



Introdução ao Método de Elementos Finitos

Parte IV

Isaac P. Santos (<https://blog.ufes.br/isaacsantos>)

Universidade Federal do Espírito Santo - UFES

18 a 22 de fevereiro de 2018

Sumário

- ▶ Introdução
- ▶ Interpolação unidimensional
- ▶ Método de elementos finitos unidimensional
- ▶ Interpolação bidimensional
- ▶ Método de elementos finitos bidimensional

Pré-requisitos

- ▶ Cálculo diferencial e integral
- ▶ Álgebra linear
- ▶ Programação
- ▶ Noções de equações diferenciais

Nível de graduação!

Introdução

Neste capítulo será apresentado o método de elementos finitos para resolver numericamente equações diferenciais parciais em duas dimensões. A ideia básica é semelhante ao caso unidimensional. Primeiro, deve-se reescrever a formulação variacional do problema e a seguir, apresentar um MEF que consiste em determinar uma solução aproximada no espaço de funções contínuas e lineares por partes.

Revisão de Cálculo Vetorial: Operadores Gradiente, Divergência e Laplaciano

O gradiente de um campo escalar fornece um campo vetorial. Se $u : \Omega \rightarrow \mathbb{R}$ é um campo escalar, então o gradiente da função $u(x, y)$ é definido por

$$\nabla u = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix}.$$

O operador de Laplace ou laplaciano de um campo escalar fornece um campo escalar. O laplaciano da função escalar $u(x, y)$ é definido por

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

Observação

Não se utiliza o operador divergência em campos escalares.

Se $\mathbf{F}(x, y) = \begin{bmatrix} F_1(x, y) \\ F_2(x, y) \end{bmatrix}$ é um campo vetorial, então o operador divergência de \mathbf{F} é definido por

$$\nabla \cdot \mathbf{F} = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y}.$$

Pode-se definir os operadores gradiente e laplaciano de campos vetoriais.

Propriedades Importantes

a) Se φ é um campo escalar e \mathbf{F} é um campo vetorial, então

$$\nabla \cdot (\varphi \mathbf{F}) = \nabla \varphi \cdot \mathbf{F} + \varphi \nabla \cdot \mathbf{F}.$$

b) A divergência do gradiente é o laplaciano: se u é uma função escalar, então

$$\nabla \cdot (\nabla u) = \Delta u.$$

De fato,

$$\nabla \cdot (\nabla u) = \nabla \cdot \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \Delta u.$$

Se o vetor $\mathbf{n}(x, y)$ é o vetor normal unitário exterior à fronteira $\partial\Omega$ de Ω em $(x, y) \in \partial\Omega$ e se ∇u é o gradiente de u ,

$$\nabla u = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix},$$

então

$$\nabla u \cdot \mathbf{n} = \frac{\partial u}{\partial \mathbf{n}}$$

é a *derivada normal* de u em $\partial\Omega$ (derivada direcional - na direção de \mathbf{n}).

Teorema da Divergência e Identidade de Green

Teorema (Divergência)

Se $\Omega \subset \mathbb{R}^2$ é um domínio com fronteira $\Gamma = \partial\Omega$ suave ou suave por partes e $\mathbf{F} : \bar{\Omega} \rightarrow \mathbb{R}^2$ é um campo vetorial, então

$$\int_{\Omega} \nabla \cdot \mathbf{F} d\Omega = \int_{\Gamma} \mathbf{F} \cdot \mathbf{n} d\Gamma, \quad (1)$$

onde \mathbf{n} é o vetor unitário à fronteira Γ na direção exterior.

Primeira Identidade de Green

Suponha que u e v sejam funções escalares (diferenciáveis) definidas em $\Omega \subset \mathbb{R}^2$:

$$\int_{\Omega} v \Delta u d\Omega + \int_{\Omega} \nabla v \cdot \nabla u d\Omega = \int_{\Gamma} v \frac{\partial u}{\partial \mathbf{n}} d\Gamma.$$

Problema Modelo

O problema a ser abordado consiste na equação diferencial parcial

$$-\nabla \cdot (\epsilon \nabla u) = f, \quad \text{em } \Omega, \quad (3)$$

$$u = g, \quad \text{em } \Gamma. \quad (4)$$

onde $\epsilon > 0$ é um parâmetro dado e f é uma função conhecida.

Este é um problema com condições de contorno de Dirichlet. Por simplicidade, consideraremos $g = 0$.

Se $f \in C^0(\bar{\Omega})$ e u resolve o problema (3)-(4) então é natural esperar que $u \in C^2(\bar{\Omega})$. Além disso, u é igual a zero na fronteira Γ .

A solução do problema (3)-(4) é procurada no espaço de funções

$$C_D^2 = \{u \in C^2(\bar{\Omega}); \quad u = 0 \text{ em } \Gamma\}.$$

Formulação Variacional

Multiplicando ambos os lados da equação (3) por uma função v (chamada *função teste*), obtém-se

$$-\nabla \cdot (\epsilon \nabla u) v = f v, \quad \text{em } \Omega.$$

Agora, integrando o resultado em Ω , resulta em

$$-\int_{\Omega} \nabla \cdot (\epsilon \nabla u) v d\Omega = \int_{\Omega} f v d\Omega. \quad (6)$$

É suficiente definir o espaço das funções testes como sendo C_D^2 , isto é, $v \in C_D^2$.

