas não permite que o usuário tenha noção de todas as questões envolvendo a solução da equação diferencial e, pior, não indica para o usuário (pelo menos diretamente) que esse elemento tem uma interpolação linear do campo de deslocamentos. Isso é ainda mais limitante em elementos finitos mais 'complicados', como no caso de problemas de elasticidade 2D/3D.

5.4 Simetria da matriz de rigidez

A matriz de rigidez é simétrica, isto é, $K_{ij} = K_{ji}$ e isso fica claro na dedução por equilíbrio, pois a reação é igual, em módulo, a força que foi aplicada para aplicar o deslocamento no nó livre. Isso está relacionado ao princípio de Maxwell-Betti, válido para problemas lineares e elásticos. No contexto da dedução mais matemática que fizemos antes, a simetria do operador é resultado do fato de termos utilizado o mesmo conjunto de bases para descrever o deslocamento e a função de ponderação e ao fato do operador

$$\int_0^L \frac{dw}{dx} E \frac{du}{dx} dx \quad (5.79)$$

ser simétrico em relação aos graus das derivadas (obtido pelo procedimento de integração por partes).

5.4.1 Sobreposição de barras

O sistema 2×2 da Eq. (5.40) descreve o equilíbrio discreto de **uma** barra, definida por **dois nós**, cada um com um **grau de liberdade** axial. Assim, temos um total de **dois graus de liberdade** (axiais) em um elemento de barra, tal que o sistema tem essa dimensão. Um elemento de barra permite a descrição dos deslocamentos e forças nodais (discretos) e também a interpolação dos deslocamentos ao longo do elemento, baseado nos deslocamentos nodais U_1 e U_2 . Um ponto importante é que não podemos informar forças aplicadas em alguma posição que não seja um dos nós, além do fato de termos utilizado a hipótese de que tanto E quanto A são constantes no domínio do elemento. Assim, se quisermos utilizar o procedimento para analisar uma estrutura como a da Fig. 5.4, que é submetido a duas forças concentradas P e Q e com propriedades diferentes, precisamos utilizar mais de um elemento.

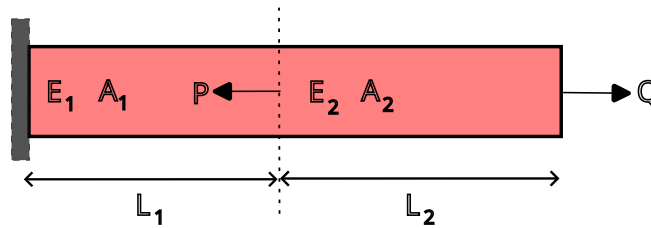


Figura 5.4: Barra com dois carregamentos axiais concentrados.

Para entendermos o procedimento a ser adotado, precisamos identificar um conjunto de **graus de liberdade discretos** do problema. No caso da teoria de barra, um grau de liberdade é um movimento ou força axial a ser observado/descrito na solução do problema. Observamos que o conjunto mínimo de graus de liberdade para descrever esse problema é 3, para que possamos descrever o apoio, a força P e a força Q . No exemplo em questão, observamos que a mudança nas propriedades coincide com a aplicação da força P . No nosso modelo discreto, vimos que os graus de liberdade estão associados aos nós do modelo, de tal forma que precisamos de três nós, como ilustrado na Figura 5.5.

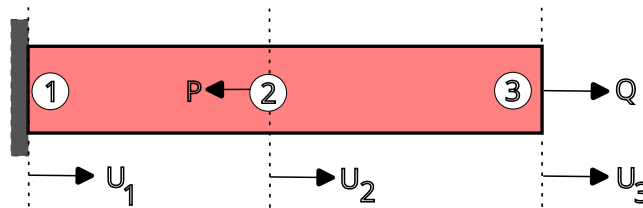


Figura 5.5: Discretização da barra em 3 nós.

Assim, dos conceitos de matriz de rigidez, vetor de deslocamentos nodais e vetor de forças nodais, verificamos que essa estrutura deve ser descrita por um sistema discreto 3×3 , na forma

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}. \quad (5.80)$$

Da definição de rigidez que vimos anteriormente, podemos identificar que K_{ij} é a relação entre o deslocamento i e a força j .

Do exemplo anterior, com uma barra, já sabemos que o deslocamento no nó do apoio é nulo, tal que $U_1 = 0$. Também sabemos que a força aplicada no nó 2 é $-P$ e a força aplicada no nó 3 é Q . No entanto, só sabemos montar a matriz de rigidez 2×2 de um elemento com 2 nós.

Vamos estudar, então, cada um dos trechos que ligam dois nós. Começando com o que vamos chamar de 'primeiro' elemento, identificado pelos nós 1 e 2 do modelo que temos a barra descrita por

$$\frac{E_1 A_1}{L_1} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} \quad (5.81)$$

e podemos 'aumentar' esse sistema (do primeiro elemento) para que tenha a dimensão do problema completo

$$\frac{E_1 A_1}{L_1} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \mathbf{F}^1 \quad (5.82)$$

sem modificar o significado do sistema 2×2 original, pois U_3 e F_3 estão sendo multiplicados por 0. Podemos identificar um 'segundo' elemento entre os nós 2 e 3, com equilíbrio descrito por

$$\frac{E_2 A_2}{L_2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_2 \\ F_3 \end{Bmatrix} \quad (5.83)$$

e podemos aumentar esse sistema, da mesma forma que fizemos no primeiro elemento

$$\frac{E_2 A_2}{L_2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \mathbf{F}^2. \quad (5.84)$$

Como já vimos em deduções anteriores, sabemos que um sistema mecânico em equilíbrio deve estar em equilíbrio de forças, tal que

$$\mathbf{F} = \mathbf{F}^1 + \mathbf{F}^2 \quad (5.85)$$

ou

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \frac{E_1 A_1}{L_1} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} + \frac{E_2 A_2}{L_2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} \quad (5.86)$$

ou, colocando o vetor de deslocamentos em evidência

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \left(\frac{E_1 A_1}{L_1} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{E_2 A_2}{L_2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix} \right) \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} \quad (5.87)$$

tal que

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \begin{bmatrix} \frac{E_1 A_1}{L_1} & -\frac{E_1 A_1}{L_1} & 0 \\ -\frac{E_1 A_1}{L_1} & \frac{E_1 A_1}{L_1} + \frac{E_2 A_2}{L_2} & -\frac{E_2 A_2}{L_2} \\ 0 & -\frac{E_2 A_2}{L_2} & \frac{E_2 A_2}{L_2} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} \quad (5.88)$$

e chamamos a matriz de rigidez do sistema completo de **matriz de rigidez global**. Voltando as informações do sistema, podemos aplicar as condições de contorno, obtendo

$$\begin{Bmatrix} ? \\ -P \\ Q \end{Bmatrix} = \begin{bmatrix} \frac{E_1 A_1}{L_1} & -\frac{E_1 A_1}{L_1} & 0 \\ -\frac{E_1 A_1}{L_1} & \frac{E_1 A_1}{L_1} + \frac{E_2 A_2}{L_2} & -\frac{E_2 A_2}{L_2} \\ 0 & -\frac{E_2 A_2}{L_2} & \frac{E_2 A_2}{L_2} \end{bmatrix} \begin{Bmatrix} 0 \\ U_2 \\ U_3 \end{Bmatrix} \quad (5.89)$$

e podemos notar que somente as equações 2 e 3 estão associadas as incógnitas, tal que o sistema pode ser reduzido a

$$\begin{Bmatrix} -P \\ Q \end{Bmatrix} = \begin{bmatrix} \frac{E_1 A_1}{L_1} + \frac{E_2 A_2}{L_2} & -\frac{E_2 A_2}{L_2} \\ -\frac{E_2 A_2}{L_2} & \frac{E_2 A_2}{L_2} \end{bmatrix} \begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix}. \quad (5.90)$$

Aqui é muito importante enfatizarmos que a matriz de rigidez global obtida antes de aplicarmos as condições de contorno de Dirichlet é **singular**, indicando que o sistema não possui solução (única). Fisicamente, isso está associado ao fato de a estrutura estar solta no espaço e a aplicação de qualquer carregamento irá implicar em movimentos de corpo rígido (sem rigidez). Assim, os deslocamentos podem ir para o infinito (como se fosse uma 'divisão' por zero). Para testar essa afirmação, podemos arbitrar valores para a matriz de rigidez 3×3 e calcular os seus autovalores. Um autovalor será nulo, indicando que a matriz é singular. Se fizermos o mesmo para a matriz 2×2 , obtida após a aplicação das condições de contorno, veremos que todos os autovalores são positivos. Isso indica que a matriz de rigidez é não singular e positivo-definida.

O sistema de equações lineares

$$\mathbf{KU} = \mathbf{F} \quad (5.91)$$

pode ser solucionado pela inversão da matriz de coeficientes

$$\mathbf{U} = \mathbf{K}^{-1} \mathbf{F} \quad (5.92)$$

resultando em

$$\begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix} = \begin{bmatrix} \frac{L_1}{E_1 A_1} & \frac{L_1}{E_1 A_1} \\ \frac{L_1}{E_1 A_1} & \frac{A_1 E_1 L_2 + A_2 E_2 L_1}{E_1 A_1 E_2 A_2} \end{bmatrix} \begin{Bmatrix} -P \\ Q \end{Bmatrix} \quad (5.93)$$

tal que

$$U_2 = \frac{L_1 Q - L_1 P}{A_1 E_1} \quad (5.94)$$

e

$$U_3 = \frac{(A_1 E_1 L_2 + A_2 E_2 L_1) Q - A_2 E_2 L_1 P}{A_1 A_2 E_1 E_2}. \quad (5.95)$$

Com isso, podemos descrever os deslocamentos axiais ao longo dos elementos como

$$u_x(x) = N_2(x) U_2 \quad x \in (0, L_1) \quad (5.96)$$

e

$$u_x(x) = N_1(x) U_2 + N_2(x) U_3 \quad x \in (0, L_2) \quad (5.97)$$

pois as funções de interpolação são definidas localmente em cada elemento.

Exercício 5.2 Assuma valores para o problema anterior e calcule os autovalores da matriz de rigidez antes e após a aplicação das condições de contorno de Dirichlet. ■

Exercício 5.3 Assuma valores para o problema anterior e obtenha valores numéricos para os deslocamentos nodais. Após, descreva matematicamente o deslocamento axial em cada trecho (elemento) e faça um gráfico do deslocamento axial. ■

Exercício 5.4 Seja uma barra de comprimento L , módulo de elasticidade e área constantes. A barra está presa em $x = 0$ e é submetida a uma força de corpo com expressão $b_x(x) = \alpha x^2$. Obtenha a solução analítica para esse problema e proponha uma discretização por elementos finitos, obtendo a solução. Qual a sua interpretação da solução discreta? Quantos elementos devemos utilizar para obter a solução analítica? ■

5.4.2 Automatização do procedimento de sobreposição de elementos

Na subseção anterior realizamos o procedimento de discretização de uma estrutura de barras, pela sobreposição de elementos finitos. O procedimento utilizado foi baseado no somatório de forças e foi realizado com detalhamento das etapas. Isso pode fazer com que o procedimento pareça trabalhoso, o que não é verdade.

Utilizando o mesmo problema definido na Fig. 5.4, podemos identificar que temos 3 nós, cada um com 1 grau de liberdade (deslocamento axial). Isso indica que temos um total de 3 graus de liberdade no problema completo, de tal forma que a nossa matriz de rigidez global será 3×3 . Também vimos que temos dois elementos, cada um com uma matriz 2×2 (isso não muda, pois vem da própria definição do elemento). O importante é identificar que cada elemento está associando graus de liberdade locais com graus de liberdade da estrutura (globais) e que termos de rigidez que compartilham um dado grau de liberdade devem ser somados (observe a sobreposição de forças na subseção anterior).

Para o primeiro elemento notamos que

- K_{11}^1 está relacionando o grau de liberdade global 1 com ele mesmo;
- K_{12}^1 está associado o grau de liberdade global 1 com o grau de liberdade global 2;
- K_{21}^1 está associado o grau de liberdade global 2 com o grau de liberdade global 1;
- K_{22}^1 está associado o grau de liberdade global 2 com ele mesmo;

e, para o segundo elemento

- K_{11}^2 está relacionando o grau de liberdade global 2 com ele mesmo;
- K_{12}^2 está associado o grau de liberdade global 2 com o grau de liberdade global 3;
- K_{21}^2 está associado o grau de liberdade global 3 com o grau de liberdade global 2;
- K_{22}^2 está associado o grau de liberdade global 3 com ele mesmo

Assim,

$$\mathbf{K} = \begin{bmatrix} K_{11}^1 & K_{12}^1 & 0 \\ K_{21}^1 & K_{22}^1 + K_{11}^2 & K_{12}^2 \\ 0 & K_{21}^2 & K_{22}^2 \end{bmatrix} \quad (5.98)$$

Como a barra 1 do exemplo anterior não tem contato com o nó 3 e a barra 2 não tem contato com o nó 1, observamos que os termos K_{13} e K_{31} da matriz de rigidez global são nulas. Isso é observado nas matrizes de rigidez globais de estruturas discretizadas pelo método dos elementos finitos e o número de posições nulas tende a aumentar quando aumentamos o número de graus de liberdade. Quando temos uma matriz com um grande número de posições nulas, chamamos essa matriz de **matriz esparsa**.

A generalização do procedimento do exemplo anterior (com duas barras) é apresentada no Alg. 1.

Algoritmo 1: Procedimento para a montagem da matriz de rigidez global com n_e elementos dispostos em linha.

```

1  $n_e$  MontaK()
2   Inicializar a matriz global
3    $\mathbf{K} \leftarrow \mathbf{0}_{n_e+1 \times n_e+1}$ 
4   Para cada elemento  $e$ 
5     for  $e \in \{1, \dots\}$  do
6       Montar a matriz local do elemento,  $\mathbf{K}^e$ , Eq. (5.68)
7       Sobrepor a matriz local nas posições globais
8        $\mathbf{K}(e, e) \leftarrow \mathbf{K}(e, e) + \mathbf{K}^e(1, 1)$ 
9        $\mathbf{K}(e, e+1) \leftarrow \mathbf{K}(e, e+1) + \mathbf{K}^e(1, 2)$ 
10       $\mathbf{K}(e+1, e) \leftarrow \mathbf{K}(e+1, e)$ 
11       $\mathbf{K}(e+1, e+1) \leftarrow \mathbf{K}(e+1, e+1) + \mathbf{K}^e(2, 2)$ 

```

■ **Exemplo 5.3** Seja uma barra de comprimento L discretizada por n_e elementos ($n_e + 1$ nós) com um elemento e sendo definido entre os nós e e $e + 1$. Com isso, cada elemento tem comprimento $l = L/n_e$. Assume-se que todos os elementos tem a mesma área e o mesmo módulo de elasticidade. Utilizando o Alg. 1, observamos que a matriz de rigidez global, neste caso, tem a forma

$$\mathbf{K} = \frac{EA}{l} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}_{n_e+1 \times n_e+1} \quad (5.99)$$

Tal matriz é **esparsa** (contém muitos zeros) e também é chamada de matriz **tridiagonal**, que é um caso particular de uma matriz do tipo **banda**. Como esperado, a matriz é simétrica. ■

Exercício 5.5 Considere uma barra discretizada por 5 elementos (ou seja, 6 nós). Realize o procedimento geral de sobreposição e obtenha a matriz de rigidez global. Observe as posições nulas. ■

Exercício 5.6 Considere uma barra discretizada por 3 elementos e submetida a uma força de corpo constante a no primeiro elemento, b no segundo elemento e c no terceiro elemento. Obtenha o vetor global de forças de corpo, detalhando o procedimento. ■

6. Elemento de barra: abordagem energética

A dedução do elemento finito de barra foi obtida a partir da forma fraca obtida por resíduos ponderados e por integração por partes. Também verificamos que a matriz de rigidez pode ser obtida diretamente por equilíbrio de forças (abordagem física, ou direta). O objetivo deste capítulo é apresentar uma outra abordagem/interpretação para a obtenção do sistema $\mathbf{KU} = \mathbf{F}$, baseada no conceito de energia.

6.1 Princípio da mínima energia potencial

Vamos começar com a definição do Princípio da mínima energia potencial, P MEP.

Definição 6.1.1 Princípio da mínima energia potencial (PMEP)

Em um **sistema conservativo**, de todos os campos de deslocamento **cinematicamente admissíveis**, aqueles correspondendo ao **equilíbrio** extremizam a **energia potencial total**. Se a condição de extremo for um mínimo, então o equilíbrio do sistema é estável.

O conceito de energia potencial total é muito importante, embora não seja muito intuitivo. Para entendermos esse conceito, vamos considerar uma mola linear com equação $ku = f$ em que f [N] é a força aplicada na extremidade da mola, k [N/m] é a rigidez da mola e u [m] o deslocamento no ponto de aplicação da força. A energia de deformação absorvida na mola é dada pela área abaixo do gráfico da curva $f \times u$, que é uma reta (curva (a) na Fig. 6.1)

$$\Delta U(u) = \frac{1}{2}fu = \frac{1}{2}ku^2 \quad [J]. \quad (6.1)$$

A energia potencial total, $\Pi(u)$, da mola é energia que a mola pode potencialmente entregar se a força for mantida constante e revertermos o deslocamento da mola. Neste caso, estaríamos aplicando um **trabalho negativo** ao sistema (pois o trabalho é o produto interno entre o vetor deslocamento e o vetor força e neste caso a força é oposta ao

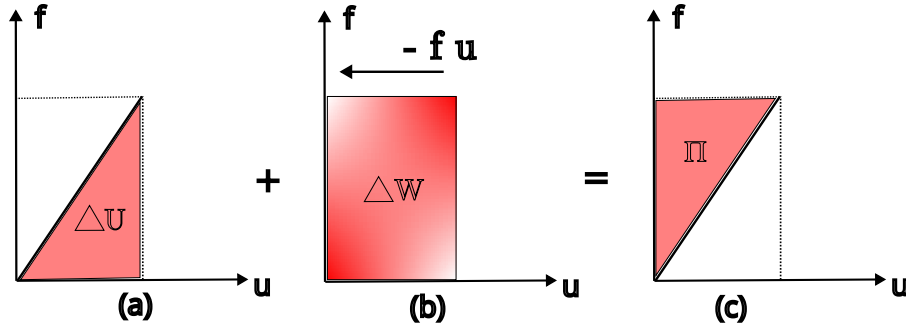


Figura 6.1: Energia interna na mola (a), trabalho realizado por uma força constante positiva em um deslocamento negativo (retorno da mola) (b) e Energia potencial total (c).

deslocamento imposto),

$$\Delta W = -fu \quad [J] \quad (6.2)$$

como ilustrado na Fig. 6.1(b). O balanço de energia, considerando a energia armazenada e o trabalho (negativo) realizado sobre a mola resulta em

$$\boxed{\Pi(u) = \frac{1}{2}ku^2 - fu} \quad [J] \quad (6.3)$$

que é o quanto de energia o sistema poderia **potencialmente** entregar. Outra maneira de interpretarmos essa medida é como sendo o negativo da energia complementar de deformação. O gráfico associado a essa medida é ilustrada na Fig. 6.1(c).

De acordo com o PMEP, o campo de deslocamento (no caso da mola, estamos falando do u da ponta) 'correto' (que equivale ao equilíbrio) é obtido pela extremização deste potencial. Do cálculo, sabemos que a condição necessária para o extremo de uma função é dada por

$$\frac{d\Pi(u)}{du} = 0 \quad (6.4)$$

e a condição suficiente para que o extremo seja um mínimo é dada por

$$\frac{d^2\Pi(u)}{du^2} > 0. \quad (6.5)$$

Assim, verificamos que

$$\frac{d\Pi(u)}{du} = \frac{d}{du} \left(\frac{1}{2}ku^2 - fu \right) = 0 \implies ku - f = 0 \implies ku = f \quad (6.6)$$

que é justamente a equação de equilíbrio da mola. Uma outra maneira de expressarmos a derivada é

$$\frac{d\Pi(u)}{du} = \frac{\partial \Pi(u)}{\partial u} \delta u \quad (6.7)$$

tal que

$$\frac{d\Pi(u)}{du} = \frac{\partial}{\partial u} \left(\frac{1}{2}ku^2 - fu \right) \delta u = \frac{1}{2}2ku\delta u - f\delta u = (ku - f)\delta u = 0 \quad (6.8)$$

e se quisermos que a relação seja válida para qualquer perturbação δu , precisamos zerar o termo entre parênteses, tal que $ku = f$.

Essa solução é estável pois

$$\frac{d^2\Pi(u)}{du^2} = k > 0 \quad (6.9)$$

lembrando que a rigidez da mola é sempre positiva.

6.2 Interpretando como um problema de perturbação da condição de equilíbrio - Princípio dos trabalhos virtuais (PTV)

Uma outra maneira de obtermos as equações anteriores é assumir que aplicamos a força na mola até obter o equilíbrio. Neste momento, a força interna da mola será

$$f_{int} = ku. \quad (6.10)$$

Se assumirmos que a mola é submetida a uma **perturbação admissível** ou **virtual**¹ δu no deslocamento, teremos uma variação de energia interna dada por

$$\delta U = ku\delta u. \quad (6.11)$$

Ao mesmo tempo, como estamos aplicando uma força externa f , teremos uma variação do trabalho externo

$$\delta W = f\delta u. \quad (6.12)$$

Como o deslocamento virtual δu , ou perturbação cinematicamente admissível, é arbitrário, não devemos ter nenhuma alteração no balanço de energia do sistema, tal que

$$\delta U = \delta W \implies ku\delta u - f\delta u = 0 \implies (ku - f)\delta u = 0 \quad (6.13)$$

que é exatamente o mesmo resultado que obtivemos na Eq. (6.8).

6.3 Utilizando o PMEP para obter o elemento de barra

A teoria de barras é uniaxial e contém somente tensões e deformações em x . Da teoria de Mecânica dos Sólidos I, sabemos que a **densidade de energia de deformação** em um ponto é dada por

$$e = \frac{1}{2}\sigma_{xx}\epsilon_{xx}, \quad [J/m^3] \quad (6.14)$$

e, usando a lei de Hooke

$$e(x) = \frac{1}{2}\epsilon_{xx}E(x)\epsilon_{xx}, \quad [J/m^3]. \quad (6.15)$$

Assim, a energia de deformação total no domínio da barra será

$$\Delta U = \int_0^L \int_A \frac{1}{2}\epsilon_{xx}E(x)\epsilon_{xx} dA dx = A \int_0^L \frac{1}{2}\epsilon_{xx}E(x)\epsilon_{xx} dx \quad (6.16)$$

¹Virtual significa potencialmente realizável

se assumirmos a área como constante. Substituindo a relação entre deformações e deslocamentos,

$$\varepsilon_{xx}(x) = \frac{du_x(x)}{dx} \quad (6.17)$$

tal que

$$\Delta U = A \int_0^L \frac{1}{2} \frac{du(x)}{dx} E(x) \frac{du(x)}{dx} dx. \quad (6.18)$$

Neste ponto, podemos assumir que o deslocamento $u(x)$ é descrito por

$$u_x(x) = \mathbf{N}(x)\mathbf{U} \quad (6.19)$$

tal que

$$\Delta U = A \int_0^L \frac{1}{2} \frac{d(\mathbf{N}\mathbf{U})}{dx} E(x) \frac{d(\mathbf{N}\mathbf{U})}{dx} dx \quad (6.20)$$

e já vimos que as derivadas das funções de interpolação podem ser escritas como uma matriz \mathbf{B} , tal que

$$\Delta U = A \int_0^L \frac{1}{2} \mathbf{U}^T \mathbf{B}^T E \mathbf{B} \mathbf{U} dx. \quad (6.21)$$

O trabalho instantâneo provocado por forças externas concentradas F_1 e F_2 pode ser escrito como

$$\Delta W = F_1 U_1 + F_2 U_2 = \mathbf{U}^T \mathbf{F}. \quad (6.22)$$

Assim, a energia potencial total da barra pode ser escrita em função dos deslocamentos discretos (nodais) \mathbf{U} como

$$\Pi(\mathbf{U}) = \mathbf{U}^T \left(A \int_0^L \frac{1}{2} \mathbf{B}^T E \mathbf{B} dx \right) \mathbf{U} - \mathbf{U}^T \mathbf{F}. \quad (6.23)$$

Assim, podemos obter o campo de deslocamentos associado ao equilíbrio da barra com

$$\frac{d\Pi(\mathbf{U})}{d\mathbf{U}} = (\delta\mathbf{U})^T \left(A \int_0^L \frac{1}{2} 2\mathbf{B}^T E \mathbf{B} dx \right) \mathbf{U} - (\delta\mathbf{U})^T \mathbf{F} = 0 \quad (6.24)$$

tal que

$$(\delta\mathbf{U})^T \left(A \int_0^L \mathbf{B}^T E \mathbf{B} dx \mathbf{U} - \mathbf{F} \right) = 0. \quad (6.25)$$

Se quisermos garantir a condição acima para qualquer valor de perturbação $\delta\mathbf{U}$, devemos igualar o termo entre parênteses ao vetor nulo, ou seja

$$\underbrace{A \int_0^L \mathbf{B}^T E \mathbf{B} dx}_{\mathbf{K}} \mathbf{U} = \mathbf{F} \quad (6.26)$$

que é a mesma equação obtida por resíduos ponderados.

Exercício 6.1 Refaça a dedução desta seção utilizando o PTV. ■

6.4 Interpretando a solução do MEF como um problema de otimização

Ao montarmos o sistema de equilíbrio global de um sistema com n graus de liberdade, podemos obter a energia potencial total

$$\Pi(\mathbf{U}) = \frac{1}{2} \mathbf{U}^T \mathbf{K} \mathbf{U} - \mathbf{U}^T \mathbf{F} \quad (6.27)$$

e sabemos que após a aplicação das condições de contorno de Dirichlet a matriz \mathbf{K} é positivo-definida. Assim, se interpretarmos $\Pi(\mathbf{U})$ como um campo escalar no espaço \mathbb{R}^n , temos que

$$\mathbf{U}^* = \operatorname{argmin} \Pi(\mathbf{U}) \quad (6.28)$$

é a solução para o problema de equilíbrio. Esse é um problema de otimização irrestrito, de uma função quadrática e convexa. Assim, a única solução (mínimo global) é obtida quando

$$\frac{d\Pi(\mathbf{U})}{d\mathbf{U}} = \mathbf{0} \implies \mathbf{K} \mathbf{U} = \mathbf{F}. \quad (6.29)$$

Neste ponto o leitor pode perceber que um mesmo assunto pode ser abordado/interpretado de várias formas diferentes.

7. Rotação de sistemas de referência

Um mesmo observador pode utilizar diferentes sistemas de referência para descrever uma grandeza vetorial. Considere o exemplo em que um observador descreve um vetor \mathbf{v} em um sistema de referência cartesiano $\mathbf{I} \times \mathbf{J}$, como ilustrado na Fig. 7.1.

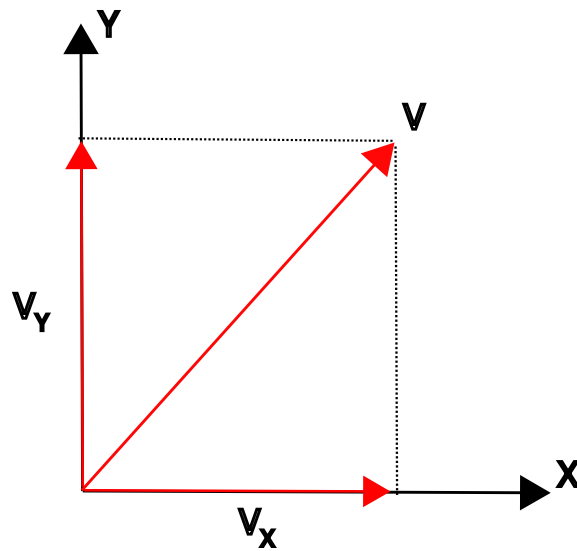


Figura 7.1: Projeção do vetor \mathbf{V} nos eixos globais X e Y .

A descrição que o observador realiza nada mais é do que

$$\mathbf{v} = v_X \mathbf{I} + v_Y \mathbf{J} \quad (7.1)$$

em que v_X é a projeção do vetor \mathbf{V} no vetor unitário \mathbf{I} e v_Y é a projeção do vetor \mathbf{V} no vetor unitário \mathbf{J} . Quando o sistema de referência é conhecido pelo leitor, podemos identificar o

vetor somente por suas componentes, na forma

$$\mathbf{V} = \begin{Bmatrix} v_X \\ v_Y \end{Bmatrix} = (v_X, v_Y). \quad (7.2)$$

Da álgebra linear (consultar o material inicial de fundamentos de matemática, no tópico sobre projeções 8.2.3), sabemos que essas projeções são descritas por

$$P_{\mathbf{I}}(\mathbf{V}) = \frac{\mathbf{V} \cdot \mathbf{I}}{\|\mathbf{V}\|} \mathbf{I} \quad (7.3)$$

com magnitude (valor da projeção)

$$v_X = \|P_{\mathbf{I}}(\mathbf{V})\| = \|\mathbf{V}\| \cos(\alpha) \quad (7.4)$$

em que α é o ângulo entre os vetores \mathbf{V} e \mathbf{I} . O cosseno deste ângulo é conhecido como **cosseno diretor** (Cossenos diretores são os cossenos dos ângulos formados entre um vetor e os eixos de coordenadas positivos). Da mesma forma,

$$P_{\mathbf{J}}(\mathbf{V}) = \frac{\mathbf{V} \cdot \mathbf{J}}{\|\mathbf{V}\|} \mathbf{J} \quad (7.5)$$

com magnitude

$$v_Y = \|P_{\mathbf{J}}(\mathbf{V})\| = \|\mathbf{V}\| \cos(\beta) \quad (7.6)$$

em que β é o ângulo entre os vetores \mathbf{V} e \mathbf{J} , definindo outro cosseno diretor.

Vamos supor que o vetor \mathbf{V} é descrito no sistema de referência global $\mathbf{I} \times \mathbf{J}$, por meio de suas projeções v_X e v_Y . Como poderíamos descrever esse mesmo vetor em um outro sistema de referência $\mathbf{i} \times \mathbf{j}$ (local), rotacionado em θ graus em relação ao sistema global, como ilustrado na Fig. 7.2.

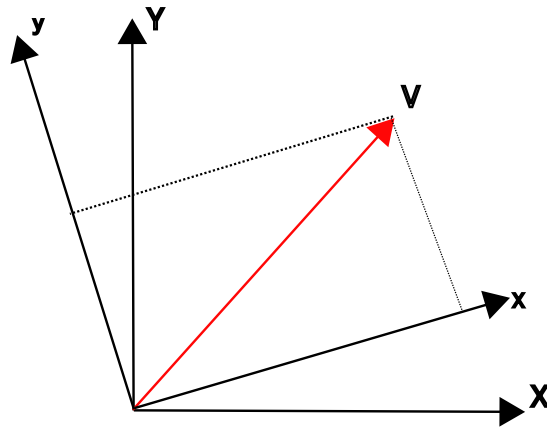


Figura 7.2: Sistema de referência global XY e sistema de referência local xy .

Para entendermos o procedimento, vamos projetar as componentes de \mathbf{V} no sistema local, v_x e v_y , nas direções globais.

Na direção X global teremos

$$v_X = v_x \cos(\theta) - v_y \sin(\theta) \quad (7.7)$$

e na direção Y global

$$v_Y = v_x \sin(\theta) - v_y \cos(\theta). \quad (7.8)$$

De forma análoga, podemos inverter as relação acima, obtendo

$$v_x = v_X \cos(\theta) + v_Y \sin(\theta) \quad (7.9)$$

e

$$v_y = -v_X \sin(\theta) + v_Y \cos(\theta). \quad (7.10)$$

O último conjunto de relações pode ser escrito em uma forma matricial

$$\begin{Bmatrix} v_x \\ v_y \end{Bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{Bmatrix} v_X \\ v_Y \end{Bmatrix} \quad (7.11)$$

ou

$$\mathbf{V}_{\text{local}} = \mathbf{R} \mathbf{V}_{\text{global}} \quad (7.12)$$

e a matriz \mathbf{R} é chamada de matriz de rotação (global/local). Essa matriz de rotação é muito importante em diversas aplicações do método dos elementos finitos, especialmente para elementos de barra, de viga e de placa.

■ **Exemplo 7.1** Rotação de sistema de referência (global para local)

Seja o vetor $\mathbf{V} = (4, 5)$ quando descrito no sistema global $X \times Y$. O mesmo vetor pode ser descrito em um sistema $x \times y$ rotacionado, por exemplo, a 90° . Neste caso,

$$\begin{Bmatrix} v_x \\ v_y \end{Bmatrix} = \begin{bmatrix} \cos(90) & \sin(90) \\ -\sin(90) & \cos(90) \end{bmatrix} \begin{Bmatrix} 4 \\ 5 \end{Bmatrix} \quad (7.13)$$

ou

$$\begin{Bmatrix} v_x \\ v_y \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{Bmatrix} 4 \\ 5 \end{Bmatrix} = \begin{Bmatrix} 5 \\ -4 \end{Bmatrix}. \quad (7.14)$$

■

É interessante notar que o conceito de cosseno diretor pode ser aplicado aqui também, onde os cossenos e senos da matriz \mathbf{R} são relativos aos cossenos diretores entre os dois sistemas de referência. De fato, a projeção de x sobre X é indicada por $\cos(\theta)$, que é muitas vezes indicado como m_{ox} , dependendo do autor. A projeção de x sobre Y é dada por $\sin(\theta)$, muitas vezes indicado por l_{ox} . Da mesma forma, a projeção de y sobre X é dada por $-\sin(\theta)$, indicada por m_{oy} e a projeção de y sobre Y é dada por $\cos(\theta)$, l_{oy} , tal que

$$\mathbf{R} = \begin{bmatrix} m_{ox} & l_{ox} \\ m_{oy} & l_{oy} \end{bmatrix}. \quad (7.15)$$

De fato, do conceito de projeção, podemos estabelecer que

$$m_{ox} = \|P_{\mathbf{I}}(\mathbf{i})\| = \mathbf{i} \cdot \mathbf{I} \quad (7.16)$$

$$l_{ox} = \|P_{\mathbf{J}}(\mathbf{i})\| = \mathbf{i} \cdot \mathbf{J} \quad (7.17)$$

$$m_{oy} = \|P_{\mathbf{I}}(\mathbf{j})\| = \mathbf{j} \cdot \mathbf{I} \quad (7.18)$$

$$l_{oy} = \|P_{\mathbf{J}}(\mathbf{j})\| = \mathbf{j} \cdot \mathbf{J}. \quad (7.19)$$

O mesmo raciocínio pode ser aplicado para um problema $3D$, com a dificuldade adicional de que precisaremos de mais de um ângulo para descrever a relação entre os sistemas de referência (ângulos de Euler), tal que

$$\mathbf{R} = \begin{bmatrix} m_{ox} & l_{ox} & n_{ox} \\ m_{oy} & l_{oy} & n_{oy} \\ m_{oz} & l_{oz} & n_{oz} \end{bmatrix}. \quad (7.20)$$

Exercício 7.1 Interprete os termos adicionais da matriz de rotação para o problema 3D.