Formulação Variacional

Como $\epsilon > 0$ é uma constante, então

$$\nabla \cdot (\epsilon \nabla u) = \epsilon \nabla \cdot (\nabla u) = \epsilon \Delta u.$$

Usando a primeira identidade de Green (2), tem-se

$$-\int_{\Omega} \nabla \cdot (\epsilon \nabla u) v d\Omega = -\int_{\Omega} \epsilon \Delta u v d\Omega \quad (7)$$

$$= \int_{\Omega} \epsilon \nabla u \cdot \nabla v d\Omega - \int_{\Gamma} \epsilon v \nabla u \cdot \mathbf{n} d\Gamma. \quad (8)$$

Substituindo esse termo na equação (6), obtemos

$$\int_{\Omega} \epsilon \nabla u \cdot \nabla v d\Omega - \int_{\Gamma} \epsilon v \nabla u \cdot \mathbf{n} d\Gamma = \int_{\Omega} f v d\Omega. \quad (9)$$

Como $v \in C_D^2$, então $v = 0$ em Γ . Portanto a formulação variacional ou forma fraca do problema (3)-(4) consiste em achar $u \in C_D^2$ tal que

$$\int_{\Omega} \epsilon \nabla u \cdot \nabla v d\Omega = \int_{\Omega} f v d\Omega, \quad \forall v \in C_D^2. \quad (10)$$

A forma variacional do problema (3)-(4) pode ser definida em termos do espaço de Sobolev $H_0^1(\Omega)$: achar $u \in H_0^1(\Omega)$ tal que

$$\int_{\Omega} \epsilon \nabla u \cdot \nabla v d\Omega = \int_{\Omega} f v d\Omega, \quad \forall v \in H_0^1(\Omega). \quad (11)$$

Neste caso, o lado direito f de (3) é assumido pertencer ao espaço $L^2(\Omega)$.

A Equação de Convecção-Difusão-Reação

A equação de convecção-difusão-reação consiste em achar a função $u = u(x, y)$, tal que

$$\nabla \cdot (-\epsilon \nabla u + u\beta) + \sigma u = f, \text{ em } \Omega; \quad (12)$$

$$u = g, \text{ em } \Gamma_D; \quad (13)$$

$$\epsilon(\nabla u \cdot \mathbf{n}) = q, \text{ em } \Gamma_N, \quad (14)$$

onde $\epsilon > 0$ é o coeficiente de difusão, considerado constante, $\sigma \geq 0$ é o coeficiente de reação, $\beta : \Omega \times \Omega \rightarrow \mathbb{R}^2$ é o campo de velocidades, $f : \Omega \rightarrow \mathbb{R}$ é o termo de fonte,

Γ_D representa a parte da fronteira onde as condições de Dirichlet são prescritas (a função u é conhecida em Γ_D),

$\Gamma_N = \Gamma \setminus \Gamma_D$ é a parte da fronteira onde as condições de Neumann são conhecidas (o fluxo de u é prescrito).

A função u pode representar a concentração de uma substância no ponto $(x, y) \in \Omega$ em um processo de transporte estacionário. Por exemplo, u pode descrever a concentração de um contaminante em um rio, de um fármaco na corrente sanguínea, etc.

Vamos assumir que o campo de velocidades é incompressível, isto é,

$$\nabla \cdot \beta = 0.$$

Segue-se que,

$$\nabla \cdot (u\beta) = u\nabla \cdot \beta + \beta \cdot \nabla u = \beta \cdot \nabla u.$$

Portanto, o problema (12)-(14) pode ser reescrito como,

$$-\epsilon \Delta u + \beta \cdot \nabla u + \sigma u = f, \text{ em } \Omega; \quad (15)$$

$$u = g \text{ em } \Gamma_D; \quad (16)$$

$$\epsilon(\nabla u \cdot \eta) = q, \text{ em } \Gamma_N. \quad (17)$$

Nessa equação, $-\epsilon \Delta u$ é o termo difusivo, $\beta \cdot \nabla u$ é o termo convectivo e σu é o termo reativo.

Dado o espaço

$$V_g = \{v \in H^1(\Omega); v = g \text{ em } \Gamma_D\},$$

o problema variacional consiste em achar $u \in V_g$ tal que

$$\int_{\Omega} (\epsilon \nabla u \cdot \nabla v + (\beta \cdot \nabla u)v + \sigma uv) d\Omega = \int_{\Omega} f v d\Omega + \int_{\Gamma_N} q v d\Gamma, \quad \forall v \in V_0. \quad (18)$$

Fazendo

$$\begin{aligned} a(u, v) &= \int_{\Omega} \left(\epsilon \nabla u \cdot \nabla v + (\beta \cdot \nabla u) v + \sigma u v \right) d\Omega; \\ \ell(v) &= \int_{\Omega} f v d\Omega + \int_{\Gamma_N} q v d\Gamma, \end{aligned}$$

o problema (19) pode ser escrito na forma compacta: achar $u \in V_g$ tal que

$$a(u, v) = \ell(v), \quad \forall v \in V_0. \tag{19}$$

Método de Elementos Finitos

O método de Galerkin consiste em converter o problema variacional contínuo, descrito em espaços de dimensão infinita, em um problema discreto, cujas funções (*admissíveis* e *teste*) pertençam ao mesmo espaço de dimensão finita.

O método de elementos finitos é uma metodologia eficiente de implementar computacionalmente o método de Galerkin.

Por simplicidade, vamos considerar que as condições de contorno de Dirichlet sejam homogêneas em (15)-(17), isto é, $g = 0$ em Γ_D . Neste caso, os espaços V_g e V_0 são iguais.

Considerando $V_{0,h}$ subespaço de dimensão finita de V_0 , o método de Galerkin consiste em restringir o problema variacional (19) a esse espaço, isto é, achar $u_h \in V_{0,h}$, tal que

$$a(u_h, v_h) = \ell(v_h), \quad \forall v_h \in V_{0,h}, \quad (20)$$

onde u_h e v_h são aproximações de u e v , respectivamente.

Obviamente, as funções u_h e v_h são construídas através de combinações lineares das funções pertencentes à base escolhida do espaço $V_{0,h}$.

Podemos dizer que o método de elementos finitos é um procedimento eficiente para construir o espaço de aproximação $V_{0,h}$, e consequentemente, resolver o problema discreto (20).