■

A matriz de rotação \mathbf{R} mapeia as componentes globais do vetor \mathbf{V} para as componentes locais no sistema $x \times y$, rotacionado em relação ao sistema global,

$$\mathbf{V}_{xy} = \mathbf{R}\mathbf{V}_{XY}. \quad (7.21)$$

Se quisermos inverter essa relação, devemos multiplicar os dois lados da equação por \mathbf{R}^{-1} , tal que

$$\mathbf{R}^{-1}\mathbf{V}_{xy} = \mathbf{R}^{-1}\mathbf{R}\mathbf{V}_{XY} = \mathbf{I}\mathbf{V}_{XY} = \mathbf{V}_{XY}, \quad (7.22)$$

tal que

$$\mathbf{V}_{XY} = \mathbf{R}^{-1}\mathbf{V}_{xy}. \quad (7.23)$$

A matriz de rotação \mathbf{R} é uma **matriz ortogonal**, tal que sua inversa é igual a sua transposta

$$\mathbf{R}^{-1} = \mathbf{R}^T, \quad (7.24)$$

de tal forma que podemos reescrever a Eq. (7.23) como

$$\mathbf{V}_{XY} = \mathbf{R}^T\mathbf{V}_{xy} \quad (7.25)$$

ou

$$\begin{Bmatrix} v_X \\ v_Y \end{Bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{Bmatrix} v_x \\ v_y \end{Bmatrix}. \quad (7.26)$$

■ **Exemplo 7.2** Rotação de sistema de referência (local para global)

Seja o vetor $\mathbf{V} = (5, -4)$ quando descrito no sistema local $x \times y$ que é rotacionado 90° em relação ao sistema global. Neste caso,

$$\begin{Bmatrix} v_X \\ v_Y \end{Bmatrix} = \begin{bmatrix} \cos(90) & -\sin(90) \\ \sin(90) & \cos(90) \end{bmatrix} \begin{Bmatrix} 5 \\ -4 \end{Bmatrix} \quad (7.27)$$

ou

$$\begin{Bmatrix} v_X \\ v_Y \end{Bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} 5 \\ -4 \end{Bmatrix} = \begin{Bmatrix} 4 \\ 5 \end{Bmatrix}. \quad (7.28)$$

O leitor deve comparar esse procedimento com o exemplo anterior.

■

8. Elemento de barra em 2D - treliças

Uma treliça pode ser definida, para os propósitos deste texto, como sendo um arranjo de

- Membros prismáticos retos;
- Conectados por rótulas;
- Submetidos somente a forças concentradas nas conexões rotuladas

conforme ilustrado na Fig. 8.1.

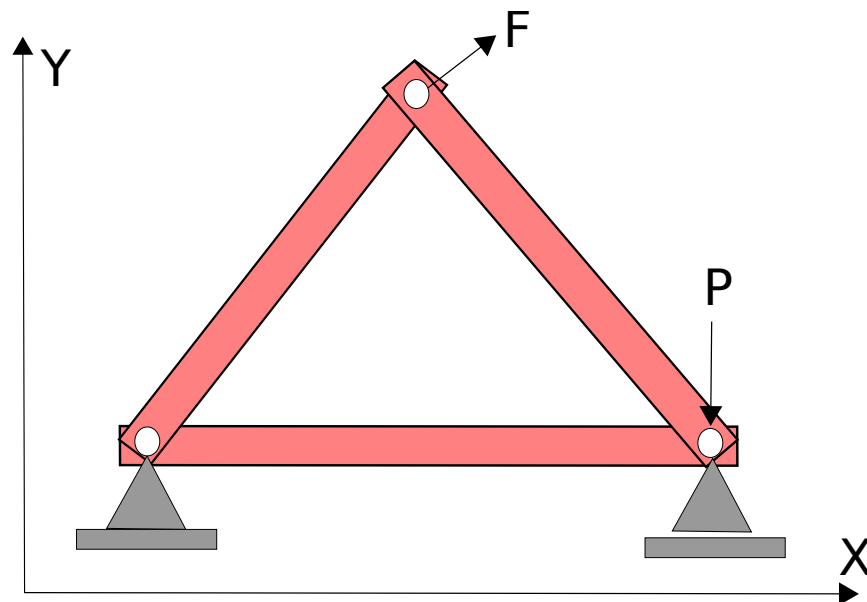


Figura 8.1: Exemplo de treliça 2D

Como consequência dessas hipóteses, observa-se que cada um dos membros da treliça (trecho reto entre duas conexões rotuladas) será submetido somente a esforços internos do tipo **Força normal** (sem cortantes ou momentos de qualquer tipo), ou seja, cada um dos

membros se comporta como uma **barra**.

Assim, seria interessante aproveitarmos toda a teoria desenvolvida no capítulo sobre elementos finitos de barras para aplicações em treliças.

8.1 Treliças bidimensionais

Embora uma treliça seja formada por barras, deve-se observar uma questão muito importante: a treliça é definida em um sistema global $X \times Y$ (Fig. 8.1), ou seja, cada ponto pode **transladar** nas direções X e Y , tal que cada ponto tem **2 graus de liberdade de translação**. No entanto, cada barra é definida em um **sistema local** de referência, indicado por x . Assim, antes de utilizar a teoria desenvolvida anteriormente, devemos nos preocupar com algumas questões importantes.

Utilizando os conceitos de discretização vistos anteriormente, podemos definir três nós nas posições das rótulas, cada um com dois graus de liberdade, como mostrado na Fig. 8.2.

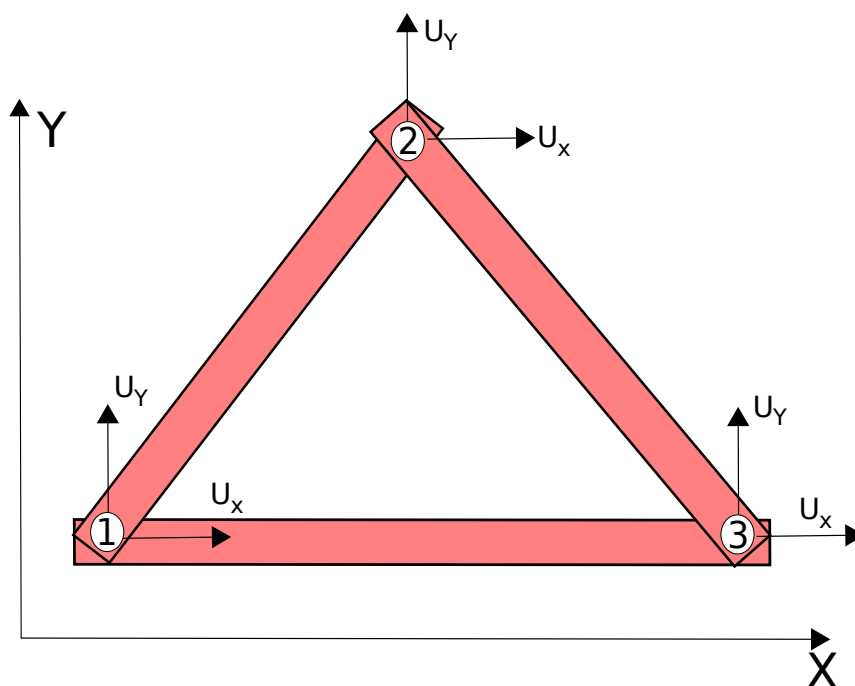


Figura 8.2: Graus de liberdade em uma treliça 2D.

Assim, já podemos verificar que o sistema tem um total de 3 (nós) vezes 2 (gl's por nó), totalizando 6 graus de liberdade (globais).

Se estudarmos somente a barra que conecta os nós 1 e 2, Figura 8.3, podemos verificar que

- Cada nó tem deslocamentos U_X e U_Y , descritos no sistema de referência global, associados a treliça;
- Os nós tem deslocamentos axiais U_1 e U_2 , que estão associados ao sistema de referência local $x \times y$, alinhado com a barra;
- A teoria de barra foi deduzida no sistema local $x \times y$, utilizando somente a variável x (não temos termos em y).

Assim, devemos relacionar os sistemas locais e globais para poder aproveitar a teoria que já desenvolvemos. A primeira coisa que podemos fazer é estender a formulação local

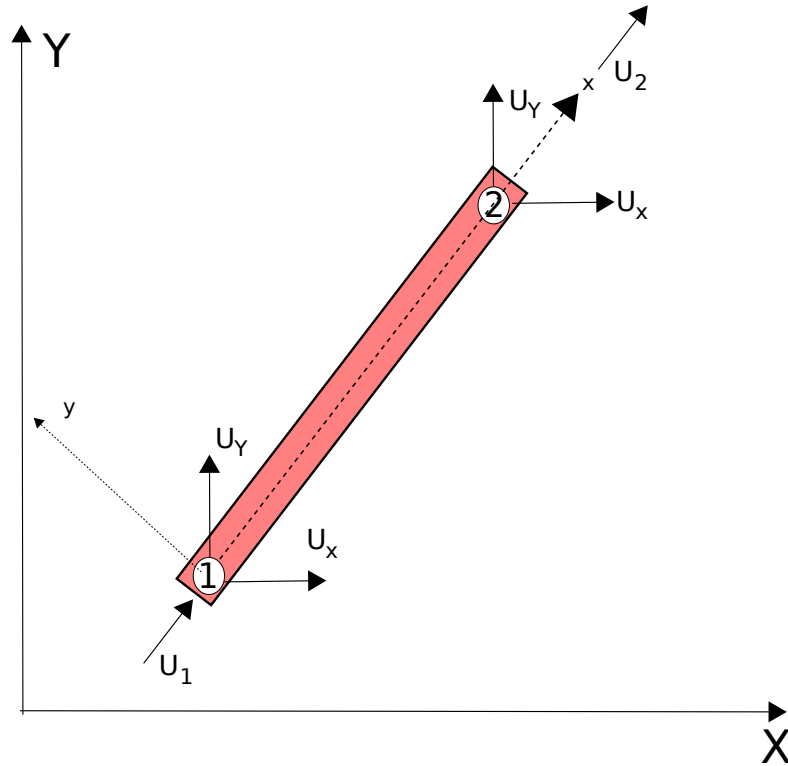


Figura 8.3: Detalhes da barra que conecta os nós 1 e 2.

para que também seja bidimensional, isto é, 'considere' x e y . Isso pode ser realizado se definirmos deslocamentos nodais neste sistema local, estendendo para o caso 2D, conforme a Fig. 8.4. Aqui é importante verificar que os deslocamentos U_1 e U_3 são os mesmos deslocamentos locais que definimos na barra e que os deslocamentos U_2 e U_4 'não existem' no modelo de barra (que é 1D).

Finalmente, se lembrarmos do significado da matriz de rigidez da barra, podemos escrever

$$\frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} \quad (8.1)$$

que, devido aos zeros que foram inseridos na matriz de rigidez, mantém exatamente o mesmo significado da matriz de rigidez obtida no capítulo do elemento de barra. Com essa modificação, temos um sistema **local** de dimensão 4×4 , pois temos 2 nós no elemento, cada um com 2 gls.

Com essa modificação, agora temos o sistema linear local de cada barra com 2 gls por nó, da mesma forma que o modelo 2D da treliça. No entanto, ainda não podemos sobrepor as barras como foi feito nos exemplos unidimensionais, pois estamos lidando com sistemas de referência diferentes.

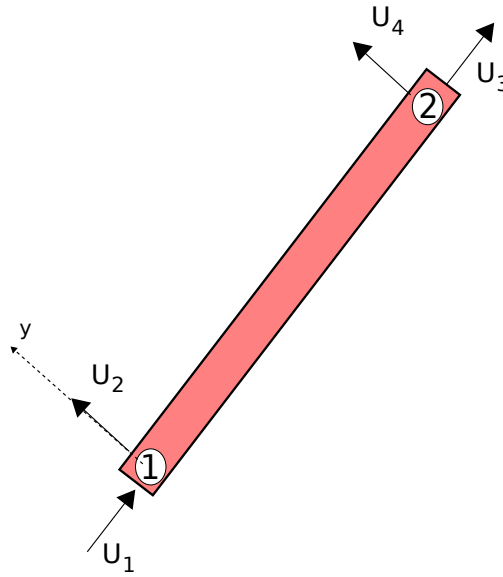


Figura 8.4: Deslocamentos no sistema local 2D.

8.1.1 Mapeamento local-global

Para entendermos o procedimento que será realizado, vamos utilizar a notação

$$\boxed{\mathbf{K}_l^e \mathbf{U}_l^e = \mathbf{F}_l^e} \quad (8.2)$$

para representar a Eq. (8.1), em que o super índice e se refere ao elemento e o índice l se refere ao sistema local do elemento. Como cada elemento tem um sistema local diferente, é interessante transformar (rotacionar) todos os sistemas locais para um sistema em comum, sendo que o sistema global de referência $X \times Y$ é a escolha mais adequada. Como vimos no capítulo anterior, um vetor \mathbf{V}_g no sistema global pode ser mapeado para um sistema local por meio da operação

$$\mathbf{V}_l = \mathbf{R} \mathbf{V}_g \quad (8.3)$$

sendo \mathbf{R} a matriz de rotação. Ainda, utilizando a Fig. 8.4, podemos verificar que em cada um dos nós temos um vetor de deslocamento com duas componentes. Assim, podemos escrever (observe bem o uso do negrito)

$$\mathbf{U}_l^e = \begin{Bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{Bmatrix} = \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} \quad (8.4)$$

ou seja, devemos aplicar a matriz de rotação a cada um dos vetores \mathbf{U}_1 e \mathbf{U}_2 , que pode ser realizada em uma só operação pelo uso da matriz

$$\boxed{\mathbf{T} = \begin{bmatrix} \mathbf{R}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{R}_{2 \times 2} \end{bmatrix}_{4 \times 4}} \quad (8.5)$$

Assim, podemos escrever

$$\boxed{\mathbf{U}_l^e = \mathbf{T}^e \mathbf{U}_g^e} \quad (8.6)$$

e

$$\boxed{\mathbf{F}_l^e = \mathbf{T}^e \mathbf{F}_g^e} \quad (8.7)$$

Por fim, vamos substituir essas duas relações na Eq. (8.2)

$$\underbrace{\mathbf{K}_l^e \mathbf{U}_l^e}_{\mathbf{T}^e \mathbf{U}_g^e} = \underbrace{\mathbf{F}_l^e}_{\mathbf{T}^e \mathbf{F}_g^e} \quad (8.8)$$

tal que

$$\mathbf{K}_l^e \mathbf{T}^e \mathbf{U}_g^e = \mathbf{T}^e \mathbf{F}_g^e \quad (8.9)$$

e, multiplicando por $(\mathbf{T}^e)^T$ (lembrando que a matriz é ortogonal)

$$(\mathbf{T}^e)^T \mathbf{K}_l^e \mathbf{T}^e \mathbf{U}_g^e = (\mathbf{T}^e)^T \mathbf{T}^e \mathbf{F}_g^e \quad (8.10)$$

ou

$$\underbrace{(\mathbf{T}^e)^T \mathbf{K}_l^e \mathbf{T}^e}_{\mathbf{K}_g^e} \mathbf{U}_g^e = \mathbf{F}_g^e \quad (8.11)$$

tal que a rotação da matriz de rigidez local para o sistema global é realizada pela operação

$$\boxed{\mathbf{K}_g^e = (\mathbf{T}^e)^T \mathbf{K}_l^e \mathbf{T}^e} \quad (8.12)$$

Salienta-se que as equações marcadas com as caixas são muito importantes e serão utilizadas extensivamente ao longo do texto.

8.1.2 Numeração dos graus de liberdade

É importante verificar que um elemento e , no caso de uma trelça, pode ter uma conectividade muito mais geral do que no caso de barras 1D. Começamos definindo que os gls da estrutura são numerados de forma sequencial, do nó 1 até o nó n . Assim, o nó 1 terá os gls 1 (direção X) e 2 (direção Y), o nó 2 terá os gls 3 (direção X) e 4 (direção Y) e assim por diante. Desta forma, os gls globais de um nó n_i podem ser definidos por

$$gl_X = 2(n_i - 1) + 1 \quad (8.13)$$

e

$$gl_Y = 2(n_i - 1) + 2 \quad (8.14)$$

de tal forma que os gls de um elemento e , com nós n_1 e n_2 , são definidos por

$$\mathbf{g}^e = \begin{Bmatrix} 2(n_1 - 1) + 1 \\ 2(n_1 - 1) + 2 \\ 2(n_2 - 1) + 1 \\ 2(n_2 - 1) + 2 \end{Bmatrix}. \quad (8.15)$$

9. Implementação computacional - treliças

O objetivo deste capítulo é discutir a estrutura de um programa (ou da análise) de Elementos Finitos. Vamos utilizar a teoria de barras aplicadas a problemas de treliças como motivação para os códigos. Como será visto mais adiante, boa parte da abstração apresentada aqui poderá ser reaproveitada para implementar análises de outras teorias estruturais. Para simplificar a apresentação, iremos considerar somente forças concentradas e condições de contorno de Dirichlet (apoios) homogêneas, isto é, nulas.

9.1 Etapas de uma análise por elementos finitos

De forma bem geral, uma análise de Elementos Finitos pode ser dividida em três partes: pré-processamento, solução e pós-processamento. O pré-processamento é a etapa em que o usuário registra informações como geometria, discretização (nós e elementos), propriedades dos materiais, apoios e forças aplicadas. Algumas informações são fornecidas diretamente pelo usuário e outras são calculadas a partir das informações básicas. Isso depende muito do tipo de modelo estrutural utilizado na análise, de tal forma que detalhes sobre essa etapa serão apresentados ao longo de todo o texto. Assim, para os propósitos deste capítulo, podemos definir a entrada de dados (para o caso de treliças 2D) separada de uma etapa de pré-processamento (cálculo dos comprimentos e dos ângulos dos elementos).

A etapa de solução utiliza as informações do pré-processamento para realizar a montagem da matriz de rigidez global e do vetor de forças para solucionar o sistema de equações $\mathbf{KU} = \mathbf{F}$.

Por fim, a etapa de pós-processamento realiza cálculos que dependem da solução do campo de deslocamentos, tais como forças internas, tensões, energias, etc. Nessa etapa são, geralmente, geradas as visualizações e relatórios de solução. Em programas modernos de Elementos Finitos, tanto a etapa de pré-processamento quanto a de pós-processamento são realizadas de forma gráfica.

É importante salientar que nem todos os programas de Elementos Finitos precisam ter as três etapas, sendo que a etapa que realmente define o método é a de solução. Existem diversos programas de realizam somente o pré-processamento (ou só o pós-processamento),

sendo que o importante é que cada uma das etapas consiga 'conversar' com as demais.

Como nosso objetivo é entender e elaborar um programa completo de análise por Elementos Finitos, vamos começar com uma implementação simples, mas funcional.

O Algoritmo 2 ilustra o programa que será desenvolvido neste capítulo.

Algoritmo 2: Estrutura geral do programa a ser desenvolvido neste capítulo

```

1 Analise()
2   Entrada de dados
3   Pré-processamento
4   Montagem da matriz de rigidez global
5   Montagem do vetor de forças global
6   Imposição das condições de contorno de Dirichlet
7   Solução do sistema global
8   Pós-processamento

```

No que segue, vamos estudar cada uma das etapas, apresentando os algoritmos.

9.2 Entrada de dados

Uma treliça é definida por barras (elementos) que são conectadas por rótulas (nós). Cada nó é definido por suas coordenadas em um sistema global de referência e cada elemento é definido por dois nós. Ainda, a cada elemento associamos um material (módulo de elasticidade) e uma propriedade geométrica da seção transversal (área), que são informações básicas sobre cada barra.

Essas informações podem ser fornecidas a um programa de computador com o uso de matrizes e vetores.

Começando com as informações sobre os nós, definimos n_n como a variável que contém o número de nós e \mathbf{XY} é uma matriz de dimensão $n_n \times 2$ que contém em cada nó (linha da matriz) a coordenada X (coluna 1 da matriz) e a coordenada Y (coluna 2 da matriz).

O número de elementos será definido pela variável n_e e as conectividades dos elementos (informações sobre quais nós estão associados a cada elemento) serão informadas na matriz \mathbf{IJ} , de dimensão $n_e \times 2$. Cada elemento (linha da matriz) é definido por um nó inicial (coluna 1 da matriz) e por um nó final (coluna 2 da matriz).

O módulo de elasticidade de cada elemento é informado em um vetor \mathbf{E} de dimensão n_e e a área de cada elemento é informada em um vetor \mathbf{A} , também de dimensão n_e .

Por fim, as n_f forças concentradas aplicadas nos nós da estrutura são informadas em uma matriz \mathbf{FC} de dimensão $n_f \times 3$, em que cada linha contém a informação sobre o nó em que a força está aplicada (coluna 1 da matriz), seu grau de liberdade (coluna 2 da matriz) e valor da força aplicada (coluna 3 da matriz).

A mesma lógica é empregada para informarmos os n_a apoios em uma matriz \mathbf{AP} , de dimensão $n_a \times 2$, em que cada linha da matriz contém o nó (primeira coluna) e grau de liberdade (segunda coluna) sendo restrito.

■ **Exemplo 9.1** Seja a treliça em Δ definida na Figura 9.1.

A treliça é definida por três nós, com coordenadas $(0,0)$ [m], $(1,1)$ [m] e $(2,0)$ [m], respectivamente. As três barras são definidas, na sequência, entre os nós 1 e 2, 2 e 3 e 1 e 3, respectivamente. Assumimos que as três barras tem o mesmo material, com 210 [GPa] e a mesma área de 0.01 [m²]. Existem duas forças concentradas aplicadas: uma força negativa de 100 [N] no sentido Y do nó 2 e uma força positiva de 200 [N] no sentido X do

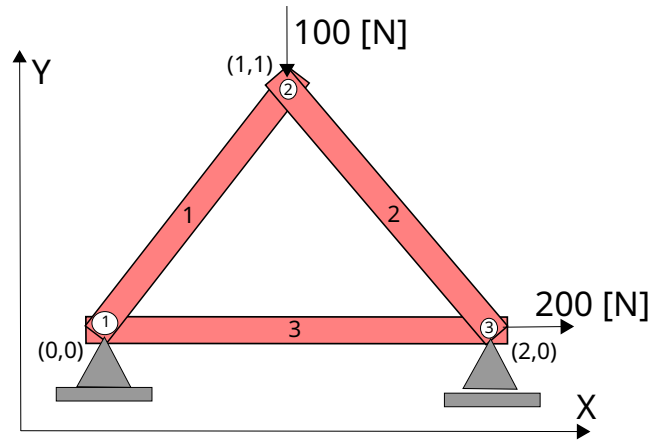


Figura 9.1: Exemplo de entrada de dados.

nó 3. Como queremos utilizar somente números para indicar os dados, utilizamos 1 para indicar X e 2 para indicar Y . Por fim, observamos que existem 3 graus de liberdade sendo restritos (apoios): direção X do nó 1, direção Y do nó 1 e direção Y do nó 3.

Dessa forma, os dados para essa treliça são:

- $n_n = 3$;
- $\mathbf{XY} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 0 \end{bmatrix}$
- $n_e = 3$;
- $\mathbf{IJ} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 1 & 3 \end{bmatrix}$;
- $\mathbf{E} = 210 \times 10^9 \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}$;
- $\mathbf{A} = 0.01 \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}$;
- $n_f = 2$
- $\mathbf{FC} = \begin{bmatrix} 2 & 2 & -100 \\ 3 & 1 & 200 \end{bmatrix}$;
- $n_a = 3$;
- $\mathbf{AP} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 3 & 2 \end{bmatrix}$;

■

9.3 Pré-processamento

Com os dados definidos na seção anterior (entrada de dados), podemos calcular informações que seriam maçantes para o usuário, mas são simples para um programa de computador. Em especial para a análise de uma treliça, temos o comprimento \mathbf{L} de cada elemento e também o ângulo θ do elemento com o sistema global.

O comprimento pode ser calculado diretamente pela aplicação do teorema de Pitágoras, pois sabemos os nós que definem cada elemento (**IJ**) e as coordenadas de cada um dos nós (**XY**). O procedimento é ilustrado no Alg. 3.

Algoritmo 3: Montagem do vetor **L**, contendo os comprimentos dos elementos

```

1 MontarL( $n_n, \mathbf{XY}, n_e, \mathbf{IJ}$ )
2   Aloca o vetor de saída
3    $\mathbf{L} \leftarrow \mathbf{0}_{n_e}$ 
4   Para cada elemento
5     for  $e \in \{1, \dots, n_e\}$  do
6       Nós do elemento
7        $n_1 \leftarrow \mathbf{IJ}(e, 1)$ 
8        $n_2 \leftarrow \mathbf{IJ}(e, 2)$ 
9       Coordenadas do nó 1
10       $x_1 \leftarrow \mathbf{XY}(n_1, 1)$ 
11       $y_1 \leftarrow \mathbf{XY}(n_1, 2)$ 
12      Coordenadas do nó 2
13       $x_2 \leftarrow \mathbf{XY}(n_2, 1)$ 
14       $y_2 \leftarrow \mathbf{XY}(n_2, 2)$ 
15      Comprimento
16       $\mathbf{L}(e) \leftarrow \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ 
17   Retorna o vetor L
18   return L

```

O procedimento para a obtenção dos ângulos θ de cada elemento também depende dos mesmos dados. Um cuidado que devemos ter ao calcular esse ângulo é que a definição de cosseno diretor implica no ângulo entre um vetor e os vetores positivos do sistema de referência global, isto é, na faixa $\theta \in [-\pi, \pi]$. Isso faz com que o uso ingênuo da função arco tangente (atan) possa causar alguns problemas de sinal, pois o uso direto de

$$\theta = \text{atan}\left(\frac{\Delta y}{\Delta x}\right) \quad (9.1)$$

retorna ângulos na faixa $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

Por isso, muitas linguagens de computador oferecem a função atan2 (ou arco tangente de dois argumentos). Procure pela ajuda da sua linguagem de programação quando for implementar o seu programa. O Algoritmo é ilustrado no Alg. 4.

Salienta-se que o ideal do ponto de vista de eficiência é implementar os Alg. 3 e 4 em uma só sub rotina, pois boa parte do procedimento é idêntica.

9.3.1 Montagem da matriz de rigidez global

A matriz de rigidez global do modelo de treliça 2D tem dimensão

$$n_{gl} = 2n \quad (9.2)$$

onde n é o número de nós. Sua montagem segue uma lógica muito parecida com o procedimento visto no capítulo do elemento de barra, com algumas modificações importantes.

O Alg. 5 ilustra de forma esquemática o procedimento de montagem da matriz de rigidez global. Cada uma das sub rotinas que são utilizadas no Alg. 5 são detalhadas nos Algoritmos 6, 7 e 8.

Algoritmo 4: Montagem do vetor $\boldsymbol{\theta}$, contendo os ângulos dos elementos em relação ao eixo X global

```

1 Monta $\Theta(n_n, \mathbf{XY}, n_e, \mathbf{IJ})$ 
2   Aloca o vetor de saída
3    $\Theta \leftarrow \mathbf{0}_{n_e}$ 
4   Para cada elemento
5     for  $e \in \{1, \dots, n_e\}$  do
6       Nós do elemento
7        $n_1 \leftarrow \mathbf{IJ}(e, 1)$ 
8        $n_2 \leftarrow \mathbf{IJ}(e, 2)$ 
9       Coordenadas do nó 1
10       $x_1 \leftarrow \mathbf{XY}(n_1, 1)$ 
11       $y_1 \leftarrow \mathbf{XY}(n_1, 2)$ 
12      Coordenadas do nó 2
13       $x_2 \leftarrow \mathbf{XY}(n_2, 1)$ 
14       $y_2 \leftarrow \mathbf{XY}(n_2, 2)$ 
15      Ângulo
16       $\theta(e) \leftarrow \text{atan2}(y_2 - y_1, x_2 - x_1)$ 
17   Retorna o vetor  $\Theta$ 
18   return  $\Theta$ 

```

9.3.2 Montagem do vetor global de forças concentradas

A montagem do vetor global de forças concentradas é bastante simples, pois as informações da matriz \mathbf{FC} já são informadas no sistema global de referência. Assim, só precisamos acessar as informações de \mathbf{FC} , isto é, cada linha da matriz, e adicionar a magnitude da força no grau de liberdade global. O Alg. 9 ilustra o procedimento.

9.3.3 Aplicação das condições de contorno de Dirichlet (apoios)

A aplicação das condições de contorno essenciais ou de Dirichlet é uma das etapas fundamentais para a solução do problema, tanto do ponto de vista físico quanto do ponto de vista matemático.

Do ponto de vista físico, a não imposição dos apoios implica em uma estrutura que apresenta modos de corpo rígido, de tal modo que os deslocamentos são indeterminados. Do ponto de vista matemático, a não imposição das condições de contorno faz com que a matriz \mathbf{K} seja singular.

Existem diversas maneiras de considerarmos essa informação, sendo que a mais simples é eliminar as linhas e colunas do sistema de equação que estejam associadas aos graus de liberdade homogêneos restritos. Isso pode ser realizado de duas formas.

Na primeira abordagem, alocamos uma nova matriz \mathbf{K} com dimensão $(2n_n - n_a) \times (2n_n - n_a)$ e copiamos somente as linhas e colunas não restritas da matriz \mathbf{K} . Fazemos o mesmo com o vetor de forças e solucionamos o sistema obtendo somente os $2n_n - n_a$ deslocamentos livres. Essa abordagem tem como ponto favorável o fato de solucionarmos um sistema com dimensão menor (o custo é proporcional a dimensão ao cubo, no pior caso). O ponto desfavorável é que temos que alocar novas matrizes e vetores e movimentar informação entre as matrizes originais e as novas. Isso pode acarretar em um custo computacional elevado.

A segunda abordagem é mais simples, pois alteramos o sistema $\mathbf{KU} = \mathbf{F}$ diretamente.

Algoritmo 5: Montagem da matriz de rigidez global em uma treliça

```

1 RigidezGlobal( $n_n, n_e, \mathbf{E}, \mathbf{A}, \mathbf{L}, \boldsymbol{\theta}, \mathbf{IJ}$ )
2   Inicializa a matriz global
3    $\mathbf{K} \leftarrow \mathbf{0}_{2n_n \times 2n_n}$ 
4   Laço nos elementos
5   for  $e \in \{1, \dots, n_e\}$  do
6     Recupera as propriedades do elemento
7      $E \leftarrow \mathbf{E}(e)$ 
8      $A \leftarrow \mathbf{A}(e)$ 
9      $L \leftarrow \mathbf{L}(e)$ 
10     $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}(e)$ 
11    Monta a matriz de rigidez local, Eq.(8.2)
12     $\mathbf{K}_l^e \leftarrow \text{RigidezBarra}(E, A, L)$ 
13    Monta a matriz de rotação do elemento, Eq.(8.5)
14     $\mathbf{T}^e \leftarrow \text{MatrizTBarra}(\boldsymbol{\theta})$ 
15    Transforma a rigidez do elemento, Eq. (8.12)
16     $\mathbf{K}_g^e \leftarrow (\mathbf{T}^e)^T \mathbf{K}_l^e \mathbf{T}^e$ 
17    Recupera os nós do elemento
18     $n_1 \leftarrow \mathbf{IJ}(e, 1)$ 
19     $n_2 \leftarrow \mathbf{IJ}(e, 2)$ 
20    Determina os gls do elemento, Eq. (8.15)
21     $\mathbf{g}^e \leftarrow \text{glsBarra}(n_1, n_2)$ 
22    Sobreposição a matriz do elemento na rigidez global
23    for  $i \in \{1, \dots, 4\}$  do
24      for  $j \in \{1, \dots, 4\}$  do
25         $\mathbf{K}(\mathbf{g}^e(i), \mathbf{g}^e(j)) \leftarrow \mathbf{K}(\mathbf{g}^e(i), \mathbf{g}^e(j)) + \mathbf{K}_g^e(i, j)$ 
26  Retorna a matriz global
27  return  $\mathbf{K}$ 

```

Algoritmo 6: Montagem da matriz de rigidez local do elemento de barra com E , A e L .

```

1 Montagem RigidezBarra( $E, A, L$ )
2   Define a matriz
3    $\mathbf{K}_l \leftarrow \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ 
4   Retorna a matriz
5   return  $\mathbf{K}_l$ 

```

Algoritmo 7: Montagem da matriz de rotação do elemento de barra com ângulo θ

```

1 MontagemMatrizTBarra( $\theta$ )
2   Calcula o seno e o cosseno
3    $s \leftarrow \sin(\theta)$ 
4    $c \leftarrow \cos(\theta)$ 
5   Define a matriz
6    $\mathbf{T} \leftarrow \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}$ 
7   Retorna a matriz
8   return  $\mathbf{T}$ 

```

Algoritmo 8: Determina os gls do elemento com nós n_1 e n_2

```

1 glsBarra( $n_1, n_2$ )
2   Define o vetor
3    $\mathbf{g} \leftarrow \begin{Bmatrix} 2(n_1 - 1) + 1 \\ 2(n_1 - 1) + 2 \\ 2(n_2 - 1) + 1 \\ 2(n_2 - 1) + 2 \end{Bmatrix}$ 
4   Retorna o vetor
5   return  $\mathbf{g}$ 

```

Algoritmo 9: Montagem do vetor global de forças concentradas

```

1 ForcaCGlobal( $n_n, n_f, FC$ )
2   Inicializa vetor global
3    $\mathbf{F} \leftarrow \mathbf{0}_{2n_n}$ 
4   Laço nas linhas de  $\mathbf{FC}$ 
5   for  $i \in \{1, \dots, n_f\}$  do
6     Nó
7      $no \leftarrow \mathbf{FC}(i, 1)$ 
8     Grau de liberdade (local)
9      $gll \leftarrow \mathbf{FC}(i, 2)$ 
10    Valor da força
11     $valor \leftarrow \mathbf{FC}(i, 3)$ 
12    Grau de liberdade global
13     $glg \leftarrow 2(no - 1) + gll$ 
14    Adiciona ao vetor global
15     $\mathbf{F}(glg) \leftarrow \mathbf{F}(glg) + valor$ 
16  Retorna o vetor de forças globais
17  return  $\mathbf{F}$ 

```

Considere o sistema com 3 equações

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} \quad (9.3)$$

e queremos impor a condição de contorno $U_2 = 0$ (eliminar a segunda equação). Para isso, zeramos a segunda linha e a segunda coluna da matriz, colocando 1 na diagonal. Além disso, impomos $F_2 = 0$

$$\begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & 1 & 0 \\ K_{31} & 0 & K_{33} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ 0 \\ F_3 \end{Bmatrix} \quad (9.4)$$

que equivale às Equações

$$\begin{cases} K_{11}U_1 + K_{13}U_3 = F_1 \\ K_{31}U_1 + K_{33}U_3 = F_3 \end{cases} \quad (9.5)$$

com $U_2 = 0$. Esse procedimento é ilustrado no Alg. 10.