O espaço $V_{0,h}$ é o espaço das funções contínuas em Ω e polinomiais em cada elemento (subdomínio) da discretização.

Para apresentar o MEF, considere a partição \mathcal{T}_h do domínio Ω em elementos triangulares (ver Fig. 1).

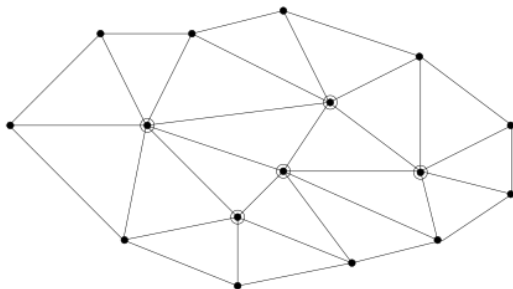


Figure: Triangulação do domínio $\bar{\Omega}$ - Figura extraída de ?

Podemos definir o espaço $V_{0,h}$ da seguinte forma,

$$V_{0,h} = \{v_h \in H^1(\Omega) \text{ tal que } u|_T \in \mathbb{P}_r, \forall T \in \mathcal{T}_h \text{ e } u|_{\Gamma_D} = 0\},$$

onde \mathbb{P}_r é o espaço de polinômios de grau $\leq r$ e cujos coeficientes são reais, definidos no elemento $T \in \mathcal{T}_h$. Definimos h como o parâmetro característico da malha. Como estamos considerando polinômios lineares em T , então $r = 1$.

Observação

Os espaços V_0 e $V_{h,0}$ são vetoriais, enquanto que V_g , com $g \neq 0$, e $V_{h,g}$ (onde $V_{h,g} \subset V_g$ tem dimensão finita) não são. De fato, se $u_1, u_2 \in V_g$, então $u_1|_{\Gamma_D} + u_2|_{\Gamma_D} = 2g \notin V_g$. Para o método de elementos finitos de Galerkin, considera-se o mesmo espaço de aproximação para as funções admissíveis e peso. Então para aplicarmos o método de elementos finitos no caso $g \neq 0$ na fronteira de Dirichlet, estendemos $g \in C(\Gamma_D)$ para o resto da fronteira Γ introduzindo a função $\tilde{g} \in C(\Gamma)$ tal que

$$\tilde{g} = g \quad \text{em } \Gamma_D.$$

Então podemos considerar

$$u_h = \tilde{u}_h + G_h,$$

tal que $\tilde{u}_h \in V_{h,0}$ e $G_h \in H^1(\Omega)$ satisfazendo $G_h = \tilde{g}$ em Γ .

Observação

Usando esse argumento, o problema discreto associado a (19) é escrito da seguinte forma: achar $\tilde{u}_h = u_h - G_h \in V_{h,0}$, tal que

$$a(\tilde{u}_h, v_h) = \tilde{\ell}(v_h), \quad \forall v_h \in V_{h,0}, \quad (21)$$

onde,

$$\tilde{\ell}(v_h) = \ell(v_h) - a(G_h, v_h). \quad (22)$$

Neste caso, os espaços das funções admissíveis e peso são iguais na formulação (21).

Para determinar o espaço V_h , devemos construir uma base para esse espaço. Seja

$$B = \{\varphi_1, \dots, \varphi_n\}$$

uma base para o espaço V_h associada a partição/triangulação \mathcal{T}_h . Denotamos por h_e o diâmetro (maior lado) do elemento triangular T e definimos $h = \max\{h_e\}$, o parâmetro característico da malha \mathcal{T}_h . A cada vértice \mathbf{x}_j não prescrito com condições de contorno de Dirichlet (marcados com o símbolo \odot na figura) associamos uma função $\varphi_j \in B$ satisfazendo

$$\varphi_j(\mathbf{x}_i) = \begin{cases} 1, & \text{se } i = j; \\ 0, & \text{se } i \neq j, \end{cases} \quad (23)$$

isto é, a função φ_j é igual a 1 no nó \mathbf{x}_j e 0 nos outros nós da partição; φ_j é uma função contínua em $\overline{\Omega}$ e afim em cada triângulo T .

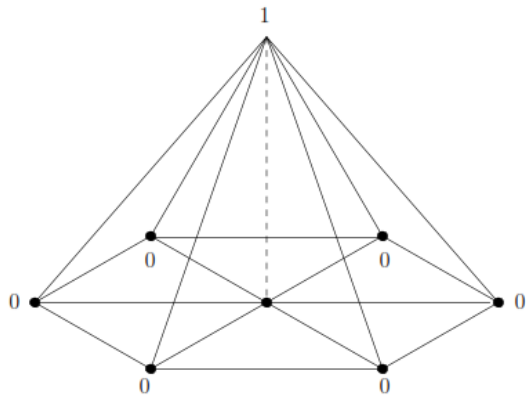


Figure: A função base φ_j - Figura extraída de ?

As funções $\varphi_1, \dots, \varphi_n$ são linearmente independentes e geram o espaço V_h . Portanto, uma função arbitrária $u_h \in V_h$ pode ser escrita como uma combinação linear dessas funções, isto é,

$$u_h(\mathbf{x}) = \sum_{j=1}^n \alpha_j \varphi_j(\mathbf{x}). \quad (24)$$

A propriedade (23) implica que em cada vértice \mathbf{x}_i , $i = 1, 2, \dots, n$ da malha,

$$u_h(\mathbf{x}_i, t) = \alpha_1 \underbrace{\varphi_1(\mathbf{x}_i)}_{=0} + \dots + \alpha_{i-1} \underbrace{\varphi_{i-1}(\mathbf{x}_i)}_{=0} + \alpha_i \underbrace{\varphi_i(\mathbf{x}_i)}_{=1} + \alpha_{i+1} \underbrace{\varphi_{i+1}(\mathbf{x}_i)}_{=0} + \dots + \alpha_n \underbrace{\varphi_n(\mathbf{x}_i)}_{=0},$$

ou seja, as coordenadas $\alpha_1, \dots, \alpha_n$ da combinação linear são dadas por

$$\alpha_i = u_h(\mathbf{x}_i).$$

Uma base satisfazendo (23) é chamada *base nodal* ou *base lagrangiana*. Substituindo (24) em (20) obtemos,

$$a\left(\sum_{j=1}^n \alpha_j \varphi_j(\mathbf{x}), v_h\right) = \ell(v_h), \quad \forall v_h \in V_h,$$

ou

$$\sum_{j=1}^n a(\varphi_j(\mathbf{x}), v_h) \alpha_j = \ell(v_h), \quad \forall v_h \in V_h.$$