Algoritmo 10: Aplicação das condições de contorno essenciais homogêneas.

```

1  AplicaCCH( $n_n, n_a, \mathbf{AP}, \mathbf{K}, \mathbf{F}$ )
2  Laço nas linhas de  $\mathbf{AP}$ 
3  for  $l \in \{1, \dots, n_a\}$  do
4      Nó
5       $no \leftarrow \mathbf{AP}(l, 1)$ 
6      Grau de liberdade (local)
7       $gll \leftarrow \mathbf{AP}(l, 2)$ 
8      Grau de liberdade global
9       $glg \leftarrow 2(no - 1) + gll$ 
10     Zera linhas e colunas da matriz
11     for  $i \in \{1, \dots, 2n_n\}$  do
12          $\mathbf{K}(i, glg) \leftarrow 0$ 
13          $\mathbf{K}(glg, i) \leftarrow 0$ 
14     Coloca 1 na diagonal
15      $\mathbf{K}(glg, glg) \leftarrow 1$ 
16     Zera a posição de  $\mathbf{F}$ 
17      $\mathbf{F}(glg) \leftarrow 0$ 
18     Retorna  $\mathbf{K}$  e  $\mathbf{F}$  modificados
19     return  $\mathbf{K}, \mathbf{F}$ 
```

A aplicação de condições de contorno essenciais não homogêneas será discutida mais adiante neste texto.

9.3.4 Pós-processamento: cálculo dos esforços internos

Treliças são formadas por barras, que são submetidas somente a esforços internos normais (tração ou compressão). Esses esforços podem ser obtidos com

$$\mathbf{F}_l^e = \mathbf{K}_l^e \mathbf{U}_l^e \quad (9.6)$$

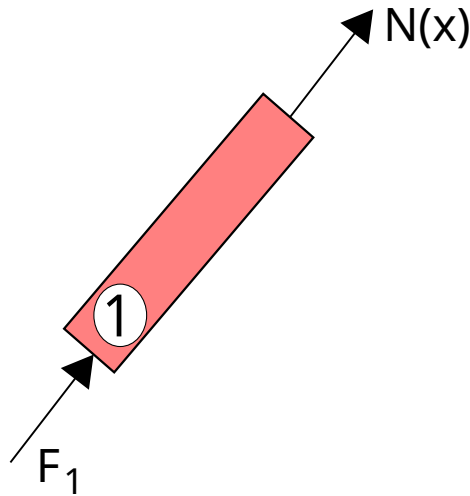


Figura 9.2: Balanço de forças axiais em uma barra.

ou seja, precisamos da matriz de rigidez do elemento e também do vetor de deslocamento do elemento. Essa é uma etapa de pós-processamento, pois para obtermos U_l^e precisamos primeiro do cálculo do vetor de deslocamentos globais U , que foi obtido na etapa de solução $KU = F$. No entanto, precisamos cuidar com a diferença de sistema de referência utilizado para descrever U (global) e U_l^e (local).

Se observarmos os graus de liberdade locais da Fig. 8.4, verificamos que as forças F_2 e F_4 serão sempre nulas (pois localmente não temos rigidez transversal em barras). Realizando um corte hipotético ao longo do comprimento, conforme ilustrado na Fig. 9.2, observamos que um somatório local de forças resulta em $N^e(x) = -F_1$.

O procedimento de cálculo é ilustrado no Alg. 11.

Exercício 9.1 Implemente os procedimentos discutidos neste capítulo em sua linguagem de programação de preferência e obtenha os deslocamentos nodais da treliça em Δ e os esforços internos. Valide a implementação de cada uma das sub rotinas implementadas (por exemplo, avaliando os comprimentos e ângulos de cada barra). ■

Algoritmo 11: Cálculo dos esforços internos dos elementos da treliça.

```

1 EsforçosInternos( $n_e, \mathbf{E}, \mathbf{A}, \mathbf{L}, \boldsymbol{\theta}, \mathbf{IJ}, \mathbf{U}$ )
2   Inicializa o vetor com os esforços normais de cada elemento
3    $\mathbf{N} \leftarrow \mathbf{0}_{n_e}$ 
4   Laço nos elementos
5   for  $e \in \{1, \dots, n_e\}$  do
6     Recupera as propriedades do elemento
7      $E \leftarrow \mathbf{E}(e)$ 
8      $A \leftarrow \mathbf{A}(e)$ 
9      $L \leftarrow \mathbf{L}(e)$ 
10     $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}(e)$ 
11    Monta a matriz de rigidez local, Eq.(8.2)
12     $\mathbf{K}_l^e \leftarrow \text{RigidezBarra}(E, A, L)$ 
13    Monta a matriz de rotação do elemento, Eq.(8.5)
14     $\mathbf{T}^e \leftarrow \text{MatrizTBarra}(\boldsymbol{\theta})$ 
15    Recupera os nós do elemento
16     $n_1 \leftarrow \mathbf{IJ}(e, 1)$ 
17     $n_2 \leftarrow \mathbf{IJ}(e, 2)$ 
18    Determina os gls do elemento, Eq. (8.15)
19     $\mathbf{g}^e \leftarrow \text{glsBarra}(n_1, n_2)$ 
20    Recupera os deslocamentos do elemento (ainda global)
21     $\mathbf{U}_g^e \leftarrow \mathbf{U}(\mathbf{g}^e)$ 
22    Transforma o deslocamento do elemento para o sistema local
23     $\mathbf{U}_l^e \leftarrow (\mathbf{T}^e) \mathbf{U}_g^e$ 
24    Calcula o vetor de forças internas do elemento
25     $\mathbf{F}_l^e \leftarrow \mathbf{K}_l^e \mathbf{U}_l^e$ 
26    Armazena o esforço interno do elemento no vetor  $\mathbf{N}$ 
27     $\mathbf{N}(e) \leftarrow -\mathbf{F}_l^e(1)$ 
28   Retorna o vetor de esforços internos
29   return  $\mathbf{N}$ 

```

10. Elemento finito de viga longa

A Equação Diferencial associada a teoria de viga longa foi apresentada no capítulo sobre a relaxação dos requisitos de continuidade, Eq. (4.33).

$$\frac{d^2}{dx^2} \left(E(x)I(x) \frac{d^2 v(x)}{dx^2} \right) = q(x) \quad x \in (0, L) \quad (10.1)$$

correspondendo a uma EDO linear de quarta ordem. A variável dependente é o deslocamento transversal $v(x)$ e a variável independente é a coordenada x (seção transversal). Essa equação diferencial foi utilizada em boa parte do texto anterior que discutiu a relaxação dos requisitos de continuidade, Capítulo 4.

Por simplicidade, vamos assumir que o módulo de elasticidade $E(x)$ e o momento de inércia $I(x)$ são constantes em x . A solução da Eq. (10.1) é a função $v(x)$ que satisfaz a EDO exatamente em todos os pontos, além de satisfazer as condições de contorno (solução forte). Após a consideração do método dos resíduos ponderados e da integração por partes, obtivemos a Eq. (4.44)

$$\int_0^L \frac{d^2 \tilde{v}(x)}{dx^2} EI \frac{d^2 w(x)}{dx^2} dx = \int_0^L w(x) q(x) dx + w(x) V(x) \Big|_0^L + \frac{dw(x)}{dx} M(x) \Big|_0^L \quad (10.2)$$

Da análise dos termos no contorno, obtidos após as duas integrações por partes, podemos verificar que temos duas condições de contorno de Dirichlet (essenciais): $v(x)$ e $\theta(x) = \frac{dv(x)}{dx}$ e duas condições de contorno de Neumann (naturais): $V(x)$ e $M(x)$. O termo 'fonte' é o carregamento distribuído $q(x)$.

Vamos seguir o mesmo procedimento utilizado no capítulo em que discutimos o método dos Elementos Finitos para barra.

10.1 Elemento Finito de Viga Longa

Começamos definindo dois nós nas extremidades da viga e descrevendo as condições de contorno essenciais nessas posições, de acordo com a Fig. 10.1. É importante notar

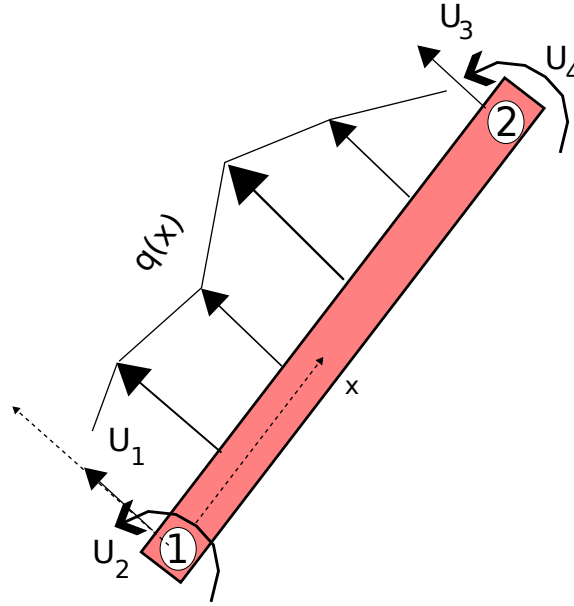


Figura 10.1: Viga descrita por um sistema de coordenadas local x .

que estamos utilizando o sistema de referência local x (centroidal). O carregamento distribuído $q(x)$ é definido na direção y local do elemento e, a princípio, pode assumir qualquer forma. Observa-se, também, que as rotações estão sendo indicadas como U_2 e U_4 .

Assim, 'sabemos' quatro informações sobre o campo de deslocamentos/rotações da viga e podemos utilizar essas informações para propor uma expressão para o deslocamento transversal. Como temos 4 informações, podemos propor um polinômio cúbico

$$v(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \quad (10.3)$$

tal que a rotação (primeira derivada de $v(x)$) é descrita por

$$\theta(x) = a_1 + 2a_2x + 3a_3x^2. \quad (10.4)$$

Assim, podemos igualar essas expressões aos valores 'conhecidos' nas posições nodais

$$v(0) = U_1 = a_0 \quad (10.5)$$

$$\theta(0) = U_2 = a_1 \quad (10.6)$$

$$v(L) = U_3 = a_0 + a_1L + a_2L^2 + a_3L^3 \quad (10.7)$$

$$\theta(L) = U_4 = a_1 + 2a_2L + 3a_3L^2 \quad (10.8)$$

resultando em

$$a_0 = U_1 \quad (10.9)$$

$$a_1 = U_2 \quad (10.10)$$

$$a_2 = -\frac{U_4L - 3U_3 + 2U_2L + 3U_1}{L^2} \quad (10.11)$$

$$a_3 = \frac{U_4L - 2U_3 + U_2L + 2U_1}{L^3} \quad (10.12)$$

tal que o deslocamento transversal pode ser interpolado ao longo do elemento com

$$v(x) = U_1 + U_2x - \left(\frac{U_4L - 3U_3 + 2U_2L + 3U_1}{L^2} \right) x^2 + \left(\frac{U_4L - 2U_3 + U_2L + 2U_1}{L^3} \right) x^3. \quad (10.13)$$

Da mesma forma que fizemos na dedução do elemento finito de barra, podemos reescrever essa interpolação como

$$v(x) = N_1(x)U_1 + N_2(x)U_2 + N_3(x)U_3 + N_4(x)U_4 \quad (10.14)$$

em que as funções $N_i(x)$ são as funções de interpolação do elemento, ou funções de forma. Após a manipulação dos termos da Eq. (10.13), colocando os deslocamentos generalizados U_i em evidência, obtemos

$$N_1(x) = \frac{2x^3}{L^3} - \frac{3x^2}{L^2} + 1 \quad (10.15)$$

$$N_2(x) = \frac{x^3}{L^2} - \frac{2x^2}{L} + x \quad (10.16)$$

$$N_3(x) = \frac{3x^2}{L^2} - \frac{2x^3}{L^3} \quad (10.17)$$

$$N_4(x) = \frac{x^3}{L^2} - \frac{x^2}{L} \quad (10.18)$$

que são **polinômios cúbicos de Hermite**, ilustrados na Fig. 10.2.

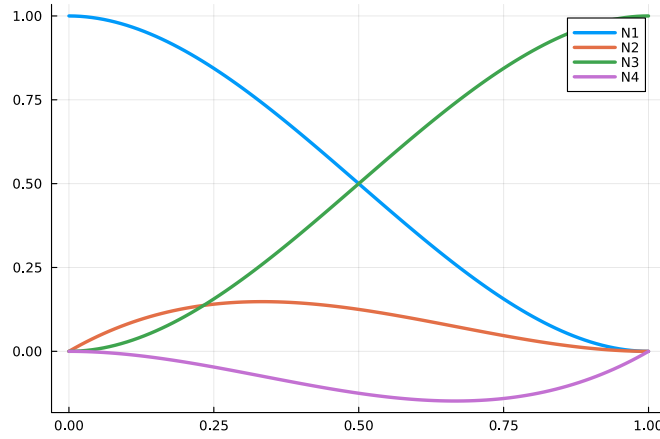


Figura 10.2: Polinômios de Hermite para $L = 1$ [m].

Lembrando da dedução do elemento finito de barra, lembramos que uma função $N_i(x)$ deve assumir valor unitário no seu nó/gl e zero nos demais nós/gl. De fato, podemos verificar que

$$N_1(0) = 1 \wedge N_1(L) = 0 \quad (10.19)$$

$$N_3(0) = 0 \wedge N_3(L) = 1 \quad (10.20)$$

pois essas funções de interpolação são relacionadas diretamente aos graus de liberdade de translação vertical (Isso pode ser verificado diretamente nos gráficos da Fig. 10.2). Talvez seja menos intuitivo que devamos verificar as propriedades de $N_2(x)$ e de $N_4(x)$ utilizando

$$\theta(x) = \frac{dN_1(x)}{dx}U_1 + \frac{dN_2(x)}{dx}U_2 + \frac{dN_3(x)}{dx}U_3 + \frac{dN_4(x)}{dx}U_4 \quad (10.21)$$

e, de fato,

$$\frac{dN_2}{dx}(0) = 1 \wedge \frac{dN_2}{dx}(L) = 0 \quad (10.22)$$

$$\frac{dN_4}{dx}(0) = 0 \wedge \frac{dN_4}{dx}(L) = 1 \quad (10.23)$$

ou seja, as **inclinações** de $N_2(0)$ e de $N_4(L)$ são unitárias.

Para utilizarmos a forma compacta (matricial) da dedução do elemento finito, vamos definir a matriz 1×4

$$\mathbf{N}(x) = [N_1(x) \quad N_2(x) \quad N_3(x) \quad N_4(x)] \quad (10.24)$$

tal que

$$v(x) = \mathbf{N}(x)\mathbf{U} \quad (10.25)$$

em que $\mathbf{U}_{4 \times 1}$ é o vetor com os deslocamentos generalizados definidos na Fig. 10.1. Da mesma forma, definimos

$$w(x) = \mathbf{N}(x)\mathbf{W}. \quad (10.26)$$

No que segue, iremos utilizar \mathbf{N} no lugar de $\mathbf{N}(x)$ para simplificar a notação.

A Eq. (10.2) contém derivadas segundas de $v(x)$ e de $w(x)$, tal que podemos definir

$$\frac{d^2 v(x)}{dx^2} = \frac{d^2}{dx^2} (\mathbf{N}\mathbf{U}) = \frac{d^2 \mathbf{N}}{dx^2} \mathbf{U} = \mathbf{B}\mathbf{U} \quad (10.27)$$

ou seja, a matriz de derivada das funções de interpolação conterá derivadas segundas das funções $N_i(x)$. Assim, podemos avaliar o primeiro termo da Eq. (10.2) como

$$\int_0^L \frac{d^2 w(x)}{dx^2} EI \frac{d^2 v(x)}{dx^2} dx = \int_0^L (\mathbf{B}\mathbf{W})^T EI (\mathbf{B}\mathbf{U}) dx = \mathbf{W}^T \int_0^L \mathbf{B}^T EI \mathbf{B} dx \mathbf{U}. \quad (10.28)$$

O primeiro termo do lado direito da Eq. (10.2) pode ser avaliado para formas gerais de $q(x)$, mas é interessante definirmos uma interpolação para o carregamento distribuído. Assumindo uma interpolação linear

$$q(x) = \left(1 - \frac{x}{L}\right) q_1 + \frac{x}{L} q_2 = \mathbf{Z}_{1 \times 2} \mathbf{q}_{2 \times 1} \quad (10.29)$$

em que q_1 e q_2 são os valores do carregamento distribuído em $x = 0$ e $x = L$, respectivamente (valores nodais). Assim,

$$\int_0^L w(x) q(x) dx = \int_0^L (\mathbf{N}\mathbf{W})^T (\mathbf{Z}\mathbf{q}) dx = \mathbf{W}^T \int_0^L \mathbf{N}^T \mathbf{Z} q dx. \quad (10.30)$$

A avaliação do segundo termo da Eq. (10.2) depende do comportamento de $w(x)$ e do cortante $V(x)$ nas extremidades da viga

$$w(x)V(x)|_0^L = w(L)V(L) - w(0)V(0) \quad (10.31)$$

e sabemos que, devido as funções de interpolação $N_i(x)$, $w(L) = W_3$ e $w(0) = W_1$. Os esforços cortantes são esforços internos, então devemos ter cuidado com a notação de sinais. Em $x = 0$ temos que $V(0) = -F_1$ e, em $x = L$, $V(L) = F_3$. Assim,

$$W_3 F_3 - W_1 (-F_1) = W_3 F_3 + W_1 F_1. \quad (10.32)$$

O mesmo procedimento é válido para o último termo da Eq. (10.2)

$$\frac{dw(x)}{dx} M(x) \Big|_0^L = W_2 F_2 + W_4 F_4 \quad (10.33)$$

em que utilizamos a notação de forças generalizadas (forças e momentos). Assim, podemos escrever os dois termos de condições de contorno em uma só expressão matricial

$$\mathbf{W}^T \mathbf{F}. \quad (10.34)$$

Finalmente, podemos utilizar os resultados anteriores para reescrever a Eq. (10.2) como

$$\mathbf{W}^T \int_0^L \mathbf{B}^T EI \mathbf{B} dx \mathbf{U} = \mathbf{W}^T \int_0^L \mathbf{N}^T \mathbf{Z} \mathbf{q} dx + \mathbf{W}^T \mathbf{F} \quad (10.35)$$

e podemos colocar os termos em comum em evidência, tal que

$$\mathbf{W}^T \left(\int_0^L \mathbf{B}^T EI \mathbf{B} dx \mathbf{U} - \int_0^L \mathbf{N}^T \mathbf{Z} \mathbf{q} dx - \mathbf{F} \right) = \mathbf{0} \quad (10.36)$$

dando origem ao sistema linear de equações

$$\boxed{\int_0^L \mathbf{B}^T EI \mathbf{B} dx = \int_0^L \mathbf{N}^T \mathbf{Z} \mathbf{q} dx + \mathbf{F}} \quad (10.37)$$

Avaliando as integrais, obtemos

$$\mathbf{K} = \int_0^L \mathbf{B}^T EI \mathbf{B} dx = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \quad (10.38)$$

e

$$\mathbf{F}_q = \int_0^L \mathbf{N}^T \mathbf{Z} \mathbf{q} dx = \begin{Bmatrix} \frac{3Lq_2 + 7Lq_1}{20} \\ \frac{2L^2q_2 + 3L^2q_1}{60} \\ \frac{7Lq_2 + 3Lq_1}{20} \\ -\frac{3L^2q_2 + 2L^2q_1}{60} \end{Bmatrix} \quad (10.39)$$

tal que o equilíbrio do elemento pode ser escrito como

$$\mathbf{KU} = \mathbf{F}_q + \mathbf{F}. \quad (10.40)$$

Exercício 10.1 Interprete fisicamente o significado dos termos da matriz de rigidez e do vetor de forças distribuídas \mathbf{F}_q (como foi realizado no capítulo de barras).

DICA: Para interpretar \mathbf{F}_q , avalie quais são as reações de uma viga bi-engastada submetida a um carregamento distribuído linear como o descrito na Eq. (10.29). ■

■ **Exemplo 10.1** Seja uma viga engastada em $x = 0$ e um momento concentrado positivo M_c em $x = L$. O sistema de referência local coincide com o sistema global.

O sistema de Equações para o elemento pode ser escrito como

$$\frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} ? \\ ? \\ 0 \\ M_c \end{Bmatrix}. \quad (10.41)$$

Como a viga está engastada no nó 1, graus de liberdade 1 (translação) e 2 (rotação), devemos eliminar as duas primeiras equações do sistema (linhas e colunas), resultando em

$$\frac{EI}{L^3} \begin{bmatrix} 12 & -6L \\ -6L & 4L^2 \end{bmatrix} \begin{Bmatrix} U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ M_c \end{Bmatrix} \quad (10.42)$$

com solução

$$\begin{Bmatrix} U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} \frac{M_c L^2}{2EI} \\ \frac{M_c L}{EI} \end{Bmatrix} \quad (10.43)$$

que são, respectivamente, o deslocamento vertical e a rotação do segundo nó. Com isso, podemos interpolar o deslocamento transversal ao longo da viga, obtendo

$$v(x) = N_3(x)U_3 + N_4(x)U_4 = \left(\frac{3x^2}{L^2} - \frac{2x^3}{L^3} \right) \frac{M_c L^2}{2EI} + \left(\frac{x^3}{L^2} - \frac{x^2}{L} \right) \frac{M_c L}{EI} \quad (10.44)$$

que pode ser simplificado para

$$v(x) = \frac{M_c}{2EI} x^2. \quad (10.45)$$

Da mesma forma que fizemos com o elemento de barra, podemos calcular os esforços nodais associados ao vetor de deslocamento. Substituindo a solução \mathbf{U} no sistema global, obtemos

$$\frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ \frac{M_c L^2}{2EI} \\ \frac{M_c L}{EI} \end{Bmatrix} = \mathbf{F} \quad (10.46)$$

resultando em

$$\mathbf{F} = \begin{Bmatrix} 0 \\ -M_c \\ 0 \\ M_c \end{Bmatrix}, \quad (10.47)$$

ou seja, conseguimos calcular a reação de momento no nó 1. ■

Exercício 10.2 Compare a solução nodal e a solução interpolada do exemplo com a solução analítica. Comente. ■

Exercício 10.3 Ainda sobre o exemplo anterior: Obtenha a solução interpolada para as rotações ao longo da viga e compare com a solução analítica. Comente. ■

Exercício 10.4 Repita o mesmo procedimento do exemplo para o caso de uma viga bi-apoiada submetida a um carregamento distribuído constante q_0 ao longo de toda a sua extensão. Obtenha, também, as reações. Compare com a solução analítica e comente. ■

10.2 Sobreposição de elementos de viga

O elemento de viga longa definido neste capítulo possui 2 nós, com 2 graus de liberdade por nó. Se tivermos mais de um elemento, devemos sobrepor os termos de rigidez e de força que compartilham graus de liberdade, tal como foi realizado no capítulo sobre elementos de Barra. Para ilustrar o procedimento, vamos considerar a estrutura definida na Fig. 10.3.

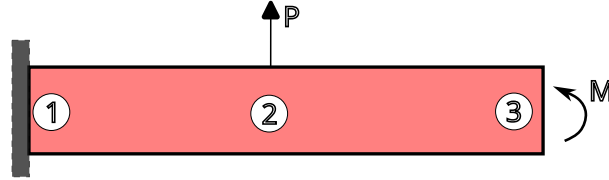


Figura 10.3: Viga engastada submetida a uma força transversal e um momento concentrado.

Essa estrutura tem uma força concentrada P aplicada na metade do seu comprimento, de tal forma que não podemos utilizar somente um elemento para descrever esse problema (podemos aplicar forças e momentos concentrados somente nos nós). Por isso, precisamos de 3 nós e 2 elementos. Como temos 3 nós, cada um com 2 graus de liberdade, teremos um sistema global com dimensão 6×6 . Começando com o elemento 1, com L_1 , E_1 e I_1 . O equilíbrio do elemento é descrito por

$$\frac{E_1 I_1}{L_1^3} \begin{bmatrix} 12 & 6L_1 & -12 & 6L_1 \\ 6L_1 & 4L_1^2 & -6L_1 & 2L_1^2 \\ -12 & -6L_1 & 12 & -6L_1 \\ 6L_1 & 2L_1^2 & -6L_1 & 4L_1^2 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} \quad (10.48)$$

e o equilíbrio do elemento 2, com L_2 , E_2 e I_2

$$\frac{E_2 I_2}{L_2^3} \begin{bmatrix} 12 & 6L_2 & -12 & 6L_2 \\ 6L_2 & 4L_2^2 & -6L_2 & 2L_2^2 \\ -12 & -6L_2 & 12 & -6L_2 \\ 6L_2 & 2L_2^2 & -6L_2 & 4L_2^2 \end{bmatrix} \begin{Bmatrix} U_3 \\ U_4 \\ U_5 \\ U_6 \end{Bmatrix} = \begin{Bmatrix} F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix}. \quad (10.49)$$

Novamente, podemos 'aumentar' as equações de ambos elementos inserindo zeros para que a dimensão de ambos seja a mesma do sistema completo (esse procedimento é puramente didático e não deve ser utilizado em um código de computador)

$$\frac{E_1 I_1}{L_1^3} \begin{bmatrix} 12 & 6L_1 & -12 & 6L_1 & 0 & 0 \\ 6L_1 & 4L_1^2 & -6L_1 & 2L_1^2 & 0 & 0 \\ -12 & -6L_1 & 12 & -6L_1 & 0 & 0 \\ 6L_1 & 2L_1^2 & -6L_1 & 4L_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} \quad (10.50)$$

e

$$\frac{E_2 I_2}{L_2^3} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 6L_2 & -12 & 6L_2 \\ 0 & 0 & 6L_2 & 4L_2^2 & -6L_2 & 2L_2^2 \\ 0 & 0 & -12 & -6L_2 & 12 & -6L_2 \\ 0 & 0 & 6L_2 & 2L_2^2 & -6L_2 & 4L_2^2 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix}. \quad (10.51)$$

Com isso, podemos escrever o equilíbrio do sistema completo como

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} = \underbrace{\frac{E_1 I_1}{L_1^3} \begin{bmatrix} 12 & 6L_1 & -12 & 6L_1 & 0 & 0 \\ 6L_1 & 4L_1^2 & -6L_1 & 2L_1^2 & 0 & 0 \\ -12 & -6L_1 & 12 & -6L_1 & 0 & 0 \\ 6L_1 & 2L_1^2 & -6L_1 & 4L_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{K}^1} + \underbrace{\frac{E_2 I_2}{L_2^3} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 6L_2 & -12 & 6L_2 \\ 0 & 0 & 6L_2 & 4L_2^2 & -6L_2 & 2L_2^2 \\ 0 & 0 & -12 & -6L_2 & 12 & -6L_2 \\ 0 & 0 & 6L_2 & 2L_2^2 & -6L_2 & 4L_2^2 \end{bmatrix}}_{\mathbf{K}^2} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{Bmatrix}$$

e podemos identificar a sobreposição nas posições K_{33} , K_{34} , K_{43} e K_{44} da matriz global de rigidez. Com isso, podemos substituir os valores de forças generalizadas do problema, tal que

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} = \begin{Bmatrix} ? \\ ? \\ P \\ 0 \\ 0 \\ M \end{Bmatrix} \quad (10.52)$$

pois sabemos que o engaste no nó 1 irá eliminar as equações globais 1 e 2. Utilizando a estratégia de zerar linhas e colunas, colocando 1 na diagonal, podemos escrever

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{33} & K_{34} & K_{35} & K_{36} \\ 0 & 0 & K_{43} & K_{44} & K_{45} & K_{46} \\ 0 & 0 & K_{53} & K_{54} & K_{55} & K_{56} \\ 0 & 0 & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ P \\ 0 \\ 0 \\ M \end{Bmatrix} \quad (10.53)$$

com

$$K_{33} = K_{33}^1 + K_{33}^2 \quad (10.54)$$

$$K_{34} = K_{34}^1 + K_{34}^2 \quad (10.55)$$

$$K_{44} = K_{44}^1 + K_{44}^2 \quad (10.56)$$

e as demais posições são advindas da matriz \mathbf{K}^2 diretamente.

Exercício 10.5 Arbitre valores para o exemplo acima e obtenha a solução do problema. Obtenha um gráfico da interpolação do deslocamento transversal e da rotação em cada um dos elementos e compare com a solução analítica. Comente. ■

Exercício 10.6 Considere o exemplo anterior, mas submetido a dois carregamentos distribuídos constantes: q no primeiro elemento e w no segundo elemento. Como realizamos a sobreposição dos vetores \mathbf{F}_q , Eq. (10.39)? Desenvolva a sobreposição utilizando o procedimento de 'aumentar' os sistemas de cada elemento. Proponha valores, solucione e compare com a solução analítica.

DICA: Isole \mathbf{F} no sistema da Eq. (10.40) e realize a sobreposição de efeitos. ■

11. Elemento Finito de Pórtico Plano

Nos capítulos anteriores vimos que o elemento finito de barra, que é 1D, pode ser 'estendido' para problemas 2D (treliças). Para isso, trabalhamos com dois conceitos: 'aumentar' o sistema local do elemento para 2D e transformar o sistema de referência local para o sistema global, para permitir a sobreposição de efeitos (sistema global de equilíbrio).

O mesmo pode ser feito com elementos de viga, mas devemos lembrar que, ao contrário de treliças, um **pórtico** não tem conexões rotuladas. Assim, existe a transferência de todos os esforços entre as conexões. Além disso, um pórtico pode ter forças distribuídas ao longo do comprimento de seus elementos, ao contrário de uma treliça. Isso faz com que tenhamos que considerar esforços internos de esforço normal (advindo do modelo de barra), esforço cortante e momento fletor (advindos do modelo de viga). Resumindo, devemos sobrepor os modelos de viga e de barra para 'aumentar' os modelos 1D para analisarmos pórticos 2D.

A definição de um elemento de pórtico 2D é ilustrada na Fig. 11.1, onde temos 2 nós e 3 graus de liberdade por nó, dois de translação (axial e transversal) e um de rotação.

Observando que já temos as definições de rigidez e de força associados a esses graus de liberdade, podemos propor um sistema 6×6 em que os graus de liberdade (locais) 1 e 4 (axiais) são associados ao modelo de barra e os graus de liberdade 2, 3, 5 e 6 são associados ao modelo de viga. É muito importante observarmos que os modelos de barra e de viga, localmente, são desacoplados (aqui estamos assumindo comportamento linear). Isso significa que a aplicação de uma força axial não provoca flexão e a aplicação de forças transversais ou momentos não provoca uma resposta axial. Utilizando o mesmo raciocínio utilizado nos procedimentos de sobreposição dos capítulos anteriores, podemos 'aumentar' os modelos de barra e de viga, obtendo

$$\frac{EA}{L} \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} \quad (11.1)$$

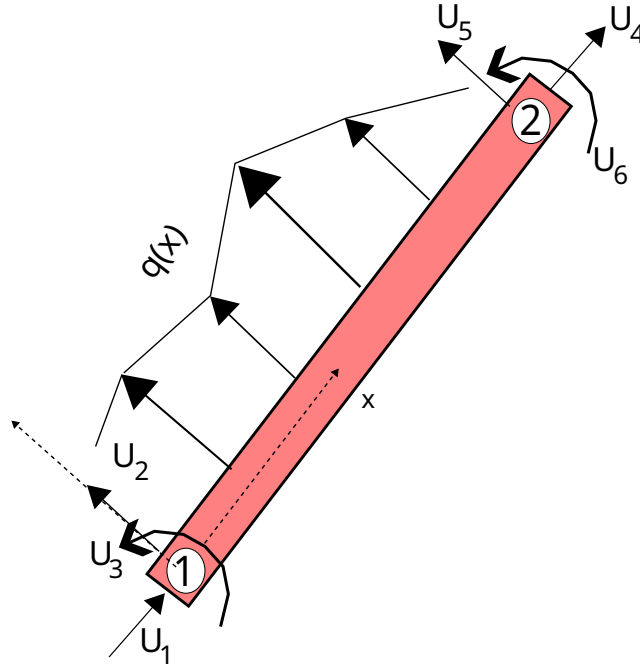


Figura 11.1: Graus de liberdade de um elemento de pórtico plano (sistema local)

e

$$\frac{EI}{L^3} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 6L & 0 & -12 & 6L \\ 0 & 6L & 4L^2 & 0 & -6L & 2L^2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ 0 & 6L & 2L^2 & 0 & -6L & 4L^2 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} + \mathbf{F}_q \quad (11.2)$$

em que

$$\mathbf{F}_q = \begin{Bmatrix} 0 \\ \frac{3Lq_2+7Lq_1}{20} \\ \frac{2L^2q_2+3L^2q_1}{60} \\ 0 \\ \frac{7Lq_2+3Lq_1}{20} \\ -\frac{3L^2q_2+2L^2q_1}{60} \end{Bmatrix}. \quad (11.3)$$

Neste ponto, o leitor deve verificar que os sistemas das Eqs. (11.1) e (11.2) são totalmente equivalentes aos sistemas locais originais de barra e de viga, devido a inclusão dos 0 nas posições adequadas das matrizes de rigidez. Assim, sobrepondo os sistemas de barra e de viga, obtemos

$$\begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} + \begin{Bmatrix} 0 \\ \frac{3Lq_2+7Lq_1}{20} \\ \frac{2L^2q_2+3L^2q_1}{60} \\ 0 \\ \frac{7Lq_2+3Lq_1}{20} \\ -\frac{3L^2q_2+2L^2q_1}{60} \end{Bmatrix} \quad (11.4)$$

ou

$$\mathbf{K}_l^e \mathbf{U}_l^e = \mathbf{F}_l^e + \mathbf{F}_{q_l}^e. \quad (11.5)$$

O procedimento de sobreposição de efeitos segue o descrito no capítulo sobre implementação computacional para problemas de treliça, com algumas particularidades. A primeira particularidade é no cálculo do grau de liberdade global. O grau de liberdade global associado a um nó n e grau de liberdade local gll (1 para translação global em X , 2 para translação global em Y e 3 para rotação)

$$gl = 3(n - 1) + gll \quad (11.6)$$

pois temos 3 graus de liberdade por nó. A outra modificação que deve ser realizada é na definição da matriz de transformação \mathbf{T} . Agora, os graus de liberdade locais em um nó podem ser interpretados como as componentes de um vetor de dimensão 3 e, como o pórtico está sempre no plano XY , não existe transformação (rotação) no eixo Z , que é o eixo da rotação. Assim, podemos aumentar a matriz de rotação \mathbf{R} para

$$\mathbf{R}_{3 \times 3} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11.7)$$

tal que

$$\mathbf{T}_{6 \times 6} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_{3 \times 3} \end{bmatrix}. \quad (11.8)$$

É importante salientar que os vetores $\mathbf{F}_{q_l}^e$ são definidos no sistema local de referência do elemento. Assim, devemos rotacionar esses vetores para sobrepor em um vetor global de carregamentos distribuídos, por meio da operação

$$\mathbf{F}_{q_g}^e = (\mathbf{T}^e)^T \mathbf{F}_{q_l}^e, \quad (11.9)$$

lembrando que a matriz \mathbf{T}^e mapeia do sistema global para o local e sua operação inversa é dada pelo transposto (já que a matriz é ortogonal).