Como a última equação é válida para toda função $v_h \in V_h$, podemos escolher $v_h = \varphi_i$, para $i = 1, 2, \dots, n$, obtendo

$$\sum_{j=1}^n a(\varphi_j, \varphi_i) \alpha_j = \ell(\varphi_i), \quad i = 1, 2, \dots, n,$$

que é um sistema de equações lineares, escrito na forma matricial como

$$\mathbf{K} \mathbf{U} = \mathbf{F} \tag{25}$$

onde $\mathbf{K} = [K_{ij}]_{n \times n} \in \mathbb{R}^{n \times n}$ é a matriz com componentes

$$K_{ij} = a(\varphi_i, \varphi_j) = \int_{\Omega} \left(\epsilon \nabla \varphi_i \cdot \nabla \varphi_j + (\boldsymbol{\beta} \cdot \nabla \varphi_j) \varphi_i + \sigma \varphi_i \varphi_j \right) d\mathbf{x},$$

$\mathbf{F} = [F_i]_n \in \mathbb{R}^n$ é o vetor com componentes

$$F_i = (f, \varphi_i) + (q, \varphi_i)_{\Gamma_N} = \int_{\Omega} f \varphi_i d\mathbf{x} + \int_{\Gamma_N} f \varphi_i ds,$$

e $U = [u_j]_n \in \mathbb{R}^n$ é o vetor incógnita com

$$u_j = \alpha_j = u_h(\mathbf{x}_j).$$

Resolvendo este sistema de equações lineares, obtém-se a solução aproximada do problema pelo MEF.

Montagem do Sistema Global

A matriz \mathbf{K} do sistema (25) é chamada de *matriz global*, enquanto o vetor \mathbf{F} é chamado de *vetor global*. A implementação computacional do método de elementos finitos é realizada a partir da construção de matrizes e vetores associados a cada elemento da malha, chamados de matrizes e vetores *locais*.

A matriz global \mathbf{K} e o vetor global \mathbf{F} são gerados a partir das matrizes e vetores locais.

Apresentaremos as matrizes locais associadas a cada elemento e um algoritmo para montar o sistema global (25).

As integrais definidas sobre o domínio Ω são divididas em integrais calculadas sobre os elementos. Por exemplo, os elementos da matriz \mathbf{K} podem ser calculados da seguinte forma,

$$\begin{aligned} K_{ij} &= \int_{\Omega} \left(\epsilon \nabla \varphi_i \cdot \nabla \varphi_j + (\boldsymbol{\beta} \cdot \nabla \varphi_j) \varphi_i + \sigma \varphi_i \varphi_j \right) d\mathbf{x} \\ &= \sum_{e=1}^{n_{el}} \int_T \left(\epsilon \nabla \varphi_i \cdot \nabla \varphi_j + (\boldsymbol{\beta} \cdot \nabla \varphi_j) \varphi_i + \sigma \varphi_i \varphi_j \right) d\mathbf{x}, \end{aligned}$$

$$\int_T \left(\epsilon \nabla \varphi_i \cdot \nabla \varphi_j + (\beta \cdot \nabla \varphi_j) \varphi_i + \sigma \varphi_i \varphi_j \right) d\mathbf{x} = 0,$$

se φ_i e/ou φ_j não forem funções associadas aos pontos nodais do elemento T .

Então, é suficiente calcular somente as integrais que não se anulam sobre os elementos e armazenar os seus valores em uma matriz 3×3 , conhecida como *matriz local*. Neste caso, podemos tratar o problema localmente, construindo as matrizes e vetores locais, para depois construir o problema global.

Para a descrição do problema local, considere a base nodal local

$$\{\lambda_1^e(x, y), \lambda_2^e(x, y), \lambda_3^e(x, y)\}$$

descrita no Capítulo 3.

Matriz Local K^e

A matriz global \mathbf{K} é formada a partir das contribuições das matrizes locais D^e , C^e e R^e , associadas aos termos de difusão, convecção e reação, respectivamente. Se K^e é a matriz local associada a \mathbf{K} , então

$$K^e = D^e + C^e + R^e. \quad (26)$$

A matriz local do termo de difusão é dada por

$$D^e = \begin{bmatrix} d_{11}^e & d_{12}^e & d_{13}^e \\ d_{21}^e & d_{22}^e & d_{23}^e \\ d_{31}^e & d_{32}^e & d_{33}^e \end{bmatrix}, \quad (27)$$

onde

$$d_{ij}^e = \int_T \epsilon \nabla \lambda_i^e(x, y) \cdot \nabla \lambda_j^e(x, y) d\mathbf{x}, \quad i, j = 1, 2, 3,$$

com

$$\nabla \lambda_s^e(x, y) = \begin{bmatrix} \frac{\partial \lambda_s^e}{\partial x} \\ \frac{\partial \lambda_s^e}{\partial y} \end{bmatrix} = \frac{1}{2A^e} \begin{bmatrix} b_s \\ c_s \end{bmatrix}.$$

Então,

$$\nabla \lambda_i^e(x, y) \cdot \nabla \lambda_j^e(x, y) = \begin{bmatrix} \frac{b_i}{2A^e} \\ \frac{c_i}{2A^e} \end{bmatrix} \cdot \begin{bmatrix} \frac{b_j}{2A^e} \\ \frac{c_j}{2A^e} \end{bmatrix} = \frac{1}{4(A^e)^2} (b_i b_j + c_i c_j).$$

A^e é a área do elemento T .

Então

$$d_{ij}^e = \int_T \nabla \lambda_i^e(x, y) \cdot \nabla \lambda_j^e(x, y) d\mathbf{x} = \frac{1}{4(A^e)^2} (b_i b_j + c_i c_j) \underbrace{\int_{\Omega_e} d\mathbf{x}}_{=A^e} = \frac{1}{4A^e} (b_i b_j + c_i c_j),$$

isto é,

$$\begin{aligned} d_{11}^e &= \frac{1}{4A^e} \left[(y_2 - y_1)(y_2 - y_1) + (x_3 - x_2)(x_3 - x_2) \right]; \\ d_{12}^e &= \frac{1}{4A^e} \left[(y_2 - y_1)(y_3 - y_1) + (x_3 - x_2)(x_1 - x_3) \right]; \\ d_{13}^e &= \frac{1}{4A^e} \left[(y_2 - y_1)(y_1 - y_2) + (x_3 - x_2)(x_2 - x_1) \right]; \end{aligned}$$

$$d_{21}^e = d_{12}^e;$$

$$d_{22}^e = \frac{1}{4A^e} \left[(y_3 - y_1)(y_3 - y_1) + (x_1 - x_3)(x_1 - x_3) \right];$$

$$d_{23}^e = \frac{1}{4A^e} \left[(y_3 - y_1)(y_1 - y_2) + (x_1 - x_3)(x_2 - x_1) \right];$$

$$d_{31}^e = d_{13}^e;$$

$$d_{32}^e = d_{23}^e;$$

$$d_{33}^e = \frac{1}{4A^e} \left[(y_1 - y_2)(y_1 - y_2) + (x_2 - x_1)(x_2 - x_1) \right].$$

A matriz local que corresponde ao termo de convecção é dada por

$$C^e = \begin{bmatrix} c_{11}^e & c_{12}^e & c_{13}^e \\ c_{21}^e & c_{22}^e & c_{23}^e \\ c_{31}^e & c_{32}^e & c_{33}^e \end{bmatrix}, \quad (28)$$

onde

$$c_{ij}^e = \int_T (\boldsymbol{\beta} \cdot \nabla \lambda_j^e(x, y)) \lambda_i^e(x, y) d\mathbf{x}.$$

com

$$\boldsymbol{\beta} \cdot \nabla \lambda_j^e(x, y) = \begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} \cdot \begin{bmatrix} \frac{b_j}{2A^e} \\ \frac{c_j}{2A^e} \end{bmatrix} = \frac{1}{2A^e} (b_j \beta_x + c_j \beta_y).$$

Considerando as velocidades β_x e β_y constantes em Ω_e , temos

$$c_{ij}^e = \frac{1}{2A^e} (b_j \beta_x + c_j \beta_y) \underbrace{\int_{\Omega_e} \lambda_i^e(x, y) d\mathbf{x}}_{=A^e/3} = \frac{1}{6} (b_j \beta_x + c_j \beta_y).$$

Se as velocidades β_x e β_y não forem constantes, podemos calcular seus valores no baricentro de cada elemento, descrevendo dessa forma, uma aproximação.

A matriz local do termo de reação é dada por

$$R^e = \begin{bmatrix} r_{11}^e & r_{12}^e & r_{13}^e \\ r_{21}^e & r_{22}^e & r_{23}^e \\ r_{31}^e & r_{32}^e & r_{33}^e \end{bmatrix}, \quad (29)$$

onde

$$r_{ij}^e = \int_T \sigma(\lambda_i^e)(\lambda_j^e) d\mathbf{x} = \begin{cases} \frac{\sigma A^e}{6}, & \text{se } i = j; \\ \frac{\sigma A^e}{12}, & \text{se } i \neq j. \end{cases}$$

Portanto,

$$R^e = \frac{\sigma A^e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Vetor Local F^e

A construção do vetor global \mathbf{F} envolve o termo de fonte e as condições de contorno de Dirichlet e Neumann.

Vamos considerar o modelo descrito na formulação (22), com $g \neq 0$ e $q = 0$. Isso significa que o vetor \mathbf{F} é associado ao termo

$$\int_{\Omega} f v_h d\mathbf{x} + \underbrace{\int_{\Gamma_N} q v_h d\Gamma}_{=0} - \int_{\Omega} (\epsilon \nabla G_h \cdot \nabla v_h + \beta \cdot \nabla G_h v_h + \sigma G_h v_h) d\mathbf{x}, \quad (30)$$

Associado a cada elemento $T \in \mathcal{T}_h$ temos um vetor local

$$F^e = \begin{bmatrix} f_1^e \\ f_2^e \\ f_3^e \end{bmatrix},$$

onde

$$f_i^e = \int_T f \lambda_i^e d\mathbf{x} - \int_T (\epsilon \nabla G_h^e \cdot \nabla \lambda_i^e + \beta \cdot \nabla G_h^e \lambda_i^e + \sigma G_h^e \lambda_i^e) d\mathbf{x},$$

$i = 1, 2, 3$, sendo G_h^e a restrição da função G_h em T .

No contexto do MEF, a função G_h é definida (para maiores detalhes ver ?) por

$$G_h(x_j, y_j) = \begin{cases} g, & \text{se } (x_j, y_j) \in \Gamma_D; \\ 0, & \text{caso contrário.} \end{cases}$$

Obviamente, as integrais (envolvendo a função f) podem ser resolvidas através de métodos de integração numérica.