Por fim, agora precisamos informar o momento de inércia de cada elemento finito e, se for o caso, o carregamento distribuído no sistema local de cada elemento.

11.1 Algoritmos para implementação computacional

Os algoritmos que foram apresentados no capítulo sobre a implementação computacional para treliças podem ser reaproveitados e/ou adaptados para a implementação de um programa de análise de elementos finitos para pórticos planos.

Os Algs. 3 e 4 não precisam de modificação, pois tanto o comprimento quanto o ângulo entre os sistemas de referência (global e local do elemento) são idênticos à análise de treliças.

A montagem da matriz de rigidez global precisa de poucas modificações. Baseado no Alg. 5, podemos propor o Alg. 12. Em relação aos argumentos da função, temos o vetor \mathbf{I}_z contendo os momentos de inércia em cada elemento. A dimensão do sistema global agora é $3n_n \times 3n_n$ e as matrizes locais tem dimensão 6×6 .

O Alg. 6 deve ser modificado de acordo com o Alg. 13.

O Alg. 7, por sua vez, deve ser modificado de acordo com o Alg. 14.

Algoritmo 12: Montagem da matriz de rigidez global em um pórtico plano

```

1 RigidezGlobalPP( $n_n, \mathbf{E}, \mathbf{A}, \mathbf{I}_z, \mathbf{L}, \boldsymbol{\theta}, \mathbf{IJ}$ )
2   Inicializa a matriz global
3    $\mathbf{K} \leftarrow \mathbf{0}_{3n_n \times 3n_n}$ 
4   Laço nos elementos
5   for  $e \in \{1, \dots, n_e\}$  do
6     Recupera as propriedades do elemento
7      $E \leftarrow \mathbf{E}(e)$ 
8      $A \leftarrow \mathbf{A}(e)$ 
9      $I_z \leftarrow \mathbf{I}_z(e)$ 
10     $L \leftarrow \mathbf{L}(e)$ 
11     $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}(e)$ 
12    Recupera os nós do elemento
13     $n_1 \leftarrow \mathbf{IJ}(e, 1)$ 
14     $n_2 \leftarrow \mathbf{IJ}(e, 2)$ 
15    Monta a matriz de rigidez local, Eq.(11.4)
16     $\mathbf{K}_l^e \leftarrow \text{RigidezPP}(E, A, I_z, L)$ 
17    Monta a matriz de rotação do elemento, Eq.(11.8)
18     $\mathbf{T}^e \leftarrow \text{MatrizTPP}(\boldsymbol{\theta})$ 
19    Transforma a rigidez do elemento
20     $\mathbf{K}_g^e \leftarrow (\mathbf{T}^e)^T \mathbf{K}_l^e \mathbf{T}^e$ 
21    Determina os gls do elemento, Eq. (8.15)
22     $\mathbf{g}^e \leftarrow \text{glsPP}(n_1, n_2)$ 
23    Sobreposição a matriz do elemento na rigidez global
24    for  $i \in \{1, \dots, 6\}$  do
25      for  $j \in \{1, \dots, 6\}$  do
26         $\mathbf{K}(g(i), g(j)) \leftarrow \mathbf{K}(g(i), g(j)) + \mathbf{K}_g^e(i, j)$ 
27  Retorna a matriz global
28  return  $\mathbf{K}$ 

```

Algoritmo 13: Montagem da matriz de rigidez local do elemento de Pórtico Plano

```

1 RigidezPP( $E, A, I, L$ )
2   Define a matriz
3    $\mathbf{K}_l \leftarrow \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$ 
4   Retorna a matriz
5   return  $\mathbf{K}_l$ 

```

Algoritmo 14: Montagem da matriz de rotação do elemento de pórtico plano com ângulo θ

```

1 MontagemMatrizTPP( $\theta$ )
2   Calcula o seno e o cosseno
3    $c \leftarrow \cos(\theta)$ 
4    $s \leftarrow \sin(\theta)$ 
5   Define a matriz
6    $\mathbf{T} \leftarrow \begin{bmatrix} c & s & 0 & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & s & 0 \\ 0 & 0 & 0 & -s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ 
7   Retorna a matriz
8   return  $\mathbf{T}$ 

```

Algoritmo 15: Determina os gls do elemento de pórtico plano com nós n_1 e n_2

```

1 glsPP( $n_1, n_2$ )
2   Define o vetor
3    $\mathbf{g} \leftarrow \begin{Bmatrix} 3(n_1 - 1) + 1 \\ 3(n_1 - 1) + 2 \\ 3(n_1 - 1) + 3 \\ 3(n_2 - 1) + 1 \\ 3(n_2 - 1) + 2 \\ 3(n_2 - 1) + 3 \end{Bmatrix}$ 
4   Retorna o vetor
5   return  $\mathbf{g}$ 

```

Algoritmo 16: Montagem do vetor global de forças concentradas para o problema de pórtico plano

```

1 ForcaCGlobalPP( $n_n, n_f, FC$ )
2   Inicializa vetor global
3    $\mathbf{F} \leftarrow \mathbf{0}_{3n_n}$ 
4   Laço nas linhas de  $\mathbf{FC}$ 
5   for  $i \in \{1, \dots, n_f\}$  do
6     Nó
7      $no \leftarrow \mathbf{FC}(i, 1)$ 
8     Grau de liberdade (local)
9      $gll \leftarrow \mathbf{FC}(i, 2)$ 
10    Magnitude
11     $valor \leftarrow \mathbf{FC}(i, 3)$ 
12    Grau de liberdade global
13     $glg \leftarrow 3(no - 1) + gll$ 
14    Adiciona ao vetor global
15     $\mathbf{F}(glg) \leftarrow \mathbf{F}(glg) + valor$ 
16  Retorna o vetor de forças globais
17  return  $\mathbf{F}$ 

```

A determinação dos graus de liberdade globais é ilustrada no Alg. 15.

A montagem do vetor de forças generalizadas globais (advindos de carregamentos concentrados), Alg. 9 deve ser modificado de acordo com o Alg. 16. As modificações são somente na dimensão do vetor global e na determinação do gl global.

Devemos, também, elaborar um Algoritmo para a montagem do vetor de forças globais associadas aos carregamentos distribuídos. Para isso, precisamos definir uma entrada de dados. Vamos definir n_q como o número de elementos submetidos a carregamento (local) distribuído. A matriz \mathbf{FQ} com n_q linhas e 3 colunas contém, na primeira coluna, o elemento e , nas segundas e terceiras colunas, os valores q_1 e q_2 que definem os valores nodais de $q(x)$. O Alg. 17 apresenta o procedimento de montagem do vetor global e o Alg.

Por fim, devemos alterar o Alg. 10 para considerar a Eq. (11.6).

Exercício 11.1 Proponha uma implementação computacional dos algoritmos discutidos neste capítulo. Após, escolha um problema de análise de um pórtico plano (deve conter ao menos dois elementos) e avalie a solução. ■

11.2 Exemplo de pórtico plano

Seja o pórtico plano da Fig. 11.2, com 3 nós e 2 elementos. Vamos assumir que as coordenadas nodais são (0,0), (0,1) e (1,1), respectivamente. Vamos assumir, também, que o primeiro elemento está definido entre os nós 12 e o segundo entre os nós 23. Com isso, podemos identificar que ambos os elementos tem comprimento unitário e que os ângulos são $\theta_1 = 90^\circ$ e $\theta_2 = 0^\circ$. O primeiro elemento é submetido a um carregamento distribuído constante que está no sentido oposto ao eixo local y do primeiro elemento, tal que $q_1 = q_2 = -q_0$ [N/m]. Por fim, observamos que existe uma força $-F$ [N] aplicada na direção Y global do segundo nó e que os gls globais 1, 2 e 7 são restritos.

Algoritmo 17: Montagem do vetor global de forças distribuídas em pórtico plano

```

1 FQPP( $n_n, \mathbf{L}, \boldsymbol{\theta}, \mathbf{IJ}, n_q, \mathbf{FQ}$ )
2   Inicializa o vetor global
3    $\mathbf{F} \leftarrow \mathbf{0}_{3n_n \times 1}$ 
4   Laço nas linhas de  $\mathbf{FQ}$ 
5   for  $l \in \{1, \dots, n_q\}$  do
6     Recupera o elemento submetido ao carregamento distribuído
7      $e \leftarrow \mathbf{FQ}(l, 1)$ 
8     Recupera os valores de  $q_1$  e  $q_2$ 
9      $q_1 \leftarrow \mathbf{FQ}(l, 2)$ 
10     $q_2 \leftarrow \mathbf{FQ}(l, 3)$ 
11    Recupera comprimento e ângulo
12     $L \leftarrow \mathbf{L}(e)$ 
13     $\theta \leftarrow \boldsymbol{\theta}(e)$ 
14    Monta o vetor local , Eq.(11.3)
15     $\mathbf{F}_{ql}^e \leftarrow \text{FqPP}(L, q_1, q_2)$ 
16    Monta a matriz de rotação do elemento, Eq.(11.8)
17     $\mathbf{T}^e \leftarrow \text{MatrizTPP}(\theta)$ 
18    Transforma o vetor para o sistema global
19     $\mathbf{F}_{qg}^e \leftarrow (\mathbf{T}^e)^T \mathbf{F}_{ql}^e$ 
20    Recupera os nós do elemento
21     $n_1 \leftarrow \mathbf{IJ}(e, 1)$ 
22     $n_2 \leftarrow \mathbf{IJ}(e, 2)$ 
23    Determina os gls do elemento
24     $\mathbf{g}^e \leftarrow \text{glsPP}(n_1, n_2)$ 
25    Sobreposição no vetor global
26    for  $i \in \{1, \dots, 6\}$  do
27       $\mathbf{F}(g(i)) \leftarrow \mathbf{F}(g(i)) + \mathbf{F}_{qg}^e(i)$ 
28  Retorna o vetor global
29  return  $\mathbf{F}$ 

```

Algoritmo 18: Monta o vetor \mathbf{F}_q local de um elemento de pórtico plano

```

1 FqPP( $L, q_1, q_2$ )
2   Define o vetor
3    $\mathbf{F}_{ql} \leftarrow \begin{Bmatrix} 0 \\ \frac{3Lq_2+7Lq_1}{20} \\ \frac{2L^2q_2+3L^2q_1}{60} \\ 0 \\ \frac{7Lq_2+3Lq_1}{20} \\ -\frac{3L^2q_2+2L^2q_1}{60} \end{Bmatrix}$ 
4   Retorna o vetor
5   return  $\mathbf{F}_{ql}$ 

```

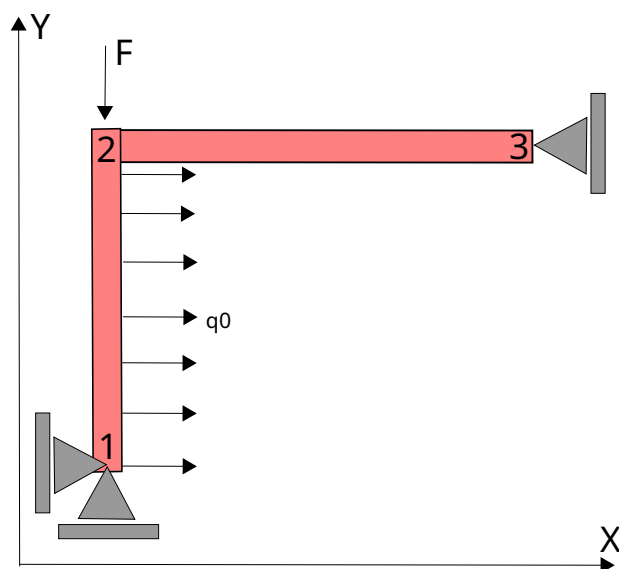


Figura 11.2: Geometria, carregamentos e apoios para o exemplo de pórtico plano.

12. C.C. de Dirichlet não homogêneas

As condições de contorno de Dirichlet, ou essenciais, correspondem a imposição de valores de deslocamento (no caso de treliças) e/ou deslocamentos e rotações (no caso de pórticos planos). Os capítulos anteriores consideraram que esses deslocamentos e/ou rotações eram nulos, também conhecidos como condições de contorno homogêneas. No entanto, é possível impor condições de contorno de Dirichlet não nulas. O objetivo deste capítulo é apresentar algumas técnicas que podem ser incorporadas ao código de elementos finitos.

12.1 Técnica 1 - Rearranjo do sistema de equações

Para ilustrar o procedimento, vamos considerar um sistema linear com 3 equações

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}. \quad (12.1)$$

Se o valor de U_2 for prescrito como \bar{U}_2 , podemos verificar que as incógnitas restantes são U_1 e U_3 . Desta forma, podemos reescrever as equações como

$$\begin{cases} K_{11}U_1 + K_{13}U_3 = F_1 - K_{12}\bar{U}_2 \\ K_{21}U_1 + K_{23}U_3 = F_2 - K_{22}\bar{U}_2 \\ K_{31}U_1 + K_{33}U_3 = F_3 - K_{32}\bar{U}_2 \end{cases} \quad (12.2)$$

e uma das equações pode ser eliminada, já que temos apenas 2 incógnitas. Se eliminarmos a equação associada a U_2 , obtemos o sistema

$$\begin{bmatrix} K_{11} & K_{13} \\ K_{31} & K_{33} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_3 \end{Bmatrix} - \begin{Bmatrix} K_{12}\bar{U}_2 \\ K_{32}\bar{U}_2 \end{Bmatrix} \quad (12.3)$$

e podemos interpretar o vetor de forças adicional como sendo as forças que devem ser aplicadas ao sistema para garantir que $U_2 = \bar{U}_2$. Como já discutimos na imposição de condições de contorno de Dirichlet homogêneas, a alteração do número de equações do sistema tem

um lado favorável (redução do número de equações) e um lado desfavorável (criação de um novo sistema, com movimentação de informações na memória do computador). Assim, é possível realizar o mesmo procedimento de zerarmos as linhas e colunas das equações com valores impostos. Para o exemplo,

$$\begin{bmatrix} K_{11} & 0 & K_{13} \\ 0 & 1 & 0 \\ K_{31} & 0 & K_{33} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ \bar{U}_2 \\ F_3 \end{Bmatrix} - \begin{Bmatrix} K_{12}\bar{U}_2 \\ 0 \\ K_{32}\bar{U}_2 \end{Bmatrix} \quad (12.4)$$

O procedimento pode ser descrito como: para cada grau de liberdade j restrito a um valor \bar{U}_j devemos realizar o produto da coluna j da matriz de rigidez \mathbf{K} pelo valor restrito. O vetor resultante deve ser descontado do vetor de forças \mathbf{F} . Após a consideração de todos os valores impostos, zeramos as linhas e colunas j de \mathbf{K} e colocamos 1 na diagonal da matriz. Por fim, para cada gl j restrito, impomos o valor \bar{U}_j na linha j de \mathbf{F} .

Deve-se observar que o procedimento descrito acaba sendo igual ao utilizado nos capítulos anteriores quando todos os $\bar{U}_j = 0$ (caso homogêneo).

■ **Exemplo 12.1** Seja o sistema

$$\begin{bmatrix} 10 & -10 & 0 & 0 \\ -10 & 20 & -10 & 0 \\ 0 & -10 & 20 & -10 \\ 0 & 0 & -10 & 10 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix}. \quad (12.5)$$

Vamos considerar a imposição das condições $U_1 = \bar{U}_1 = 0.1$ e $U_3 = \bar{U}_3 = -0.1$. A primeira condição gera um vetor de forças

$$\mathbf{F}_{U_1} = \bar{U}_1 \begin{Bmatrix} K_{11} \\ K_{21} \\ K_{31} \\ K_{41} \end{Bmatrix} = 0.1 \begin{Bmatrix} 10 \\ -10 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{Bmatrix}. \quad (12.6)$$

A segunda condição gera um vetor de forças

$$\mathbf{F}_{U_3} = \bar{U}_3 \begin{Bmatrix} K_{13} \\ K_{23} \\ K_{33} \\ K_{43} \end{Bmatrix} = -0.1 \begin{Bmatrix} 0 \\ -10 \\ 20 \\ -10 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ -2 \\ 1 \end{Bmatrix}. \quad (12.7)$$

Assim, o sistema pode ser reescrito como

$$\begin{bmatrix} 10 & -10 & 0 & 0 \\ -10 & 20 & -10 & 0 \\ 0 & -10 & 20 & -10 \\ 0 & 0 & -10 & 10 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix} - \begin{Bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 1 \\ -2 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 2 \\ 5 \\ 3 \end{Bmatrix}. \quad (12.8)$$

A partir deste momento, podemos obter um sistema reduzido pela eliminação das equações 1 e 3

$$\begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix} \begin{Bmatrix} U_2 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 3 \end{Bmatrix} \quad (12.9)$$

que resulta em $U_2 = 0.1$ e $U_4 = 0.3$.

A outra alternativa é utilizar a técnica de zerar as linhas e colunas, colocando 1 na diagonal e os valores prescritos no lado direito do sistema

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} 0.1 \\ 2 \\ -0.1 \\ 3 \end{Bmatrix} \quad (12.10)$$

tal que

$$\mathbf{U} = \begin{Bmatrix} 0.1 \\ 0.1 \\ -0.1 \\ 0.3 \end{Bmatrix}. \quad (12.11)$$

■

O Alg. 19 ilustra o procedimento. Uma alteração que deve ser feita na entrada de dados é informar o valor do deslocamento/rotação prescrita. Neste caso, basta adicionarmos uma coluna à matriz \mathbf{AP} , contendo o valor a ser imposto (terceira coluna).

12.2 Técnica 2 - Lagrangiano

É possível interpretar a solução do sistema de equações $\mathbf{KU} = \mathbf{F}$ como sendo a minimização da energia potencial total, uma função quadrática e convexa (pois a matriz de rigidez deve ser positivo-definida)

$$\Pi(\mathbf{U}) = \frac{1}{2} \mathbf{U}^T \mathbf{KU} - \mathbf{U}^T \mathbf{F}. \quad (12.12)$$

Da teoria de cálculo sabemos que o mínimo de uma função quadrática com matriz positivo-definida ocorre quando

$$\nabla_{\mathbf{U}} \Pi(\mathbf{U}) = \mathbf{KU} - \mathbf{F} = \mathbf{0} \implies \mathbf{KU} = \mathbf{F}. \quad (12.13)$$

Quando o sistema de n equações lineares tem r variáveis restritas $\bar{\mathbf{U}}$ (como no caso da imposição e condições de contorno de Dirichlet), podemos escrever essas restrições de uma forma matricial

$$\mathbf{G}_{r \times n} \mathbf{U}_{n \times 1} = \bar{\mathbf{U}}_{r \times 1}. \quad (12.14)$$

Da teoria de otimização, podemos montar a função Lagrangiana do problema

$$\mathcal{L}(\mathbf{U}, \boldsymbol{\lambda}) = \Pi(\mathbf{U}) + \boldsymbol{\lambda}^T (\mathbf{GU} - \bar{\mathbf{U}}) \quad (12.15)$$

em que $\boldsymbol{\lambda}$ é um vetor com r variáveis adicionais (multiplicadores de Lagrange). Novamente, podemos investigar as condições para o mínimo da função Lagrangiana

$$\nabla_{\mathbf{U}} \mathcal{L}(\mathbf{U}, \boldsymbol{\lambda}) = \mathbf{0} \implies \mathbf{KU} - \mathbf{F} + \mathbf{G}^T \boldsymbol{\lambda} = \mathbf{0} \quad (12.16)$$

e

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{U}, \boldsymbol{\lambda}) = \mathbf{0} \implies \mathbf{GU} - \bar{\mathbf{U}} = \mathbf{0} \quad (12.17)$$

que combinadas resultam em um sistema de equações aumentado

$$\begin{bmatrix} \mathbf{K} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \bar{\mathbf{U}} \end{Bmatrix} \quad (12.18)$$

em que o bloco nulo na matriz de coeficientes tem dimensão $r \times r$.

Algoritmo 19: Aplicação de condições de contorno essenciais utilizando o cálculo das forças de reação devido aos deslocamentos/rotações impostas. n_{gln} contém o número de graus de liberdade por nó do elemento que está sendo utilizado na malha.

```

1 AplicaCC( $n_n, n_a, n_{gln}, \mathbf{AP}, \mathbf{K}, \mathbf{F}$ )
2   Laço nas linhas de  $\mathbf{AP}$  para corrigirmos  $\mathbf{F}$ 
3   for  $l \in \{1, \dots, n_a\}$  do
4       Nó
5        $no \leftarrow \mathbf{AP}(l, 1)$ 
6       Grau de liberdade (local)
7        $gll \leftarrow \mathbf{AP}(l, 2)$ 
8       Valor
9        $val \leftarrow \mathbf{AP}(l, 3)$ 
10      Grau de liberdade global
11       $glg \leftarrow n_{gln}(no - 1) + gll$ 
12      Calcula a força associada e esta cc e soma à  $\mathbf{F}$ 
13      for  $i \in \{1, \dots, n_{gln}n_n\}$  do
14           $\mathbf{F}(i) \leftarrow \mathbf{F}(i) - val \cdot \mathbf{K}(i, glg)$ 
15  Novo laço para zerarmos linhas e colunas
16  for  $l \in \{1, \dots, n_a\}$  do
17      Nó
18       $no \leftarrow \mathbf{AP}(l, 1)$ 
19      Grau de liberdade (local)
20       $gll \leftarrow \mathbf{AP}(l, 2)$ 
21      Valor
22       $val \leftarrow \mathbf{AP}(l, 3)$ 
23      Zera linhas e colunas da matriz
24      for  $i \in \{1, \dots, n_{gln}n_n\}$  do
25           $\mathbf{K}(i, glg) \leftarrow 0$ 
26           $\mathbf{K}(glg, i) \leftarrow 0$ 
27      Coloca 1 na diagonal
28       $\mathbf{K}(glg, glg) \leftarrow 1$ 
29      Informa o valor
30       $\mathbf{F}(glg) \leftarrow valor$ 
31  Retorna  $\mathbf{K}$  e  $\mathbf{F}$  modificados
32  return  $\mathbf{K}, \mathbf{F}$ 

```

■ **Exemplo 12.2** Seja o sistema

$$\begin{bmatrix} 10 & -10 & 0 & 0 \\ -10 & 20 & -10 & 0 \\ 0 & -10 & 20 & -10 \\ 0 & 0 & -10 & 10 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix}. \quad (12.19)$$

Vamos considerar a imposição das condições $U_1 = \bar{U}_1 = 0.1$ e $U_3 = \bar{U}_3 = -0.1$. Neste caso,

$$\mathbf{G}\mathbf{U} = \bar{\mathbf{U}} \implies \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{U} = \begin{Bmatrix} 0.1 \\ -0.1 \end{Bmatrix}. \quad (12.20)$$

Com isso, o sistema aumentado pode ser escrito como

$$\begin{bmatrix} 10 & -10 & 0 & 0 & 1 & 0 \\ -10 & 20 & -10 & 0 & 0 & 0 \\ 0 & -10 & 20 & -10 & 0 & 1 \\ 0 & 0 & -10 & 10 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 0.1 \\ -0.1 \end{Bmatrix} \quad (12.21)$$

tal que

$$\begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} 0.1 \\ 0.1 \\ -0.1 \\ 0.3 \\ 1.0 \\ 9.0 \end{Bmatrix} \quad (12.22)$$

e estamos interessados apenas nos valores de \mathbf{U} (os λ são apenas um recurso para a imposição das restrições de deslocamento).

■

Essa abordagem é muito interessante, pois além de permitir a imposição de valores associados a apoios também permite associar o comportamento de diferentes graus de liberdade do problema, ao que chamamos de **multi point constraints**. Essa é a técnica utilizada por muitos programas de elementos finitos para relacionar graus de liberdade de partes importadas para montar um conjunto.

■ **Exemplo 12.3** Seja o sistema

$$\begin{bmatrix} 10 & -10 & 0 & 0 \\ -10 & 20 & -10 & 0 \\ 0 & -10 & 20 & -10 \\ 0 & 0 & -10 & 10 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix}. \quad (12.23)$$

Vamos considerar a imposição da condição $U_2 = 0.5$ e a igualdade $U_1 = U_4$, que pode ser escrita como $U_1 - U_4 = 0$. Neste caso,

$$\mathbf{G}\mathbf{U} = \bar{\mathbf{U}} \implies \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \mathbf{U} = \begin{Bmatrix} 0.5 \\ 0 \end{Bmatrix}. \quad (12.24)$$

Com isso, o sistema aumentado pode ser escrito como

$$\begin{bmatrix} 10 & -10 & 0 & 0 & 0 & 1 \\ -10 & 20 & -10 & 0 & 1 & 0 \\ 0 & -10 & 20 & -10 & 0 & 0 \\ 0 & 0 & -10 & 10 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 0.5 \\ 0 \end{Bmatrix} \quad (12.25)$$

com solução

$$\begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} 0.933 \\ 0.5 \\ 0.866 \\ 0.933 \\ 10.0 \\ -3.333 \end{Bmatrix}. \quad (12.26)$$

■

Exercício 12.1 Suponha que deseja-se modelar uma treliça em Δ utilizando elementos de viga, pois alguns elementos tem carregamentos transversais distribuídos. No entanto, o uso de elementos de viga faz com que as conexões não sejam rotuladas. Proponha uma estratégia para desacoplar os graus de liberdade de rotação utilizando a segunda técnica vista neste capítulo. ■

Algoritmo 20: Aplicação de condições de contorno essenciais utilizando multiplicadores de Lagrange. n_{gln} contém o número de graus de liberdade por nó do elemento que está sendo utilizado na malha.

```

1 AplicaCCLagr( $n_n, n_a, n_{gln}, \mathbf{AP}, \mathbf{K}, \mathbf{F}$ )
2   Aloca uma matriz de rigidez aumentada
3    $\mathbf{KA} \leftarrow \mathbf{0}_{n_{gln}n_n + n_a \times n_{gln}n_n + n_a}$ 
4   Aloca um vetor de forças aumentado
5    $\mathbf{FA} \leftarrow \mathbf{0}_{n_{gln}n_n + n_a \times 1}$ 
6   Copia a matriz  $\mathbf{K}$  para a matriz aumentada
7   for  $i \in \{1, \dots, n_{gln}n_n\}$  do
8     for  $j \in \{1, \dots, n_{gln}n_n\}$  do
9        $\mathbf{KA}(i, j) \leftarrow \mathbf{K}(i, j)$ 
10  Laço pelas condições de contorno
11  for  $l \in \{1, \dots, n_a\}$  do
12    Nó
13     $no \leftarrow \mathbf{AP}(l, 1)$ 
14    Grau de liberdade (local)
15     $gll \leftarrow \mathbf{AP}(l, 2)$ 
16    Valor
17     $val \leftarrow \mathbf{AP}(l, 3)$ 
18    Grau de liberdade global
19     $glg \leftarrow n_{gln}(no - 1) + gll$ 
20    Coloca 1 nas posições de  $\mathbf{G}$  e  $\mathbf{G}^T$ 
21     $\mathbf{KA}(glg, n_{gln}n_n + l) \leftarrow 1$ 
22     $\mathbf{KA}(n_{gln}n_n + l, glg) \leftarrow 1$ 
23    Coloca o valor prescrito em  $\mathbf{FA}$ 
24     $\mathbf{FA}(n_{gln}n_n + l) \leftarrow val$ 
25  Retorna  $\mathbf{KA}$  e  $\mathbf{FA}$  modificados
26  return  $\mathbf{KA}, \mathbf{FA}$ 

```

Elementos para elasticidade

13	Uma breve revisão de elasticidade linear	119
13.1	Relações entre deslocamentos e deformações	
13.2	Relações entre deformações e tensões	
13.3	Estados planos	
14	Elementos Finitos aplicados à Elasticidade	125
14.1	Método dos resíduos ponderados	
14.2	Método dos elementos finitos	
14.3	Pós-processamento	
15	Elemento bilinear isoparamétrico de 4 nós	129
15.1	Funções de interpolação	
15.2	Cálculo das derivadas e mapeamento isoparamétrico	
15.3	Cálculo das integrais de domínio	
15.4	Cálculo das integrais de contorno	
15.5	Cálculo de tensões	
15.6	Exemplo	
16	Cisalhamento parasita	151
16.1	Modos de deformação de um elemento finito	
16.2	Problemas dominados por flexão	
17	Implementação computacional	159
17.1	Entrada de dados	
17.2	Montagem da matriz de rigidez local de um elemento	
17.3	Montagem da matriz de rigidez global	
17.4	Montagem do vetor de forças distribuídas	
17.5	Cálculo de tensões em um elemento	

13. Uma breve revisão de elasticidade linear

O objetivo deste capítulo é revisar alguns conceitos de elasticidade linear, visando definir uma notação para os próximos capítulos. Espera-se que o leitor tenha noções de mecânica do contínuo, tais como tensão, deformação e relações constitutivas.

Como este texto inicial lida somente com aplicações do método dos elementos finitos para elasticidade linear, iremos trabalhar com deformações infinitesimais e material no regime linear elástico.

13.1 Relações entre deslocamentos e deformações

O estado de deformação em um ponto de um corpo é um tensor de segunda ordem

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{bmatrix} \quad (13.1)$$

em que ε_{ii} são as deformações normais e ε_{ij} , para $i \neq j$, são as deformações angulares médias, definidas como

$$\varepsilon_{ij} = \frac{1}{2} \gamma_{ij} \quad (13.2)$$

e γ_{ij} são as deformações angulares. Cada uma das componentes de deformação pode ser associada ao campo de deslocamentos por meio das relações diferenciais

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{du_i}{dj} + \frac{du_j}{di} \right) \quad i, j = x, y, z. \quad (13.3)$$

Por exemplo,

$$\varepsilon_{xx} = \frac{1}{2} \left(\frac{du_x}{dx} + \frac{du_x}{dx} \right) = \frac{du_x}{dx} \quad (13.4)$$

e

$$\epsilon_{yz} = \frac{1}{2} \left(\frac{du_y}{dz} + \frac{du_z}{dy} \right) = \frac{1}{2} \gamma_{yz}. \quad (13.5)$$

Como o estado de deformação é simétrico, podemos utilizar uma **notação** compacta, chamada de **notação de Voigt**. Vamos introduzir uma notação que será muito útil na formulação de elementos finitos

$$\boldsymbol{\epsilon}^v = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{yz} \\ 2\epsilon_{xz} \\ 2\epsilon_{xy} \end{Bmatrix}, \quad (13.6)$$

lembrando que $\boldsymbol{\epsilon}$ é um tensor (tem dois índices) e que essa notação em forma de 'vetor' é somente uma maneira prática de representar a medida. As relações diferenciais da Eq. (13.3) podem ser expressas em uma forma compacta como

$$\boldsymbol{\epsilon}(\mathbf{x}) = \mathbf{L}(\mathbf{u}) = \frac{1}{2} \left[\nabla \mathbf{u}(\mathbf{x}) + (\nabla \mathbf{u}(\mathbf{x}))^T \right] \quad (13.7)$$

em que \mathbf{L} é um operador diferencial (linear) e \mathbf{u} é o vetor de deslocamentos. Aqui devemos lembrar que o gradiente de um vetor é um tensor de segunda ordem, pois

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{du_x}{dx} & \frac{du_y}{dx} & \frac{du_z}{dx} \\ \frac{du_x}{dy} & \frac{du_y}{dy} & \frac{du_z}{dy} \\ \frac{du_x}{dz} & \frac{du_y}{dz} & \frac{du_z}{dz} \end{bmatrix}. \quad (13.8)$$

Também podemos escrever a relação da Eq. (13.7) na notação de Voigt

$$\boldsymbol{\epsilon}^v = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{yz} \\ 2\epsilon_{xz} \\ 2\epsilon_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{d}{dx} & 0 & 0 \\ 0 & \frac{d}{dy} & 0 \\ 0 & 0 & \frac{d}{dz} \\ 0 & \frac{d}{dz} & \frac{d}{dy} \\ \frac{d}{dz} & 0 & \frac{d}{dx} \\ \frac{d}{dy} & \frac{d}{dx} & 0 \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} = \mathbf{L}^v \mathbf{u} \quad (13.9)$$

em que \mathbf{L}^v é uma matriz.

13.2 Relações entre deformações e tensões

O estado de tensão em um ponto é um tensor de segunda ordem simétrico

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{bmatrix}, \quad (13.10)$$

tal que também pode ser representado pela notação de Voigt

$$\boldsymbol{\sigma}^v = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{Bmatrix}. \quad (13.11)$$

A relação entre tensão e deformação em um ponto de um material linear elástico pode ser escrita como

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \quad (13.12)$$

ou, em notação indicial

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl} \quad (13.13)$$

tal que \mathbf{C} é um tensor de quarta ordem (4 índices). Como a tensão e a deformação são tensores simétricos de segunda ordem, pode-se mostrar que \mathbf{C} possui diversas simetrias em relação aos seus índices, também permitindo o uso da notação compacta de Voigt. Desta forma, podemos escrever

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix} = \begin{bmatrix} C_{xxxx} & C_{xxyy} & C_{xxzz} & C_{xxyz} & C_{xxxz} & C_{xxxxy} \\ C_{xxyy} & C_{yyyy} & C_{yyzz} & C_{yyyz} & C_{yyxz} & C_{yyxy} \\ C_{xxzz} & C_{yyzz} & C_{zzzz} & C_{zzyz} & C_{zzxz} & C_{zzxy} \\ C_{xxyz} & C_{yyyz} & C_{zzzy} & C_{yzyz} & C_{yzxz} & C_{yzxy} \\ C_{xxyx} & C_{yyxy} & C_{zzxz} & C_{yzxz} & C_{xzxz} & C_{yzxy} \\ C_{xxxxy} & C_{yyxy} & C_{zzxy} & C_{yzxy} & C_{yzxy} & C_{xyxy} \end{bmatrix} \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{xz} \\ 2\varepsilon_{xy} \end{pmatrix} \quad (13.14)$$

ou seja, podemos tratar o tensor de quarta ordem como uma matriz simétrica 6×6 , tal que

$$\boldsymbol{\sigma}^v = \mathbf{C}^v \boldsymbol{\varepsilon}^v \quad (13.15)$$

passa a ser o produto entre uma matriz e um vetor.

A matriz \mathbf{C}^v 'cheia' é associada a um material triclinico (ou anisotrópico) geral. Materiais **isotrópicos** são materiais que apresentam 'infinitas' simetrias e tem propriedades invariantes com qualquer tipo de transformação. Neste caso,

$$\mathbf{C}_{\text{iso}}^v = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix} \quad (13.16)$$

em que E é o módulo de elasticidade longitudinal e ν é o coeficiente de Poisson.

13.3 Estados planos

Embora a geometria das peças e componentes seja tridimensional, existem situações em que é possível considerar modelos simplificados bi- ou até uni-dimensionais. Em se tratando da redução de dimensionalidade, temos três opções muito comuns na literatura: estado plano de tensão, estado plano de deformação e axi-simetria. O nosso objetivo inicial é dar um tratamento unificado para os dois primeiros casos: estado plano de tensão e estado plano de deformação. Problemas axi-simétricos serão tratados em um capítulo separado.

13.3.1 Estado plano de tensão - EPT

Assumirmos que o domínio que está sendo analisado possui uma dimensão muito menor do que as demais. Neste texto iremos adotar que a dimensão Z é muito menor do que as dimensões em X e Y e iremos adotar o conceito de espessura como sendo o comprimento na direção Z . Ainda, assumimos que os carregamentos que são aplicados no domínio atuem somente no plano XY , ou seja, não temos componentes de forças na direção Z . Com isso, podemos observar que as componentes de tensão σ_{xz} , σ_{yz} e σ_{zz} são nulas. Assim, considerando um material isotrópico, podemos escrever

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ 0 \\ 0 \\ 0 \\ \sigma_{xy} \end{pmatrix} = \begin{bmatrix} C_{xx} & C_{xy} & C_{xy} & 0 & 0 & 0 \\ C_{xy} & C_{xx} & C_{xy} & 0 & 0 & 0 \\ C_{xy} & C_{xy} & C_{xx} & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{yz} \\ 2\epsilon_{xz} \\ 2\epsilon_{xy} \end{pmatrix}. \quad (13.17)$$

Assim, observamos que pode existir deformação normal ϵ_{zz} mesmo que não tenhamos esforços nesta direção (devido ao Poisson). Utilizando a terceira equação (terceira linha) e isolando o ϵ_{zz} , obtemos

$$\epsilon_{zz} = -\frac{C_{xy}\epsilon_{xx} + C_{xy}\epsilon_{yy}}{C_{xx}} \quad (13.18)$$

tal que podemos simplificar as relações para calcularmos as tensões no plano em função das deformações no plano, obtendo

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \end{pmatrix} \quad (13.19)$$

tal que

$$\boxed{\mathbf{C}_{EPT}^{\nu} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}}. \quad (13.20)$$

Estado plano de deformação - EPD

Neste caso, a dimensão na direção Z é muito maior do que as dimensões no plano XY . No entanto, continuamos assumindo que os carregamentos são aplicados apenas no plano XY e são constantes ao longo de toda a direção Z . Com isso, agora observamos que as deformações ϵ_{zz} , ϵ_{xz} e ϵ_{yz} são nulas. Assim, considerando um material isotrópico, podemos escrever

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{pmatrix} = \begin{bmatrix} C_{xx} & C_{xy} & C_{xy} & 0 & 0 & 0 \\ C_{xy} & C_{xx} & C_{xy} & 0 & 0 & 0 \\ C_{xy} & C_{xy} & C_{xx} & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 0 \\ 0 \\ 0 \\ 2\epsilon_{xy} \end{pmatrix} \quad (13.21)$$

que permite obter a relação

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-2\nu \end{bmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \end{pmatrix}. \quad (13.22)$$

Assim,

$$\mathbf{C}_{EPD}^v = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-2\nu \end{bmatrix}. \quad (13.23)$$

Particularização da matriz \mathbf{L}^v para os estados planos

Podemos verificar que tanto em EPT quanto em EPD, precisamos apenas das deformações ϵ_{xx} , ϵ_{yy} e ϵ_{xy} (a deformação ϵ_{zz} pode ser pós-processada no caso EPT). Assim, podemos particularizar a matriz \mathbf{L}^v para

$$\mathbf{L}^v = \begin{bmatrix} \frac{d}{dx} & 0 \\ 0 & \frac{d}{dy} \\ \frac{d}{dy} & \frac{d}{dx} \end{bmatrix} \quad (13.24)$$

tal que

$$\boldsymbol{\epsilon}^v = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{d}{dx} & 0 \\ 0 & \frac{d}{dy} \\ \frac{d}{dy} & \frac{d}{dx} \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} = \mathbf{L}^v \mathbf{u}. \quad (13.25)$$

14. Elementos Finitos aplicados à Elasticidade

O objetivo deste capítulo é apresentar uma formulação geral para o método dos elementos finitos aplicados a problemas de elasticidade linear. Formulações específicas serão apresentadas nos próximos capítulos.

14.1 Método dos resíduos ponderados

Vimos que a equação diferencial que descreve o equilíbrio estático em um ponto de um sólido é

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}) + \mathbf{b}(\mathbf{x}) = \mathbf{0} \quad (14.1)$$

em que \mathbf{b} é o vetor de forças de corpo, com unidade $[N/m^3]$, e \mathbf{x} um ponto do domínio. Essa equação diferencial parcial de primeira ordem pode ser transformada em uma equação diferencial parcial de segunda ordem se utilizarmos a relação constitutiva

$$\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{C}(\mathbf{x}) : \boldsymbol{\varepsilon}(\mathbf{x}) \quad (14.2)$$

e o operador diferencial que relaciona deslocamentos e deformações

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \mathbf{L}(\mathbf{u}(\mathbf{x})) \quad (14.3)$$

tal que

$$\nabla \cdot (\mathbf{C}(\mathbf{x}) : \mathbf{L}(\mathbf{u}(\mathbf{x}))) + \mathbf{b}(\mathbf{x}) = \mathbf{0}. \quad (14.4)$$

A Eq. (14.4) é uma EDP de segunda ordem, com variável dependente \mathbf{u} e variáveis independentes as coordenadas de cada ponto do domínio Ω . Assim, a solução 'forte' deve satisfazer a Eq. (14.4) em **todos** os pontos (\mathbf{x}) do domínio.

Se utilizarmos a notação de Voigt, podemos reescrever a Eq. (14.4) como

$$(\mathbf{L}^v)^T (\mathbf{C}^v \mathbf{L}^v \mathbf{u}(\mathbf{x})) + \mathbf{b}(\mathbf{x}) = \mathbf{0}. \quad (14.5)$$

com Se propusermos uma solução $\tilde{\mathbf{u}}(\mathbf{x})$ iremos gerar um resíduo

$$\mathbf{r}(\mathbf{x}) = (\mathbf{L}^v)^T (\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}(\mathbf{x})) + \mathbf{b}(\mathbf{x}) \quad (14.6)$$

que, ao contrário dos exemplos estudados nos capítulos anteriores, é um vetor. Assim, vamos precisar definir um **vetor de funções testes** $\mathbf{w}(\mathbf{x})$. O produto interno no espaço das funções agora pode ser escrito como

$$\int_{\Omega} \mathbf{w}(\mathbf{x}) \cdot \mathbf{r}(\mathbf{x}) d\Omega = 0. \quad (14.7)$$

Se substituirmos a Eq. (14.6) na Eq. (14.7), obtemos

$$\int_{\Omega} \mathbf{w}(\mathbf{x}) \cdot \left((\mathbf{L}^v)^T (\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}(\mathbf{x})) + \mathbf{b}(\mathbf{x}) \right) d\Omega = 0 \quad (14.8)$$

ou

$$\int_{\Omega} \mathbf{w}(\mathbf{x}) \cdot \left((\mathbf{L}^v)^T (\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}(\mathbf{x})) \right) d\Omega = - \int_{\Omega} \mathbf{w}(\mathbf{x}) \cdot \mathbf{b}(\mathbf{x}) d\Omega. \quad (14.9)$$

Neste ponto, podemos notar que o o lado esquerdo da Eq. (14.9) pode ser integrado por partes para equalizar os requisitos de diferenciabilidade de $\mathbf{w}(\mathbf{x})$ e de $\tilde{\mathbf{u}}(\mathbf{x})$. Embora pareça algo bem mais complicado do que o utilizado em capítulos anteriores, podemos identificar a estrutura

$$\int_{\Omega} \underbrace{\mathbf{w}(\mathbf{x})}_{\mathbf{u}} \cdot \left(\left(\underbrace{\mathbf{L}^v}_D \right)^T \underbrace{(\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}(\mathbf{x}))}_{\mathbf{v}} \right) d\Omega = \int_{\Gamma} \mathbf{u} \cdot (\mathbf{v} \cdot \mathbf{n}) d\Gamma - \int_{\Omega} D(\mathbf{u}) \cdot \mathbf{v} d\Omega \quad (14.10)$$

obtemos

$$\int_{\Gamma} \mathbf{w}(\mathbf{x}) \cdot [(\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}(\mathbf{x})) \cdot \mathbf{n}] d\Gamma - \int_{\Omega} (\mathbf{L}^v \mathbf{w}(\mathbf{x})) \cdot (\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}(\mathbf{x})) d\Omega \quad (14.11)$$

Inserindo o resultado da integração por partes na Eq. (14.9), e deixando de indicar a dependência com \mathbf{x} para simplificar a notação, obtemos

$$\int_{\Omega} (\mathbf{L}^v \mathbf{w}) \cdot (\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}) d\Omega = \int_{\Omega} \mathbf{w} \cdot \mathbf{b} d\Omega + \int_{\Gamma} \mathbf{w} \cdot [(\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}) \cdot \mathbf{n}] d\Gamma. \quad (14.12)$$

Lembrando de mecânica do contínuo

$$(\mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}}) \cdot \mathbf{n} = \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t} \quad (14.13)$$

conhecido como vetor tração no contorno, associado a carregamento distribuído aplicado em algumas regiões do contorno, que chamaremos de Γ_t . Por fim, lembramos que o produto interno de dois vetores pode ser escrito como

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} \quad (14.14)$$

resultando em

$$\boxed{\int_{\Omega} (\mathbf{L}^v \mathbf{w})^T \mathbf{C}^v \mathbf{L}^v \tilde{\mathbf{u}} d\Omega = \int_{\Omega} \mathbf{w}^T \mathbf{b} d\Omega + \int_{\Gamma} \mathbf{w}^T \mathbf{t} d\Gamma.} \quad (14.15)$$

A Equação (14.15) é a base para a dedução dos elementos finitos para problemas de elasticidade, podendo ser aplicado tanto para problemas 3D quanto para problemas 2D. Enfatiza-se que a Eq. (14.15) considera todo o domínio Ω e todo o contorno Γ do problema.

14.2 Método dos elementos finitos

Utilizando a filosofia do método dos elementos finitos, podemos descrever os deslocamentos $\tilde{\mathbf{u}}$ e as funções \mathbf{w} em função de valores discretos em alguns pontos da estrutura. Assim, se definirmos um sub-domínio $\Omega_e \subseteq \Omega$ (elemento finito), podemos escrever

$$\tilde{\mathbf{u}}(\mathbf{x}) = \mathbf{N}(\mathbf{x})\mathbf{U}_e \quad (14.16)$$

e

$$\mathbf{w}(\mathbf{x}) = \mathbf{N}(\mathbf{x})\mathbf{W}_e \quad (14.17)$$

em que \mathbf{U}_e e \mathbf{W}_e são, respectivamente, valores de deslocamentos e da função de ponderação em pontos discretos de Ω_e (que chamaremos de nós).

A matriz $\mathbf{N}(\mathbf{x})$ é uma matriz contendo funções de interpolação, que devem ser definidas somente dentro de Ω_e , tal que

$$\mathbf{x} \cap \Omega_e = \emptyset \implies \mathbf{N}(\mathbf{x}) = \mathbf{0}. \quad (14.18)$$

A essa propriedade chamamos de **funções com suporte compacto**, ou seja, a função é nula em qualquer ponto que não esteja em Ω_e . Por fim, consideramos a hipótese de que o domínio de dois elementos adjacentes só tenham pontos de seus contornos em comum (os domínios de dois elementos não se sobrepõem).

Assim, com essas propriedades, podemos afirmar que

$$\int_{\Omega} f(\Omega) d\Omega = \sum_e \int_{\Omega_e} f_e(\Omega_e) d\Omega_e \quad (14.19)$$

onde $f_e(\Omega_e)$ são funções escalares com suporte compacto em Ω_e , ou partições de $f(\Omega)$.

Com essas definições, podemos reescrever a Eq. (14.15) como

$$\sum_e \int_{\Omega_e} (\mathbf{L}^v \mathbf{N} \mathbf{W}_e)^T \mathbf{C}^v \mathbf{L}^v \mathbf{N} \mathbf{U}_e d\Omega_e = \sum_e \int_{\Omega_e} (\mathbf{N} \mathbf{W}_e)^T \mathbf{b} d\Omega_e + \sum_e \int_{\Gamma_e} (\mathbf{N} \mathbf{W}_e)^T \mathbf{t} d\Gamma_e \quad (14.20)$$

ou seja, podemos estudar as integrais de um elemento genérico e . Operando com os termos, obtemos

$$\int_{\Omega_e} (\mathbf{L}^v \mathbf{N} \mathbf{W}_e)^T \mathbf{C}^v \mathbf{L}^v \mathbf{N} \mathbf{U}_e d\Omega_e = \mathbf{W}_e^T \int_{\Omega_e} (\mathbf{L}^v \mathbf{N})^T \mathbf{C}^v \mathbf{L}^v \mathbf{N} d\Omega_e \mathbf{U}_e = \mathbf{W}_e^T \underbrace{\int_{\Omega_e} \mathbf{B}^T \mathbf{C}^v \mathbf{B} d\Omega_e}_{\mathbf{K}_e} \mathbf{U}_e,$$

$$\int_{\Omega_e} (\mathbf{N} \mathbf{W}_e)^T \mathbf{b} d\Omega_e = \mathbf{W}_e^T \int_{\Omega_e} \mathbf{N}^T \mathbf{b} d\Omega_e = \mathbf{W}_e^T \mathbf{F}_{e_b} \quad (14.21)$$

e

$$\int_{\Gamma_e} (\mathbf{N} \mathbf{W}_e)^T \mathbf{t} d\Gamma_e = \mathbf{W}_e^T \int_{\Gamma_e} \mathbf{N}^T \mathbf{t} d\Gamma_e = \mathbf{W}_e^T \mathbf{F}_{e_t} \quad (14.22)$$

tal que, em um elemento e

$$\mathbf{W}_e^T (\mathbf{K}_e \mathbf{U}_e - \mathbf{F}_{e_b} - \mathbf{F}_{e_t}) = 0 \quad (14.23)$$

levando a um sistema de equações de equilíbrio para o elemento e

$$\mathbf{K}_e \mathbf{U}_e = \mathbf{F}_e. \quad (14.24)$$

Resumindo, as expressões necessárias para calcularmos o equilíbrio de um elemento finito de elasticidade linear são:

Matriz de rigidez

$$\mathbf{K}_e = \int_{\Omega_e} \mathbf{B}^T \mathbf{C}^v \mathbf{B} d\Omega_e, \quad (14.25)$$

vetor de forças de corpo

$$\mathbf{F}_{e_b} = \int_{\Omega_e} \mathbf{N}^T \mathbf{b} d\Omega_e \quad (14.26)$$

e vetor de forças de superfície

$$\mathbf{F}_{e_t} = \int_{\Gamma_e} \mathbf{N}^T \mathbf{t} d\Gamma_e. \quad (14.27)$$

14.3 Pós-processamento

Após obtermos o vetor global de deslocamentos \mathbf{U} , podemos retornar a cada elemento finito Ω_e da malha para realizarmos cálculos que tenham significado local, isto é, em $\mathbf{x} \in \Omega_e$. Exemplos de tais cálculos são a interpolação de deslocamentos no elemento

$$\mathbf{u}(\mathbf{x}) = \mathbf{N}(\mathbf{x})\mathbf{U}_e, \quad (14.28)$$

o cálculo das deformações no elemento

$$\boldsymbol{\varepsilon}^v(\mathbf{x}) = \mathbf{L}^v \mathbf{u}(\mathbf{x}) = \mathbf{L}^v \mathbf{N}(\mathbf{x})\mathbf{U}_e = \mathbf{B}(\mathbf{x})\mathbf{U}_e, \quad (14.29)$$

e o cálculo das tensões no elemento

$$\boldsymbol{\sigma}^v(\mathbf{x}) = \mathbf{C}^v \boldsymbol{\varepsilon}^v(\mathbf{x}) = \mathbf{C}^v \mathbf{B}(\mathbf{x})\mathbf{U}_e. \quad (14.30)$$

15. Elemento bilinear isoparamétrico de 4 nós

Seja um domínio bidimensional Ω e um subdomínio Ω_e delimitado por 4 pontos distintos (nós), conectados por lados retos (arestas). Assumindo que o comportamento no domínio é descrito pela equação de balanço da elasticidade linear, observamos que cada ponto possui dois graus de liberdade de translação, u_x e u_y .

Podemos organizar os valores discretos de deslocamentos de cada nó em um vetor \mathbf{U}_e . Como temos 4 nós e cada nó tem 2 graus de liberdade, teremos um total de 8 valores discretos (graus de liberdade). Vamos assumir que esses deslocamentos são agrupados por nó/gl em sequência, tal que U_1 é o deslocamento em X do nó 1, U_2 é o deslocamento em Y do nó 1, U_3 é o deslocamento em X do nó 2 ... U_8 é o deslocamento em Y do nó 4. Também vamos assumir que os nós são numerados no sentido anti-horário.

Recomenda-se a leitura do capítulo de integração numérica nas notas de aula de Fundamentos de Matemática, especialmente os métodos de quadratura.

15.1 Funções de interpolação

Vamos estudar a interpolação do deslocamento \tilde{u}_x em Ω_e . Como a nossa definição da geometria do elemento é bem geral, observamos que a proposta de funções de base para a interpolação pode ser bastante complicada. Por esse motivo, podemos considerar um mapeamento entre a geometria geral e uma geometria normalizada, de acordo com a Fig. 15.1.

A geometria normalizada é definida em um domínio $r \times s$, no intervalo $[-1, 1] \times [-1, 1]$. Esse sistema é colinear com o sistema XY , de tal forma que o mapeamento é relativo somente às dimensões, sem a necessidade de lidarmos com matrizes de rotação.

Utilizando essas variáveis, podemos escrever um **polinômio bilinear** em rs

$$\tilde{u}_x(r, s) = a_0 + a_1 r + a_2 s + a_3 rs \quad (15.1)$$

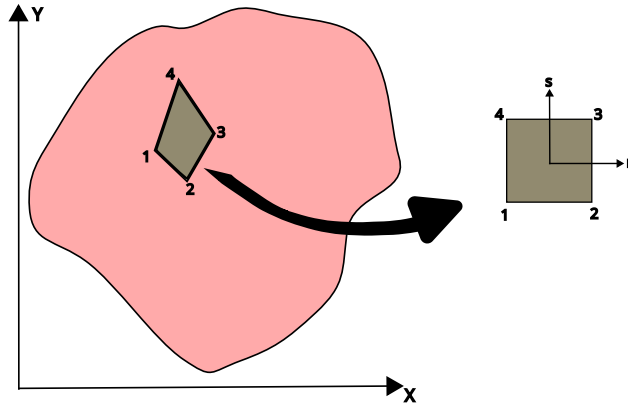


Figura 15.1: Mapeamento de domínio

e, como conhecemos os deslocamentos em X nas posições nodais, temos que

$$\text{nó 1} \quad \tilde{u}_x(-1, -1) = a_0 - a_1 - a_2 + a_3 = U_1 \quad (15.2)$$

$$\text{nó 2} \quad \tilde{u}_x(+1, -1) = a_0 + a_1 - a_2 - a_3 = U_3 \quad (15.3)$$

$$\text{nó 3} \quad \tilde{u}_x(+1, +1) = a_0 + a_1 + a_2 + a_3 = U_5 \quad (15.4)$$

$$\text{nó 4} \quad \tilde{u}_x(-1, +1) = a_0 - a_1 + a_2 - a_3 = U_7 \quad (15.5)$$

ou seja, temos um sistema com 4 equações e 4 incógnitas a_i . A solução desse sistema e reagrupamento dos termos, permite obter

$$\tilde{u}_x(r, s) = N_1(r, s)U_1 + N_2(r, s)U_3 + N_3(r, s)U_5 + N_4(r, s)U_7 \quad (15.6)$$

em que

$$N_1(r, s) = \frac{1}{4}(1 - r)(1 - s) \quad (15.7)$$

$$N_2(r, s) = \frac{1}{4}(1 + r)(1 - s) \quad (15.8)$$

$$N_3(r, s) = \frac{1}{4}(1 + r)(1 + s) \quad (15.9)$$

$$N_4(r, s) = \frac{1}{4}(1 - r)(1 + s) \quad (15.10)$$

são as funções de interpolação. Se assumirmos uma interpolação de mesma base para os deslocamentos em y ,

$$\tilde{u}_y(r, s) = b_0 + b_1 r + b_2 s + b_3 rs \quad (15.11)$$

e, como conhecemos os deslocamentos em Y nas posições nodais, temos que

$$\text{nó 1} \quad \tilde{u}_y(-1, -1) = b_0 - b_1 - b_2 + b_3 = U_2 \quad (15.12)$$

$$\text{nó 2} \quad \tilde{u}_y(+1, -1) = b_0 + b_1 - b_2 - b_3 = U_4 \quad (15.13)$$

$$\text{nó 3} \quad \tilde{u}_y(+1, +1) = b_0 + b_1 + b_2 + b_3 = U_6 \quad (15.14)$$

$$\text{nó 4} \quad \tilde{u}_y(-1, +1) = b_0 - b_1 + b_2 - b_3 = U_8 \quad (15.15)$$

Isso mostra que a estrutura das interpolações é a mesma (compartilham da mesma base), de tal forma que podemos escrever

$$\tilde{u}_y(r, s) = N_1(r, s)U_2 + N_2(r, s)U_4 + N_3(r, s)U_6 + N_4(r, s)U_8. \quad (15.16)$$

Agrupando então as duas interpolações, podemos escrever

$$\begin{Bmatrix} \tilde{u}_x(r,s) \\ \tilde{u}_y(r,s) \end{Bmatrix} \begin{bmatrix} N_1(r,s) & 0 & N_2(r,s) & 0 & N_3(r,s) & 0 & N_4(r,s) & 0 \\ 0 & N_1(r,s) & 0 & N_2(r,s) & 0 & N_3(r,s) & 0 & N_4(r,s) \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \\ U_8 \end{Bmatrix}$$

ou

$$\tilde{\mathbf{u}}(r,s) = \mathbf{N}(r,s) \mathbf{U}_e \quad (15.17)$$

e chamamos $\mathbf{N}(r,s)$ de matriz das funções de interpolação. Como estamos assumindo o mesmo conjunto de bases para as funções de ponderação, podemos escrever

$$\mathbf{w}(r,s) = \mathbf{N}(r,s) \mathbf{W}_e. \quad (15.18)$$

15.2 Cálculo das derivadas e mapeamento isoparamétrico

Com a interpolação definida, devemos proceder com o cálculo dos termos das integrais. A matriz de rigidez de um elemento finito é dada por

$$\mathbf{K}_e = \int_{\Omega_e} \mathbf{B}^T \mathbf{C}^v \mathbf{B} d\Omega_e \quad (15.19)$$

com

$$\mathbf{B}(r,s) = \mathbf{L}^v \mathbf{N}(r,s) = \begin{bmatrix} \frac{d}{dx} & 0 \\ 0 & \frac{d}{dy} \\ \frac{d}{dy} & \frac{d}{dx} \end{bmatrix} \mathbf{N}(r,s) = \begin{bmatrix} \frac{dN_1}{dx} & 0 & \dots & \frac{dN_4}{dx} & 0 \\ 0 & \frac{dN_1}{dy} & \dots & 0 & \frac{dN_4}{dy} \\ \frac{dN_1}{dy} & \frac{dN_1}{dx} & \dots & \frac{dN_4}{dy} & \frac{dN_4}{dx} \end{bmatrix}_{3 \times 8} (r,s) \quad (15.20)$$

de tal forma que temos derivadas em relação a X e Y de funções definidas em r e s . Utilizando a regra da cadeia para uma função de interpolação N_i

$$\frac{dN_i(r,s)}{dr} = \frac{dN_i(r,s)}{dx} \frac{dx}{dr} + \frac{dN_i(r,s)}{dy} \frac{dy}{dr} \quad (15.21)$$

e

$$\frac{dN_i(r,s)}{ds} = \frac{dN_i(r,s)}{dx} \frac{dx}{ds} + \frac{dN_i(r,s)}{dy} \frac{dy}{ds} \quad (15.22)$$

que podem ser agrupadas em uma forma matricial

$$\begin{Bmatrix} \frac{dN_i(r,s)}{dr} \\ \frac{dN_i(r,s)}{ds} \end{Bmatrix} = \begin{bmatrix} \frac{dx}{dr} & \frac{dy}{dr} \\ \frac{dx}{ds} & \frac{dy}{ds} \end{bmatrix} \begin{Bmatrix} \frac{dN_i(r,s)}{dx} \\ \frac{dN_i(r,s)}{dy} \end{Bmatrix} \quad (15.23)$$

em que a matriz de derivadas é chamada de **matriz Jacobiana** da transformação, ou **J**. Como pode ser verificado, a matriz Jacobiana relaciona diferenciais nos dois sistemas de referência, sendo que cada coluna da matriz é um gradiente (taxa de variação de x e y nas

direções de r e s). Assim, se realizarmos um produto vetorial desses vetores gradientes (que estão no plano)

$$\left\{ \frac{dx}{dr} \right\} \times \left\{ \frac{dy}{ds} \right\} = \left(\frac{dx}{dr} \frac{dy}{ds} - \frac{dx}{ds} \frac{dy}{dr} \right) \mathbf{k} \quad (15.24)$$

obtemos um vetor perpendicular ao plano, cuja magnitude tem o significado de área. Essa magnitude é igual ao **determinante** da matriz \mathbf{J} , que iremos indicar por $|\mathbf{J}|$ ao longo deste texto. Como a matriz \mathbf{J} relaciona diferenciais no plano, observa-se que o mapeamento entre os diferenciais de área é

$$dA^{(xy)} = |\mathbf{J}| dA^{(rs)}. \quad (15.25)$$

É muito importante observar que, devido ao seu significado geométrico de mapeamento de áreas, não podemos ter um determinante nulo e/ou negativo.

A relação entre os diferenciais das funções de interpolação, Eq. (15.23), será denotada em uma forma compacta por

$$d\mathbf{N}_i^{(rs)}(r, s) = \mathbf{J}(r, s) d\mathbf{N}_i^{(xy)}(r, s) \quad (15.26)$$

e a relação inversa resulta em

$$d\mathbf{N}_i^{(xy)}(r, s) = \mathbf{J}^{-1}(r, s) d\mathbf{N}_i^{(rs)}(r, s). \quad (15.27)$$

Assim, para calcularmos a matriz $\mathbf{B}(r, s)$, precisamos calcular as derivadas das funções de interpolação em relação a rs , obter a matriz \mathbf{J} e sua inversa e realizar as correções das derivadas, de acordo com a Eq. (15.27).

No entanto, os termos da matriz Jacobiana são derivadas das coordenadas x e y dentro de Ω_e em relação as coordenadas normalizadas r e s , mas não temos expressões de $x(r, s)$ e $y(r, s)$ para calcular as derivadas. A solução para esse problema foi proposta por Bruce Irons e Iang Taig no começo da década de 70, chamado de **mapeamento isoparamétrico** (mesma parametrização). A ideia é interpolar a geometria do elemento a partir das coordenadas nodais utilizando o mesmo conjunto de funções de interpolação do campo de deslocamento, tal que

$$x(r, s) = \sum_{i=1}^4 N_i(r, s) X_i \quad (15.28)$$

e

$$y(r, s) = \sum_{i=1}^4 N_i(r, s) Y_i \quad (15.29)$$

onde X_i e Y_i são, respectivamente, as coordenadas nodais dos nós do elemento. Com essa definição, podemos calcular os termos da matriz Jacobiana, tal que

$$\frac{dx}{dr} = \sum_{i=1}^4 \frac{dN_i(r, s)}{dr} X_i, \quad (15.30)$$

$$\frac{dy}{dr} = \sum_{i=1}^4 \frac{dN_i(r, s)}{dr} Y_i, \quad (15.31)$$

$$\frac{dx}{ds} = \sum_{i=1}^4 \frac{dN_i(r,s)}{ds} X_i, \quad (15.32)$$

e

$$\frac{dy}{ds} = \sum_{i=1}^4 \frac{dN_i(r,s)}{ds} Y_i \quad (15.33)$$

resultando em

$$\mathbf{J}(r,s) = \begin{bmatrix} \sum_{i=1}^4 \frac{dN_i(r,s)}{dr} X_i & \sum_{i=1}^4 \frac{dN_i(r,s)}{dr} Y_i \\ \sum_{i=1}^4 \frac{dN_i(r,s)}{ds} X_i & \sum_{i=1}^4 \frac{dN_i(r,s)}{ds} Y_i \end{bmatrix} \quad (15.34)$$

ou, desenvolvendo as derivadas e os somatórios

$$\mathbf{J}(r,s) = \begin{bmatrix} \frac{s(X_1+X_3-X_2-X_4)+(X_2+X_3-X_1-X_4)}{4} & \frac{s(Y_1+Y_3-Y_2-Y_4)+(Y_2+Y_3-Y_1-Y_4)}{4} \\ \frac{r(X_1+X_3-X_2-X_4)+(X_4+X_3-X_2-X_1)}{4} & \frac{r(Y_1+Y_3-Y_4-Y_2)+(Y_4+Y_3-Y_2-Y_1)}{4} \end{bmatrix} \quad (15.35)$$

■ **Exemplo 15.1** Considere um elemento com coordenadas $(0,0)$, $(1,0)$, $(1,1)$ e $(0,1)$ para os nós 1, 2, 3 e 4, respectivamente. Como o elemento é um quadrado no domínio original, observamos que a matriz Jacobiana se torna

$$\mathbf{J} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \quad (15.36)$$

independente de (r,s) , com determinante $|\mathbf{J}| = \frac{1}{4}$. Só temos termos na diagonal da matriz, pois ocorre somente um 'alongamento' dos comprimentos em X e Y , que são 1, para os comprimentos em r e s , que são 2. Esses elementos diferenciais de comprimento não se inclinam, de tal forma que os termos fora da diagonal são nulos. O determinante da matriz Jacobiana nos mostra que a relação entre a área em XY , $1m^2$ e a área em rs , que é sempre $4m^2$.

■

■ **Exemplo 15.2** Considere um elemento com coordenadas $(0,0)$, $(4,1)$, $(7,4)$ e $(2,3)$ para os nós 1, 2, 3 e 4, respectivamente. A matriz Jacobiana resulta em

$$\mathbf{J}(r,s) = \begin{bmatrix} \frac{9+s}{4} & \frac{1}{2} \\ \frac{5+r}{4} & \frac{3}{2} \end{bmatrix} \quad (15.37)$$

com determinante

$$|\mathbf{J}(r,s)| = \frac{22+2s-r}{8}. \quad (15.38)$$

Isso mostra que para uma geometria original 'distorcida', teremos um mapeamento que depende da posição.

■

Exercício 15.1 Interprete os resultados do exemplo anterior.

■

Exercício 15.2 Seja um elemento quadrado com lados de $1m$. O que ocorre com o determinante da matriz Jacobiana da transformação se organizarmos os nós no sentido horário? Interprete. ■

Tendo definido o procedimento de cálculo para obtermos $\mathbf{B}(r,s)$, podemos proceder com o cálculo da integral no domínio do elemento, Eq. (15.19).

15.3 Cálculo das integrais de domínio

Como podemos ver na Fig. 15.1, a integral em Ω_e deve ser realizada na configuração original do elemento e não na mapeada. No entanto, todas as nossas grandezas estão descritas em $r \times s \in [-1, 1] \times [-1, 1]$. Na verdade, o fato de termos realizado o mapeamento não só facilita a definição das funções de interpolação como também permite a utilização de métodos para integração numérica, como os descritos nas notas de aula de Fundamentos de Matemática - Métodos para Integração Numérica.

Iniciamos observando que é possível reescrever a integral no domínio Ω_e no domínio rs , desde que seja considerada uma correção dos diferenciais, conforme a Eq. (15.25)

$$\mathbf{K}_e = \int_{\Omega_e} \mathbf{B}(r,s)^T \mathbf{C}^v \mathbf{B}(r,s) d\Omega_e = t_e \int_{-1}^1 \int_{-1}^1 \mathbf{B}(r,s)^T \mathbf{C}^v \mathbf{B}(r,s) |\mathbf{J}(r,s)| dr ds \quad (15.39)$$

e observamos que a integral de domínio é relativa ao volume do elemento, mas a espessura t_e é constante, de tal forma que precisamos calcular uma integral bidimensional (área).

A integral pode ser calculada de diversas formas, mas lembramos que se o integrando for polinomial (ou, no caso, uma matriz de termos polinomiais), a utilização da **quadratura de Gauss-Legendre** é a abordagem mais econômica (menor número de avaliações do núcleo da integral). Desta forma, devemos avaliar o núcleo da integral, que é composta do produto $\mathbf{B}^T \mathbf{C}^v \mathbf{B} |\mathbf{J}|$. Se cada posição desta matriz for formada por polinômios, devemos descobrir o grau destes para especificar o número de pontos de integração. Para facilitar a análise, vamos utilizar um elemento quadrado com lado unitário e material com $E = 1$ e $\nu = 0$. Neste caso, o primeiro termo do núcleo da integral tem a forma (os demais mantêm uma estrutura similar ou de grau menor)

$$N_{11}(r,s) = \frac{2s^2 - 4s + r^2 - 2r + 3}{32} \quad (15.40)$$

que é um polinômio quadrático em r e em s . A integral analítica desta função é

$$\int_{-1}^1 \int_{-1}^1 N_{11}(r,s) dr ds = 0.5 \quad (15.41)$$

e, utilizando uma quadratura 2×2 ,

$$N_{11}\left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) + N_{11}\left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) + N_{11}\left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) + N_{11}\left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) = 0.5. \quad (15.42)$$

ou seja, integramos a expressão corretamente.