Estratégia utilizada: aproximaremos a função f em T por sua interpolante linear $\pi f \in \mathbb{P}_1(T)$ e calcularemos a integral de forma exata em função de πf .

$$\pi f(x, y) = \sum_{i=1}^3 f_i \lambda_i^e(x, y),$$

onde $f_i = f(x_i, y_i)$ é o valor da função f no vértice $N_i = (x_i, y_i)$, $i = 1, 2, 3$ do triângulo T .

Portanto,

$$\int_T \lambda_i^e(x, y) f(x, y) d\mathbf{x} = \left(\int_T \lambda_i^e \lambda_1^e d\mathbf{x} \right) f_1 + \left(\int_T \lambda_i^e \lambda_2^e d\mathbf{x} \right) f_2 + \left(\int_T \lambda_i^e \lambda_3^e d\mathbf{x} \right) f_3.$$

Calculando esses integrais para $i = 1, 2, 3$ obtemos o vetor local associado ao termo $\int_T f v_h d\mathbf{x}$,

$$\frac{A^e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \frac{A^e}{12} \begin{bmatrix} 2f_1 + f_2 + f_3 \\ f_1 + 2f_2 + f_3 \\ f_1 + f_2 + 2f_3 \end{bmatrix},$$

onde $f_i = f(x_i, y_i)$.

Analogamente, a função G_h^e é descrita da seguinte forma

$$G_h^e = g_1 \lambda_1^e(x, y) + g_2 \lambda_2^e(x, y) + g_3 \lambda_3^e(x, y),$$

onde

$$g_i = g(x_i, y_i)$$

é o valor que G_h^e assume no vértice (x_i, y_i) , $i = 1, 2, 3$ do triângulo T .

Se $(x_i, y_i) \notin \Gamma_D$, então $g_i = 0$, caso contrário, $g_i = g(x_i, y_i)$.

Então, o vetor local associado ao termo que representa a condição de contorno de Dirichet no lado direito de (30) é dada por

$$\begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} E_{11}g_1 + E_{12}g_2 + E_{13}g_3 \\ E_{21}g_1 + E_{22}g_2 + E_{23}g_3 \\ E_{31}g_1 + E_{32}g_2 + E_{33}g_3 \end{bmatrix},$$

onde E é a matriz local associada a $a(G_h, v_h)$.

Essa matriz é a mesma matriz local K^e descrita em (26).

Vale ressaltar que $g_i = G_h(x_i, y_i)$ é diferente de zero apenas se o ponto nodal (x_i, y_i) é um vértice pertencente a fronteira prescrita de Dirichlet.

Portanto, o vetor local F^e , para $q = 0$, é dado por

$$F^e = \begin{bmatrix} f_1^e \\ f_2^e \\ f_3^e \end{bmatrix} = \frac{A^e}{12} \begin{bmatrix} 2f_1 + f_2 + f_3 \\ f_1 + 2f_2 + f_3 \\ f_1 + f_2 + 2f_3 \end{bmatrix} - \begin{bmatrix} E_{11}g_1 + E_{12}g_2 + E_{13}g_3 \\ E_{21}g_1 + E_{22}g_2 + E_{23}g_3 \\ E_{31}g_1 + E_{32}g_2 + E_{33}g_3 \end{bmatrix},$$

onde $E_{ij} = K_{ij}^e$.

Implementação Computacional

A implementação computacional de um código de elementos finitos envolve alguns passos importantes. Primeiro destacaremos algumas variáveis úteis que são utilizadas na implementação.

- ▶ *nel*: essa variável representa o número de elementos da malha \mathcal{T}_h ;
- ▶ *neq*: representa o número de equações, isto é, a ordem do sistema global a ser resolvido;
- ▶ *nnos*: é o número de pontos nodais da malha \mathcal{T}_h .

Além das três variáveis descritas acima, destacaremos também algumas estruturas de dados utilizadas: vetores e matrizes

- ▶ *COORD*: *matriz coordenada* - matriz que armazena as coordenadas (x, y) dos pontos nodais da malha;
- ▶ *ID*: vetor que identifica a equação associada a cada vértice global não prescrito;
- ▶ *IEN*: *matriz de conectividade* - matriz que associa a cada elemento seu vértices globais;
- ▶ *LM*: *matriz de localização* - essa matriz associa os vértices (pontos nodais) locais do elemento ao número da equação correspondente;
- ▶ *BOUNDCOND*: vetor que armazena os valores prescritos da fronteira (condições de contorno de Dirichlet).

A consideração de condições de fronteiras de Neumann envolve o uso de mais estruturas de dados, e não abordaremos isso nesse trabalho. Em geral, um programa de elementos finitos é dividido em três partes:

- ▶ *Pre-processamento*: consiste na geração de malha, construção das estruturas de dados e cálculos relacionados aos elementos;
- ▶ *Processamento*: consiste na montagem e solução do sistema global $KU = F$ para problemas estacionários;
- ▶ *Pós-processamento*: saída dos dados calculados e visualização gráfica da solução e outras quantidades de interesse.

Faremos um exemplo simples para ilustrar o funcionamento do método de elementos finitos na montagem do sistema (25). Considere a malha representada na Fig. 3.

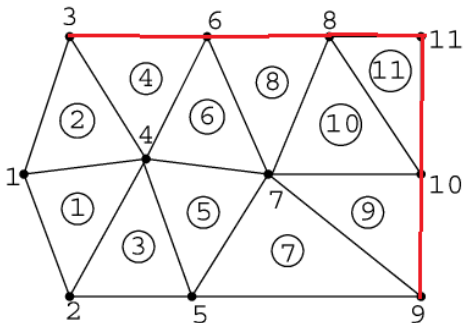


Figure: Malha de elementos finitos com 11 elementos e 11 vértices (pontos nodais)

Essa malha possui 11 elementos triangulares e 11 vértices (pontos nodais).