Se modificarmos as coordenadas dos nós do elemento para $(0,0)$, $(1,0.5)$, $(2,1)$ e $(0.5,0.5)$, temos que o determinante da matriz Jacobiana é

$$|\mathbf{J}| = \frac{0.25s - 0.25r + 0.5}{8} \quad (15.43)$$

que é nulo quando $r = 1$ e $s = -1$, indicando que o mapeamento não é válido ao longo de todo o domínio do elemento. De fato, para este caso,

$$N_{11}(r, s) = \frac{12s^2 + (8 - 32r)s + 22r^2 - 12r + 2}{s - r + 2} \quad (15.44)$$

que não é definido quando $r = 1$ e $s = -1$ (o denominador vai para 0). A integral analítica de N_{11} é de 41.81 mas a integral obtida por quadratura 2×2 resulta em 28.66, um erro relativo de 31%.

Isso mostra que a distorção do elemento implica em um núcleo não polinomial e que pode apresentar descontinuidades ao longo do domínio, violando as hipóteses do procedimento de integração numérica.

As mesmas observações podem ser realizadas para o termo de forças de corpo, Eq. (14.26)

$$\mathbf{F}_{eb} = \int_{\Omega_e} \mathbf{N}^T \mathbf{b} d\Omega_e = t_e \int_{-1}^1 \int_{-1}^1 \mathbf{N}^T(r, s) \mathbf{b}(r, s) |\mathbf{J}(r, s)| dr ds \quad (15.45)$$

e precisamos de uma expressão para $\mathbf{b}(r, s)$. Como no método dos elementos finitos descrevemos todas as grandezas a partir dos nós, podemos assumir que o valor em um ponto (r, s) do domínio pode ser obtido por uma interpolação bilinear de valores nodais, tal que

$$\begin{Bmatrix} b_x(r, s) \\ b_y(r, s) \end{Bmatrix} = \mathbf{N}(r, s) \begin{Bmatrix} Bx_1 \\ By_1 \\ Bx_2 \\ By_2 \\ Bx_3 \\ By_3 \\ Bx_4 \\ By_4 \end{Bmatrix} = \mathbf{N}(r, s) \mathbf{b}_e \quad (15.46)$$

em que, por exemplo, Bx_1 é o valor da força de corpo na direção X , nó 1 e By_3 é o valor da força de corpo na direção Y do nó 3. Um caso particular e muito comum é termos a força da gravidade na direção Y negativa, tal que

$$\mathbf{b}_e = -g\rho \begin{Bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \quad [N/m^3] \quad (15.47)$$

em que g [m/s^2] é a aceleração da gravidade e ρ [kg/m^3] a massa específica. Neste caso, com um elemento quadrado de lado unitário, temos que os termos do núcleo da integral da força de corpo tem a forma

$$-g\rho \frac{(r-1)s + (1-r)}{16} \quad (15.48)$$

tal que podem ser integrados pela mesma regra de quadratura utilizada para a matriz de rigidez.

15.4 Cálculo das integrais de contorno

O elemento tem 4 lados (arestas) e as integrais de contorno são relativas a cada uma das arestas. Vamos numerar as arestas seguindo a mesma lógica da numeração nodal, tal que

- Aresta 1) nó 1 ao nó 2;
- Aresta 2) nó 2 ao nó 3;
- Aresta 3) nó 3 ao nó 4;
- Aresta 4) nó 4 ao nó 1

conforme ilustrado na Fig. 15.2

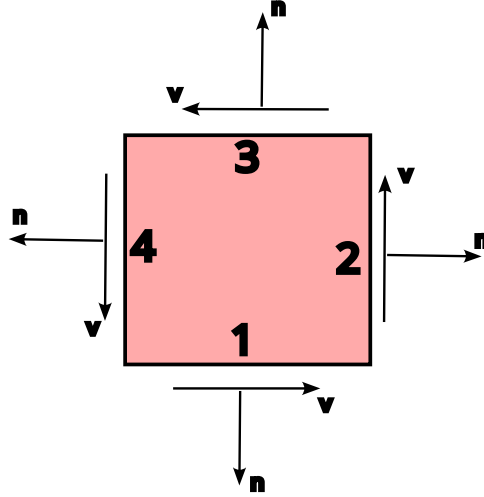


Figura 15.2: Definição das arestas de um elemento bilinear isoparamétrico de 4 nós e de seus sistemas locais.

O carregamento distribuído \mathbf{t}_i [N/m^2] aplicado em uma aresta i pode ser informado no sistema global XY ou em um sistema local $\mathbf{n} \times \mathbf{v}$ da aresta (muito comum quando estamos aplicando pressão a uma superfície). Irei utilizar a letra \mathbf{v} para denotar o vetor tangencial à aresta, além do tradicional \mathbf{n} para denotar o vetor normal à aresta.

Vamos estudar com cuidado o tratamento a ser dado a cada aresta.

15.4.1 Aresta 1

Para entendermos o procedimento, vamos estudar como realizar a integral na aresta 1. Começamos notando que no domínio mapeado rs a aresta tem coordenada fixa $s = -1$, de tal forma que as funções de interpolação relativa aos nós 3 e 4 serão sempre nulas. Do mapeamento isoparamétrico, lembramos que

$$x(r, -1) = N_1(r, -1)X_1 + N_2(r, -1)X_2 \quad (15.49)$$

$$y(r, -1) = N_1(r, -1)Y_1 + N_2(r, -1)Y_2 \quad (15.50)$$

ou

$$x(r) = \frac{1-r}{2}X_1 + \frac{1+r}{2}X_2 \quad (15.51)$$

$$y(r) = \frac{1-r}{2}Y_1 + \frac{1+r}{2}Y_2 \quad (15.52)$$

que nada mais são do que interpolações lineares em cada coordenada. Com isso, podemos definir um vetor tangencial à aresta 1 como

$$\left\{ \frac{dx(r)}{dr} \right\} = \left\{ \frac{X_2 - X_1}{2} \right\} \quad (15.53)$$

com norma

$$\frac{1}{2} \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} = \frac{1}{2} L^1, \quad (15.54)$$

em que L^1 é o comprimento da aresta 1.

Por questões de notação, vamos definir

$$\Delta X_{21} = X_2 - X_1 \quad (15.55)$$

$$\Delta Y_{21} = Y_2 - Y_1 \quad (15.56)$$

Com isso, normalizando o vetor para que fique unitário

$$\mathbf{v}^1 = \frac{1}{\sqrt{\Delta X_{21}^2 + \Delta Y_{21}^2}} \begin{Bmatrix} \Delta X_{21} \\ \Delta Y_{21} \end{Bmatrix}. \quad (15.57)$$

Sabendo que o problema é 2D (temos um vetor obrigatoriamente na direção \mathbf{k}), podemos obter o vetor normal à aresta pelo produto vetorial

$$\mathbf{n}^1 \times \mathbf{v}^1 = \mathbf{k} \implies \mathbf{n}^1 = -\mathbf{k} \times \mathbf{v}^1 \quad (15.58)$$

tal que

$$\mathbf{n}^1 = - \begin{vmatrix} i & j & k \\ 0 & 0 & 1 \\ v_x & v_y & 0 \end{vmatrix} = v_y i - v_x j. \quad (15.59)$$

Assim,

$$\mathbf{n}^1 = \frac{1}{\sqrt{\Delta X_{21}^2 + \Delta Y_{21}^2}} \begin{Bmatrix} \Delta Y_{21} \\ -\Delta X_{21} \end{Bmatrix}. \quad (15.60)$$

■ **Exemplo 15.3** Seja a aresta 1 de um elemento bilinear isoparamétrico de 4 nós. Assuma que as coordenadas dos nós 1 e 2 sejam (0,0) e (1,0), respectivamente. Neste caso,

$$\Delta X_{21} = 1 - 0 = 1 \quad (15.61)$$

$$\Delta Y_{21} = 0 - 0 = 0 \quad (15.62)$$

tal que

$$\mathbf{v}^1 = \frac{1}{\sqrt{1^2 + 0^2}} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \quad (15.63)$$

que está correto. Da mesma forma,

$$\mathbf{n}^1 = \frac{1}{\sqrt{1^2 + 0^2}} \begin{Bmatrix} 0 \\ -1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -1 \end{Bmatrix} \quad (15.64)$$

ou seja, a normal 'aponta para baixo', o que está correto (ela deve apontar para 'fora' do elemento). ■

■ **Exemplo 15.4** Seja a aresta 1 de um elemento bilinear isoparamétrico de 4 nós. Assuma que as coordenadas dos nós 1 e 2 sejam $(0,0)$ e $(1,1)$, respectivamente. Neste caso,

$$\Delta X_{21} = 1 - 0 = 1 \quad (15.65)$$

$$\Delta Y_{21} = 1 - 0 = 1 \quad (15.66)$$

$$(15.67)$$

tal que

$$\mathbf{v}^1 = \frac{1}{\sqrt{1^2 + 1^2}} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} = \frac{1}{\sqrt{2}} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \quad (15.68)$$

que está correto, pois é um vetor que está inclinado com a aresta (45°) . Da mesma forma,

$$\mathbf{n}^1 = \frac{1}{\sqrt{1^2 + 1^2}} \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} = \frac{1}{\sqrt{2}} \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} \quad (15.69)$$

e novamente notamos que a normal aponta para 'fora' da aresta (-45°) . ■

Com o sistema de referência da aresta bem definido, podemos nos preocupar com o cálculo da integral ao longo da aresta. Novamente, queremos utilizar um procedimento de integração por quadratura, de tal forma que a integral deve ser escrita como

$$t_e \int_{\Gamma^1} \mathbf{N}^T \mathbf{t} |\mathbf{J}^1| dr \quad (15.70)$$

em que a correção do diferencial agora é uma relação entre o comprimento da aresta no domínio original e o comprimento da aresta no domínio mapeado, que é sempre 2. Aqui, estamos utilizando a notação \mathbf{J}^1 para representar o mapeamento da aresta, que não é o mesmo do domínio completo. De fato, se integrarmos uma constante no domínio original

$$\int_0^{L^1} \alpha d\zeta = \alpha L^1 \quad (15.71)$$

e, no domínio mapeado (aresta)

$$\int_{-1}^1 \alpha |\mathbf{J}^1| dr = 2\alpha |\mathbf{J}^1| \quad (15.72)$$

tal que

$$|\mathbf{J}^1| = \frac{L^1}{2}. \quad (15.73)$$

Com isso, podemos identificar que

$$|\mathbf{J}^1| = \frac{1}{2} \sqrt{\Delta X_{21}^2 + \Delta Y_{21}^2} \quad (15.74)$$

ou seja, é a própria Eq. (15.54).

Com isso, assumindo que o vetor \mathbf{t}^1 contém valores constantes de trações no contorno Γ^1 , podemos calcular a integral numericamente como

$$\int_{-1}^1 \mathbf{N}^T(r, -1) \mathbf{t}^1 |\mathbf{J}^1| dr. \quad (15.75)$$

Outra opção de entrada de dados é informar um escalar de pressão (normal) à superfície, $p(r)$ $[N/m^2]$, tal que

$$\int_{-1}^1 \mathbf{N}^T(r, -1) p(r) \mathbf{n}^1 |\mathbf{J}^1| dr \quad (15.76)$$

sendo que podemos fazer o mesmo com um carregamento que atue na direção tangencial, como um atrito.

Ambas operações irão gerar um vetor 8×1 com valores não nulos somente nos gls associados aos nós da aresta 1.

15.4.2 Aresta 2

Essa aresta, que liga os nós 2 e 3, tem coordenada fixa $r = 1$. Assim, as funções de interpolação relativa aos nós 1 e 4 serão sempre nulas. Do mapeamento isoparamétrico, lembramos que

$$x(1, s) = N_2(1, s)X_2 + N_3(1, s)X_3 \quad (15.77)$$

$$y(1, s) = N_2(1, s)Y_2 + N_3(1, s)Y_3 \quad (15.78)$$

ou

$$x(s) = \frac{1-s}{2}X_2 + \frac{1+s}{2}X_3 \quad (15.79)$$

$$y(s) = \frac{1-s}{2}Y_2 + \frac{1+s}{2}Y_3 \quad (15.80)$$

que nada mais são do que interpolações lineares em cada coordenada. Com isso, podemos definir um vetor tangencial à aresta 2 como

$$\left\{ \frac{dx(s)}{ds}, \frac{dy(s)}{ds} \right\} = \left\{ \frac{X_3 - X_2}{2}, \frac{Y_3 - Y_2}{2} \right\} \quad (15.81)$$

com norma

$$\frac{1}{2} \sqrt{\Delta X_{32}^2 + \Delta Y_{32}^2} \quad (15.82)$$

e, normalizando o vetor para que fique unitário

$$\mathbf{v}^2 = \frac{1}{\sqrt{\Delta X_{32}^2 + \Delta Y_{32}^2}} \begin{Bmatrix} \Delta X_{32} \\ \Delta Y_{32} \end{Bmatrix}. \quad (15.83)$$

Lembrando o que fizemos na face 1, podemos obter o vetor normal à aresta pelo produto vetorial

$$\mathbf{n}^2 \times \mathbf{v}^2 = \mathbf{k} \implies \mathbf{n}^2 = -\mathbf{k} \times \mathbf{v}^2 \quad (15.84)$$

tal que

$$\mathbf{n}^2 = \frac{1}{\sqrt{\Delta X_{32}^2 + \Delta Y_{32}^2}} \begin{Bmatrix} \Delta Y_{32} \\ -\Delta X_{32} \end{Bmatrix} \quad (15.85)$$

e

$$|\mathbf{J}^2| = \frac{1}{2} \sqrt{\Delta X_{32}^2 + \Delta Y_{32}^2}. \quad (15.86)$$

■ **Exemplo 15.5** Seja a aresta 2 de um elemento bilinear isoparamétrico de 4 nós. Assuma que as coordenadas dos nós 2 e 3 sejam $(1,0)$ e $(1,1)$, respectivamente. Neste caso,

$$\Delta X_{32} = 1 - 1 = 0 \quad (15.87)$$

$$\Delta Y_{32} = 1 - 0 = 1 \quad (15.88)$$

tal que

$$\mathbf{v}^2 = \frac{1}{\sqrt{0^2 + 1^2}} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \quad (15.89)$$

que está correto, pois é um vetor que aponta para cima. Da mesma forma,

$$\mathbf{n}^2 = \frac{1}{\sqrt{0^2 + 1^2}} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \quad (15.90)$$

que é um vetor que aponta para a direita (para fora do elemento), o que está correto. ■

15.4.3 Aresta 3

Essa aresta, que liga os nós 4 e 3, tem coordenada fixa $s = 1$. Assim, as funções de interpolação relativa aos nós 1 e 2 serão sempre nulas.

Aqui devemos ter um cuidado com a definição do vetor tangente, que é uma derivada em relação à coordenada r . No entanto, como a coordenada r é positiva para a direita, termos um vetor tangente nessa direção e um vetor normal apontando para dentro do elemento.

Para evitarmos esse problema, podemos realizar um mapeamento 'invertido'

$$x(r, 1) = N_4(r, 1)X_3 + N_3(r, 1)X_4 \quad (15.91)$$

$$y(r, 1) = N_4(r, 1)Y_3 + N_3(r, 1)Y_4 \quad (15.92)$$

ou

$$x(r) = \frac{1-r}{2}X_3 + \frac{1+r}{2}X_4 \quad (15.93)$$

$$y(r) = \frac{1-r}{2}Y_3 + \frac{1+r}{2}Y_4 \quad (15.94)$$

Com isso, podemos definir um vetor tangencial à aresta 3 como

$$\left\{ \frac{dx(r)}{dr}, \frac{dy(r)}{dr} \right\} = \left\{ \frac{X_4 - X_3}{2}, \frac{Y_4 - Y_3}{2} \right\} \quad (15.95)$$

com norma

$$\frac{1}{2} \sqrt{\Delta X_{43}^2 + \Delta Y_{43}^2} \quad (15.96)$$

e, normalizando o vetor para que fique unitário

$$\mathbf{v}^3 = \frac{1}{\sqrt{\Delta X_{43}^2 + \Delta Y_{43}^2}} \begin{Bmatrix} \Delta X_{43} \\ \Delta Y_{43} \end{Bmatrix}. \quad (15.97)$$

Lembrando o que fizemos nas faces anteriores, podemos obter o vetor normal à aresta pelo produto vetorial

$$\mathbf{n}^3 \times \mathbf{v}^3 = \mathbf{k} \implies \mathbf{n}^3 = -\mathbf{k} \times \mathbf{v}^3 \quad (15.98)$$

tal que

$$\mathbf{n}^3 = \frac{1}{\sqrt{\Delta X_{43}^2 + \Delta Y_{43}^2}} \begin{Bmatrix} \Delta Y_{43} \\ -\Delta X_{43} \end{Bmatrix} \quad (15.99)$$

e

$$|\mathbf{J}^3| = \frac{1}{2} \sqrt{\Delta X_{43}^2 + \Delta Y_{43}^2}. \quad (15.100)$$

■ **Exemplo 15.6** Seja a aresta 3 de um elemento bilinear isoparamétrico de 4 nós. Assuma que as coordenadas dos nós 3 e 4 sejam (1,1) e (0,1), respectivamente. Neste caso,

$$\Delta X_{43} = 0 - 1 = -1 \quad (15.101)$$

$$\Delta Y_{43} = 1 - 1 = 0 \quad (15.102)$$

tal que

$$\mathbf{v}^3 = \frac{1}{\sqrt{(-1)^2 + 0^2}} \begin{Bmatrix} -1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 0 \end{Bmatrix} \quad (15.103)$$

que é um vetor que aponta para a esquerda.

Com isso,

$$\mathbf{n}^3 = \frac{1}{\sqrt{(-1)^2 + 0^2}} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \quad (15.104)$$

é um vetor que aponta para cima, ou seja, para fora do elemento. ■

15.4.4 Aresta 4

Essa aresta, que liga os nós 4 e 1, tem coordenada fixa $r = -1$. Assim, as funções de interpolação relativa aos nós 2 e 3 serão sempre nulas.

Aqui cabe o mesmo comentário da aresta 3. Para obtermos uma orientação normal para fora do elemento, devemos utilizar

$$x(-1, s) = N_1(-1, s)X_4 + N_4(-1, s)X_1 \quad (15.105)$$

$$y(-1, s) = N_1(-1, s)Y_4 + N_4(-1, s)Y_1 \quad (15.106)$$

ou

$$x(s) = \frac{1-s}{2}X_4 + \frac{1+s}{2}X_1 \quad (15.107)$$

$$y(s) = \frac{1-s}{2}Y_4 + \frac{1+s}{2}Y_1. \quad (15.108)$$

Com isso, podemos definir um vetor tangencial à aresta 4 como

$$\begin{Bmatrix} \frac{dx(s)}{ds} \\ \frac{dy(s)}{ds} \end{Bmatrix} = \begin{Bmatrix} \frac{X_1 - X_4}{2} \\ \frac{Y_1 - Y_4}{2} \end{Bmatrix} \quad (15.109)$$

com norma

$$\frac{1}{2} \sqrt{\Delta X_{14}^2 + \Delta Y_{14}^2} \quad (15.110)$$

e, normalizando o vetor para que fique unitário

$$\mathbf{v}^4 = \frac{1}{\sqrt{\Delta X_{14}^2 + \Delta Y_{14}^2}} \begin{Bmatrix} \Delta X_{14} \\ \Delta Y_{14} \end{Bmatrix}. \quad (15.111)$$

Novamente, podemos calcular a normal como

$$\mathbf{n}^4 \times \mathbf{v}^4 = \mathbf{k} \implies \mathbf{n}^4 = -\mathbf{k} \times \mathbf{v}^4 \quad (15.112)$$

tal que

$$\mathbf{n}^4 = \frac{1}{\sqrt{\Delta X_{14}^2 + \Delta Y_{14}^2}} \begin{Bmatrix} \Delta Y_{14} \\ -\Delta X_{14} \end{Bmatrix} \quad (15.113)$$

e

$$|\mathbf{J}^4| = \frac{1}{2} \sqrt{\Delta X_{14}^2 + \Delta Y_{14}^2}. \quad (15.114)$$

■ **Exemplo 15.7** Seja a aresta 4 de um elemento bilinear isoparamétrico de 4 nós. Assuma que as coordenadas dos nós 4 e 1 sejam (0,1) e (0,0), respectivamente. Neste caso,

$$\Delta X_{14} = 0 - 0 = 0 \quad (15.115)$$

$$\Delta Y_{14} = 0 - 1 = -1 \quad (15.116)$$

tal que

$$\mathbf{v}^4 = \frac{1}{\sqrt{0^2 + (-1)^2}} \begin{Bmatrix} 0 \\ -1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -1 \end{Bmatrix} \quad (15.117)$$

que aponta para baixo. O vetor normal à face é obtido com

$$\mathbf{n}^4 = \frac{1}{\sqrt{0^2 + (-1)^2}} \begin{Bmatrix} -1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 0 \end{Bmatrix} \quad (15.118)$$

que aponta para a esquerda, ou seja, para fora do elemento. ■

Exercício 15.3 Obtenha o vetor de forças de superfície em um elemento quadrado de 1 m de lado e espessura t_e , submetido a

- um vetor \mathbf{t} [N/m^2] com ambas as componentes não nulas;
- a uma pressão normal p_0 [N/m^2] constante
- um atrito q_0 [N/m^2] constante

em cada uma das arestas. Interprete os resultados. ■

Exercício 15.4 É possível observar um padrão em comum para os resultados obtidos para as diferentes faces. De fato, se definirmos indicadores p_1 e p_2 , podemos representar os resultados de dJ^i , \mathbf{v}^i e \mathbf{n}^i apenas em função da diferenças de coordenadas nestes dois indicadores, para qualquer face i . Investigue essas relações. ■

15.5 Cálculo de tensões

Já vimos que o estado de tensões em um ponto de um elemento finito pode ser calculado em uma etapa de pós-processamento, por meio da expressão

$$\boldsymbol{\sigma}^v(r, s) = \mathbf{C}^v \underbrace{\mathbf{B}(r, s) \mathbf{U}_e}_{\boldsymbol{\varepsilon}^v(r, s)} \quad (15.119)$$

com $t \times s \in [-1, 1] \times [-1, 1]$ e $\mathbf{B}(r, s)$ a matriz com as derivadas das funções de interpolação. Assim, se o nosso elemento tem uma interpolação bilinear para os deslocamentos, terá uma interpolação com grau menor para as tensões e deformações, devido às derivadas. De fato, considerando um elemento com geometria quadrada de lado unitário, obtemos

$$\mathbf{B}(r, s) = \frac{1}{2} \begin{bmatrix} s-1 & 0 & 1-s & 0 & s+1 & 0 & -s-1 & 0 \\ 0 & r-1 & 0 & -r-1 & 0 & r+1 & 0 & 1-r \\ r-1 & s-1 & -r-1 & 1-s & r+1 & s+1 & 1-r & -s-1 \end{bmatrix} \quad (15.120)$$

que tem termos lineares em r ou em s , e não bilineares como na matriz $\mathbf{N}(r, s)$.

Outra questão muito importante é o fato de que a interpolação dos deslocamentos garante a compatibilidade dos deslocamentos entre elementos adjacentes, não só entre os nós, mas também entre as faces. Isso quer dizer que se interpolarmos o deslocamento \mathbf{u} dentro de dois elementos adjacentes, obteremos o mesmo valor (a geometria deformada de um elemento não sobrepõe a dos elementos adjacentes). O mesmo não é observado, obrigatoriamente, no cálculo das tensões e das deformações, devido à matriz \mathbf{B} .

De fato, Considere que temos dois elementos quadrados de lados unitários, ligados pelas arestas 2 (elemento da esquerda) e 4 (elemento da direita). Como ambos elementos tem a mesma geometria, terão a mesma matriz \mathbf{B} da Eq. (15.120). No entanto, se considerarmos o ponto central da aresta em comum, teremos

$$\mathbf{B}_1(1, 0) = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 & 0 & 0 \\ 0 & -1 & -2 & 1 & 2 & 1 & 0 & -1 \end{bmatrix} \quad (15.121)$$

para o elemento da esquerda e

$$\mathbf{B}_2(-1, 0) = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 2 \\ -2 & -1 & 0 & 1 & 0 & 1 & 2 & -1 \end{bmatrix} \quad (15.122)$$

para o elemento da direita. Como os elementos compartilham uma face, terão deslocamentos nodais iguais nos nós da face (gls 3 e 4 do elemento da esquerda serão os gls 1 e 2 do elemento da direita e gls 5 e 6 do elemento da esquerda serão iguais aos gls 7 e 8 do elemento da direita). Arbitrando os deslocamentos

$$\mathbf{U}_1 = \begin{Bmatrix} -1 \\ -2 \\ 1 \\ 1 \\ 2 \\ 1 \\ 3 \\ -1 \end{Bmatrix} \quad \text{e} \quad \mathbf{U}_1 = \begin{Bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 3 \\ 4 \\ 2 \\ 1 \end{Bmatrix}. \quad (15.123)$$

Assim, os estados de deformações no ponto em comum ao contorno de ambos os elementos serão

$$\boldsymbol{\epsilon}_1^v(1,0) = \mathbf{B}_1(1,0)\mathbf{U}_1 = \begin{Bmatrix} 0.5 \\ 0 \\ 3.5 \end{Bmatrix} \neq \boldsymbol{\epsilon}_2^v(-1,0) = \mathbf{B}_2(-1,0)\mathbf{U}_2 = \begin{Bmatrix} -0.5 \\ 0 \\ 1.5 \end{Bmatrix} \quad (15.124)$$

tal que as tensões também serão diferentes. Isso obviamente conflita com os conceitos da mecânica do contínuo, mas observamos que os valores tendem (no limite) a coincidir a medida que o tamanho dos elementos tende a zero.

Por fim, existem infinitos pontos dentro de um elemento finito e sabemos que tensão e deformação são medidas que variam ponto a ponto. Com isso, surge a questão de em que pontos $(r,s) \in \Omega$ devemos calcular as tensões. Nesta altura o leitor deve estar pensando em utilizar as coordenadas nodais $r,s = \pm 1$, mas infelizmente essa não é uma boa opção em termos de exatidão da estimativa de tensões. De fato, Barlow¹ mostrou que existem pontos dentro do elemento, chamados de Pontos de Barlow, que permitem um cálculo mais exato das tensões e deformações. No caso do elemento bilinear isoparamétrico de 4 nós, o ponto central $r = s = 0$ é considerado o melhor ponto. No entanto, é muito comum vermos a utilização dos pontos de Gauss para o cálculo de tensões (isso é muito comum em programas comerciais). A utilização desses pontos não é um problema em si, principalmente para o cálculo das tensões normais, mas pode acarretar um erro um pouco maior para a tensão cisalhante.

Iremos discutir os pontos de Barlow em um capítulo específico.

15.6 Exemplo

Vamos utilizar um exemplo para reforçar os conceitos vistos neste capítulo.

■ **Exemplo 15.8** Considere um domínio com dimensões $1 \times 0.1 \times 0.01$ [m³]. Sabe-se que todos os pontos em $X = 0$ são restritos em translação na direção X e que os pontos em $Y = 0$ são restritos em translação na direção Y . Todos os pontos em $X = 1$ são submetidos a uma força normal de 1000 [N/m²] na direção positiva do eixo X . O material do domínio é isotrópico e assumimos que está no regime linear elástico, com $E = 1 \times 10^9$ [Pa] e $\nu = 0.3$. Obtenha os deslocamentos no domínio.

Dos dados podemos identificar que estamos lidando com um problema de Estado Plano de Tensões, EPT, com $t = 0.01$ [m]. Podemos discretizar o domínio com dois elementos finitos bilineares isoparamétricos de 4 nós, tal que $n_n = 6$ e $n_e = 2$, conforme a Fig. 15.3.

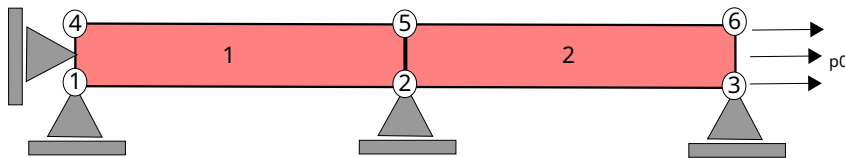


Figura 15.3: Domínio discretizado com dois elementos finitos de 4 nós.

¹Barlow, J. (1976), Optimal stress locations in finite element models. Int. J. Numer. Meth. Engng., 10: 243-251. <https://doi.org/10.1002/nme.1620100202>

As coordenadas nodais são descritas por

$$\mathbf{XY} = \begin{bmatrix} 0.0 & 0.0 \\ 0.5 & 0.0 \\ 1.0 & 0.0 \\ 0.0 & 0.1 \\ 0.5 & 0.1 \\ 1.0 & 0.1 \end{bmatrix} \quad (15.125)$$

e as conectividades dos elementos são

$$\mathbf{IJ} = \begin{bmatrix} 1 & 2 & 5 & 4 \\ 2 & 3 & 6 & 5 \end{bmatrix} \quad (15.126)$$

e os dados sobre material e espessura são

$$\mathbf{MAT} = \begin{bmatrix} 1 \times 10^9 & 0.3 \\ 1 \times 10^9 & 0.3 \end{bmatrix} \quad (15.127)$$

e

$$\mathbf{ESP} = \begin{Bmatrix} 0.01 \\ 0.01 \end{Bmatrix}. \quad (15.128)$$

As condições de contorno de Dirichlet, apoios, são aplicadas nos nós 1 (gls X e Y), 2 (gl Y), 3 (gl Y) e 4 (gl X), tal que

$$\mathbf{AP} = \begin{bmatrix} 1 & 1 & 0.0 \\ 1 & 2 & 0.0 \\ 2 & 2 & 0.0 \\ 3 & 2 & 0.0 \\ 4 & 1 & 0.0 \end{bmatrix} \quad (15.129)$$

e $n_a = 5$. Por fim, temos a aplicação de uma pressão positiva na direção normal da aresta 2 do elemento 2

$$\mathbf{P} = [2 \quad 2 \quad 1000] \quad (15.130)$$

e $n_p = 1$.

Devido a geometria bem simples dos elementos (que tem as mesmas dimensões e mesmas propriedades), podemos calcular

$$\mathbf{J} = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.05 \end{bmatrix} \quad (15.131)$$

com

$$|\mathbf{J}| = 0.0125 \quad (15.132)$$

e

$$\mathbf{J}^{-1} = \begin{bmatrix} 4 & 0 \\ 0 & 20 \end{bmatrix}. \quad (15.133)$$

A matriz \mathbf{B} se torna

$$\mathbf{B} = \begin{bmatrix} s-1 & 0 & 1-s & 0 & s+1 & 0 & -s-1 & 0 \\ 0 & 5r-5 & 0 & -5r-5 & 0 & 5r+5 & 0 & 5-5r \\ 5r-5 & s-1 & -5r-5 & 1-s & 5r+5 & s+1 & 5-5r & -s-1 \end{bmatrix} \quad (15.134)$$

e a rigidez local se torna

$$\mathbf{K}_e = \begin{bmatrix} 7.14 & 1.79 & 2.47 & -0.137 & -3.57 & -1.79 & -6.04 & 0.137 \\ 1.79 & 18.6 & 0.137 & 8.9 & -1.79 & -9.29 & -0.137 & -18.2 \\ 2.47 & 0.137 & 7.14 & -1.79 & -6.04 & -0.137 & -3.57 & 1.79 \\ -0.137 & 8.9 & -1.79 & 18.6 & 0.137 & -18.2 & 1.79 & -9.29 \\ -3.57 & -1.79 & -6.04 & 0.137 & 7.14 & 1.79 & 2.47 & -0.137 \\ -1.79 & -9.29 & -0.137 & -18.2 & 1.79 & 18.6 & 0.137 & 8.9 \\ -6.04 & -0.137 & -3.57 & 1.79 & 2.47 & 0.137 & 7.14 & -1.79 \\ 0.137 & -18.2 & 1.79 & -9.29 & -0.137 & 8.9 & -1.79 & 18.6 \end{bmatrix} \times 10^6$$

onde o número de casas decimais foi reduzida para fins de visualização.

Os graus de liberdade globais para os elementos são

$$\mathbf{g}^1 = \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 9 \\ 10 \\ 7 \\ 8 \end{Bmatrix} \text{ e } \mathbf{g}^2 = \begin{Bmatrix} 3 \\ 4 \\ 5 \\ 6 \\ 11 \\ 12 \\ 9 \\ 10 \end{Bmatrix} \quad (15.135)$$

que serão utilizados para posicionar as matrizes locais 8×8 na matriz global 12×12 . De forma esquemática, podemos representar as posições da matriz do primeiro elemento por 1, do segundo elemento por 2 e a sobreposição das matrizes por 3. Com isso, teremos a estrutura

$$\mathbf{K} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 3 & 3 & 2 & 2 & 1 & 1 & 3 & 3 & 2 & 2 \\ 1 & 1 & 3 & 3 & 2 & 2 & 1 & 1 & 3 & 3 & 2 & 2 \\ 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 3 & 3 & 2 & 2 & 1 & 1 & 3 & 3 & 2 & 2 \\ 1 & 1 & 3 & 3 & 2 & 2 & 1 & 1 & 3 & 3 & 2 & 2 \\ 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 2 & 2 \end{bmatrix} \quad (15.136)$$

na matriz global.

Para calcularmos o vetor de forças distribuídas, identificamos que a pressão de 1000 $[N/m^2]$ é aplicada na face 2 do segundo elemento. A aresta 2 do elemento 2 é descrita por

$$\mathbf{v}^2 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \quad (15.137)$$

e

$$\mathbf{n}^2 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \quad (15.138)$$

com

$$|\mathbf{J}|^2 = \frac{1}{2} \sqrt{(0-0)^2 + (0.1-0)^2} = \frac{0.1}{2}. \quad (15.139)$$

O vetor de forças para o elemento 2 é calculado com

$$t_e \int_{-1}^1 \mathbf{N}^T(1,s) p(s) \mathbf{n}^2 |\mathbf{J}|^2 ds \quad (15.140)$$

ou

$$0.01 \int_{-1}^1 p_0 \frac{1}{2} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1-s & 0 \\ 0 & 1-s \\ 1+s & 0 \\ 0 & 1+s \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \frac{0.1}{2} ds = 0.01 \int_{-1}^1 p_0 \frac{1}{2} \begin{Bmatrix} 0 \\ 0 \\ 1-s \\ 0 \\ 1+s \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \frac{0.1}{2} ds = 0.01 \cdot 1000 \frac{1}{2} \frac{0.1}{2} \begin{Bmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

ou

$$\mathbf{F}_2 = \begin{Bmatrix} 0 \\ 0 \\ 0.5 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} [N]. \quad (15.141)$$

Esse vetor 8×1 deve ser posicionado no vetor 12×1 de forças globais, de acordo com o vetor \mathbf{g}^2 . Com isso, teremos um vetor global

$$\mathbf{F} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}. \quad (15.142)$$

Após a montagem do sistema global, podemos eliminar as linhas e colunas associadas aos gls restritos (matriz \mathbf{AP}), ou seja, os gls 1,2,4,6 e 7. Com isso, a solução do sistema

linear 12×12 resulta em

$$\mathbf{U} = \begin{Bmatrix} 0.0 \\ 0.0 \\ 0.5 \\ 0 \\ 1 \\ 0 \\ 0.0 \\ -0.03 \\ 0.5 \\ -0.03 \\ 1 \\ -0.03 \end{Bmatrix} \times 10^{-6}. \quad (15.143)$$

A situação analisada neste exemplo nada mais é do que uma barra com $L = 1$ [m], $A = 0.1^2$ [m²] e $E = 1 \times 10^9$ [Pa] submetida a uma força de $p_0 A$ [N]. Da teoria de barras, sabemos que o deslocamento em $x = 1$ [m] é dado por

$$u_x(1) = \frac{FL}{EA} = \frac{p_0 L}{E} = \frac{1000}{1 \times 10^9} = 1 \times 10^{-6}. \quad (15.144)$$

Esse é exatamente o deslocamento obtido nos graus de liberdade X dos nós 3 e 6. Da teoria de barra também sabemos que o deslocamento varia linearmente com o comprimento, o que de fato é observado nos graus de liberdade X dos nós 2 e 5, que são a metade do deslocamento da ponta. Embora a teoria de barra não considere o efeito do coeficiente de Poisson, podemos utilizar a sua definição

$$\nu = -\frac{\epsilon_{yy}}{\epsilon_{xx}} \quad (15.145)$$

em que

$$\epsilon_{xx} = \frac{\Delta L}{L} = 1 \times 10^{-6}. \quad (15.146)$$

Com isso,

$$\epsilon_{yy} = -0.3 \times 10^{-6} \quad (15.147)$$

e, por definição

$$\epsilon_{yy} = \frac{\Delta H}{H} = \frac{\Delta H}{0.1} \quad (15.148)$$

tal que

$$\Delta H = -0.03 \times 10^{-6} \quad (15.149)$$

que é o deslocamento vertical (gl Y) dos nós 4, 5 e 6.

Por fim, podemos calcular as tensões nos elementos. Como estamos calculando a resposta de uma barra, sabemos que a única tensão que existe é a tensão normal

$$\sigma_{xx} = \frac{F}{A} = \frac{p_0 A}{A} = p_0 \quad (15.150)$$

que será igual a tração aplicada no contorno. A tensão no centro de um elemento finito pode ser calculada por

$$\boldsymbol{\sigma}^v(0,0) = \mathbf{C}^v \mathbf{B}(0,0) \mathbf{U}_e \quad (15.151)$$

com

$$\mathbf{C}_{EPT}^v = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} = \frac{1 \times 10^9}{1-0.3^2} \begin{bmatrix} 1 & 0.3 & 0 \\ 0.3 & 1 & 0 \\ 0 & 0 & \frac{1-0.3}{2} \end{bmatrix} \quad (15.152)$$

e

$$\mathbf{B}(0,0) = \begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -5 & 0 & -5 & 0 & 5 & 0 & 5 \\ -5 & -1 & -5 & 1 & 5 & 1 & 5 & -1 \end{bmatrix} \quad (15.153)$$

já que os elementos são iguais. O que muda é o vetor de deslocamentos de cada um. Assim, para o primeiro elemento (graus de liberdade 1,2,3,4,9,10,7,8)

$$\boldsymbol{\sigma}_1^v(0,0) = \frac{1 \times 10^9}{1-0.3^2} \begin{bmatrix} 1 & 0.3 & 0 \\ 0.3 & 1 & 0 \\ 0 & 0 & \frac{1-0.3}{2} \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -5 & 0 & -5 & 0 & 5 & 0 & 5 \\ -5 & -1 & -5 & 1 & 5 & 1 & 5 & -1 \end{bmatrix} \begin{Bmatrix} 0.0 \\ 0.0 \\ 0.5 \\ 0.0 \\ 0.5 \\ -0.03 \\ 0.0 \\ -0.03 \end{Bmatrix} \times 10^{-6}$$

resultando em

$$\boldsymbol{\sigma}_1^v(0,0) = \begin{Bmatrix} 1000 \\ 0 \\ 0 \end{Bmatrix}, \quad (15.154)$$

como esperado. Verifique o resultado para o segundo elemento. ■

16. Cisalhamento parasita

O elemento bilinear isoparamétrico de 4 nós é um dos elementos mais utilizados para solucionar problemas de mecânica dos sólidos. Por esse motivo, é muito importante entendermos as suas características, pontos fortes e pontos fracos.

Do exemplo 15.8, podemos notar que obtivemos a solução analítica com apenas dois elementos. Isso pode ser explicado pelo fato de o deslocamento analítico ao longo do comprimento e da altura do domínio serem lineares, assim como as funções de interpolação do elemento. Portanto, como vimos nos capítulos iniciais deste texto, estamos utilizando um conjunto correto de bases para descrever o problema.

Suponha agora que um material isotrópico, linear e elástico seja submetido à flexão, como ilustrado na Fig. 16.1.

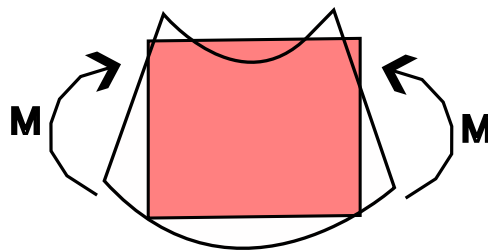


Figura 16.1: Resposta de um material linear e elástico submetido à flexão pura

Como temos um meio contínuo submetido a uma flexão pura (momento constante), podemos utilizar alguns conceitos da teoria de vigas. Para facilitar a comparação com o elemento bilinear isoparamétrico, vamos utilizar o sistema rs para descrever a geometria da Fig. 16.1.

Da relação momento \times curvatura, sabemos que

$$\kappa = \frac{d^2 u_y(r)}{dr^2} = \frac{M}{EI} = \alpha \quad (16.1)$$

onde α é uma constante e u_y o deslocamento na direção Y (vertical). A solução dessa EDO é

$$u_y(r) = \frac{\alpha}{2} r^2 + C_1 r + C_2 \quad (16.2)$$

ou seja, temos um deslocamento transversal **quadrático** em r (lembre-se que, da hipótese de vigas, $\mathbf{v} = 0$ e a altura não varia. Por isso, podemos utilizar o conceito de linha elástica e não temos a dependência de s aqui).

A rotação da seção transversal é dada por

$$\theta(r) = \alpha r + C_1 \quad (16.3)$$

e, da hipótese de seções planas (que é válida, pois estamos em flexão pura),

$$u_x(r, s) = -\theta(r)s \quad (16.4)$$

Impondo $u_y(-1) = u_y(1) = \bar{v}$, obtemos

$$u_y(r) = \frac{\alpha}{2} r^2 + \frac{2\bar{v} - \alpha}{2} \quad (16.5)$$

e

$$\theta(r) = \alpha r \quad (16.6)$$

tal que

$$u_x(r, s) = -\alpha r s. \quad (16.7)$$

Neste ponto podemos entender que o **elemento finito bilinear não deve ser capaz de representar corretamente o deslocamento transversal $u_y(r)$ em flexão**, pois utiliza bases bilineares.

Para investigarmos a resposta de um elemento finito bilinear isoparamétrico de 4 nós submetido à flexão, vamos considerar um elemento quadrado de lado unitário, $E = 1$, $\mathbf{v} = 0$. Como condições de contorno de Dirichlet, vamos impor que os deslocamentos verticais nos nós 6 e 8 são iguais e que o deslocamento horizontal do nó 1 é nulo (com isso eliminamos os 3 modos de corpo rígido).

Quando esse elemento é submetido a um binário unitário

$$\mathbf{F} = \left\{ \begin{array}{c} -1 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{array} \right\} \quad (16.8)$$

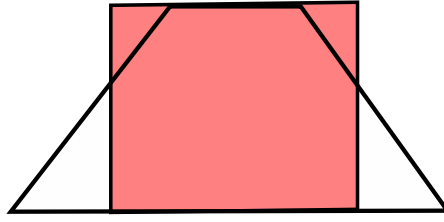


Figura 16.2: Resposta de um elemento bilinear isoparamétrico submetido à flexão pura.

obtemos

$$\mathbf{U} = \begin{Bmatrix} 0 \\ b \\ c \\ b \\ 0 \\ b \\ c \\ b \end{Bmatrix} \quad (16.9)$$

tal que

$$u_y(r, s) = N_1(r, s)b + N_2(r, s)b + N_3(r, s)b + N_4(r, s)b = b \quad (16.10)$$

ou seja, **todos os pontos do elemento tem o mesmo deslocamento vertical b** . Isso mostra que o elemento não flete, o que é um resultado surpreendente.

O deslocamento horizontal no interior do elemento é

$$u_x(r, s) = N_1(r, s)0 + N_2(r, s)c + N_3(r, s)0 + N_4(r, s)c = \frac{c}{2}(1 - rs) \quad (16.11)$$

que é bilinear como a Eq. (16.7) (as condições de contorno não são exatamente as mesmas, o que justifica a diferença nas constantes). A Figura 16.2 ilustra o padrão de deslocamentos relativo à aplicação de um carregamento de flexão pura em um elemento bilinear isoparamétrico de 4 nós.

O padrão geométrico obtido não é o esperado, estando de acordo com o que se esperaria da resposta à aplicação de esforços cisalhantes. Por esse motivo, dizemos que o elemento responde **erroneamente** em cisalhamento ao invés de flexão, ao que chamamos de cisalhamento parasita ou *parasitic shear*.

16.1 Modos de deformação de um elemento finito

Já sabemos que a relação entre forças e deslocamentos é representado, na forma discreta, por

$$\mathbf{K}_e \mathbf{U}_e = \mathbf{F}_e. \quad (16.12)$$

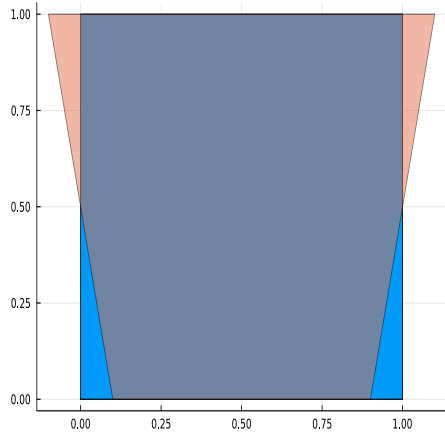
Também sabemos (Consultar notas de aula de Fundamentos de Matemática), que o problema de autovalores e autovetores permite transformar o sistema acima para uma forma em que a matriz de coeficientes é diagonal, contendo os autovalores. Assim, os autovalores da matriz de rigidez estão associados a valores principais de rigidez.

Também sabemos que os autovetores estão associados aos 'modos' de resposta do problema, que no nosso caso, estão associados ao padrão de deslocamento nodal devido a uma rigidez principal.

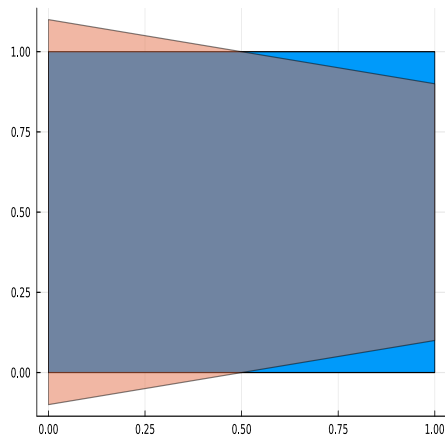
De fato, se considerarmos o mesmo elemento bilinear isoparamétrico da seção anterior, mas sem a aplicação das condições de contorno de Dirichlet (elemento está 'solto' no plano), iremos obter 8 autovalores

$$\boldsymbol{\lambda} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0.5 \\ 0.5 \\ 1.0 \\ 1.0 \\ 1.0 \end{Bmatrix}. \quad (16.13)$$

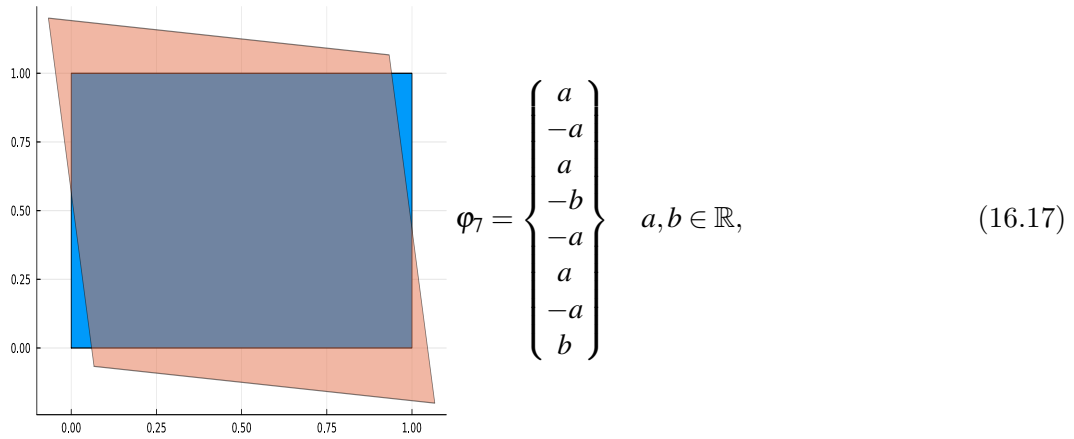
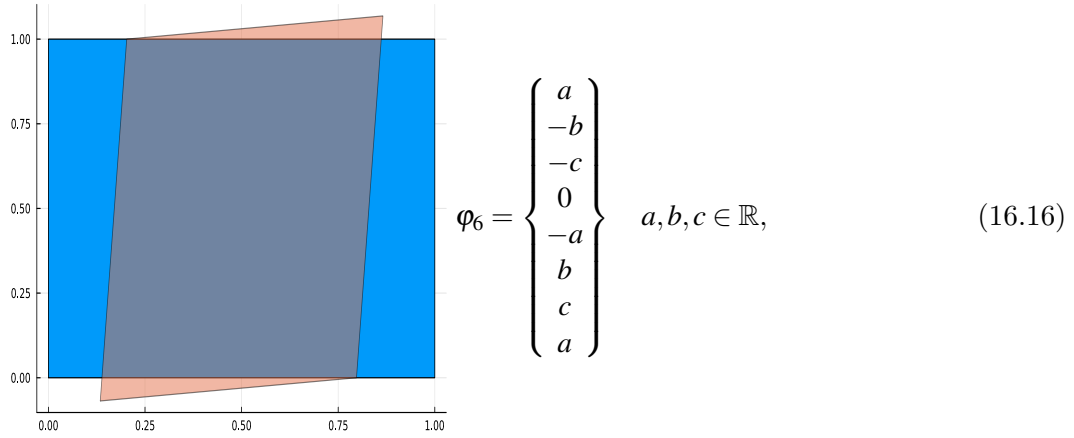
Os três primeiros modos não apresentam rigidez, pois são modos de corpo rígido (translação em X , translação em Y e rotação). Os outros 5 modos são modos que apresentam rigidez (são modos que realmente contribuem para a rigidez do elemento) e tem autovetores



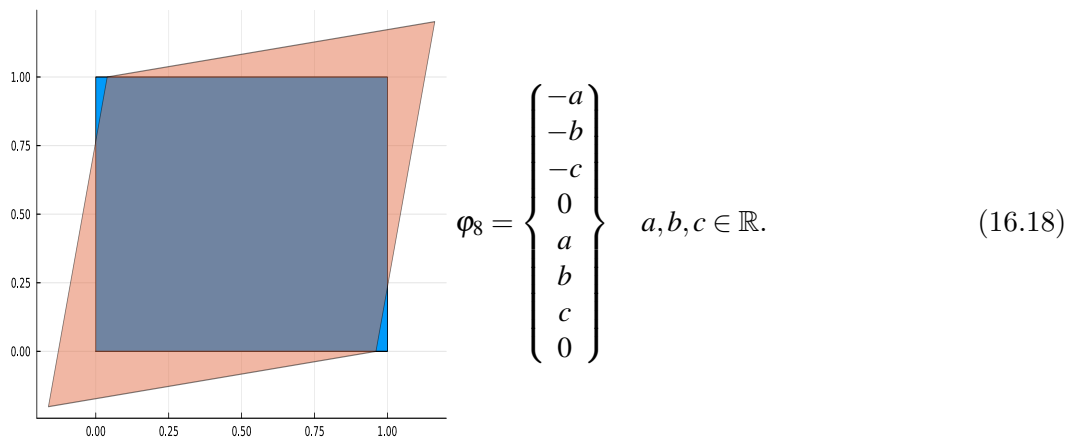
$$\varphi_4 = \begin{Bmatrix} a \\ 0 \\ -a \\ 0 \\ a \\ 0 \\ -a \\ 0 \end{Bmatrix} \quad a \in \mathbb{R}, \quad (16.14)$$



$$\varphi_5 = \begin{Bmatrix} 0 \\ -a \\ 0 \\ a \\ 0 \\ 0 \\ -a \\ a \end{Bmatrix} \quad a \in \mathbb{R}, \quad (16.15)$$



e



Como podemos notar, qualquer resposta em deslocamento do nosso elemento bilinear será uma **combinação linear** destes modos fundamentais de deslocamento. Por exemplo, a resposta em cisalhamento parasita nada mais é do que o modo 4 diretamente, e não existe nenhuma combinação linear dos modos que resulte no padrão quadrático de deslocamentos da Eq. (16.5)

16.2 Problemas dominados por flexão

Dos resultados anteriores podemos notar que o elemento não é eficaz em descrever, sozinho, a flexão. Portanto, se tivermos um problema dominado por flexão, devemos esperar que os elementos respondam em cisalhamento, que é mais rígido do que a flexão. De fato, vamos considerar um domínio $1 \times 0.1 \times 0.1$ [m], material com $E = 10 \times 10^3$ [Pa] e $\nu = 0$. Esse domínio está restrito em translação em todos os pontos com $X = 0$ e está submetido a uma força concentrada de 1 [N] em $X = 1$ e $Y = 0.1$. Da teoria de vigas longas sabemos que o deslocamento na ponta será

$$u_y(1) = \frac{FL^3}{3EI} = \frac{12}{3(10000)(0.1)^4} = 4 \text{ [m]}. \quad (16.19)$$

Vamos estudar a utilização de discretizações $10n \times n$ elementos ($10n$ elementos no comprimento e n elementos na altura). O código 16.1 ilustra a utilização do pacote LFEM para realizarmos os cálculos e o Gráfico da Fig. 16.3 ilustra os resultados obtidos para um número cada vez maior de elementos na altura.

Listing 16.1: Código em Julia para estudar o refino da viga engastada em flexão.

```
using LFEM
using TMeshes
using Plots

# Número de divisões
N = 10

# Aloca o vetor de saída
desloc = zeros(n)

# Laço em n
for n=1:N

    # Cria a malha utilizando o TMeshes
    mesh = Cantilever_beam_bottom2D(10*n,n,:solid2D,Lx=1.0,
                                     Ly=0.1, force=-1.0,Ex=10E3,vxy=0.0,thickness=0.1
    );

    # Seleciona o elemento bilinear isoparamétrico puro
    delete!(mesh.options,:INCOMPATIBLE)

    # Soluciona o problema usando o LFEM
    U,_ = Solve_linear(mesh);

    # Armazena o resultado do deslocamento na ponta livre
    desloc[n]=U[end]
end

# Gera o gráfico
plot(1:N,desloc,linewidth=4,label="",xlabel="n",ylabel="uy")
```

Esse resultado é muito importante, pois indica que **em problemas dominados por flexão** precisamos de **um grande número de elementos bilineares** ao longo da altura (no sentido da flexão) para que a resposta seja calculada corretamente. Embora também tenhamos refinado o número de elementos na direção X (para evitar grandes razões de aspecto), a grande influência aqui se deve ao número de elementos ao longo da altura. A explicação para essa melhora é bem simples: a medida em que aumentamos o número de elementos ao longo da altura, submetemos esses elementos cada vez mais a somente tração ou compressão simples (lembre-se que em vigas temos apenas σ_{xx}) e já sabemos que o elemento é capaz de descrever esse comportamento corretamente.

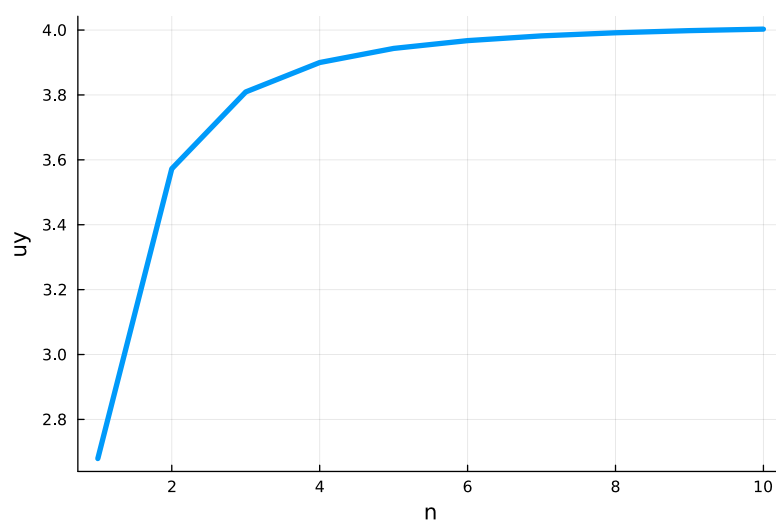


Figura 16.3: Deslocamento vertical em função do número de elementos na altura

17. Implementação computacional

O objetivo deste capítulo é apresentar os algoritmos necessários para a implementação computacional do elemento bilinear isoparamétrico de 4 nós. Por simplicidade, vamos considerar Estado Plano de Tensão (EPT), embora a consideração de EPD seja trivial.

17.1 Entrada de dados

Começando com as informações sobre os nós, definimos n_n como a variável que contém o número de nós e \mathbf{XY} é uma matriz de dimensão $n_n \times 2$ que contém em cada nó (linha da matriz) a coordenada X (coluna 1 da matriz) e a coordenada Y (coluna 2 da matriz). Essa entrada de dados é idêntica ao visto no capítulo sobre a implementação computacional para treliças 2D.

O número de elementos será definido pela variável n_e e as conectividades dos elementos (informações sobre quais nós estão associados a cada elemento) serão informadas na matriz \mathbf{IJ} , de dimensão $n_e \times 4$, pois cada elemento tem 4 nós.

As propriedades do material de cada elemento são o módulo de elasticidade longitudinal e o coeficiente de Poisson. Essas informações serão fornecidas em uma matriz \mathbf{MAT} com n_e linhas e duas colunas (E e ν).

A informação 'geométrica' de cada elemento, no caso de EPT, é a sua espessura. Essa informação é fornecida em um vetor \mathbf{ESP} com n_e posições.

As n_f forças concentradas aplicadas nos nós da estrutura são informadas em uma matriz \mathbf{FC} de dimensão $n_f \times 3$, em que cada linha contém a informação sobre o nó em que a força está aplicada (coluna 1 da matriz), seu grau de liberdade (coluna 2 da matriz) e valor da força aplicada (coluna 3 da matriz).

Os n_p carregamentos distribuídos normais serão fornecidos em uma matriz \mathbf{P} com n_p linhas e 3 colunas (elemento, face, valor).

As n_a condições de contorno de Dirichlet são informadas na matriz \mathbf{AP} , de dimensão $n_a \times 3$ (nó, gl, valor).

17.2 Montagem da matriz de rigidez local de um elemento

A matriz de rigidez local de um elemento, Eq. (15.39), depende de uma série de operações. Relembrando

$$\mathbf{K}_e = t_e \int_{-1}^1 \int_{-1}^1 \mathbf{B}(r,s)^T \mathbf{C}^v \mathbf{B}(r,s) |\mathbf{J}(r,s)| dr ds. \quad (17.1)$$

Começando com a determinação da matriz \mathbf{C}^v para EPT, Alg. 21.

Algoritmo 21: Montagem da matriz constitutiva EPT de um elemento com E e ν .

```

1 MontaCEPT( $E, \nu$ )
2   Calcula termo em comum
3    $c \leftarrow \frac{E}{1-\nu^2}$ 
4   Define a matriz
5    $\mathbf{C}^v \leftarrow \begin{bmatrix} c & c\nu & 0 \\ c\nu & c & 0 \\ 0 & 0 & c\frac{1-\nu}{2} \end{bmatrix}$ 
6   Retorna a matriz
7   return  $\mathbf{C}^v$ 
```

A montagem da matriz $\mathbf{B}(r,s)$ depende não só das derivadas das funções de interpolação, mas também da montagem da matriz Jacobiana e das correções. Por isso, vamos implementar cada uma das operações separadamente. Vamos assumir que os vetores $\mathbf{X}_{4 \times 1}$ e $\mathbf{Y}_{4 \times 1}$ contêm as coordenadas do elemento.

Começamos com o cálculo das derivadas das funções de interpolação em relação as variáveis r e s , Alg. 22.

Algoritmo 22: Retorna a derivada das funções de interpolação em relação a r e s como uma matriz 2×4 .

```

1 MontadN( $r, s$ )
2   Matriz com as derivadas em relação a  $r$  (primeira linha) e  $s$ 
   (segunda linha)
3    $\mathbf{dNrs} \leftarrow \frac{1}{4} \begin{bmatrix} s-1 & 1-s & 1+s & -(1+s) \\ r-1 & -(1+r) & 1+r & 1-r \end{bmatrix}$ 
4   Retorna a matriz
5   return  $\mathbf{dNrs}$ 
```

De posse das derivadas e das coordenadas nodais do elemento, podemos calcular a matriz Jacobiana, Alg. 24

Com isso, podemos calcular a matriz \mathbf{B} em um ponto (r,s) , Alg.

Por fim, podemos proceder com a montagem da matriz de rigidez local do elemento

17.3 Montagem da matriz de rigidez global

A montagem da matriz de rigidez global é praticamente igual ao visto no capítulo de implementação computacional para treliças.

A montagem do vetor de forças globais devidas a pressão normal aplicada na face de alguns elementos está ilustrada no Alg. 34, que depende de uma série de algoritmos

Algoritmo 23: Retorna a matriz **J** no ponto (r, s) .

```

1 MontaJ(dNrs,X,Y)
2   Aloca a matriz J
3    $\mathbf{J} \leftarrow \mathbf{0}_{2 \times 2}$ 
4   Laço para montar a matriz
5   for  $i \in \{1, \dots, 4\}$  do
6        $\mathbf{J}(1,1) \leftarrow \mathbf{J}(1,1) + \mathbf{X}(i)\mathbf{dNrs}(1,i)$ 
7        $\mathbf{J}(1,2) \leftarrow \mathbf{J}(1,2) + \mathbf{Y}(i)\mathbf{dNrs}(1,i)$ 
8        $\mathbf{J}(2,1) \leftarrow \mathbf{J}(2,1) + \mathbf{X}(i)\mathbf{dNrs}(2,i)$ 
9        $\mathbf{J}(2,2) \leftarrow \mathbf{J}(2,2) + \mathbf{Y}(i)\mathbf{dNrs}(2,i)$ 
10  Retorna a matriz
11  return J

```

Algoritmo 24: Mapeia as derivadas de rs para xy .

```

1 CorrigedN(dNrs,J)
2   Inverte a matriz
3    $\mathbf{invJ} \leftarrow \mathbf{J}^{-1}$ 
4   Calcula a correção
5    $\mathbf{dNxy} \leftarrow \mathbf{J}^{-1}\mathbf{dNrs}$ 
6   Retorna a matriz
7   return dNxy

```

Algoritmo 25: Retorna a matriz **B** no ponto (r, s) .

```

1 MontaB(dNxy)
2   Aloca a matriz
3    $\mathbf{B} \leftarrow \mathbf{0}_{3 \times 8}$ 
4   Indicador de início de bloco
5    $c \leftarrow 1$ 
6   Laço para montar a matriz
7   for  $i \in \{1, \dots, 4\}$  do
8        $\mathbf{B}(1,c) \leftarrow \mathbf{dNxy}(1,i)$ 
9        $\mathbf{B}(2,c+1) \leftarrow \mathbf{dNxy}(2,i)$ 
10       $\mathbf{B}(3,c) \leftarrow \mathbf{B}(2,c+1)$ 
11       $\mathbf{B}(3,c+1) \leftarrow \mathbf{B}(2,c)$ 
12      Atualiza o bloco
13       $c \leftarrow c+2$ 
14  Retorna a matriz
15  return B

```

Algoritmo 26: Retorna a matriz \mathbf{K}_e do elemento bilinear isoparamétrico de 4 nós.

```

1 MontaKe( $\mathbf{X}, \mathbf{Y}, E, \nu, t_e$ )
2   Aloca a matriz
3    $\mathbf{K}_e \leftarrow \mathbf{0}_{8 \times 8}$ 
4   Matriz constitutiva do elemento
5    $\mathbf{C}_\nu \leftarrow \text{MontaCEPT}(E, \nu)$ 
6   Define os pontos de Gauss
7    $\mathbf{pg} \leftarrow \frac{1}{\sqrt{3}} \begin{bmatrix} -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$ 
8   Laço pelos pontos de Gauss
9   for  $p \in \{1, \dots, 4\}$  do
10      Coordenadas do PG
11       $r \leftarrow \mathbf{pg}(1, p)$ 
12       $s \leftarrow \mathbf{pg}(2, p)$ 
13      Calcula as derivadas das funções de interpolação
14       $\mathbf{dNrs} \leftarrow \text{MontadN}(r, s)$ 
15      Calcula a matriz Jacobiana
16       $\mathbf{J} \leftarrow \text{MontaJ}(\mathbf{dNrs}, \mathbf{X}, \mathbf{Y})$ 
17      Calcula o determinante da matriz  $\mathbf{J}$ 
18       $dJ \leftarrow \det(\mathbf{J})$ 
19      Mapeia as derivadas para  $xy$ 
20       $\mathbf{dNxy} \leftarrow \text{CorrigedN}(\mathbf{dNrs}, \mathbf{J})$ 
21      Monta a matriz  $\mathbf{B}$ 
22       $\mathbf{B} \leftarrow \text{MontaB}(\mathbf{dNxy})$ 
23      Acumula na matriz de rigidez
24       $\mathbf{K}_e \leftarrow \mathbf{K}_e + \mathbf{B}^T \mathbf{C}_\nu \mathbf{B} t_e dJ$ 
25   Retorna a matriz
26   return  $\mathbf{K}_e$ 

```

auxiliares que são definidos nesta seção. Em especial, é interessante notar que esses algoritmos auxiliares servem para sistematizar os resultados individuais que foram obtidos no capítulo anterior para cada uma das faces do elemento.

Algoritmo 27: Montagem dos vetores com as coordenadas dos nós do elemento

```

1 MontaXY( $e, IJ, XY$ )
2   Inicializa os vetores
3    $\mathbf{X} \leftarrow \mathbf{0}_4$ 
4    $\mathbf{Y} \leftarrow \mathbf{0}_4$ 
5   Preenche os vetores com as coordenadas
6   for  $i \in \{1, \dots, 4\}$  do
7     Nó
8      $no \leftarrow IJ(e, i)$ 
9     Coordenadas
10     $\mathbf{X}(i) \leftarrow \mathbf{XY}(no, 1)$ 
11     $\mathbf{Y}(i) \leftarrow \mathbf{XY}(no, 2)$ 
12   Retorna os vetores com as coordenadas do elemento
13   return  $\mathbf{X}, \mathbf{Y}$ 

```

Algoritmo 28: Montagem do vetor com os graus de liberdade do elemento

```

1 Montagls( $e, IJ$ )
2   Inicializa o vetor
3    $\mathbf{gls} \leftarrow \mathbf{0}_8$ 
4   Contador
5    $c \leftarrow 1$ 
6   Calcula as posições
7   for  $i \in \{1, \dots, 4\}$  do
8     Nó
9      $no \leftarrow IJ(e, i)$ 
10    Laço pelos graus de liberdade do nó
11    for  $j \in \{1, \dots, 2\}$  do
12      Calcula o gl
13       $\mathbf{gls}(c) \leftarrow 2(no - 1) + j$ 
14      Incrementa o contador
15       $c \leftarrow c + 1$ 
16   Retorna o vetor com os gls
17   return  $\mathbf{gls}$ 

```

17.4 Montagem do vetor de forças distribuídas

A montagem do vetor de forças concentradas é idêntica ao que vimos para treliças. A montagem do vetor de forças concentradas necessita das definições das arestas, tal que vamos precisar das definições que vimos no capítulo anterior.