Os vértices 3, 6, 8, 9, 10 e 11 são prescritos com condições de contorno de Dirichlet (fronteira em vermelho).

Os outros vértices chamaremos de *vértices livres*. Então teremos os seguintes valores para as variáveis *nel*, *neq* e *nnos*:

- ▶ *nel*: 11(número de elementos);
- ▶ *neq*: 5 (número de equações ou de vértices livres);
- ▶ *nnos*: 11 (número total de vértices).

O próximo passo é preencher a matriz *COORD* que é utilizada para armazenar as coordenadas dos vértices da malha.

Essa matriz possui *nnos* linhas e 2 colunas, e associa a cada ponto nodal suas coordenadas *x* e *y*, isto é, $COORD[i][j]$ representa a *j*-ésima coordenada (*x* ou *y*) do *i*-ésimo vértice da malha.

O vetor *ID* que identifica a equação associada a cada vértice global *livre* é um vetor de tamanho *nnos*, onde

$$ID[i] = \begin{cases} eq, & \text{se } i \text{ é um ponto nodal livre;} \\ 0, & \text{se } i \text{ é um ponto nodal prescrito,} \end{cases}$$

sendo que $eq > 0$ é o número da equação associada ao nó *i*.

$$ID = \begin{bmatrix} 1 & 2 & 0 & 3 & 4 & 0 & 5 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

Neste caso, os vértices *livres* 1, 2, 4, 5 e 7 estão associados às equações de número 1, 2, 3, 4, 5, respectivamente.

A matriz IEN possui nel linhas e 3 colunas e associa cada elemento $e = 1, 2, \dots, nel$ a seus respectivos pontos nodais (vértices) globais.

$$IEN = \begin{bmatrix} z_{1,1} & z_{1,2} & z_{1,3} \\ z_{2,1} & z_{2,2} & z_{2,3} \\ \vdots & \vdots & \vdots \\ z_{nel,1} & z_{nel,2} & z_{nel,3} \end{bmatrix},$$

onde $z_{i,j}$ é o ponto nodal global do elemento T_i associado ao ponto nodal local j , ou seja,

$$IEN[elemento][noLocal] = noGlobal.$$

A matriz IEN é dada por

$$IEN = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 4 & 3 \\ 2 & 5 & 4 \\ 3 & 4 & 6 \\ 4 & 5 & 7 \\ 4 & 7 & 6 \\ 5 & 9 & 7 \\ 6 & 7 & 8 \\ 7 & 9 & 10 \\ 7 & 10 & 8 \\ 8 & 10 & 11 \end{bmatrix} .$$

A matriz de localização LM possui *nel* linhas e 3 colunas, e associa os vértices locais de cada elemento ao número da equação correspondente. Para a malha em estudo, temos

$$LM = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 0 \\ 2 & 4 & 3 \\ 0 & 3 & 0 \\ 3 & 4 & 5 \\ 3 & 5 & 0 \\ 4 & 0 & 5 \\ 0 & 5 & 0 \\ 5 & 0 & 0 \\ 5 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

A matriz de localização LM pode ser construída a partir do vetor ID e da matriz IEN através da expressão

$$LM[elem][noLocal] = ID[IEN[elem][noLocal]].$$

O vetor $BOUNDCOND$ armazena os valores prescritos da fronteira (condições de contorno de Dirichlet) e é dado por

$$BOUNDCOND = \begin{bmatrix} x & x & g_3 & x & x & g_6 & x & g_8 & g_9 & g_{10} & g_{11} \end{bmatrix}^T,$$

onde $g_i = g(x_i, y_i)$.

O valor descrito por x pode ser qualquer número real, uma vez que essas posições do vetor não são utilizadas, pois estão associadas aos vértices não-prescritos, isto é, aos números das equações que são representadas pelo vetor ID .

O preenchimento do vetor *BOUNDCOND* usando o vetor *ID* é descrito pelo algoritmo a seguir,

```
para i = 1 até nnos faça
    se ID[i] = 0 então
        BOUNDCOND[i] ←  $g(x_i, y_i)$ ;
    fim-se
fim-para
```

Algoritmo: monta matriz local do termo difusivo

Entrada: elemento e , matriz D^e , coeficiente $\epsilon > 0$, matrizes IEN e $COORD$

Saída: matriz D^e preenchida

```

aloca vetores x e y de ordem 3    % armazena coordenadas dos nós do elemento
para noLocal = 1,2,3 faça        % loop sobre os vértices do elemento e
    noGlobal  $\leftarrow IEN[e][noLocal]$ ;    % associação entre os nós global e local
    x[noLocal]  $\leftarrow COORD[noGlobal][1]$ ;    % coordenada x do nó global
    y[noLocal]  $\leftarrow COORD[noGlobal][2]$ ;    % coordenada y do nó global
fim-para
b[1]  $\leftarrow y[2] - y[3]$ ;    b[2]  $\leftarrow y[3] - y[1]$ ;    b[3]  $\leftarrow y[1] - y[2]$ ;
c[1]  $\leftarrow x[3] - x[2]$ ;    c[2]  $\leftarrow x[1] - x[3]$ ;    c[3]  $\leftarrow x[2] - x[1]$ ;
Ae  $\leftarrow (c[3] * b[2] - c[2] * b[3]) / 2.0$ ;    % área do elemento
para i = 1,2,3 faça
    para j = 1,2,3 faça
        D[i][j]  $\leftarrow (\epsilon / (4.0 * A^e)) * (b[i] * b[j] + c[i] * c[j])$ ;
    fim-para
fim-para
    
```

Algoritmo: monta matriz local do termo convectivo

Entrada: elemento e , matriz C^e , vetor $\beta = (\beta_x, \beta_y)$, matrizes IEN e $COORD$

Saída: matriz C^e preenchida

```
aloca vetores x e y de ordem 3    % armazena coordenadas dos nós do elemento
para noLocal = 1,2,3 faça        % loop sobre os vértices do elemento e
    noGlobal  $\leftarrow IEN[e][noLocal]$ ;    % associação entre os nós global e local
    x[noLocal]  $\leftarrow COORD[noGlobal][1]$ ;    % coordenada x do nó global
    y[noLocal]  $\leftarrow COORD[noGlobal][2]$ ;    % coordenada y do nó global
fim-para
b[1]  $\leftarrow y[2] - y[3]$ ;    b[2]  $\leftarrow y[3] - y[1]$ ;    b[3]  $\leftarrow y[1] - y[2]$ ;
c[1]  $\leftarrow x[3] - x[2]$ ;    c[2]  $\leftarrow x[1] - x[3]$ ;    c[3]  $\leftarrow x[2] - x[1]$ ;
Ae  $\leftarrow (c[3] * b[2] - c[2] * b[3]) / 2.0$ ;
para i = 1,2,3 faça
    para j = 1,2,3 faça
        C[1][i]  $\leftarrow C[2][i] \leftarrow C[3][i] \leftarrow (\beta_x / 6.0) * b[i] + (\beta_y / 6.0) * c[i]$ ;
    fim-para
fim-para
```

Algoritmo: monta matriz local do termo reativo

Entrada: elemento e , matriz R^e , coeficiente $\sigma \geq 0$, matrizes IEN e $COORD$

Saída: matriz R^e preenchida

```
aloca vetores x e y de ordem 3    % armazena coordenadas dos nós do elemento
para noLocal = 1, 2, 3 faça      % loop sobre os vértices do elemento e
    noGlobal ← IEN[e][noLocal];  % associação entre os nós global e local
    x[noLocal] ← COORD[noGlobal][1]; % coordenada x do nó global
    y[noLocal] ← COORD[noGlobal][2]; % coordenada y do nó global
fim-para
b[1] ← y[2] - y[3];    b[2] ← y[3] - y[1];    b[3] ← y[1] - y[2];
c[1] ← x[3] - x[2];    c[2] ← x[1] - x[3];    c[3] ← x[2] - x[1];
Ae ← (c[3] * b[2] - c[2] * b[3]) / 2.0;
para i = 1, 2, 3 faça
    para j = 1, 2, 3 faça
        se i = j então
            R[i][j] ← σ * Ae / 6.0;
        senão
            R[i][j] ← σ * Ae / 12.0;
        fim-se
    fim-para
fim-para
```

Algoritmo: monta vetor local

Entrada: elemento e , vetor F^e , vetores ID e $BOUNDCOND$, matrizes IEN e $COORD$, função de fonte f . Além desse dados, considere a matriz $K^e = D^e + C^e + R^e$

Saída: vetor F^e preenchida

aloca matriz M de ordem 3×3 % armazena matriz de massa (matriz R^e com $\sigma = 1$)

$M \leftarrow R^e$ com $\sigma = 1$ % chama a rotina que monta a matriz R^e

% Construção da parte de F^e associada ao termo de fonte

para $i = 1, 2, 3$ faça

$F^e[i] = 0.0$

para $j = 1, 2, 3$ faça

$noGlobal = IEN[e][j];$

$x = COORD[noGlobal][1];$

$y = COORD[noGlobal][2];$

$F^e[i] = F^e[i] + M[i][j] * f(x, y);$

fim-para

fim-para

% Construção da parte de F^e associada a condição de contorno de Dirichlet

para $i = 1, 2, 3$ faça

$noGlobal = IEN[e][i];$

se $ID[noGlobal] \neq 0$ então

$F^e[i] = F^e[i] - (BOUNDCOND[IEN[e][1]] * K^e[i][1] + BOUNDCOND[IEN[e][2]] * K^e[i][2]$

$BOUNDCOND[IEN[e][3]] * K^e[i][3]);$

fim-se

fim-para

Algoritmo: monta matriz e vetor globais

Entrada: nel , matriz LM , matriz K e vetor F

Saída: matriz K e vetor F preenchidos

```

 $K \leftarrow 0$ ;  $F \leftarrow 0$ ;
aloca matriz  $K^e$  (de ordem  $3 \times 3$ ) e vetor  $F^e$  (de ordem 3);
para  $e = 1, 2, 3 \dots nel$  faça % percorrendo todos os elementos da malha
    monta matriz local  $K^e$  e vetor local  $F^e$  para o elemento atual;
    para  $i = 1, 2, 3$  faça
        se  $LM[e][i] \neq 0$  então;
             $F[LM[e][i]] \leftarrow F[LM[e][i]] + F^e[i]$ ;
            para  $j = 1, 2, 3$  faça
                se  $LM[e][j] \neq 0$  então
                     $K[LM[e][i]][LM[e][j]] \leftarrow K[LM[e][i]][LM[e][j]] + K^e[i][j]$ ;
            fim-se
        fim-para
    fim-se
fim-para

```

Muito obrigado!



The poster for the LNCC Summer Program 2018 features a blue background. At the top left is the LNCC logo. In the center, the text 'PROGRAMA DE VERÃO' is displayed. On the right side, the year '2018' is written vertically. Below the title, there is a collage of images showing various activities: a large building, a person working on a laptop, and a group of people. To the right of the collage, a list of activities is provided.

LNCC

**PROGRAMA
DE
VERÃO**

2018

Escola Supercomputador SDumont
Jornada em Ciência de Dados
Jornada em Ecologia Teórica
Jornada de Iniciação Científica e Tecnológica
XI Encontro Acadêmico de Modelagem Computacional
IV Encontro em Modelagem Matemática do
Crescimento Tumoral
Minicursos Avulsos