Algoritmo 29: Montagem da matriz de rigidez global

```

1 RigidezGlobal( $n_n, n_e, \mathbf{MAT}, \mathbf{ESP}, \mathbf{XY}, \mathbf{IJ}$ )
2   Inicializa a matriz global
3    $\mathbf{K} \leftarrow \mathbf{0}_{2n_n \times 2n_n}$ 
4   Laço nos elementos
5   for  $e \in \{1, \dots, n_e\}$  do
6     Recupera as propriedades do elemento
7      $E \leftarrow \mathbf{MAT}(e, 1)$ 
8      $v \leftarrow \mathbf{MAT}(e, 2)$ 
9      $t_e \leftarrow \mathbf{ESP}(e)$ 
10    Recupera as coordenadas do elemento
11     $\mathbf{X}, \mathbf{Y} \leftarrow \text{MontaXY}(e, \mathbf{IJ}, \mathbf{XY})$ 
12    Monta a matriz de rigidez local
13     $\mathbf{K}_e \leftarrow \text{MontaKe}(\mathbf{X}, \mathbf{Y}, E, v, t_e)$ 
14    Determina os gls globais do elemento
15     $\mathbf{g} \leftarrow \text{Montagls}(e, \mathbf{IJ})$ 
16    Sobrepe a matriz do elemento na rigidez global
17    for  $i \in \{1, \dots, 8\}$  do
18      for  $j \in \{1, \dots, 8\}$  do
19         $\mathbf{K}(\mathbf{g}(i), \mathbf{g}(j)) \leftarrow \mathbf{K}(\mathbf{g}(i), \mathbf{g}(j)) + \mathbf{K}_e(i, j)$ 
20  Retorna a matriz global
21  return  $\mathbf{K}$ 

```

Algoritmo 30: Definições das faces do elemento bilinear isoparamétrico de 4 nós.

```

1 Faces(face)
2   Retorna dois índices para o cálculo dos vetores locais da face
3   if face = 1 then
4      $p_1 \leftarrow 1$ 
5      $p_2 \leftarrow 2$ 
6   if face == 2 then
7      $p_1 \leftarrow 2$ 
8      $p_2 \leftarrow 3$ 
9   if face == 3 then
10     $p_1 \leftarrow 3$ 
11     $p_2 \leftarrow 4$ 
12   if face == 4 then
13     $p_1 \leftarrow 4$ 
14     $p_2 \leftarrow 1$ 
15   Retorna as posições
16   return  $p_1, p_2$ 

```

Algoritmo 31: Retorna a matriz de funções de interpolação do elemento na coordenada ζ dependendo da face.

```

1 MontaN( $\zeta, p_1, p_2$ )
2   Aloca a matriz N
3    $\mathbf{N} \leftarrow \mathbf{0}_{2 \times 8}$ 
4   Funções de interpolação não nulas
5    $N_1 \leftarrow \frac{1-\zeta}{2}$ 
6    $N_2 \leftarrow \frac{1+\zeta}{2}$ 
7   Posiciona na matriz N
8    $\mathbf{N}(1, 2p_1 - 1) \leftarrow N_1$ 
9    $\mathbf{N}(2, 2p_1) \leftarrow N_1$ 
10   $\mathbf{N}(1, 2p_2 - 1) \leftarrow N_2$ 
11   $\mathbf{N}(2, 2p_2) \leftarrow N_2$ 
12  Retorna a matriz
13  return N

```

Algoritmo 32: Sistema local de referência na face do elemento bilinear isoparamétrico de 4 nós .

```

1 Facesnv( $p_1, p_2, e, \mathbf{IJ}, \mathbf{XY}$ )
2   Recupera os nós da face
3    $n_1 \leftarrow \mathbf{IJ}(e, p_1)$ 
4    $n_2 \leftarrow \mathbf{IJ}(e, p_2)$ 
5   Recupera as coordenadas
6    $x_1 \leftarrow \mathbf{XY}(n_1, 1)$ 
7    $y_1 \leftarrow \mathbf{XY}(n_1, 2)$ 
8    $x_2 \leftarrow \mathbf{XY}(n_2, 1)$ 
9    $y_2 \leftarrow \mathbf{XY}(n_2, 2)$ 
10  Deltas
11   $\Delta x \leftarrow x_2 - x_1$ 
12   $\Delta y \leftarrow y_2 - y_1$ 
13  Comprimento
14   $L \leftarrow \sqrt{\Delta x^2 + \Delta y^2}$ 
15  Determinante do Jacobiano
16   $dJ \leftarrow L/2$ 
17  Vetor tangente à face
18   $\mathbf{v} \leftarrow \frac{1}{L} \begin{Bmatrix} \Delta x \\ \Delta y \end{Bmatrix}$ 
19  Vetor normal à face
20   $\mathbf{n} \leftarrow \frac{1}{L} \begin{Bmatrix} \Delta y \\ -\Delta x \end{Bmatrix}$ 
21  Retorna  $dJ$  e os vetores
22  return  $dJ, \mathbf{n}, \mathbf{v}$ 

```

Algoritmo 33: Calcula o vetor de forças distribuídas normais (pressão) na face do elemento.

```

1 Fpbilinear( $dJ, \mathbf{n}, \mathbf{v}, p_1, p_2, p_0, t_e$ )
2   Pontos de Gauss
3    $\mathbf{pg} \leftarrow \frac{1}{\sqrt{3}} \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$ 
4   Aloca o vetor
5    $\mathbf{F} \leftarrow \mathbf{0}_8$ 
6   Laço nos pontos de Gauss
7   for  $i \in \{1, \dots, 2\}$  do
8     Recupera o ponto de Gauss
9      $\zeta \leftarrow \mathbf{pg}(i)$ 
10    Monta a matriz  $\mathbf{N}$ 
11     $\mathbf{N} \leftarrow \text{MontaN}(\zeta, p_1, p_2)$ 
12    Acumula no vetor
13     $\mathbf{F} \leftarrow \mathbf{F} + p_0 \mathbf{N}^T \mathbf{n} dJ t_e$ 
14   Retorna o vetor local de força normal distribuída
15   return  $\mathbf{F}$ 

```

Algoritmo 34: Montagem do vetor global de forças distribuídas nas faces dos elementos

```

1 ForcapGlobal( $n_n, \mathbf{ESP}, \mathbf{XY}, \mathbf{IJ}, n_p, \mathbf{P}$ )
2   Inicializa o vetor global
3    $\mathbf{F} \leftarrow \mathbf{0}_{2n_n}$ 
4   Laço nas linhas de  $\mathbf{P}$ 
5   for  $i \in \{1, \dots, n_p\}$  do
6     Recupera os dados do carregamento
7      $ele \leftarrow \mathbf{P}(i, 1)$ 
8      $face \leftarrow \mathbf{P}(i, 2)$ 
9      $p_0 \leftarrow \mathbf{P}(i, 3)$ 
10    Espessura do elemento
11     $t_e \leftarrow \mathbf{ESP}(ele)$ 
12    Ponteiros da face
13     $p_1, p_2 \leftarrow \text{Faces}(face)$ 
14    Vetores da face
15     $dJ, \mathbf{n}, \mathbf{v} \leftarrow \text{Facesnv}(p_1, p_2, ele, \mathbf{IJ}, \mathbf{XY})$ 
16    Vetor da face
17     $\mathbf{F}_e \leftarrow \text{Fpbilinear}(dJ, \mathbf{n}, \mathbf{v}, p_1, p_2, p_0, t_e)$ 
18    Determina os gls do elemento
19     $\mathbf{g} \leftarrow \text{Montagls}(ele, \mathbf{IJ})$ 
20    Sobreposição no vetor global
21    for  $j \in \{1, \dots, 8\}$  do
22       $\mathbf{F}(\mathbf{g}(j)) \leftarrow \mathbf{F}(\mathbf{g}(j)) + \mathbf{F}_e(j)$ 
23   Retorna o vetor global de forças distribuídas
24   return  $\mathbf{F}$ 

```

17.5 Cálculo de tensões em um elemento


O Algoritmo 35 assume que os vetores \mathbf{X} e \mathbf{Y} já foram obtidos para o elemento e que os deslocamentos \mathbf{U}_e , de dimensão, 8×1 já foi extraído do vetor global \mathbf{U} .

Algoritmo 35: Retorna o vetor com as tensões em um ponto (r,s) de um elemento bilinear isoparamétrico de 4 nós.

```

1  CalculaTensaoElemento( $r,s,\mathbf{X},\mathbf{Y},E,\nu,\mathbf{U}_e$ )
2  Matriz constitutiva do elemento
3   $\mathbf{C}_\nu \leftarrow \text{MontaCEPT}(E,\nu)$ 
4  Calcula as derivadas das funções de interpolação
5   $\mathbf{dNrs} \leftarrow \text{MontadN}(r,s)$ 
6  Calcula a matriz Jacobiana
7   $\mathbf{J} \leftarrow \text{MontaJ}(\mathbf{dNrs},\mathbf{X},\mathbf{Y})$ 
8  Mapeia as derivadas para  $xy$ 
9   $\mathbf{dNxy} \leftarrow \text{CorrigedN}(\mathbf{dNrs},\mathbf{J})$ 
10 Monta a matriz  $\mathbf{B}$ 
11  $\mathbf{B} \leftarrow \text{MontaB}(\mathbf{dNxy})$ 
12 Calcula os produtos
13  $\boldsymbol{\sigma}_e \leftarrow \mathbf{C}_\nu \mathbf{B} \mathbf{U}_e$ 
14 Retorna o vetor
15 return  $\boldsymbol{\sigma}_e$ 
```

Exercício 17.1 Desenvolva um programa de computador com os algoritmos deste capítulo. Utilize os dados do Exemplo 15.8 para validar o seu programa. ■



Exemplos de implementação computacional na linguagem Julia

18	Treliças 2D	171
19	Pórtico plano	177
20	Bilinear isoparamétrico de 4 nós	183

18. Treliças 2D

Este capítulo tem como objetivo apresentar a implementação em Julia dos algoritmos discutidos no capítulo sobre a implementação computacional de treliças. A preocupação é apresentar uma listagem o mais parecida possível com os algoritmos, sem utilizar algumas facilidades da linguagem Julia.

Listing 18.1: Rotina principal

```
function main()

    # "Leitura" dos dados
    nn, XY, ne, IJ, E, A, na, AP, nf, FC = Dados()

    # Pre-processamento
    L = MontaL(nn,XY,ne,IJ)
    Θ = MontaΘ(nn,XY,ne,IJ)

    # Montagem da matriz global de rigidez
    K = RigidezGlobal(nn,ne,E,A,L,Θ,IJ)

    # Montagem do vetor global de forças
    F = ForcaCGlobal(nn,nf,FC)

    # Aplica as condições de contorno de Dirichlet
    K, F = AplicaCCH(nn,na,AP,K,F)

    # Soluciona o sistema
    U = K\F

    # Pos-processamento
    N = EsforçosInternos(ne,E,A,L,Θ,IJ,U)

end
```

Listing 18.2: Entrada de dados

```

function Dados()

# Numero de nos
nn = 3

# Coordenadas
XY = [0.0 0.0 ;
      1.0 1.0 ;
      2.0 0.0 ]

# Numero de elementos
ne = 3

# Conectividades
IJ = [1 2 ;
      2 3 ;
      1 3]

# Dados sobre o material de
# cada elemento
E = [210E9;
     210E9;
     210E9]

# Area de cada elemento
A = [0.01 ;
     0.01 ;
     0.01 ]

# Numero de apoios
na = 3

# Apoios
AP = [1 1 0.0 ;
      1 2 0.0 ;
      3 2 0.0 ]

# Numero de forças concentradas
nf = 2

# Informacao sobre as forças concentradas
FC = [2 2 -100 ;
      3 1 200 ]

# Retorna as informacoes
nn, XY, ne, IJ, E, A, na, AP, nf, FC
end

```

Listing 18.3: Calcula o comprimento dos elementos

```

function MontaL(nn,XY,ne,IJ)

# Aloca o vetor de saida
L = zeros(ne)

# Laço pelos elementos
for e=1:ne

# Nos do elemento
n1 = IJ[e,1]
n2 = IJ[e,2]

# Coordenadas do no 1
x1 = XY[n1,1]
y1 = XY[n1,2]

# Coordenadas do no 2

```

```

x2 = XY[n2,1]
y2 = XY[n2,2]

# Comprimento
L[e] = sqrt( (x2-x1)^2 + (y2-y1)^2 )

end

# Retorna o vetor L
return L

end

```

Listing 18.4: Calcula os ângulos dos elementos

```

function MontaTheta(nn,XY,ne,IJ)

# Aloca o vetor de saída
theta = zeros(ne)

# Laço pelos elementos
for e=1:ne

# Nos do elemento
n1 = IJ[e,1]
n2 = IJ[e,2]

# Coordenadas do nó 1
x1 = XY[n1,1]
y1 = XY[n1,2]

# Coordenadas do nó 2
x2 = XY[n2,1]
y2 = XY[n2,2]

# Calcula o ângulo
theta[e] = atan(y2 - y1, x2 - x1)

end

# Retorna o vetor
return theta

end

```

Listing 18.5: Montagem da matriz de rigidez local do elemento de barra

```

function RigidezBarra(E,A,L)

return (E*A/L)*[1 0 -1 0 ;
                 0 0 0 0 ;
                 -1 0 1 0 ;
                 0 0 0 0 ]

end

```

Listing 18.6: Montagem da matriz de rotação

```

function MatrizTBarra(theta)

# Calcula o seno e o cosseno

```

```

s = sin(θ)
c = cos(θ)

return [c s 0 0 ;
        -s c 0 0 ;
         0 0 c s ;
         0 0 -s c]

end

```

Listing 18.7: Determina os gls do elemento

```

function glsBarra(n1,n2)

return [2*(n1-1)+1 ; 2*(n1-1)+2; 2*(n2-1)+1; 2*(n2-1)+2]

end

```

Listing 18.8: Montagem da matriz de rigidez global da trelça

```

function RigidezGlobal(nn,ne,E,A,L,θ,IJ)

# Inicializa a matriz global
K = zeros(2*nn, 2*nn)

# Laço pelos elementos
for e=1:ne

    # Recupera as propriedades do elemento
    Ee = E[e]
    Ae = A[e]
    Le = L[e]
    θe = θ[e]

    # Monta a matriz de rigidez local
    Kel = RigidezBarra(Ee,Ae,Le)

    # Monta a matriz de rotação
    Te = MatrizTBarra(θe)

    # Transforma a matriz do elemento para o sistema global
    Keg = Te'*Kel*Te

    # Recupera os nos do elemento
    n1 = IJ[e,1]
    n2 = IJ[e,2]

    # Determina os gls do elemento
    ge = glsBarra(n1,n2)

    # Sobreponha na matriz global
    for i=1:4
        for j=1:4
            K[ge[i],ge[j]] = K[ge[i],ge[j]] + Keg[i,j]
        end
    end

end

# Retorna a matriz global de rigidez
return K

end

```

Listing 18.9: Montagem do vetor de forças globais concentradas

```

function ForcaGlobal(nn,nf,FC)

# Inicializa o vetor global
F = zeros(2*nn)

# Laço pelas linhas de FC
for i=1:nf

    # No
    no = Int(FC[i,1])

    # Grau de liberdade (local)
    gll = Int(FC[i,2])

    # Valor da força
    val = FC[i,3]

    # Grau de liberdade global
    glg = 2*(no-1) + gll

    # Adiciona ao vetor global
    F[glg] = F[glg] + val

end

# Retorna o vetor de forças globais
return F

end

```

Listing 18.10: Aplica as condições de contorno de Dirichlet - somente homogêneas

```

function AplicaCCH(nn,na,AP,K,F)

# Para cada informação de apoio, zera linhas e colunas
# colocando 1 na diagonal
for ap = 1:na

    # Recupera as informações
    no = Int(AP[ap,1])
    gl = Int(AP[ap,2])

    # Calcula o gl global
    glg = 2*(no-1)+gl

    # Zera linhas e colunas
    for i=1:2*nn
        K[glg,i] = 0
        K[i,glg] = 0
    end

    # Coloca 1 na diagonal de K
    K[glg,glg] = 1

    # E zera a posição em F
    F[glg] = 0

end

# Retorna K e F modificados
return K, F

end

```

Listing 18.11: Calcula os esforços internos

```
function EsforçosInternos(ne,E,A,L,θ,IJ,U)

# Inicializa o vetor de forças internas
N = zeros(ne)

# Laço pelos elementos
for e=1:ne

    # Recupera as propriedades do elemento
    Ee = E[e]
    Ae = A[e]
    Le = L[e]
    θe = θ[e]

    # Monta a matriz de rigidez local
    Kel = RigidezBarra(Ee,Ae,Le)

    # Monta a matriz de rotação
    Te = MatrizTBarra(θe)

    # Recupera os nos do elemento
    n1 = IJ[e,1]
    n2 = IJ[e,2]

    # Determina os gls do elemento
    ge = glsBarra(n1,n2)

    # Recupera os deslocamentos do elemento (ainda global)
    ueg = U[ge]

    # Transforma os deslocamentos para o sistema local
    uel = Te*ueg

    # Calcula o vetor de forças nodais
    Fel = Kel*uel

    # Armazena o esforço interno do elemento
    N[e] = -Fel[1]

end

# Retorna o vetor de forças internas
return N

end
```


19. Pórtico plano

Este capítulo tem como objetivo apresentar a implementação em Julia dos algoritmos para pórticos planos. A preocupação é apresentar uma listagem o mais parecida possível com os algoritmos, sem utilizar algumas facilidades da linguagem Julia. Algumas subrotinas são aproveitadas do capítulo de treliças.

Listing 19.1: Rotina principal

```
function main()

    # "Leitura" dos dados
    nn, XY, ne, IJ, E, Iz, A, na, AP, nf, FC, nq, FQ = Dados()

    # Pre-processamento
    L = MontaL(nn,XY,ne,IJ)
    Θ = MontaΘ(nn,XY,ne,IJ)

    # Montagem da matriz global de rigidez
    K = RigidezGlobalPP(nn,ne,E,Iz,A,L,Θ,IJ)

    # Montagem do vetor global de forças
    Fc = ForcaCGlobalPP(nn,nf,FC)

    # Montagem do vetor global de forças distribuidas
    Fq = FQPP(nn,L,Θ,IJ,nq,FQ)

    # Sobrepor as forças
    F = Fc + Fq

    # Aplica as condições de contorno de Dirichlet
    K, F = AplicaCCHPP(nn,na,AP,K,F)

    # Soluciona o sistema
    U = K\F

end
```

Listing 19.2: Entrada de dados

```

function Dados()

# Numero de nos
nn = 3

# Coordenadas
XY = [0.0 0.0 ;
      0.0 1.0 ;
      2.0 1.0 ]

# Numero de elementos
ne = 2

# Conectividades
IJ = [1 2 ;
      2 3 ]

# Dados sobre o material de
# cada elemento
E = [210E9;
     210E9]

# Momento de inercia de cada elemento
Iz = [8.33E-6;
      8.33E-6]

# Area de cada elemento
A = [0.01 ;
     0.01 ]

# Numero de apoios
na = 3

# Apoios
AP = [1 1 0.0 ;
      1 2 0.0 ;
      3 1 0.0 ]

# Numero de forças/momentos concentrados
nf = 2

# Informacao sobre as forças/momentos concentrados
FC = [2 1 -100 ;
      2 2 100 ]

# Numero de carregamentos (locais) distribuídos
nq = 1

# Informações sobre os carregamentos distribuidos
FQ = [2 100 100]

# Retorna as informações
nn, XY, ne, IJ, E, Iz, A, na, AP, nf, FC, nq, FQ

end

```

Listing 19.3: Montagem da matriz de rigidez local do elemento de pórtico plano

```

function RigidezPP(E,A,Iz,L)

# Rigidez axial
ka = E*A/L

# Termos de rigidez de viga
k1 = 12*E*Iz/L^3
k2 = 6*E*Iz/L^2

```

```

k3 = 4*E*Iz/L
k4 = 2*E*Iz/L

return [ka 0 0 -ka 0 0 ;
        0 k1 k2 0 -k1 k2 ;
        0 k2 k3 0 -k2 k4 ;
        -ka 0 0 ka 0 0 ;
        0 -k1 -k2 0 k1 -k2 ;
        0 k2 k4 0 -k2 k3 ]

end

```

Listing 19.4: Montagem da matriz de rotação para o elemento de pórtico plano

```

function MatrizTPP(theta)

# Calcula o seno e o cosseno
s = sin(theta)
c = cos(theta)

return [c s 0 0 0 0 ;
        -s c 0 0 0 0 ;
        0 0 1 0 0 0 ;
        0 0 0 c s 0 ;
        0 0 0 -s c 0 ;
        0 0 0 0 0 1]

end

```

Listing 19.5: Determina os gls do elemento de pórtico plano

```

function glsPP(n1,n2)

return [3*(n1-1)+1 ; 3*(n1-1)+2; 3*(n1-1)+3;
        3*(n2-1)+1 ; 3*(n2-1)+2; 3*(n2-1)+3]

end

```

Listing 19.6: Montagem da matriz de rigidez global do pórtico plano

```

function RigidezGlobalPP(nn,ne,E,Iz,A,L,theta,IJ)

# Inicializa a matriz global
K = zeros(3*nn, 3*nn)

# Laço pelos elementos
for e=1:ne

# Recupera as propriedades do elemento
Ee = E[e]
Ae = A[e]
Ie = Iz[e]
Le = L[e]
thetae = theta[e]

# Monta a matriz de rigidez local
Kel = RigidezPP(Ee,Ie,Ae,Le)

# Monta a matriz de rotação
Te = MatrizTPP(thetae)

```

```

# Transforma a matriz do elemento para o sistema global
Keg = Te'*Kel*Te

# Recupera os nos do elemento
n1 = IJ[e,1]
n2 = IJ[e,2]

# Determina os gls do elemento
ge = glsPP(n1,n2)

# Sobrepor na matriz global
for i=1:6
    for j=1:6
        K[ge[i],ge[j]] = K[ge[i],ge[j]] + Keg[i,j]
    end
end

end

# Retorna a matriz global de rigidez
return K

end

```

Listing 19.7: Montagem do vetor de forças globais concentradas para pórtico plano

```

function ForcaCGlobalPP(nn,nf,FC)

# Inicializa o vetor global
F = zeros(3*nn)

# Laço pelas linhas de FC
for i=1:nf

    # No
    no = Int(FC[i,1])

    # Grau de liberdade (local)
    gll = Int(FC[i,2])

    # Valor da força
    val = FC[i,3]

    # Grau de liberdade global
    glg = 3*(no-1) + gll

    # Adiciona ao vetor global
    F[glg] = F[glg] + val

end

# Retorna o vetor de forças globais
return F

end

```

Listing 19.8: Calcula o vetor de forças consistentes para pórtico plano

```

# Vetor de carregamento distribuído local
function FqPP(Le,q1,q2)

return [ (3*Le*q2 + 7*Le*q1)/20 ;

```

```

        (2*(Le^2)*q2 + 3*(Le^2)*q1)/60 ;
    0 ;
    (7*Le*q2 + 3*Le*q1)/20 ;
    -(3*(Le^2)*q2 + 2*(Le^2)*q1)/60 ;
    0]
end

```

Listing 19.9: Monta o vetor global de forças distribuídas

```

function FQPP(nn,L,θ,IJ,nq,FQ)

# Inicializa o vetor global
F = zeros(3*nn)

# Laço pelas linhas de FQ
for i=1:nq

    # Recupera o elemento submetido ao carregamento
    e = Int(FQ[i,1])

    # Recupera os valores de q1 e q2
    q1 = FQ[i,2]
    q2 = FQ[i,3]

    # Recupera comprimento e angulo
    Le = L[e]
    θe = θ[e]

    # Vetor de forças local
    Fl = FqPP(Le,q1,q2)

    # Monta a matriz de rotaçao
    Te = MatrizTPP(θe)

    # Transforma a matriz do elemento para o sistema global
    Fg = Te'*Fl

    # Recupera os nos do elemento
    n1 = IJ[e,1]
    n2 = IJ[e,2]

    # Determina os gls do elemento
    ge = glsPP(n1,n2)

    # Sobrepe no vetor global
    for j=1:6
        F[ge[j]] = F[ge[j]] + Fg[j]
    end

end

# Retorna o vetor global
return F

end

```

Listing 19.10: Aplica as condições de contorno de Dirichlet - somente homogêneas para pórtico plano

```

function AplicaCCHPP(nn,na,AP,K,F)

# Para cada informação de apoio, zera linhas e colunas
# colocando 1 na diagonal

```

```
for ap = 1:na

    # Recupera as informações
    no = Int(AP[ap,1])
    gl = Int(AP[ap,2])

    # Calcula o gl global
    glg = 3*(no-1)+gl

    # Zera linhas e colunas
    for i=1:3*nn
        K[glg,i] = 0
        K[i,glg] = 0
    end

    # Coloca 1 na diagonal de K
    K[glg,glg] = 1

    # E zera a posição em F
    F[glg] = 0

end

# Retorna K e F modificados
return K, F

end
```

20. Bilinear isoparamétrico de 4 nós

Este capítulo tem como objetivo apresentar a implementação em Julia dos algoritmos do Cap. 17. A preocupação é apresentar uma listagem o mais parecida possível com os algoritmos, sem utilizar algumas facilidades da linguagem Julia.

Listing 20.1: Programa principal

```
# Rotina principal
function main()

    # "Leitura" dos dados
    nn, XY, ne, IJ, MAT, ESP, na, AP, np, P = Dados()

    # Montagem da matriz global de rigidez
    K = RigidezGlobal(nn,ne,MAT,ESP,XY,IJ)

    # Montagem do vetor global de forças
    F = ForcapGlobal(nn,ESP,XY,IJ,np,P)

    # Aplica as condições de contorno de Dirichlet
    K, F = AplicaCCH(nn,na,AP,K,F)

    # Soluciona o sistema
    U = K\F

end
```

Listing 20.2: Exemplo de entrada de dados

```
#
# Entrada de dados
#
function Dados()

    # Numero de nós
```

```

nn = 6

# Coordenadas
XY = [0.0 0.0 ;
      0.5 0.0 ;
      1.0 0.0 ;
      0.0 0.1 ;
      0.5 0.1 ;
      1.0 0.1]

# Número de elementos
ne = 2

# Conectividades
IJ = [1 2 5 4 ;
      2 3 6 5 ]

# Dados sobre o material de
# cada elemento
MAT = [1E9 0.3 ;
       1E9 0.3 ]

# Espessura de cada elemento
ESP = [0.01 ;
       0.01 ]

# Numero de apoios
na = 5

# Apoios
AP = [1 1 0.0 ;
      1 2 0.0 ;
      2 2 0.0 ;
      3 2 0.0 ;
      4 1 0.0 ]

# Numero de faces sobre pressao
# normal
np = 1

# Informacao sobre as pressoes
P = [2 2 1000]

# Retorna as informacoes
nn, XY, ne, IJ, MAT, ESP, na, AP, np, P
end

```

Listing 20.3: Monta a matriz constitutiva em EPT

```

function MontaCEPT(E,v)

# Calcula o termo em comum
c = E/(1-v^2)

# Define a matriz
C = [ c      c*v      0 ;
      c*v     c      0 ;
      0      0  c*(1-v)/2]

# Retorna a matriz
return C

end

```


Listing 20.4: Derivada das funções de interpolação

```
function MontadN(r,s)

# Matriz com as derivadas em relacao a r (primeira linha)
# e a s (segunda linha)
dNrs = (1/4)*[ s-1    1-s    1+s    -(1+s) ;
               r-1    -(1+r)  1+r     1-r ]

# Retorna a matriz
return dNrs

end
```

Listing 20.5: Matriz Jacobiana

```
function MontaJ(dNrs,X,Y)

# Aloca a matriz
J = zeros(2,2)

# Laco para montar a matriz
for i=1:4
    J[1,1] = J[1,1] + X[i]*dNrs[1,i]
    J[1,2] = J[1,2] + Y[i]*dNrs[1,i]
    J[2,1] = J[2,1] + X[i]*dNrs[2,i]
    J[2,2] = J[2,2] + Y[i]*dNrs[2,i]
end

# Retorna a matriz
return J

end
```

Listing 20.6: Corrige as derivadas de rs para xy

```
function CorrigedN(dNrs,J)

# Calcula a correcao
dNxy = J\dNrs

# Retorna a matriz
return dNxy

end
```

Listing 20.7: Matriz B

```
function MontaB(dNxy)

# Aloca a matriz
B = zeros(3,8)

# Indicador de início de bloco
c = 1

# Laco para montar a matriz
for i=1:4

# Copia as derivadas para as posicoes no bloco
B[1,c] = dNxy[1,i]
```

```

    B[2,c+1] = dNxy[2,i]
    B[3,c]   = B[2,c+1]
    B[3,c+1] = B[2,c]

    # Incrementa o contador
    c = c + 2

end

# Retorna a matriz
return B

end

```

Listing 20.8: Monta a matriz de rigidez do elemento bilinear isoparamétrico de 4 nós

```

function MontaKe(X,Y,E,v,te)

    # Aloca a matriz
    Ke = zeros(8,8)

    # Matriz constitutiva do elemento
    C = MontaCEPT(E,v)

    # Define os pontos de Gauss
    pg = (1/sqrt(3))*[-1  1  1 -1 ;
                     -1 -1  1  1 ]

    # Laco pelos pontos de Gauss
    for p=1:4

        # Coordenadas do PG
        r = pg[1,p]
        s = pg[2,p]

        # Derivadas da funcoes de interpolacao
        dNrs = MontadN(r,s)

        # Matriz Jacobiana
        J = MontaJ(dNrs,X,Y)

        # Determinante de J
        dJ = det(J)

        # Mapeia as derivadas para xy
        dNxy = CorrigedN(dNrs,J)

        # Monta a matriz B
        B = MontaB(dNxy)

        # Acumula na matriz de rigidez
        Ke .= Ke .+ B'*C*B*dJ*te

    end

    # Retorna a matriz
    return Ke

end

```

Listing 20.9: Monta os vetores com as coordenadas do elemento

```

function MontaXY(e,IJ,XY)

```

```

# Inicializa os vetores
X = zeros(4)
Y = zeros(4)

# Preenche os vetores com as coordenadas
for i=1:4

    # No
    no = IJ[e,i]

    # Coordenadas
    X[i] = XY[no,1]
    Y[i] = XY[no,2]

end

# Retorna os vetores
return X,Y

end

```

Listing 20.10: Monta o vetor com os gls do elemento

```

function Montagls(e,IJ)

# Inicializa o vetor
gls = zeros(Int,8)

# Contador
c = 1

# Calcula as posicoes
for i=1:4

    # No
    no = IJ[e,i]

    # Laco pelos gls
    for j=1:2

        # Calcula o gl
        gls[c] = 2*(no-1)+j

        # Incrementa o contador
        c = c + 1

    end

end

# Retorna o vetor com os gls
return gls

end

```

Listing 20.11: Monta a matriz de rigidez global

```

function RigidezGlobal(nn,ne,MAT,ESP,XY,IJ)

# Inicializa a matriz global
K = zeros(2*nn,2*nn)

# Laco pelos elementos da malha
for e=1:ne

```

```

# Recupera as propriedades do elemento
E = MAT[e,1]
v = MAT[e,2]
te = ESP[e]

# Recupera as coordenadas do elemento
X,Y = MontaXY(e,IJ,XY)

# Monta a matriz de rigidez local
Ke = MontaKe(X,Y,E,v,te)

# Determina os gls globais do elemento
gls = Montagls(e,IJ)

# Sobrepor a matriz local na rigidez global
for i=1:8
    for j=1:8
        K[gls[i],gls[j]] = K[gls[i],gls[j]] + Ke[i,j]
    end
end

end

# Retorna a matriz global
return K

end

```

Listing 20.12: Retorna os ponteiros para os cálculos nas faces do elemento

```

function Faces(face)

# Retorna 2 indices para o cálculo dos vetores locais da face
if face==1
    p1 = 1
    p2 = 2
elseif face==2
    p1 = 2
    p2 = 3
elseif face==3
    p1 = 3
    p2 = 4
else
    p1 = 4
    p2 = 1
end

# Retorna as posicoes
return p1,p2

end

```

Listing 20.13: Retorna os ponteiros para os cálculos nas faces do elemento

```

function MontaN(ζ,p1,p2)

# Aloca a matriz N
N = zeros(2,8)

# Funcoes de interpolacao nao nulas
N1 = (1-ζ)/2
N2 = (1+ζ)/2

# Posiciona na matriz N

```

```

N[1,2*p1 - 1] = N1
N[2,2*p1]     = N1

N[1,2*p2 - 1] = N2
N[2,2*p2]     = N2

# Retorna a matriz
return N

end

```

Listing 20.14: Retorna o dJ e os vetores do sistema local de uma aresta de um elemento

```

function Facesnv(p1,p2,e,IJ,XY)

# Recupera os nos da face
n1 = IJ[e,p1]
n2 = IJ[e,p2]

# Recupera as coordenadas
x1 = XY[n1,1]
y1 = XY[n1,2]
x2 = XY[n2,1]
y2 = XY[n2,2]

# Deltas
Δx = x2 - x1
Δy = y2 - y1

# Comprimento
L = sqrt(Δx^2 + Δy^2)

# Determinante do Jacobiano
dJ = L/2

# Vetor tangente a face
v = (1/L)*[Δx ; Δy]

# Vetor normal a face
n = (1/L)*[Δy ; -Δx]

# Retorna dJ e os vetores
return dJ, n, v

end

```

Listing 20.15: Retorna o dJ e os vetores do sistema local de uma aresta de um elemento

```

function Fpbilinear(dJ,n,v,p1,p2,p0,te)

# Pontos de Gauss
pg = (1/sqrt(3))*[-1;1]

# Aloca o vetor
F = zeros(8)

# Laco pelos pontos de Gauss
for i=1:2

# Recupera o ponto de Gauss
ζ = pg[i]

# Monta a matriz N
N = MontaN(ζ,p1,p2)

```

```

        # Acumula o vetor
        F .= F .+ p0*N'*n*dJ*te

    end

    # Retorna o vetor
    return F

end

```

Listing 20.16: Montagem do vetor global de forças distribuídas

```

function ForcapGlobal(nn,ESP,XY,IJ,np,P)

    # Inicializa o vetor global
    F = zeros(2*nn)

    # Laco pelas linhas de P
    for i=1:np

        # Recupera os dados do carregamento
        ele = Int(P[i,1])
        face = Int(P[i,2])
        p0 = P[i,3]

        # Espessura do elemento
        te = ESP[ele]

        # Ponteiros para a face
        p1,p2 = Faces(face)

        # Vetores da face
        dJ,n,v = Facesnv(p1,p2,ele,IJ,XY)

        # Vetor de forca da face
        Fe = Fpbilinear(dJ,n,v,p1,p2,p0,te)

        # Graus de liberdade globais do elemento
        gls = Montagls(ele,IJ)

        # Sobrepor no vetor global
        for j=1:8
            F[gls[j]] = F[gls[j]] + Fe[j]
        end

    end

    # Retorna o vetor global de forças distribuidas
    return F

end

```

Listing 20.17: Aplica condições de contorno de Dirichlet homogêneas

```

function AplicaCCH(nn,na,AP,K,F)

    # Para cada informacao de apoio, zera linhas e colunas
    # colocando 1 na diagonal
    for ap = 1:na

        # Recupera as informacoes
        no = Int(AP[ap,1])
        gl = Int(AP[ap,2])
    end

```

```
# Calcula o gl global
glg = 2*(no-1)+gl

# Zera linhas e colunas
for i=1:2*nn
    K[glg,i] = 0
    K[i,glg] = 0
end

# Coloca 1 na diagonal de K
K[glg,glg] = 1

# E zera a posicao em F
F[glg] = 0

end

# Retorna K e F modificados
return K, F

end
```

